IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

Gulcin Uluyazi

17.06.2022

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 Interactive Map with Folium
- Building a Dashboard with Plotly Dash
- Predictive Analysis

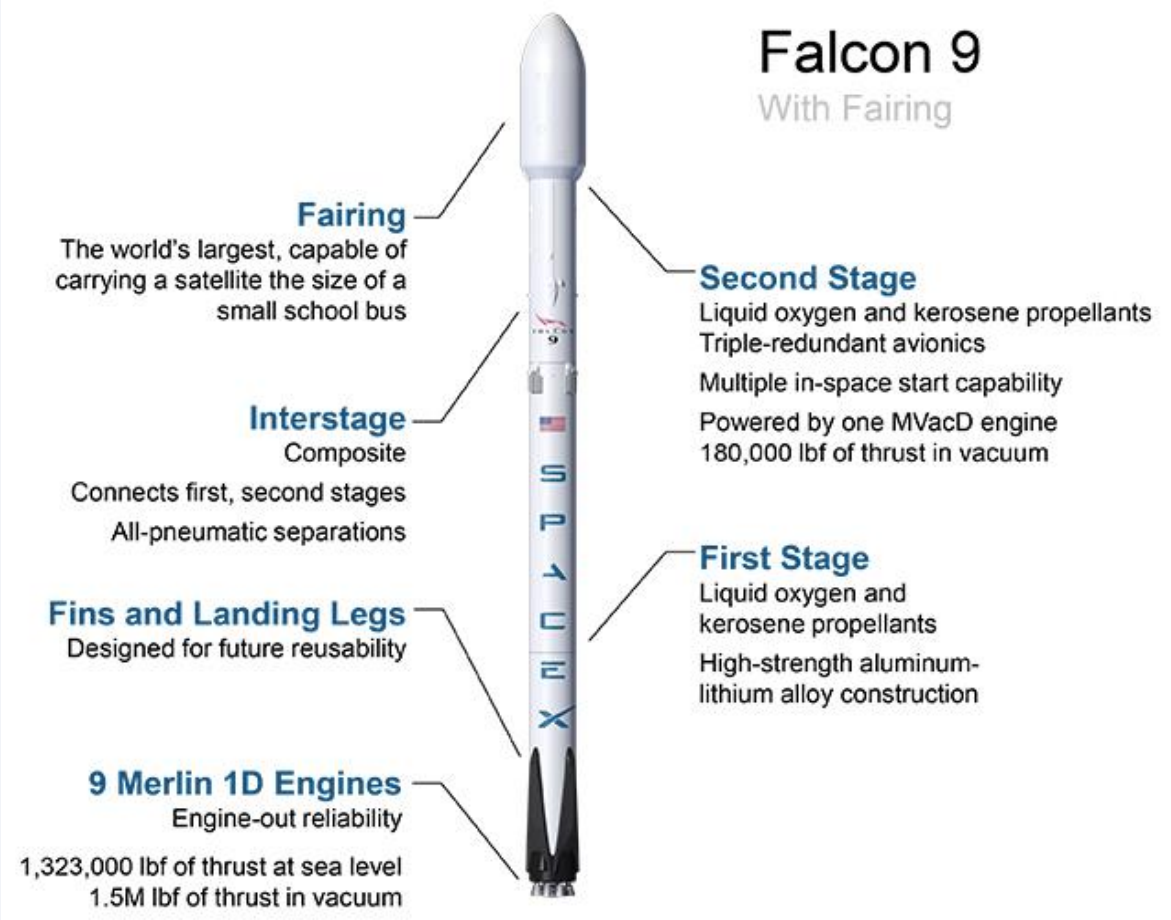
Summary of all results

- EDA results
- Interactive Analytics Graphs
- Predictive Analysis Findings

Introduction

Project background and context

- There are several companies which trying to make space travels affordable for everyone, such as Virgin Galactic, Rocket Lab, Blue Origin and SpaceX.
- SpaceX launches rockets (Falcon 9) relatively inexpensive then others.
- SpaceX advertises Falcon 9 rocket launches on its website with a cost of 62 million dollars; other providers cost upwards of 165 million dollars each, much of the savings is because SpaceX can reuse the first stage.
- Therefore, if we can determine if the first stage will land, we can determine the cost of a launch. This information can be used if an alternate company wants to bid against SpaceX for a rocket launch.


Falcon 9
With Fairing
Fairing
The world's largest, capable of carrying a satellite the size of a small school bus
Interstage
Composite
Connects first, second stages
All-pneumatic separations
Fins and Landing Legs
Designed for future reusability
9 Merlin 1D Engines
Engine-out reliability
1,323,000 lbf of thrust at sea level
1.5M lbf of thrust in vacuum
Second Stage
Liquid oxygen and kerosene propellants
Triple-redundant avionics
Multiple in-space start capability
Powered by one MVacD engine
180,000 lbf of thrust in vacuum
First Stage
Liquid oxygen and kerosene propellants
High-strength aluminum-lithium alloy construction
SPACEX

Problems you want to find answers

- What is the price of each launch?
- Will first stage of rockets reuse by SpaceX? What is the estimation of Machine Learning Models about it?

Section 1

Methodology

Methodology

Executive Summary

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

Data Collection

- SpaceX launch data that is gathered from an API, specifically the SpaceX REST API.
- This API will give us data about launches, including information about the rocket used, payload delivered, launch specifications, landing specifications, and landing outcome.
- Each contents has different endpoints such as

  api.spacexdata.com/v4/capsules

  api.spacexdata.com/v4/cores

- We will use this api.spacexdata.com/v4/launches/past
- We get data using a get request from this api in the form of JSON.
- Then convert JSON to a DATAFRAME

1

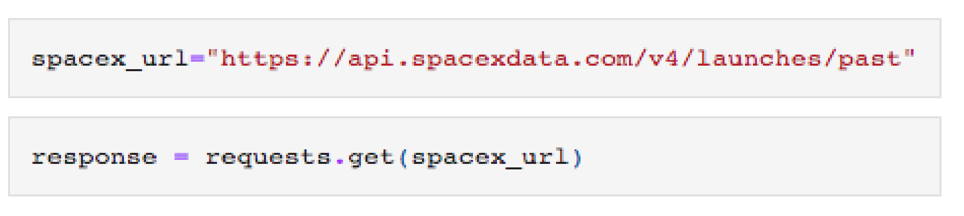
- Another popular data source for obtaining Falcon 9 Launch data is web scraping related Wiki pages.
- We get data by using Python BeautifulSoup package to web scrape some HTML tables that contain valuable Falcon 9 launch records.
- Then we parse the data from those tables and convert them into a Pandas data frame
- Finally we transform this raw data into a clean dataset which provides meaningful data.

2

Data Collection – SpaceX API

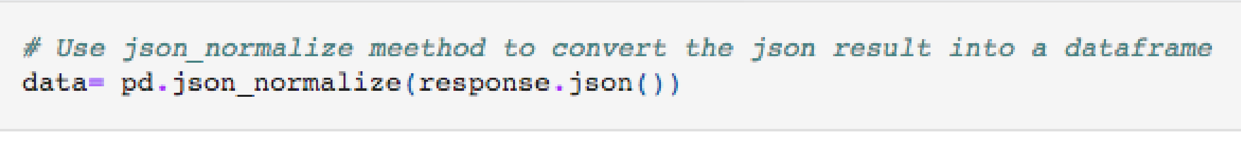
- https://github.com/GulcinBU/GulcinBU/blob/main/jupyter-labs-spacex-data-collection-api.ipynb

1. Request and parse the SpaceX launch data , converting JSON file



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

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



```
# Use json_normalize meethod to convert the json result into a dataframe
data= pd.json_normalize(response.json())
```

