Nadia Marina Ramasawmy(Mosoiu)
Dec 9 2023

OUTLINE

FIGURE TABLE OF CONTENTS

1. EXECUTIVE SUMMARY

This is a capstone project for data analysis with python delivered on EdX The training materials were initial very well thought, the challenges are that the journals are not updated. Some libraries are obsolete. The data sets are not consistent for all analyses

1.1. Summary of all results

We need to see where SpaceX underperforms. All four US launching sitees are further to the equator than the EU launch site in Guyana. Right now, due to the current political situation the Soyuz launch pad is not used by the Russian counterparts. Orinoco could use the Guyana site **Figure 22** for heavy payloads given that less fuel is needed for lift. If we limit the analysis to the US, we notice that we can launch heavy payloads from Florida and moderate payloads from California. The site from California is not in the hurricane path and it is attractive for customers from the Americas and from the Far East. Here you can see the location: **Vandenberg CA Site analysis**
Spacex does not have a good record for placing payloads in the the GTO orbit. More recent launches show that this rate is improving, **Figure 10** We could court potential clients for this orbit placement. Here is the success rate for orbit placement for SpaceX**: Figure 9.** We also notice that the payload for this orbit is between 4000 and 6000 kg, ideal from launching from California. See **Figure 11**Figure 11
According to SQL analysis SpaceX has tremendous success using F9 FT boosters F9 B4 boosters and F9 B5 boosters. The F5 boosters are used for heavy payloads The dashboard shows another picture, B4 having a 50% success rate **Figure 29** but, as stated – the set of data for the dashboard is incomplete. See **Figure 1**
If we were to predict a successful launch based on historic data several classification models can be used. For the moment three models have the same accuracy rate of .83% . See **Accuracy of the Model**
One can use the model they are more familiar with. The confusion matrix shows similar data. False positives are encountered for all three successful models as it can be seen in **Figure 30**
One should not use node clustering since false positives and false negatives are present

1.2. Summary of methodologies

Data collection methodology
Perform data wrangling
Perform **exploratory data analysis (EDA) using visualization and SQL**
Perform interactive visual analytics using Folium and Plotly Dash
Perform predictive analysis using classification models
Perform **exploratory data analysis (EDA) using visualization and SQL**

2. INTRODUCTION

2.1. Project background and context

The data analysis department at ORINOCO Space produced this report at the request of CFO Artu Ditto . The findings of the report will be used to design a strategy to compete with SPACEX. We have been tasked to analyze if the launch costs our competitor, SPACEX is providing in bidding documents are substantiated by the successful reutilization of the first stage of the rockets the competitor uses.

2.2. Problems answered

We need to see where spaceX underperforms. Falcon 9 is a two steps rocket. It is the flagship of SpaceX. A launch of payload costs on average 62 million dollars; other providers cost upward of 165 million dollars each. Much of the savings is because SpaceX can reuse the first stage. We need to determine if the first stage will land. Then we can determine the cost of a launch. We can also determine which launch pads can accommodate ORINOCO launches to compete with SPACEX

3. METHODOLOGY

3.1. Executive Summary

Data collection methodology

The data sources were the SpaceX API and HTML tables extracted from Falcon 9 Wikipedia pages. Apart this Skills Network provided some data for SQL and the Dash dashboard We designed our Application Programming Interface using known Python libraries such as: requests pandas, NumPy, datetime, beautiful soup to acquire and transform the data into data frames. We focused on obtaining relevant Falcon 9 data and eliminating noise. This data is ready for conversion into a usable form – a process known as wrangling.

Perform data wrangling

We discovered we need to define the success rate for the rocket launches. We transformed the data frame to contain this new column and we addressed missing data

Perform exploratory data analysis (EDA) using visualization and SQL

Using the wrangled data from SpaceX We analyzed the success or failure of launch depending on Launch site, Orbit and Payload The Spacex dataset was then exported to a CSV file. This csv file was again loaded as a pandas data frame. We used new libraries sqlalchemy and iphytonsql to convert the pandas data frame in a database and perform SQL queries. sqlite3 provides a SQL-like interface to read, query, and write SQL databases from Python. I have created a new table from all records in the SQL database named SPACEXNR to perform queries

Mission success is extremely high. I have competed failures over the whole data set (and not only as suggested by skillset until 2017)   We tried to detect which boosters have success in ground pad landing for a payload mass where EDA visualization showed most failure outcomes. Outside the Skillset template notebook, I've analyzed the number of failures over time.

Perform interactive visual analytics using Folium and Plotly Dash

We used data offered by Skillsets. It contains the longitude, latitude, launch site code and outcome of landing. Using the Folium library, the launch sites were identified on the US map. We clustered the launches setting successful and unsuccessful data markers. We assessed how far are launch sites are from cities, railways, highways and the coastline. WE used polyline to calculate distances and to mark the equator line on the map.

For dash, I have not used the virtual environment offered by Skillset. I have just installed the dash library on my system and followed the instructions in the assignment journal.

I used the data from skillsets which is incomplete.

```
# Dash Data , sql conflicitng results
urldash = "https://cf-courses-data.s3.us.cloud-object-storage.appdomain.cloud/IBM-DS0321EN-SkillsNetwork/datasets/spacex_launch_dash.csv"
dash_df = pd.read_csv(urldash)
dash_df.tail(5)
print("See Dash Dataframe Features",dash_df.shape)

sql_df = pd.read_csv("https://cf-courses-data.s3.us.cloud-object-storage.appdomain.cloud/IBM-DS0321EN-SkillsNetwork/labs/module_2/data/Spacex.cs
print("See SQL DataFrame Features", sql_df.shape)
print(dash_df.tail(1))
print(sql_df.tail(1))

✓ 0.3s
```

```
See Dash Dataframe Features (56, 7)
See SQL DataFrame Features (101, 10)
    Unnamed: 0  Flight Number   Launch Site  class  Payload Mass (kg)  \
55          55             56  CCAFS SLC-40      0             5384.0

   Booster Version Booster Version Category
55  F9 B4  B1040.2                       B4
          Date Time (UTC) Booster_Version Launch_Site        Payload  \
100  2020-12-06   16:17:08  F9 B5 B1058.4   KSC LC-39A  SpaceX CRS-21

     PAYLOAD_MASS__KG_      Orbit     Customer Mission_Outcome Landing_Outcome
100               2972  LEO (ISS)  NASA (CRS)         Success         Success
```

Figure 1

In support of this statement I am providing this snapshot. As you can see, there are more than 100 records for SQL analysis and only 55 for the Dash dashboard building. We used the pandas library to process data and the plotly library for plotting graphs. The dash library was used in order to create an interactive dashboard presenting the SPACEX correlation between payload and success for a specific site and booster type. A slider allows to set the payload mass within a range.

Perform predictive analysis using classification models

We tried to find the best Hyperparameter for Support Vector Machine, Classification Trees, Node Clustering and Logistic Regression. We split the data into training and testing data. The data is divided into validation data, a second set used for training data; then the models are trained and hyperparameters are selected using GridSearchCV. The accuracy of each model was assessed against the validation data. For each model a confusion matrix was created.

4. DATA COLLECTION

4.1. Data Collection – SpaceX API.

The completed SpaceX API data collection process can be found here: [Data Collection SpaceX API]
We designed our Application Programming Interface using the python library requests obtaining a json file. Then we used functions from pandas, numpy and datetime Python libraries to transform the data into data frames for SPACEX API. This allows for flexibility both ways. it is easy to eliminate un-necessary data. We only looked for Falcon 9 booster data and we have expanded the data frame querying the SPACEX API for launch longitude and latitude, payload ,payload orbit , launch date, Launch Site, outcome with helper functions provided by IBM Skillset

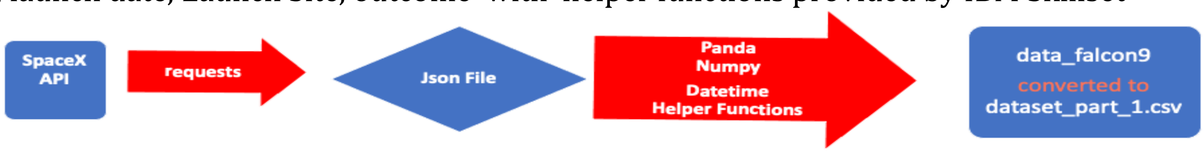


Figure 2

Here is the head of dataset_part_1.csv

dataset_part_1

FlightNumber	Date	BoosterVersion	PayloadMass	Orbit	LaunchSite	Outcome	Flights	GridFins	Reused	Legs	LandingPad	Block	ReusedCount	Serial	Longitude	Latitude
1	2010-06-04	Falcon 9	6123.547647058820	LEO	CCSFS SLC 40	None None	1	FALSE	FALSE	FALSE		1.0	0	B0003	-80.577366	28.5618571
2	2012-05-22	Falcon 9	525.0	LEO	CCSFS SLC 40	None None	1	FALSE	FALSE	FALSE		1.0	0	B0005	-80.577366	28.5618571
3	2013-03-01	Falcon 9	677.0	ISS	CCSFS SLC 40	None None	1	FALSE	FALSE	FALSE		1.0	0	B0007	-80.577366	28.5618571
4	2013-09-29	Falcon 9	500.0	PO	VAFB SLC 4E	False Ocean	1	FALSE	FALSE	FALSE		1.0	0	B1003	-120.610829	34.632093

Figure 3

4.2. Data Collection – Scraping.

The completed web scraping notebook process can be found here: [Data Collection Web Scraping]

4.3. A Wiki Falcon9 web page that was scrapped of information We designed our Application Programming Interface using known Python libraries such as requests obtaining a json file. Then we used the beautiful soup library to define a soup object – a HTML table. This was the basis of creating a dictionary of useful data. We used helper functions provided by IBM Skillset to process web scraped data. We eliminated type errors in the process of creating the dictionary. The dictionary is converted to a data frame. We notice the data columns have some common elements with the data from SpaceX API but data is formatted differently.