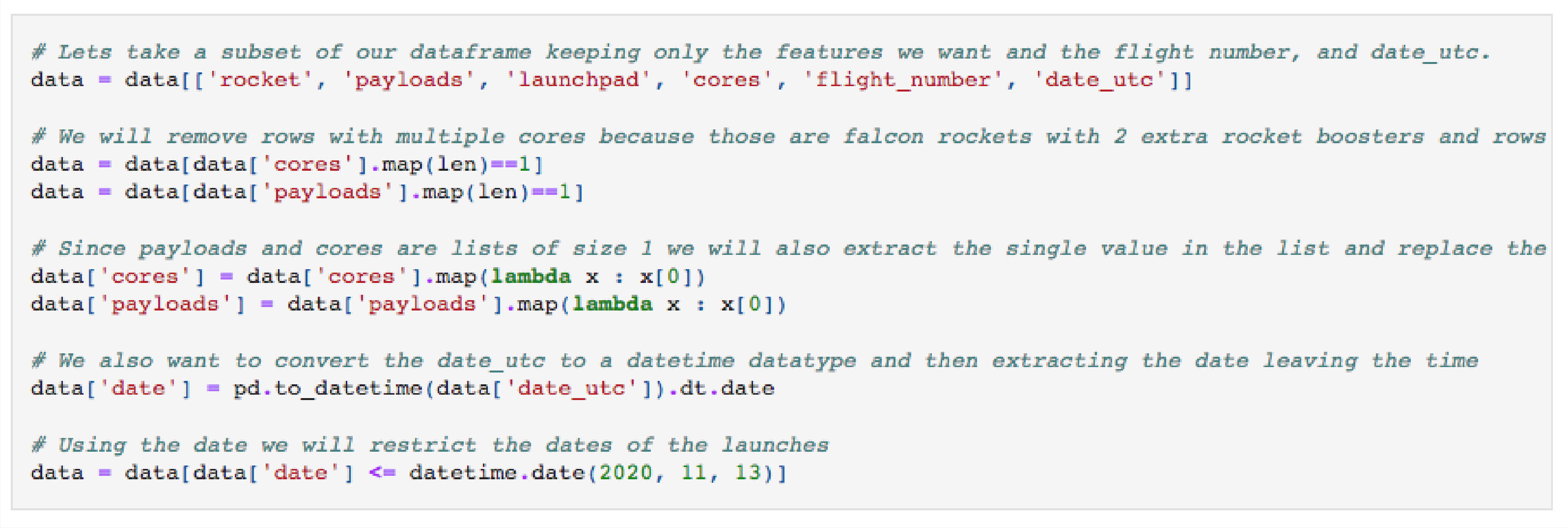
2. Getting subset of DF only for columns:

rocket,

payloads,

launchpad

cores.



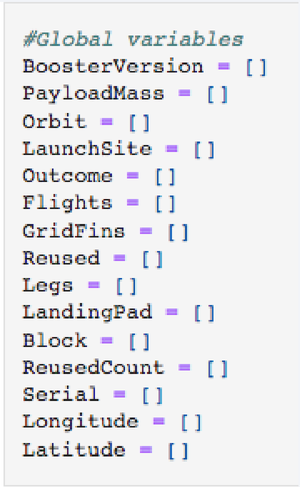
```
# Lets take a subset of our dataframe keeping only the features we want and the flight number, and date_utc.
data = data[['rocket', 'payloads', 'launchpad', 'cores', 'flight_number', 'date_utc']]

# We will remove rows with multiple cores because those are falcon rockets with 2 extra rocket boosters and rows
data = data[data['cores'].map(len)==1]
data = data[data['payloads'].map(len)==1]

# Since payloads and cores are lists of size 1 we will also extract the single value in the list and replace the
data['cores'] = data['cores'].map(lambda x : x[0])
data['payloads'] = data['payloads'].map(lambda x : x[0])

# We also want to convert the date_utc to a datetime datatype and then extracting the date leaving the time
data['date'] = pd.to_datetime(data['date_utc']).dt.date

# Using the date we will restrict the dates of the launches
data = data[data['date'] <= datetime.date(2020, 11, 13)]
```

Data Collection – SpaceX API

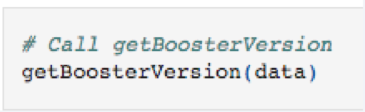
3. Getting some stored data

for a new DF

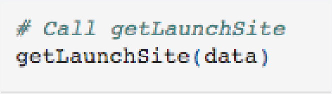
by these functions



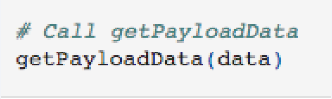
```
#Global variables
BoosterVersion = []
PayloadMass = []
Orbit = []
LaunchSite = []
Outcome = []
Flights = []
GridFins = []
Reused = []
Legs = []
LandingPad = []
Block = []
ReusedCount = []
Serial = []
Longitude = []
Latitude = []
```



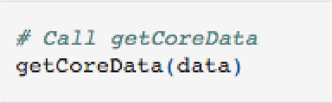
```
# Call getBoosterVersion
getBoosterVersion(data)
```



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



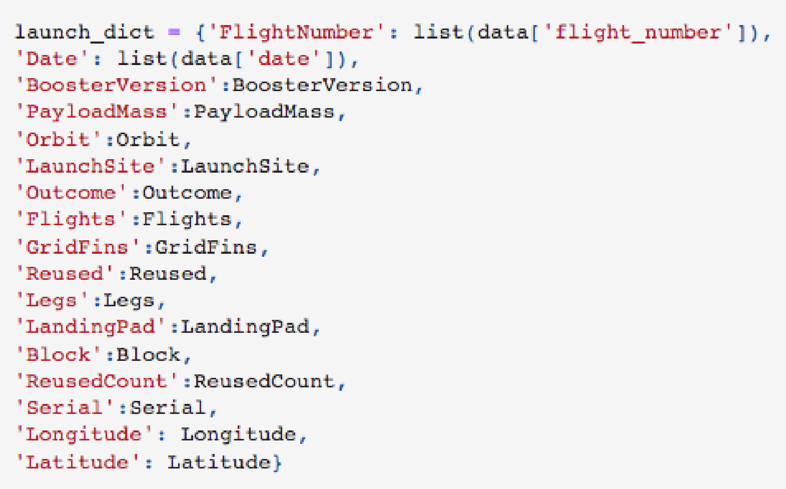
```
# Call getPayloadData
getPayloadData(data)
```



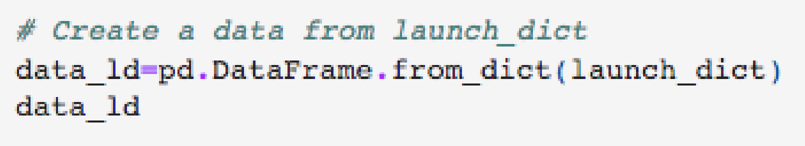
```
# Call getCoreData
getCoreData(data)
```

4.Constructing our dataset using

the data we have obtained



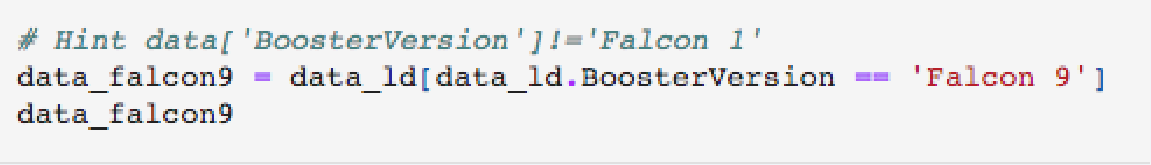
```
launch_dict = {'FlightNumber': list(data['flight_number']),
'Date': list(data['date']),
'BoosterVersion':BoosterVersion,
'PayloadMass':PayloadMass,
'Orbit':Orbit,
'LaunchSite':LaunchSite,
'Outcome':Outcome,
'Flights':Flights,
'GridFins':GridFins,
'Reused':Reused,
'Legs':Legs,
'LandingPad':LandingPad,
'Block':Block,
'ReusedCount':ReusedCount,
'Serial':Serial,
'Longitude': Longitude,
'Latitude': Latitude}
```



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

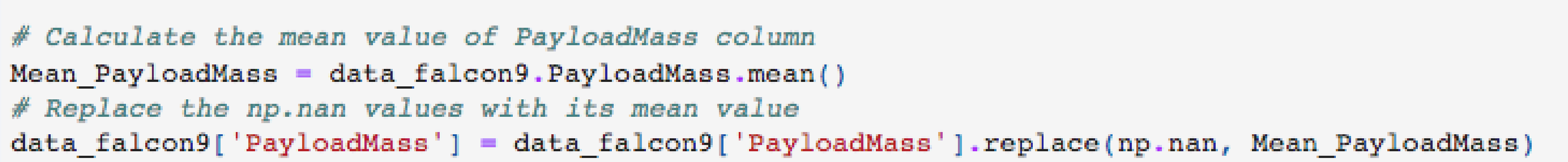
Data Collection – SpaceX API

5.Filter the dataframe to only include Falcon 9 launches

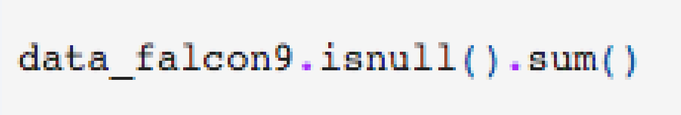


```
# Hint data['BoosterVersion']!='Falcon 1'
data_falcon9 = data_ld[data_ld.BoosterVersion == 'Falcon 9']
data_falcon9
```

6. Dealing with missing values

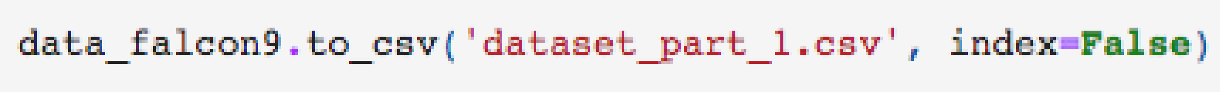


```
# Calculate the mean value of PayloadMass column
Mean_PayloadMass = data_falcon9.PayloadMass.mean()
# Replace the np.nan values with its mean value
data_falcon9['PayloadMass'] = data_falcon9['PayloadMass'].replace(np.nan, Mean_PayloadMass)
```



```
data_falcon9.isnull().sum()
```

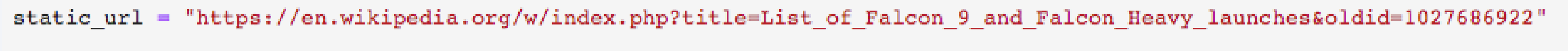
7. Exporting DF to CSV file



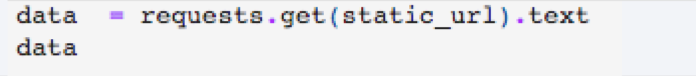
```
data_falcon9.to_csv('dataset_part_1.csv', index=False)
```

Data Collection - Scraping

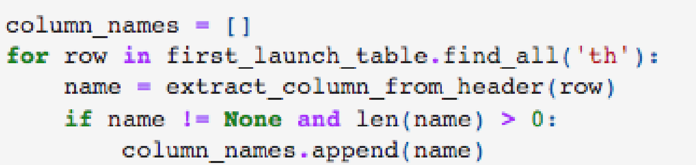
- https://github.com/GulcinBU/GulcinBU/blob/main/jupyter-labs-webscraping.ipynb



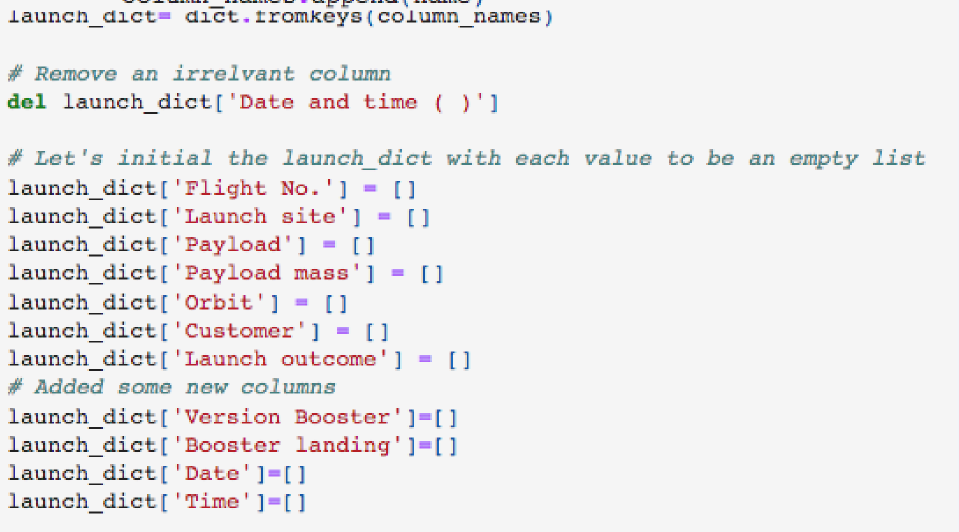
```
static_url = "https://en.wikipedia.org/w/index.php?title=List_of_Falcon_9_and_Falcon_Heavy_launches&oldid=1027686922"
```



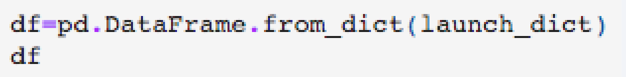
```
data  = requests.get(static_url).text
data
```



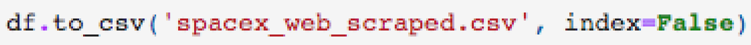
```
column_names = []
for row in first_launch_table.find_all('th'):
    name = extract_column_from_header(row)
    if name != None and len(name) > 0:
        column_names.append(name)
```



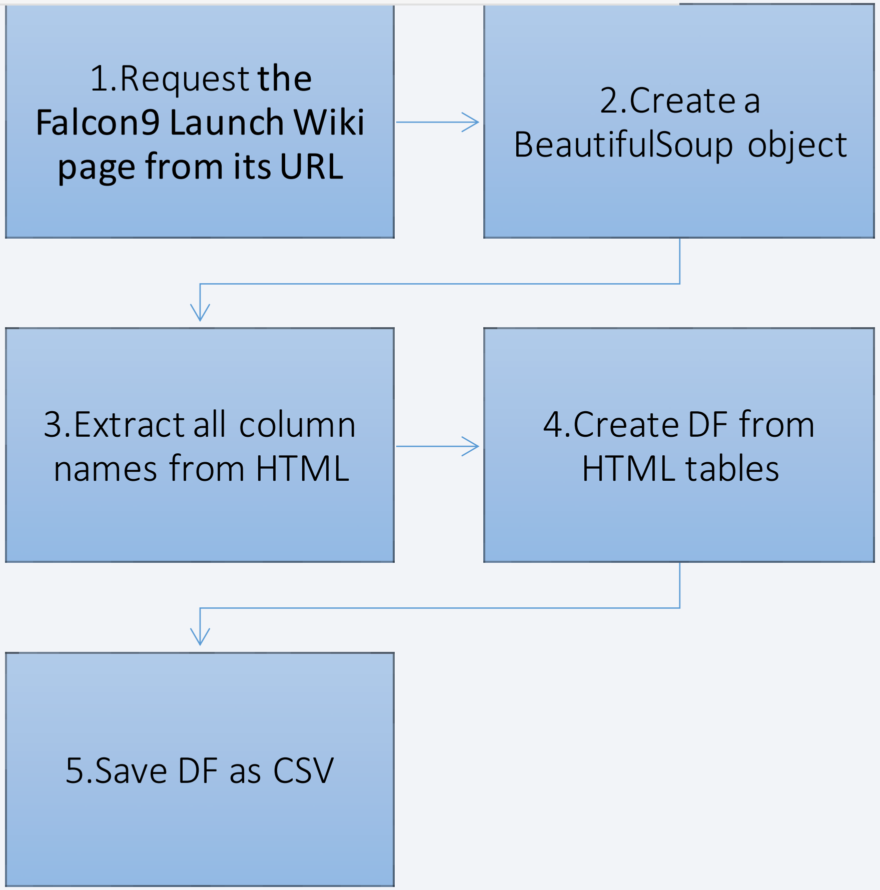
```
launch_dict= dict.fromkeys(column_names)

# Remove an irrelvant column
del launch_dict['Date and time ( )']

# Let's initial the launch_dict with each value to be an empty list
launch_dict['Flight No.'] = []
launch_dict['Launch site'] = []
launch_dict['Payload'] = []
launch_dict['Payload mass'] = []
launch_dict['Orbit'] = []
launch_dict['Customer'] = []
launch_dict['Launch outcome'] = []
# Added some new columns
launch_dict['Version Booster']=[]
launch_dict['Booster landing']=[]
launch_dict['Date']=[]
launch_dict['Time']=[]
```