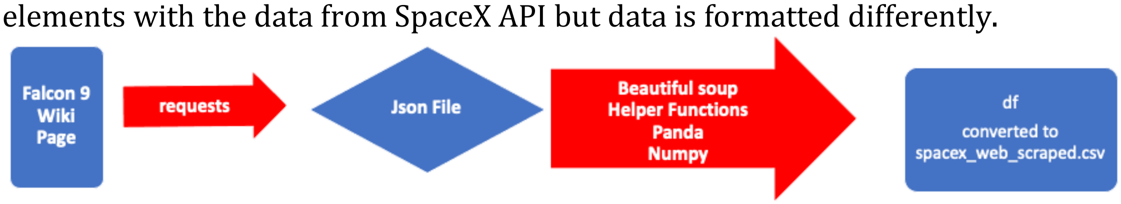


Figure 4

5. DATA WRANGLING

The complete Data Wrangling notebook can be found here: **Data Wrangling**
There are four steps in the data wrangling:
Discovery: Define success rate for Falcon9 rockets
Transformation
- Correct missing data.
- add several columns Orbit, Class(landing outcome) to the data frame
- transform objects in the data frame in string or integers

Validation: check landing success rate
Publishing – output data frames to csv file for Estimated Data Analysis
Here is a comparison between the Data Collecting output and data Wrangling output for SpaceX

Data Frame – Collected Data – Frame Head

	FlightNumber	Date	BoosterVersion	PayloadMass	Orbit	LaunchSite	Outcome	Flights	GridFins	Reused	Legs	LandingPad	Block	ReusedCount	Serial	Longitude	Latitude
4	1	2010-06-04	Falcon 9	NaN	LEO	CCSFS SLC 40	None None	1	False	False	False	None	1.0	0	B0003	-80.577366	28.561857
5	2	2012-05-22	Falcon 9	525.0	LEO	CCSFS SLC 40	None None	1	False	False	False	None	1.0	0	B0005	-80.577366	28.561857
6	3	2013-03-01	Falcon 9	677.0	ISS	CCSFS SLC 40	None None	1	False	False	False	None	1.0	0	B0007	-80.577366	28.561857

DATA WRANGLING

Data Frame –Wrangled Data – Head and Tail

	FlightNumber	Date	BoosterVersion	PayloadMass	Orbit	LaunchSite	Outcome	Flights	GridFins	Reused	Legs	LandingPad	Block	ReusedCount	Serial	Longitude	Latitude	Class
0	1	2010-06-04	Falcon 9	6104.959412	LEO	CCAFS SLC 40	None None	1	False	False	False	NaN	1.0	0	B0003	-80.577366	28.561857	0
1	2	2012-05-22	Falcon 9	525.000000	LEO	CCAFS SLC 40	None None	1	False	False	False	NaN	1.0	0	B0005	-80.577366	28.561857	0

	FlightNumber	Date	BoosterVersion	PayloadMass	Orbit	LaunchSite	Outcome	Flights	GridFins	Reused	Legs	LandingPad	Block	ReusedCount	Serial	Longitude	Latitude	Class
88	89	2020-10-24	Falcon 9	15400.0	VLEO	CCAFS SLC 40	True ASDS	3	True	True	True	5e9e3033383ecbb9e534e7cc	5.0	2	B1060	-80.577366	28.561857	1
89	90	2020-11-05	Falcon 9	3681.0	MEO	CCAFS SLC 40	True ASDS	1	True	False	True	5e9e3032383ecb6bb234e7ca	5.0	0	B1062	-80.577366	28.561857	1

Figure 5

6. EXPLORATORY DATA ANALYSIS

6.1. EDA with Data Visualization

The complete Exploratory Data Analysis notebook can be found here: EDA with data visualization notebook
Using the wrangled data from SpaceX We visualized the dependency of success failure of independent variables: We used pandas, numpy to process data and matplotlib and seaborn to build graphs. We created the following

- plot out the FlightNumber vs. PayloadMass and overlay the outcome of the launch
- plot the relationship between Flight Number and Launch Site and overlay the outcome of the launch
- plot the relationship between launch sites and the payload mass of each launch and overlay the outcome of the launch
- plot the relationship between success rate of each Orbit type
- plot the relationship between Flight Number and Orbit type and overlay the outcome of the launch
- plot the relationship between Orbit type and payload mass of each launch overlay the outcome of the launch
- plot the trend of launch success over time

The plotted graph model offered by Skillset a length/height ratio of 5:1 making observation very difficult.
The variation of success/failure depending on Flight Number and Payload can be better discerned using the 2:1 proportion in the scatter plot style.

Figure 6

Based on these visual cues we have generated a data frame with containing variables that affect the success rate. The variables chosen are :
FlightNumber
PayloadMass
Orbit
LaunchSite
Flights
GridFins
Reused
Legs
LandingPad
Block
ReusedCount
Serial

We then expanded several columns creating dummy variables to categorical columns Orbit LaunchSite, LandingPad and Serial

6.2. Exploratory Data Analysis with SQL

The complete notebook with Exploratory Data Analysis with SQL can be found here: EDA with SQL

SQL queries perform much faster than manipulating csv rows. For the current project we could have extracted the following data using pandas data frames:

- Display the names of the unique launch sites in the space mission
- Display 5 records where launch sites begin with the string 'KSC'
- 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 where the successful landing outcome in drone ship was achieved.
- List the names of the boosters which have success in ground pad and have payload mass greater than 4000 but less than 6000
  List the total number of successful and failure mission outcomes
- List the names of the booster versions which have carried the maximum payload mass. Use a subquery
- List the records which will display the month names, successful landing outcomes in ground pad, booster versions, launch site for the months in year 2017
- List the records which will display the month names, successful landing outcomes in ground pad, booster versions, launch site for the months in year 2017
- List the records which will display the month names, successful landing outcomes in ground pad, booster versions, launch site for the months in year 2017

7. BUILD AN INTERACTIVE MAP WITH FOLIUM

The complete notebook with Folium Map can be found here: Folium Launch Site Map

We analyzed the existing launch site locations
We started by marking all launch sites on a folium map. We defined circles using the site coordinates, added markers with the site names and placed the objects on the map.
Then, for each site, a cluster is created. It is comprised of failed and successful launches
I focused on the VAFB site. It is the further site from the equator but not in the path of hurricanes. I measured distances to the closest city, highway, railway and coastline. You will not be able to see the folium map in git – one would need to download the journal and run locally. Map snapshots are presented in the visualization section below.

8. BUILD A DASHBOARD WITH PLOTLY DASH

The script generating the dashboard can be found here: **Site Launch Dasboard**

The data set offered by Skils Network is half the size of the data that is analyzed with other methods. **The results obtained here conflict with the sql data interpretation.** We used the pandas library to process data, plotly for plotting graphs and dash in order to create an interactive dashboard presenting the SPACEX correlation between payload and success for a specific site and booster type. A slider allows to set the payload mass within a range. Map visualization is quite cumbersome in this document

9. PREDICTIVE ANALYSIS

The complete notebook with Predictive Analysis can be found here: **Predicted Analysis**

We used pandas, and numpy to process data. Seaborn was useful to normalize data. The sklearn library is deprecated. Used scikit-learn instead in order to split our data into test and train sets, sift through classification algorithms to find the best among these algorithms: Support Vector Machine, Classification Trees, Node Clustering and Logistic Regression.

SECTION 2

INSIGHTS DRAWN FROM EDA

Flight Number vs. Launch Site

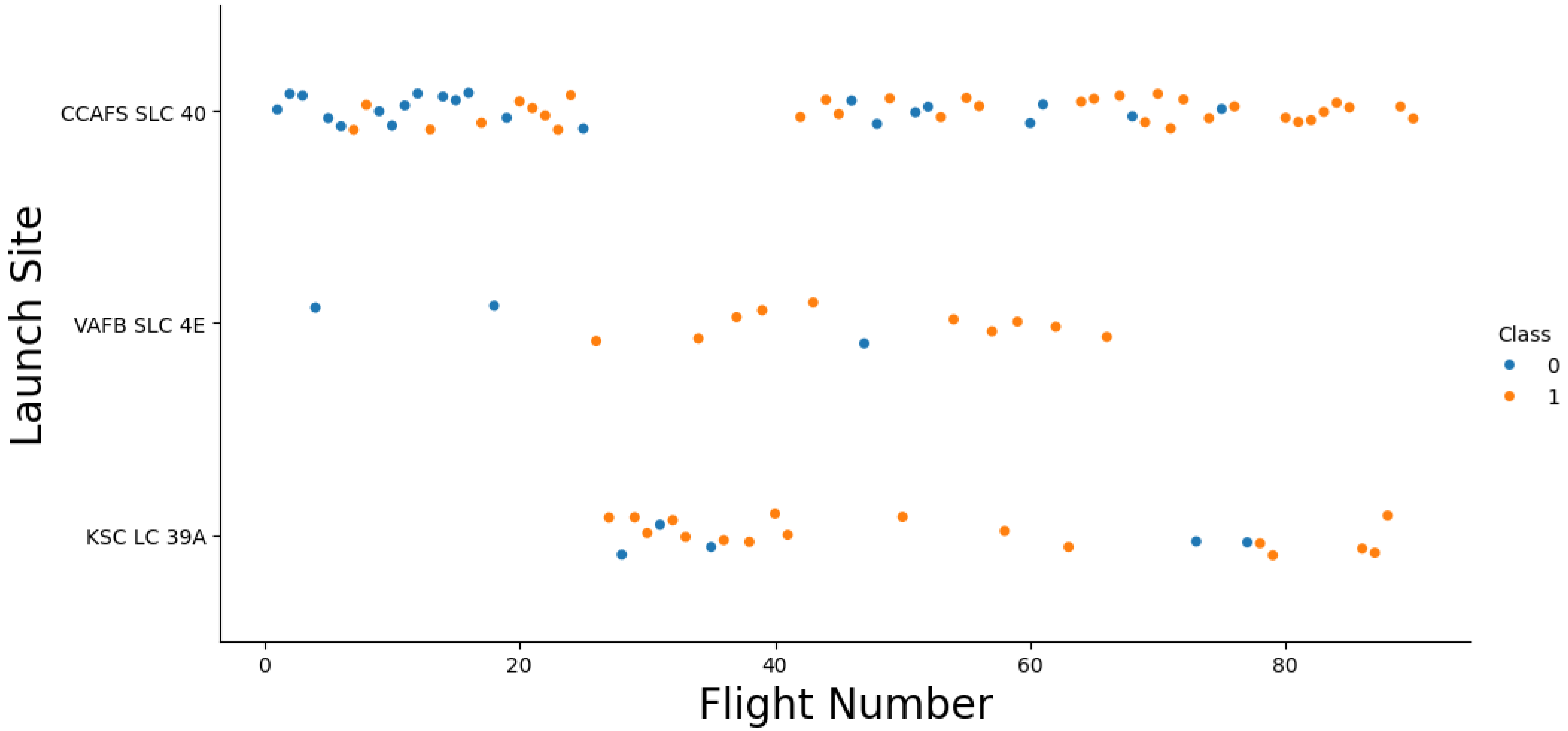


Figure 7

Payload vs. Launch Site

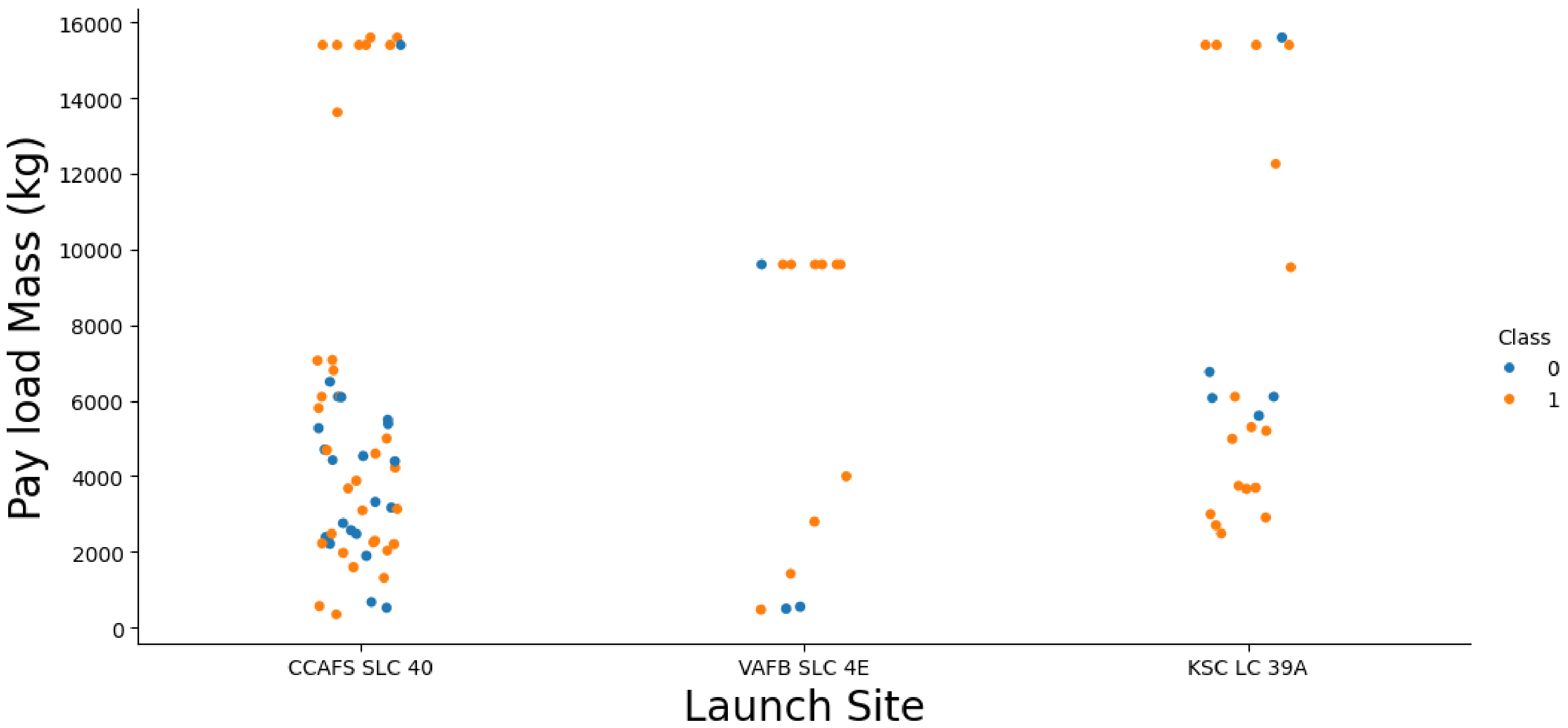


Figure 8

Now if you observe Payload Vs. Launch Site scatter point chart you will find for the VAFB-SLC launchsite there are no rockets launched for heavypayload mass(greater than 10000).