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



```
df.to_csv('spacex_web_scraped.csv', index=False)
```



1.Request the Falcon9 Launch Wiki page from its URL

2.Create a BeautifulSoup object

3.Extract all column names from HTML

4.Create DF from HTML tables

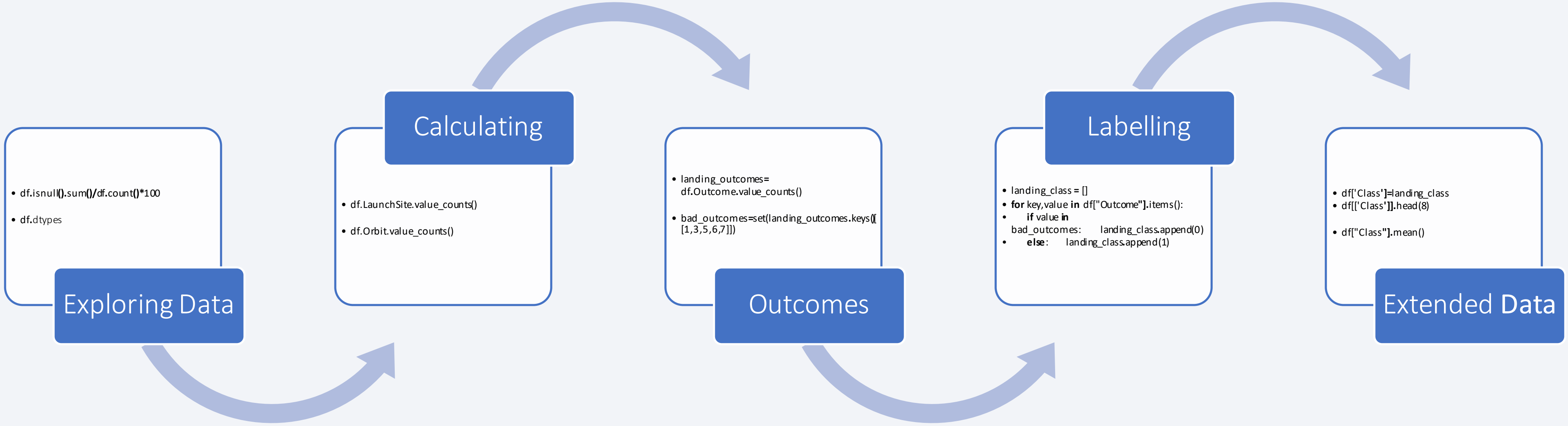
5.Save DF as CSV

Data Wrangling

https://github.com/GulcinBU/GulcinBU/blob/main/Capstone%20Data%20Wrangling%20W1-3.ipynb

- Our Aim: predicting Space X Falcon 9 First Stage Landing.

1. Finding some patterns in the data by EDA (Exploratory Data Analysis)
2. Assigning labels [0,1] for outcomes to make further trainings of data.



```
• df.isnull().sum()/df.count()*100
• df.dtypes
```

```
• df.LaunchSite.value_counts()
• df.Orbit.value_counts()
```

```
• landing_outcomes=
  df.Outcome.value_counts()
• bad_outcomes=set(landing_outcomes.keys()[
  [1,3,5,6,7]])
```

```
• landing_class = []
• for key,value in df["Outcome"].items():
•      if value in
  bad_outcomes:      landing_class.append(0)
•      else:      landing_class.append(1)
```

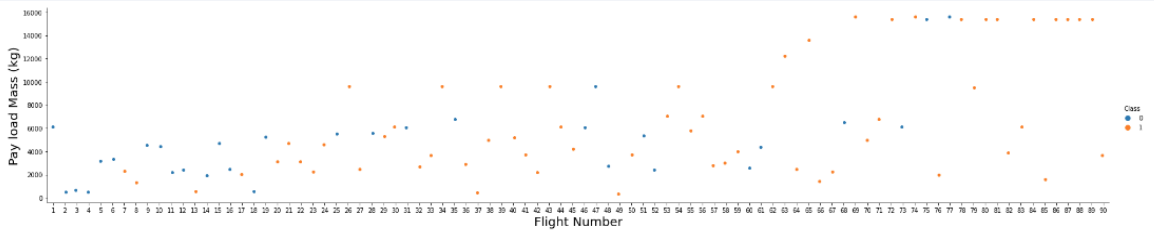
```
• df['Class']=landing_class
• df[['Class']].head(8)

• df["Class"].mean()
```

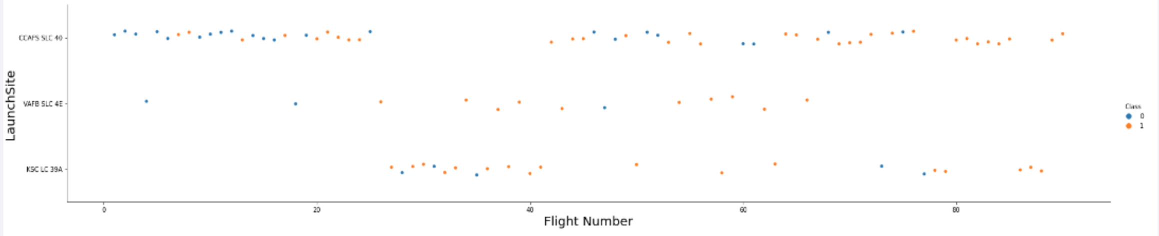
EDA with Data Visualization

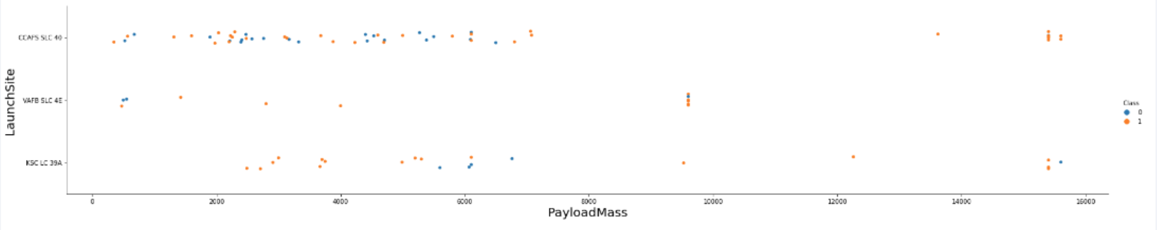
- Scatter Plots for [FlightNumber,PayloadMass], [FlightNumber,LaunchSite], [PayloadMass,LaunchSite],[FlightNumber,Orbit], [PayloadMass,Orbit]

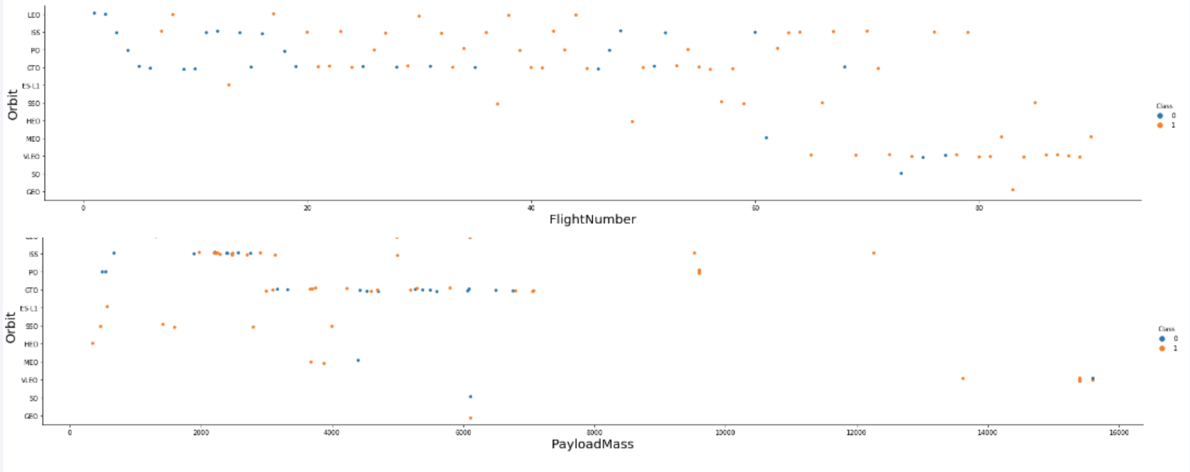
- Bar Chart for the sucess rate of each orbit