Success Rate vs. Orbit Type

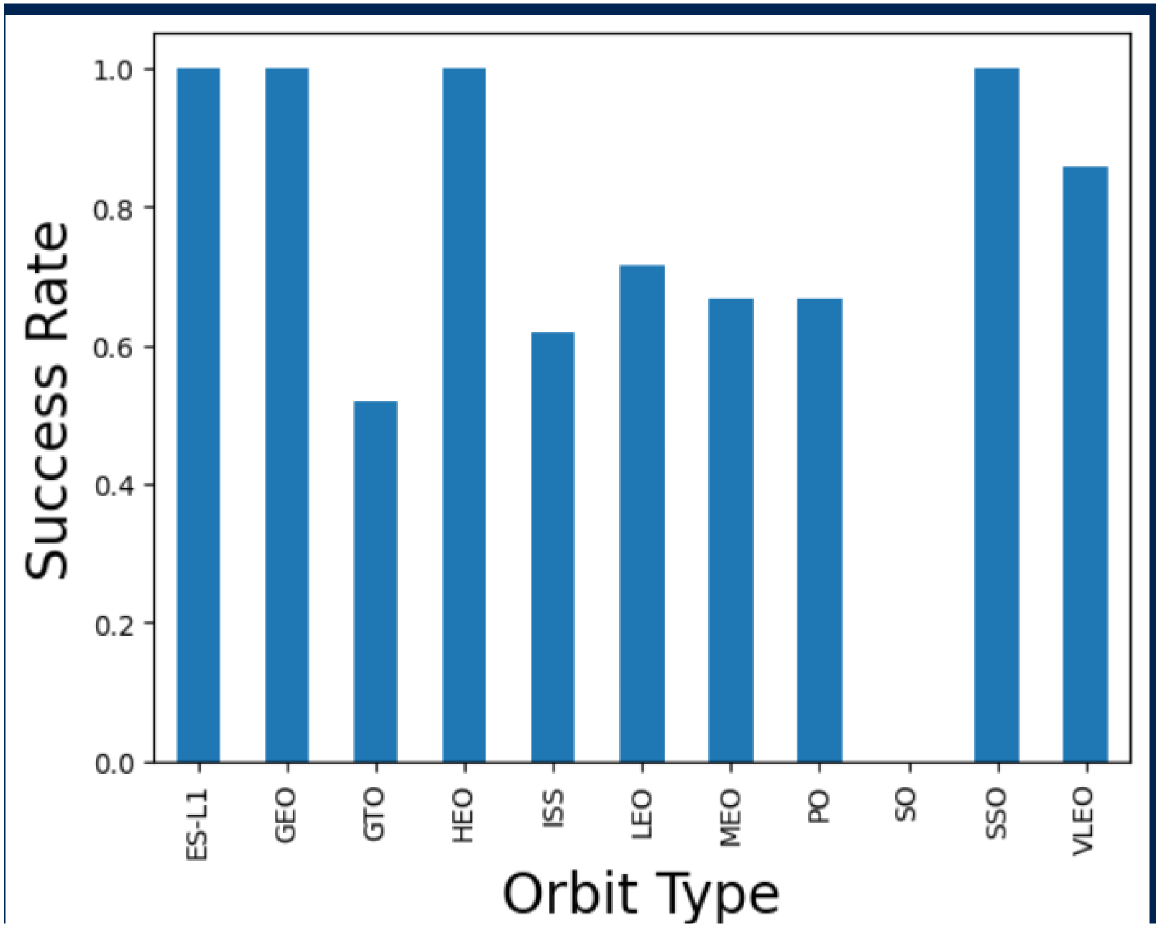


Figure 9

Flight Number vs. Orbit Type

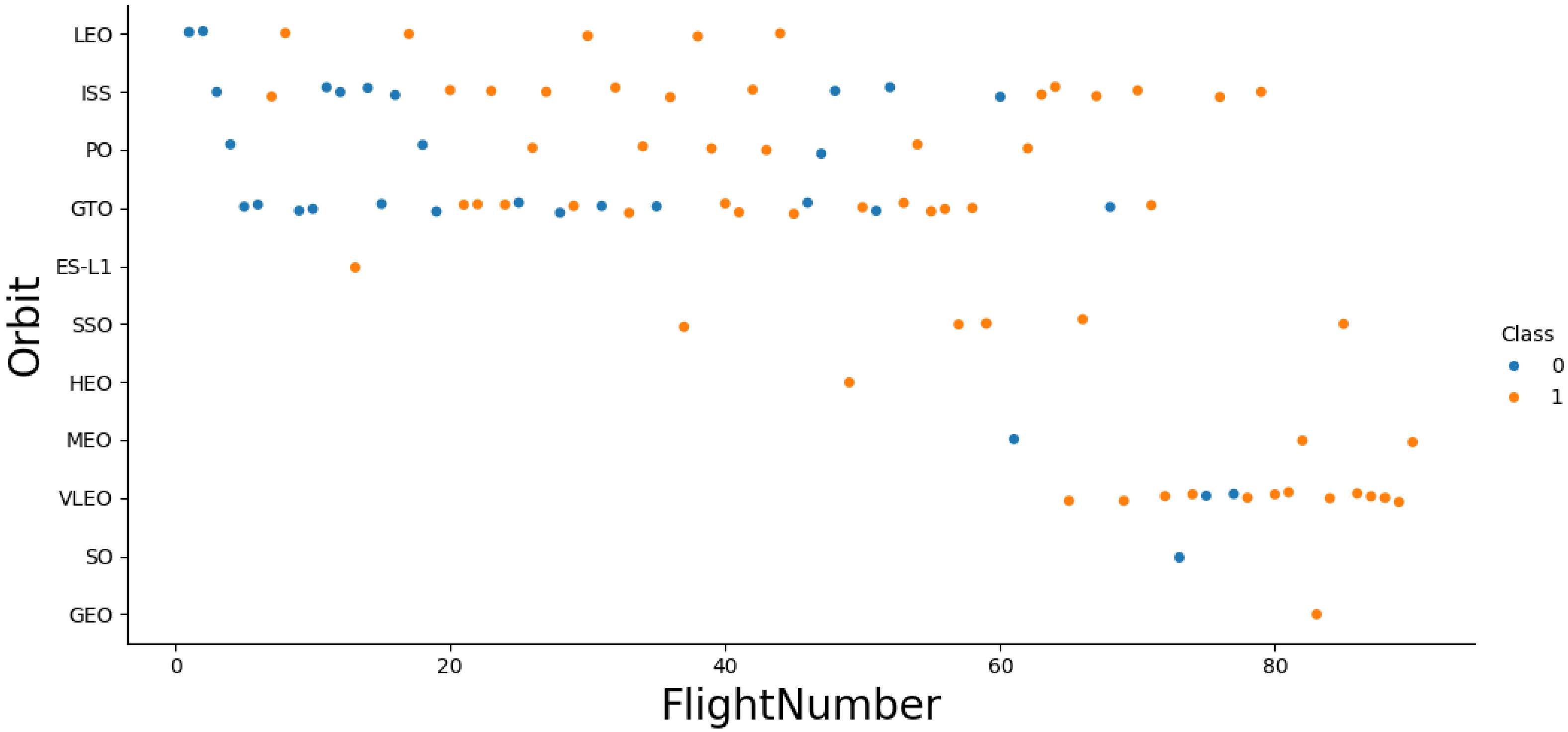


Figure 10

You should see that in the LEO orbit the Success appears related to the number of flights; on the other hand, there seems to be no relationship between flight number when in GTO orbit.