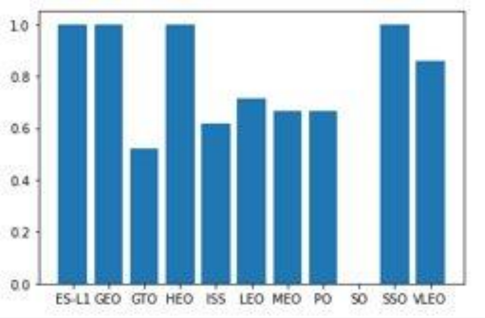
- A line chart for yearly average success rate

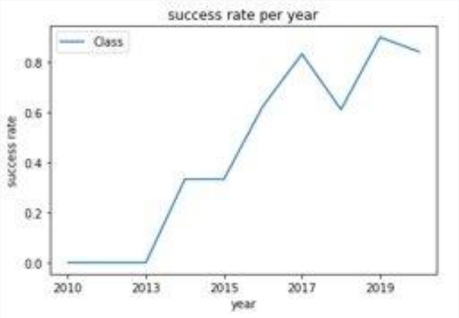












https://github.com/GulcinBU/GulcinBU/blob/main/jupyter-labs-eda-dataviz.ipynb

EDA with SQL

Used SQL queries to answer following questions:

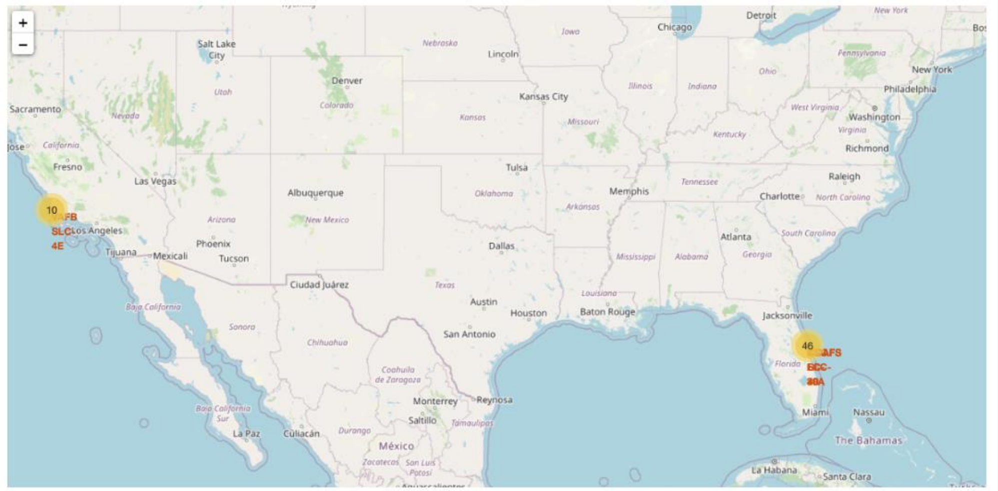
- Display the names of the unique launch sites in the space mission
- Display 5 records where launch sites begin with the string 'CCA'
- Display the total payload mass carried by boosters launched by NASA (CRS)
- Display average payload mass carried by booster version F9 v1.1
- List the date when the 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 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.
- 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 the date 04-06-2010 and 20-03-2017 in descending order.

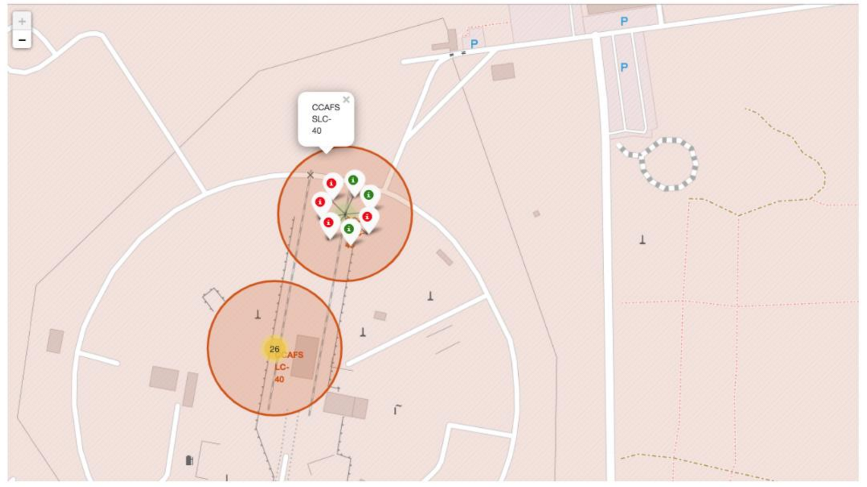
https://github.com/GulcinBU/GulcinBU/blob/main/jupyter-labs-eda-sql-coursera_sqllite.ipynb

Build an Interactive Map with Folium

- The launch success rate may depend on the location and proximities of a launch site. We could discover some of the factors by analyzing the existing launch site locations.

- We use Folium which is an interactive mapping library in Python.
- We add markers to mark all launch sites and the success/failed launches on a map.
- We add circles for highlighted circle area with a text label on a specific coordinate.
- We add lines to calculate the distances between a launch site to its proximities.



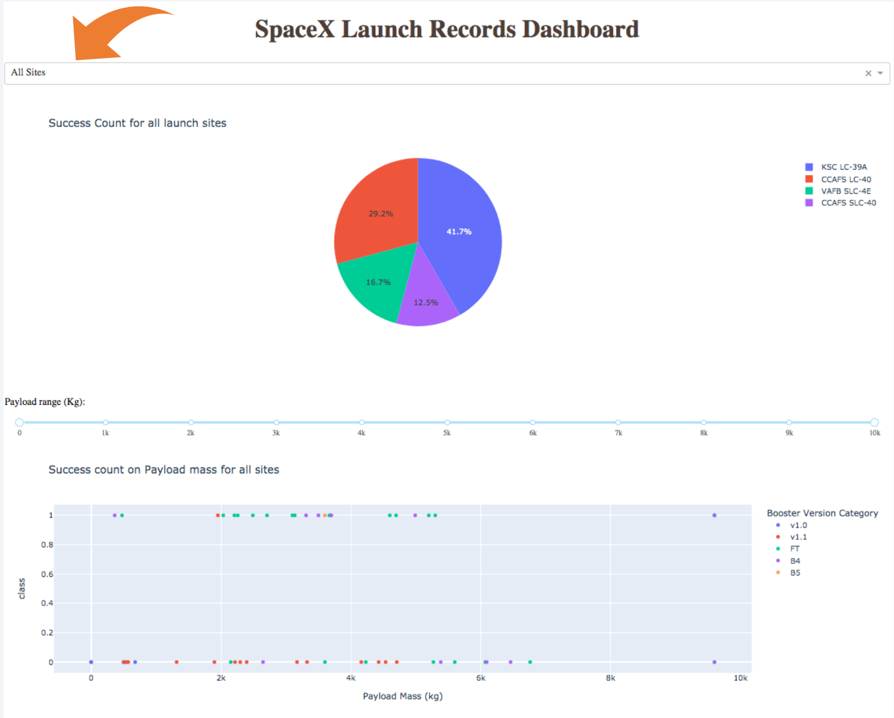


https://github.com/GulcinBU/GulcinBU/blob/main/Interactive%20Visual%20Analytics%20with%20Folium%20lab.ipynb

Build a Dashboard with Plotly Dash

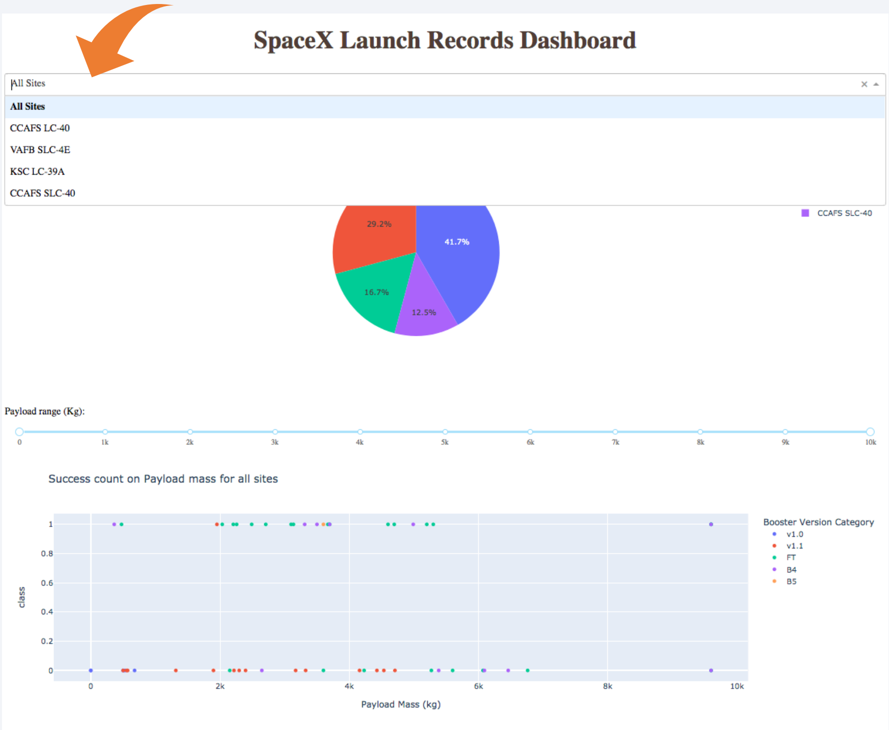
- We use a dashboard to show success-pie-chart and success-payload-scatter-chart based on selected site dropdown.

- We can see both graphs for total launch sites and each sites separately by using PLOTLY DASH



SpaceX Launch Records Dashboard

All Sites

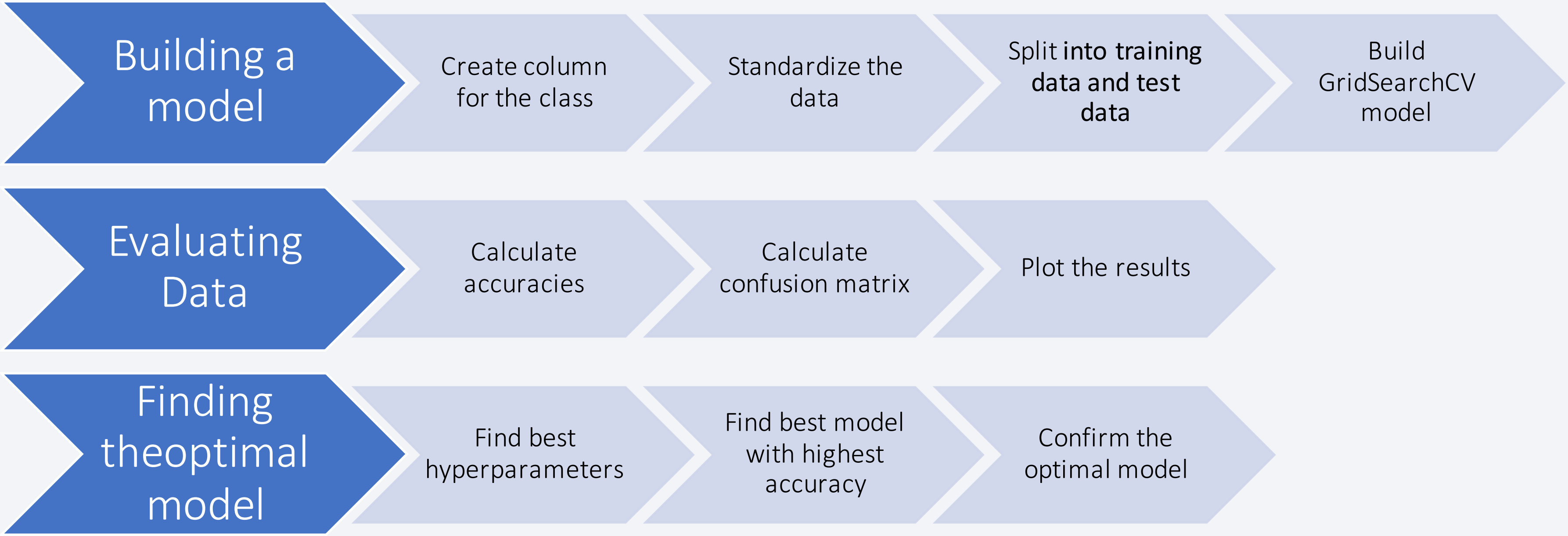


SpaceX Launch Records Dashboard

All Sites

- https://github.com/GulcinBU/GulcinBU/blob/main/spacex_dash_app.py

Predictive Analysis (Classification)



Building a model

Create column for the class

Standardize the data

Split into training data and test data

Build GridSearchCV model

Evaluating Data

Calculate accuracies

Calculate confusion matrix

Plot the results

Finding theoptimal model

Find best hyperparameters

Find best model with highest accuracy

Confirm the optimal model

- https://github.com/GulcinBU/GulcinBU/blob/main/SpaceX_Machine%20Learning%20Prediction_Part_5.ipynb

Results

Exploratory data analysis results:

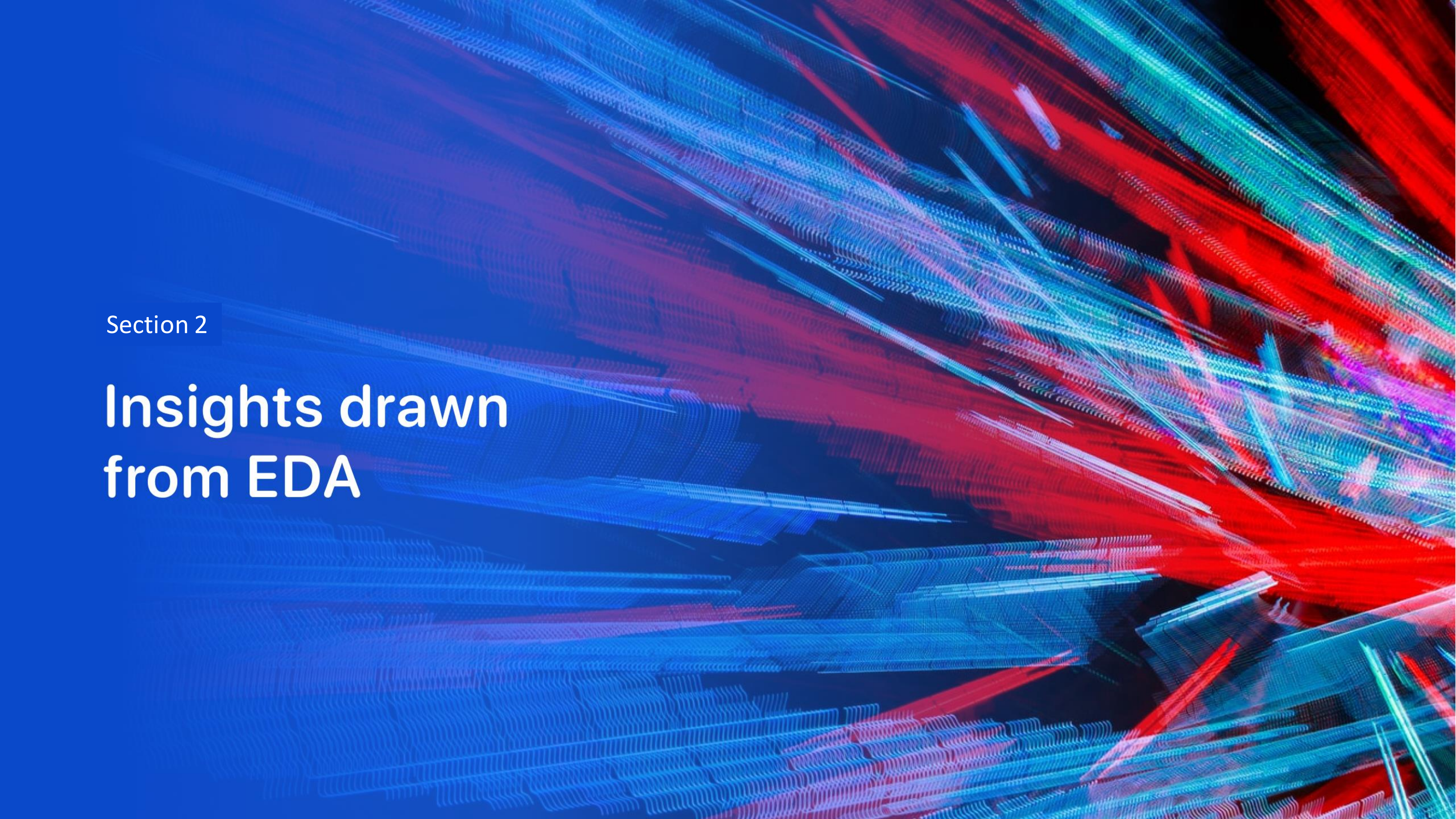
- Average launch success kept increasing since 2013 till 2020.
- Low weighted payloads are more successful than heavier payloads.
- ES-L1, GEO,HEO,SSO,VLEO orbits have highest success rate.
- Flight Number vs. Launch Site scatter point plot says that more there is no relationship between high flight number and success rate.