Payload Mass vs Orbit Type

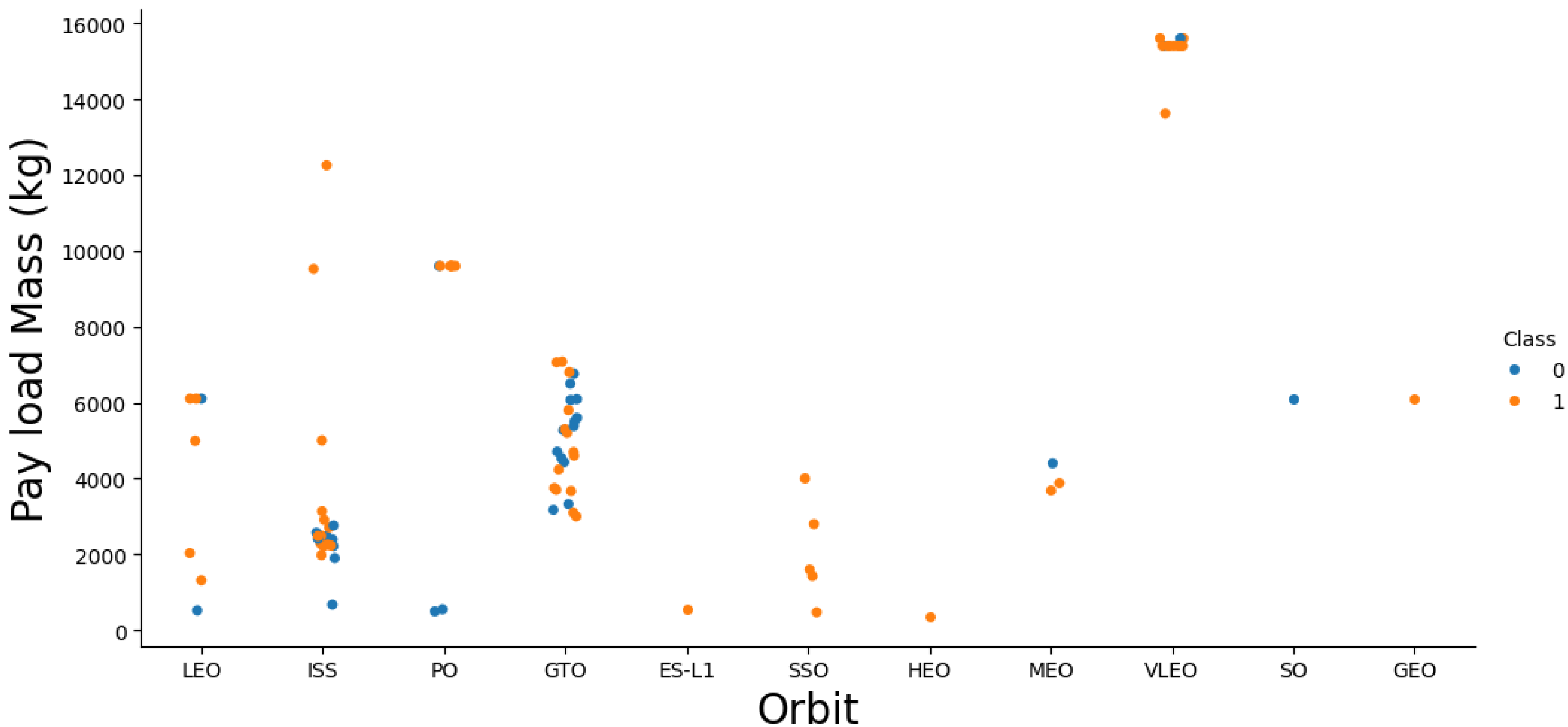


Figure 11

Launch Success Yearly Trend

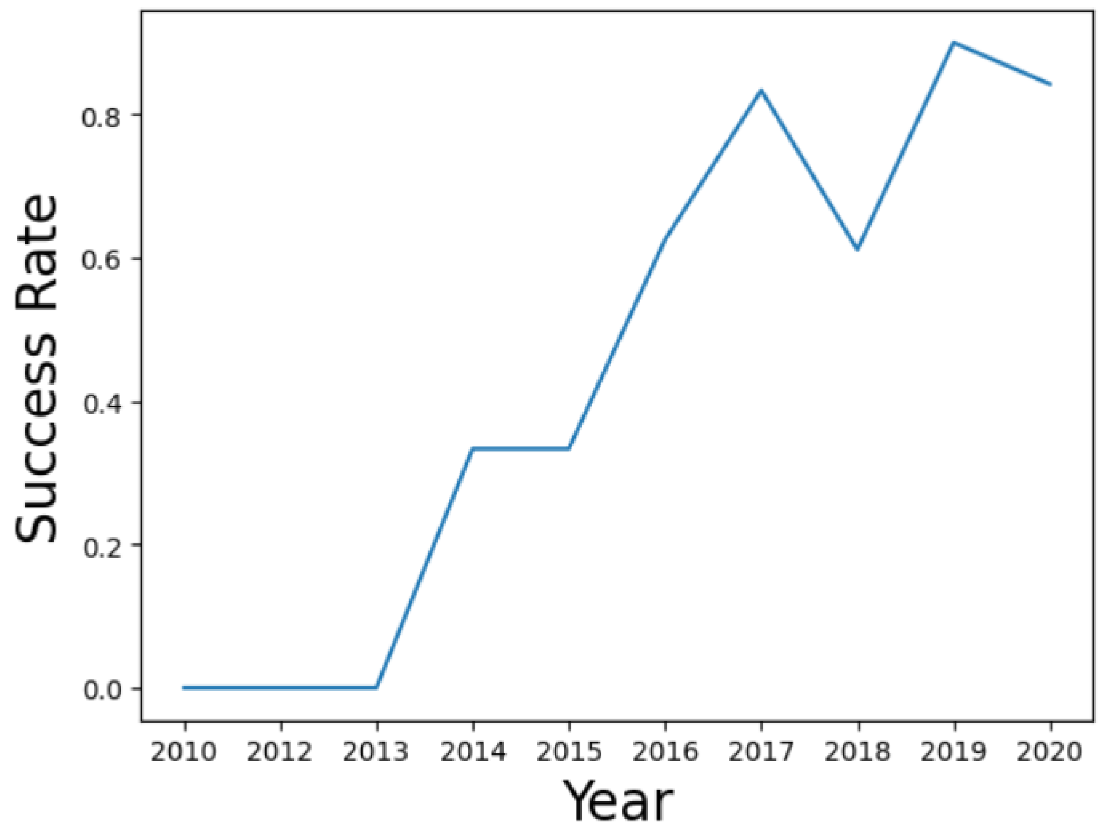


Figure 12

you can observe that the sucess rate since 2013 kept increasing till 2020

All Launch Site Names

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

```
[77] %sql select distinct Launch_Site from SPACEXNR
```

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

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

Figure 13

Launch Site Names Begin with 'KSC

```
%sql select * from SPACEXNR where Launch_Site like "KSC%" limit 5
```

Python

* sqlite:///my_data1.db
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	6:00:00	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

Figure 14

Total Payload Mass

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

```
[79]
%sql select sum(PAYLOAD_MASS__KG_) from SPACEXNR WHERE customer = 'NASA (CRS)'
```

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

sum(PAYLOAD_MASS__KG_)
45596

Figure 15

Average Payload Mass by F9 v1.1

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

```
%sql select avg(PAYLOAD_MASS__KG_) from SPACEXNR WHERE booster_version = "F9 v1.1"
```

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

avg(PAYLOAD_MASS__KG_)
2928.4

Figure 16

First Successful Ground Landing Date

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

Hint:Use min function

[81]

```
1  %sql select min(Date) from SPACEXNR WHERE Landing_Outcome = "Success (drone ship)"
```

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

min(Date)
2016-04-08

Figure 17

Successful Booster Names for Successful Ground Pad Landing with Payload between 4 and 6 tons

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

```
%sql select Booster_Version from SPACEXNR WHERE (( PAYLOAD_MASS__KG_ between 4000 and 6000) and (Landing_Outcome = "Success (ground pad)") )
```

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

Booster_Version
F9 FT B1032.1
F9 B4 B1040.1
F9 B4 B1043.1

Figure 18

Total Number of Successful and Failure Mission Outcomes

```
%sql select Mission_Outcome, count(Mission_Outcome) from SPACEXNR GROUP BY Mission_Outcome
```

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

Mission_Outcome	count(Mission_Outcome)
Failure (in flight)	1
Success	98
Success	1
Success (payload status unclear)	1

Figure 19

Boosters Carried Maximum Payload

```
%sql select Booster_Version , PAYLOAD_MASS__KG_ from SPACEXNR\
where  PAYLOAD_MASS__KG_ = (select max( PAYLOAD_MASS__KG_) from SPACEXNR)
```

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

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

Figure 20

Impossible requirement

This data scientist package course has been plagued by a chronic lack of data maintaining. This is the original requirement in the PowerPoint Capstone document – Skillset competence at its best. Please see the code for 2017 in my journal EDA with SQL

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

Rank Landing Outcomes Between 2010-06-04 and 2017-03-20 – Also for more recent data

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

```
%sql select Landing_Outcome, count(Landing_Outcome) from SPACEXNR \
where ((Date >= '2010-06-04' and Date <= '2017-03-20') and (Landing_Outcome="Failure (drone ship)" or Landing_Outcome="Success (ground pad)")) \
GROUP BY Landing_Outcome \
order by (Landing_Outcome) desc
```

Python

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

Landing_Outcome	count(Landing_Outcome)
Success (ground pad)	3
Failure (drone ship)	5

```
%sql select  Landing_Outcome,count(Landing_Outcome) from SPACEXNR \
where ((Date >= '2010-06-04' and Date <= '2020-12-31') and (Landing_Outcome="Failure (drone ship)" or Landing_Outcome="Success (ground pad)")) \
GROUP BY Landing_Outcome \
order by (Landing_Outcome) desc

```

Python

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

Landing_Outcome	count(Landing_Outcome)
Success (ground pad)	9
Failure (drone ship)	5

Figure 21

10.

11. LAUNCH SITES PROXIMITY ANALYSIS

11.1. Global Map of All launch sites' location

Here is a location of the three sites from the data set. (CA, Fl – 2 sites). The map was embedded with the equator and a location of another launch site (Guyana) was plotted. All launch sites are in the proximity of the coast, the site furthest to the Equator is the Ca site. Guyana space center is very close to the Equator

Figure 22

11.2. Launch outcomes for each launch center

Looking on the data it seems that the number of failures is concentrated in Space Launch complex 40 in Florida and at the Vandenberg AFB. The perspective is skewed, keep in mind that no data until 2020 is included in the data frame provided by Skils Network