Interactive analytics results

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

Predictive analysis results

- Decision Tree model gives the best accuracy with 0.8333

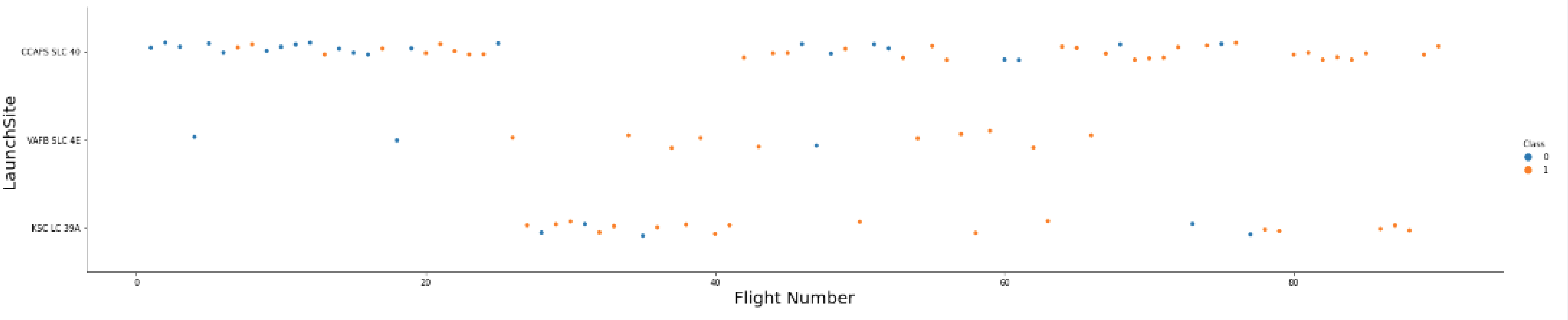


Section 2

Insights drawn from EDA

Flight Number vs. Launch Site

- Flight Number vs. Launch Site scatter point plot says us that ***number of CCAF5 SLC 40 is the most amongst others but rate of success is lower than others. We see that VAFB SLC 4E has very low number of flights but mostly successful (~77 %) and KSC LC 39A has similar success but more flight numbers.***

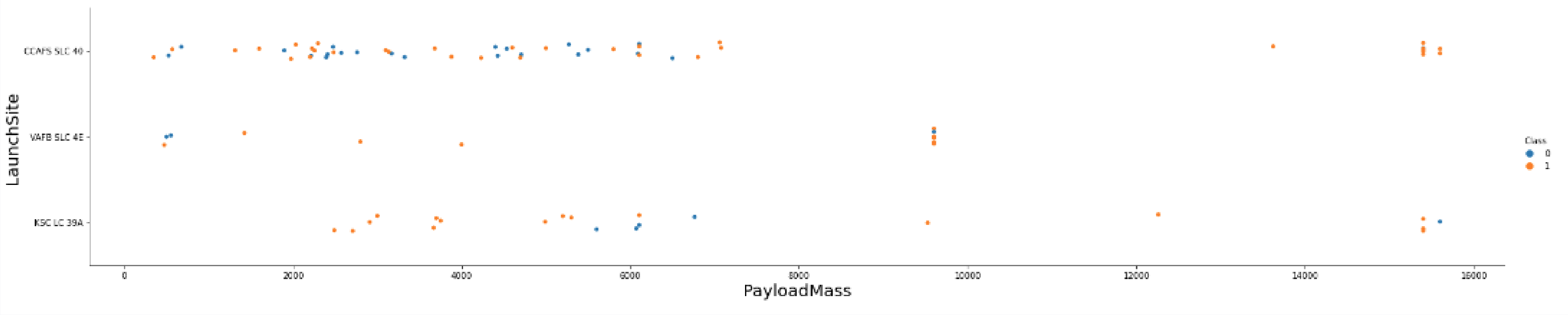


Blue dots: false landing

Orange dots: successful landing

Payload vs. Launch Site

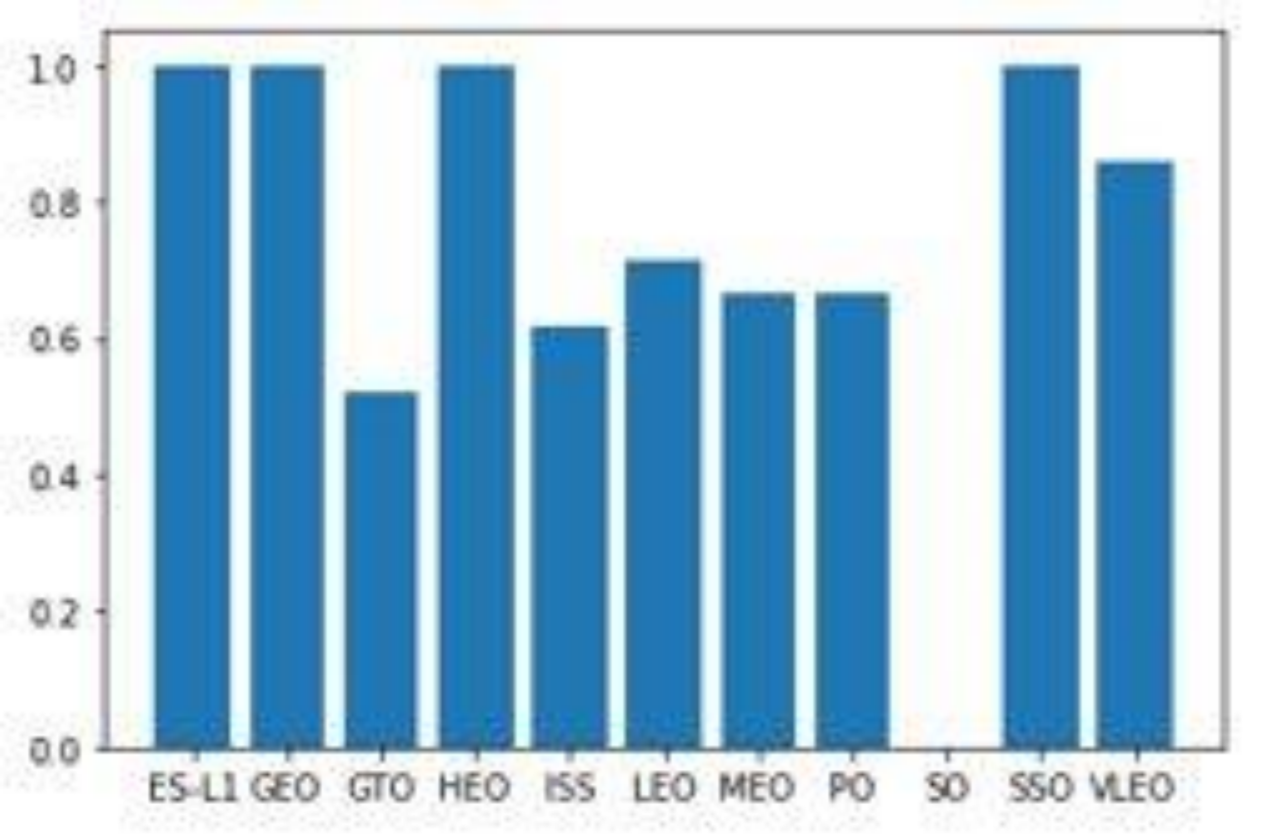
- The plot showing relationship between launch sites and their payload mass make us find ***for the VAFB-SLC launch site there are no rockets launched for heavy payload mass(greater than 10000).***