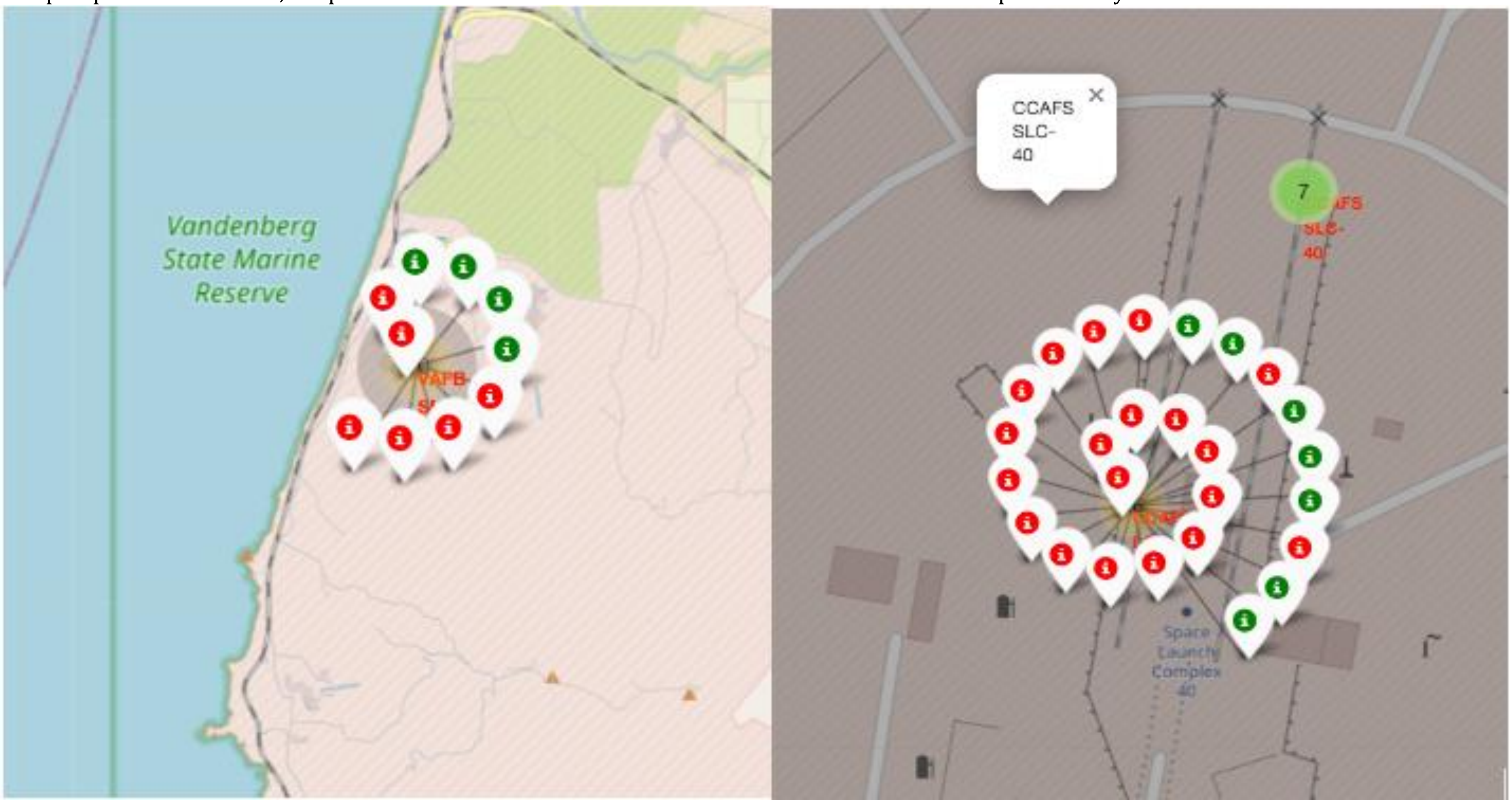
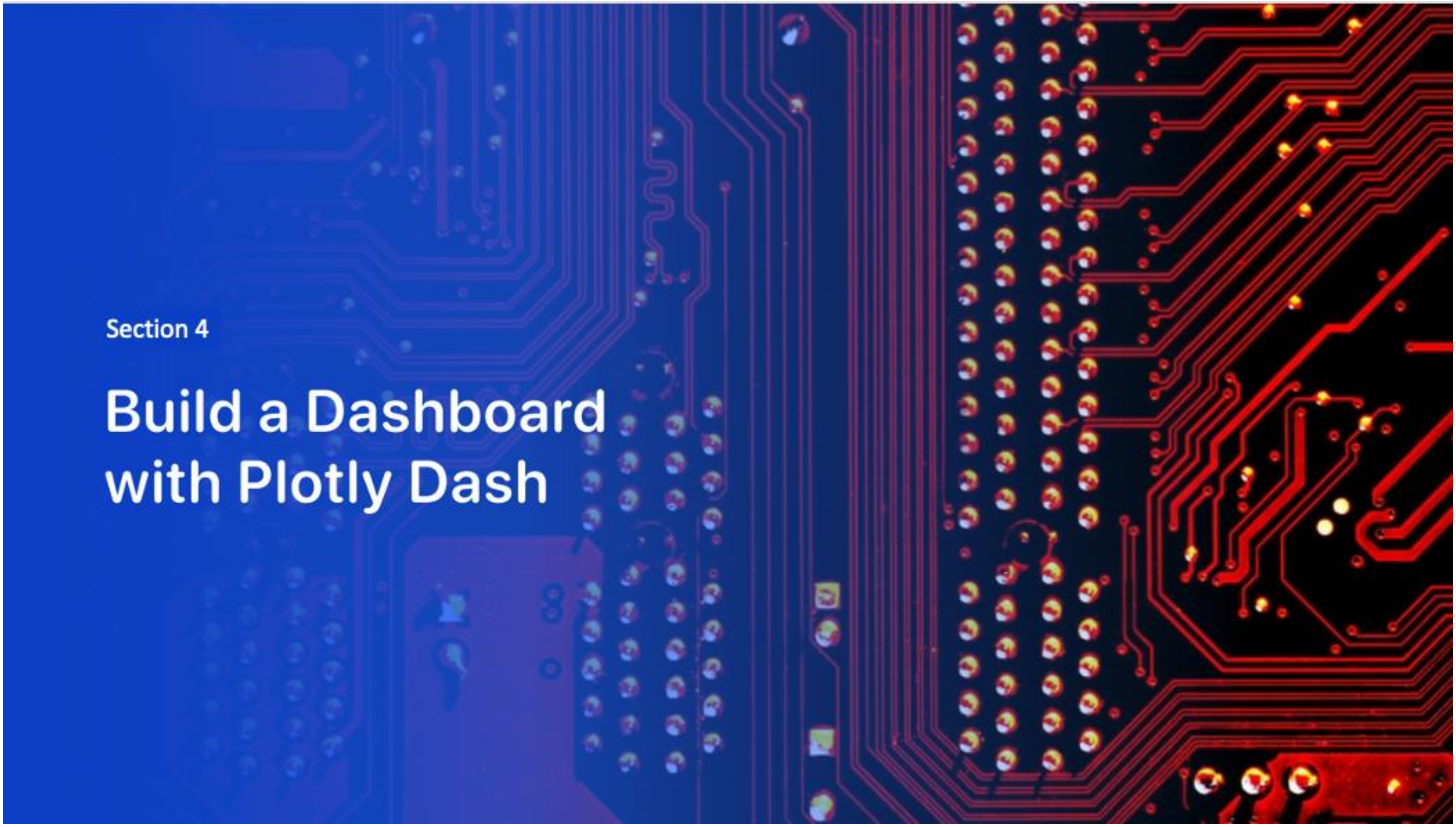


Figure 23

11.3. Vandenberg CA Site analysis

The coastline, highway and railway are very close to the launching site. This is shown in the top magnified map. The distance to the closest city, Lompoc CA from the launch site is 14.11 km (or less than 10 miles for those who use Imperial measuring units). The two images showing proximity are not at scale Given the proximity to the city and the fact that this site is the most distant from the Equator, payloads should be restricted in size, security should be provided when launches take place to prevent accidents and restrict access

Figure 24


Section 4
Build a Dashboard
with Plotly Dash

12. PLOTLY DASHBOARD

I have installed Dash on my system and I can run the dashboard locally. I advise you to do the same instead of using the virtual environment offered by skillsets

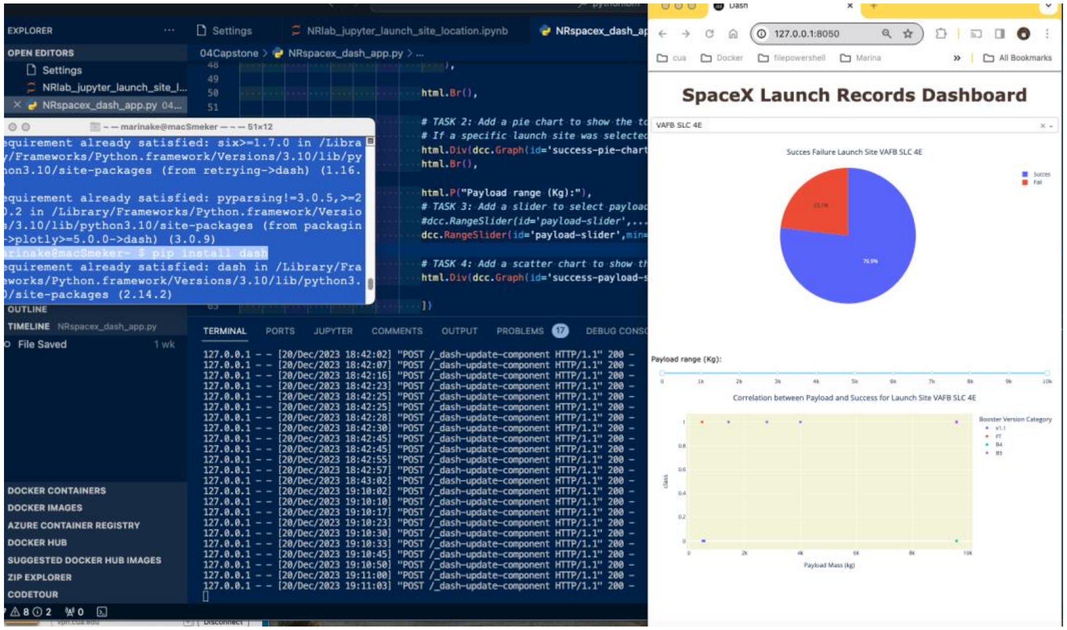


Figure 25

The dashboard displays a success/failure pie chart and a scatter graph that shows the correlation between payload and success. A slide button can adjust the payload for a range. Booster type is color coded. The data frame contains 87 records

12.1. Launch Records - Total Launches by Site Name

The payload is adjusted to a range between 2000 and 3000 kg. You can see that the pie chart shows the launches for the CCAFS SLC 40 site is 33. This is the site with most launches but the number of launches in the scatter plot below is limited by the slider

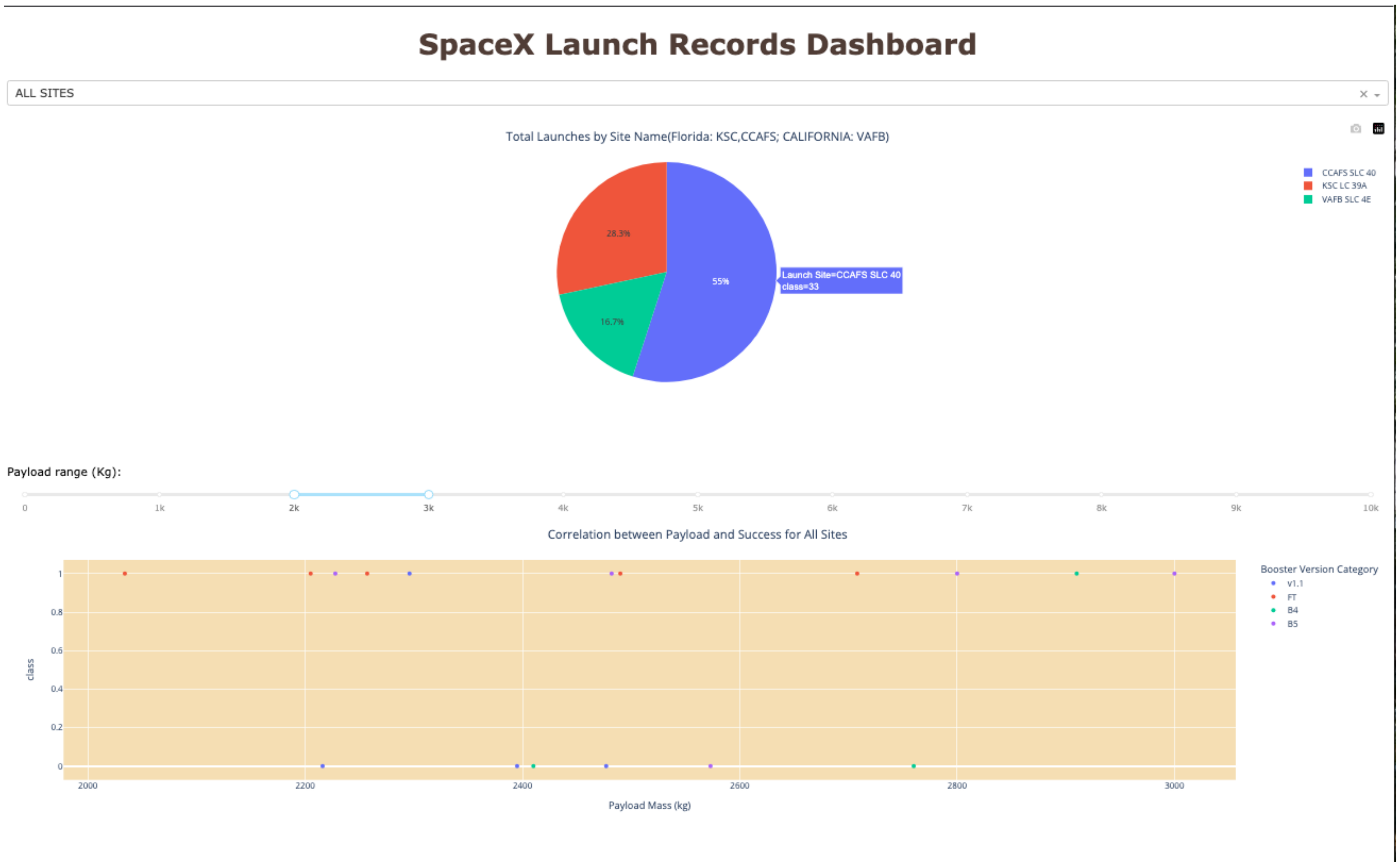


Figure 26

12.2. Launch Records Most Successful Site

The most successful launch site for launches is KSC LC 39A. There are 17 successful launches

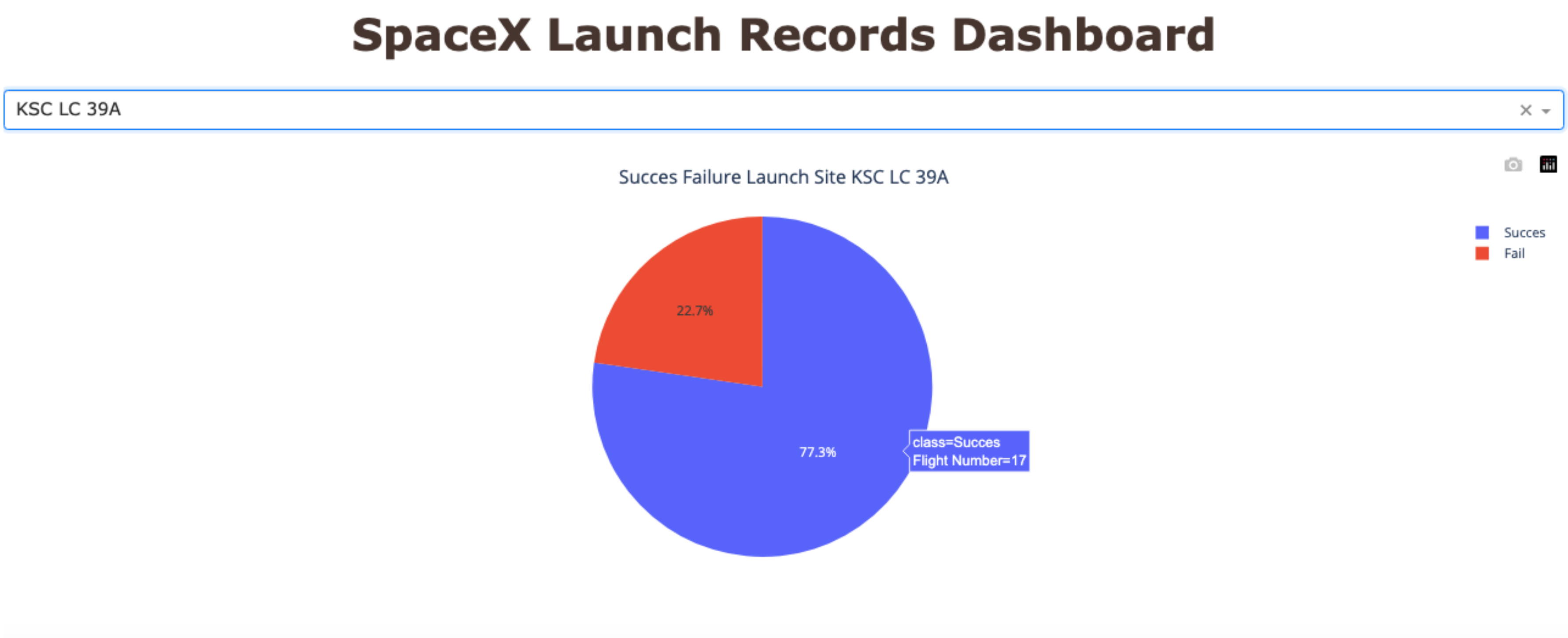


Figure 27

12.3. Payload vs Launch Outcome for all Sites

This is quite a crowded graph – we can notice most failures are due to early booster versions. We notice that FT and B5 boosters perform very well for a wide range of payloads

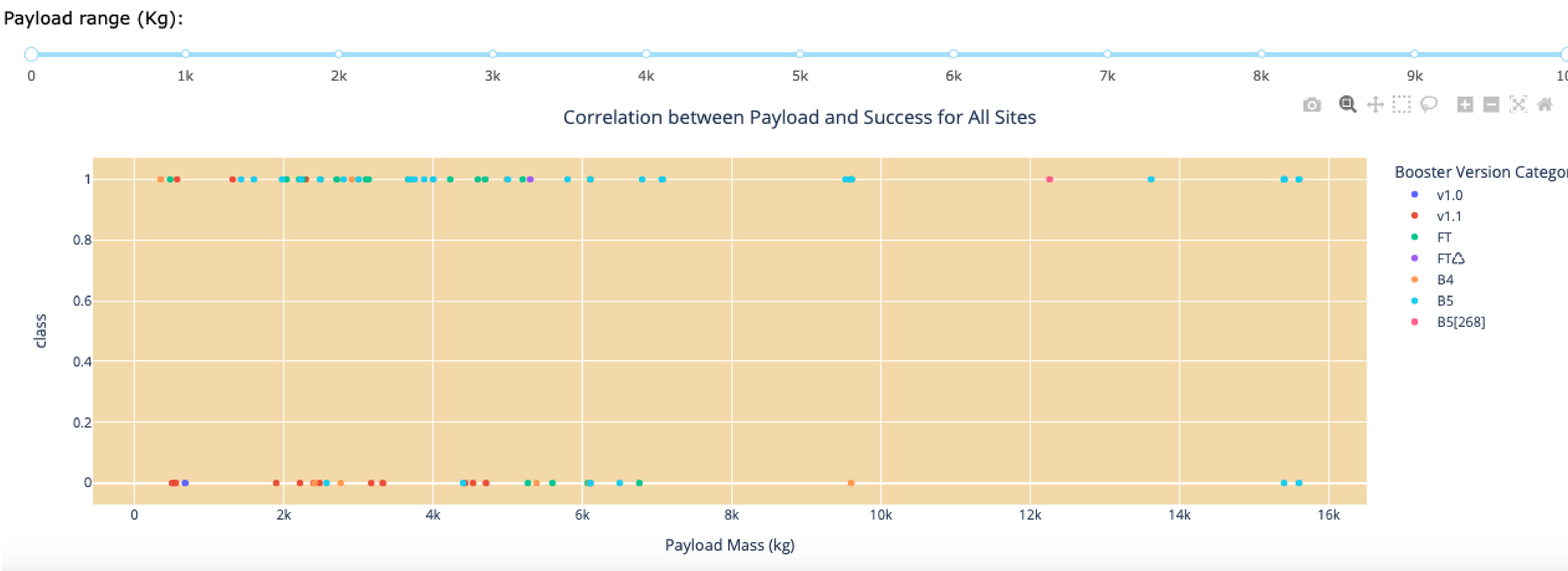


Figure 28

If we want to dig more, we can choose to focus on a certain booster type. As expected, version V.11 shows lots of failures. B4 booster are quit lame too, only 50% success rate is shown. Even for the most successful site, B4 has a poor record. But as we recall from the sql analysis B4 boosters were very successful in the 4k to 6k range. One explanation is that the data frame offered for sql analysis is not the same with the data frame w

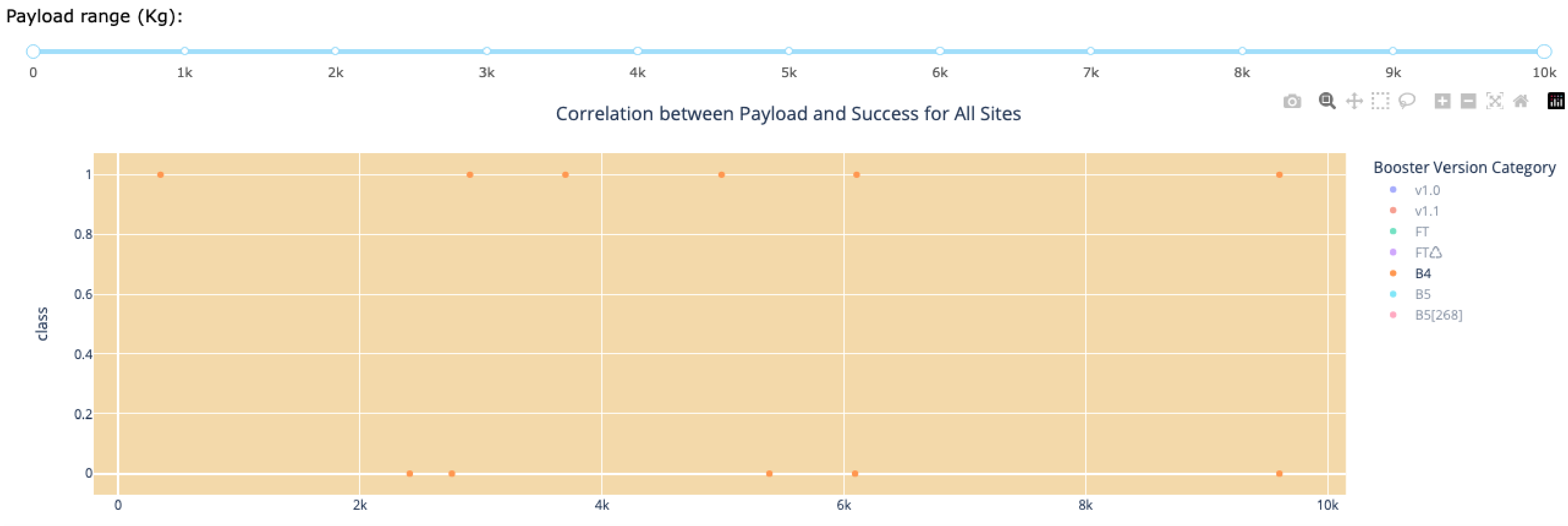


Figure 29


Section 5
Predictive Analysis
(Classification)

13. PREDICTIVE ANAYSIS – CLASSIFICATION

13.1. Accuracy of the Model

The four models and the accuracy for the test and train sets are summarized below. Logistic Regression. SVM and decision tree algorithms have similar highest accuracy for the test sample and decision tree has the highest train accuracy among the three methods. Node clustering should not be used based only by accuracy

	Classification Method Name	Test Accuracy	Train Accuracy
0	Logistic regression	0.833333	0.850000
1	supportVectorMachine	0.833333	0.864286
2	decisiontree	0.833333	0.892857
3	nodeclustering	0.777778	0.864286

	Classification Method Name	Best Params
0	Logistic regression	{'C': 0.1, 'penalty': 'l2', 'solver': 'lbfgs'}
1	supportVectorMachine	{'C': 1.0, 'gamma': 0.03162277660168379, 'kernel': 'sigmoid'}
2	decisiontree	{'criterion':'gini','max_depth': 2, 'max_features': 'log2', 'min_samples_leaf': 1, 'min_samples_split': 2, 'splitter': 'best'}
3	nodeclustering	{'algorithm': 'auto', 'n_neighbors': 4, 'p': 1}

Let us see what the confusion matrix shows

13.

Fo

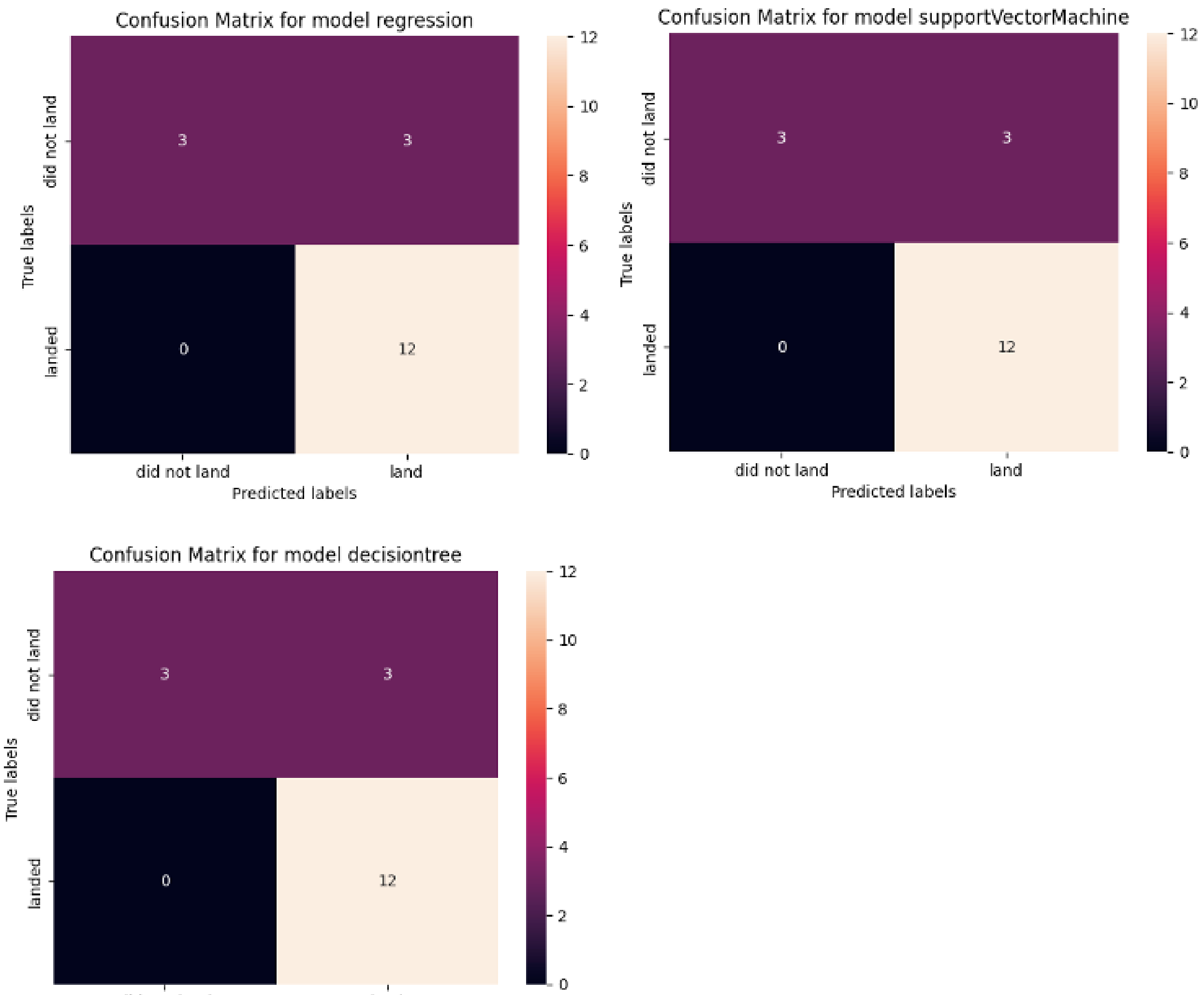


Figure 30

For the clustering model we notice that there will be both successful landings predicted as failed and failed landings predicted as successful . We get both false negatives and false positives

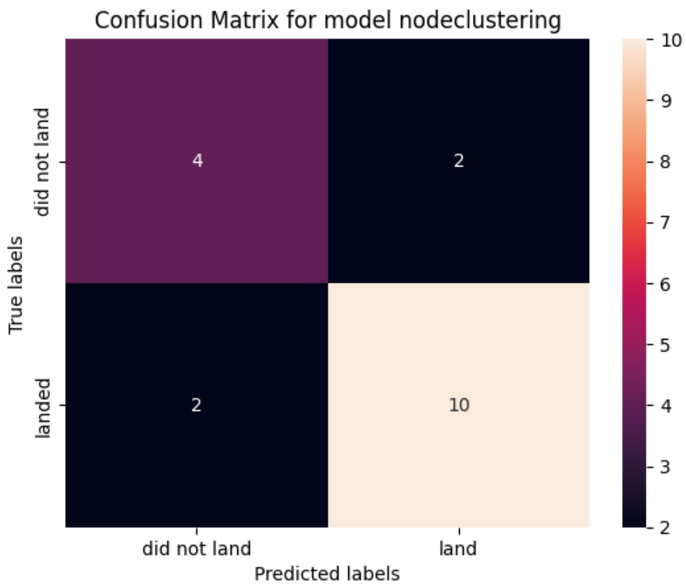


Figure 31