Blue dots: false landing   Orange dots: successful landing

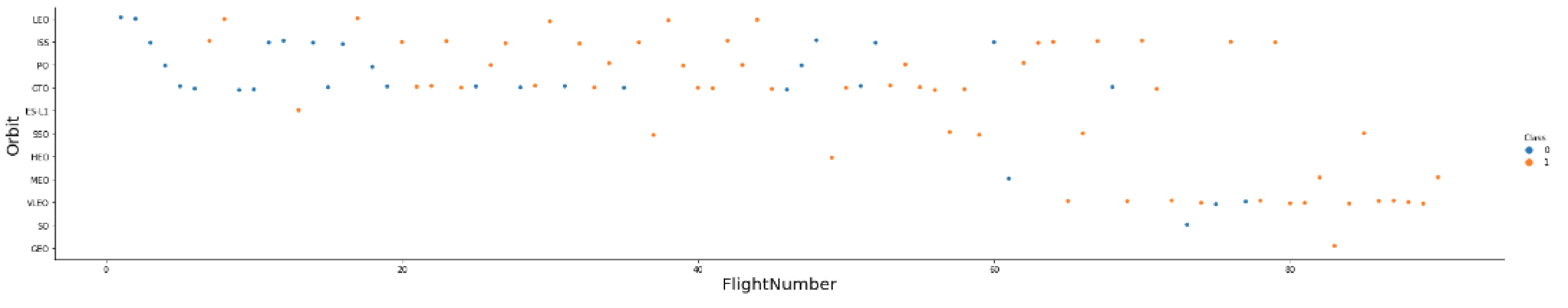
Success Rate vs. Orbit Type

- Success rate and orbit type bar chart gives us the result that ***ES-L1, GEO,HEO,SSO orbits have highest success rate with 100%.***
- ***VLEO is also have a high rate with ~90%.***
- ***GTO has the lowest rate which is 50%.***



Flight Number vs. Orbit Type

- Flight Number and Orbit type plot indicates that ***for 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.***

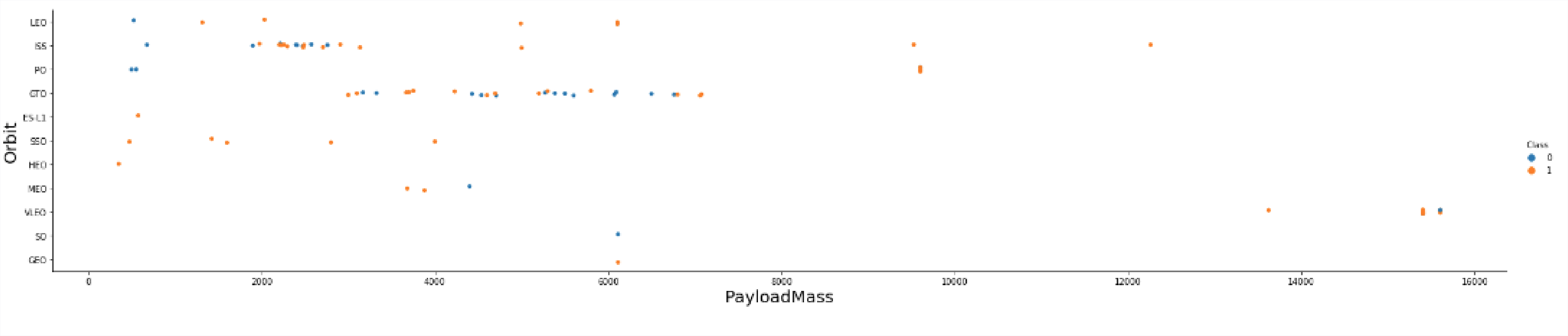


Blue dots: false landing

Orange dots: successful landing

Payload vs. Orbit Type

- Payload and Orbit type plot says that ***with heavy payloads the successful landing or positive landing rate are more for Polar, LEO and ISS.***

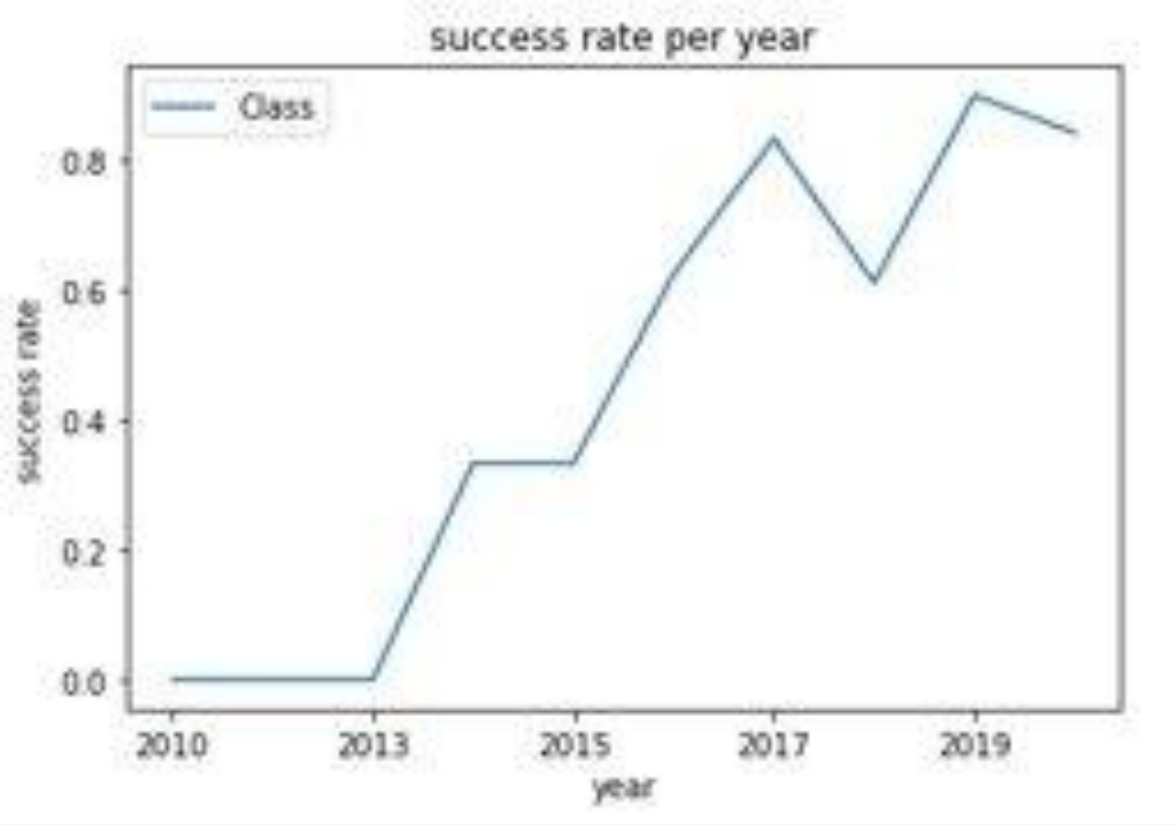


Blue dots: false landing

Orange dots: successful landing

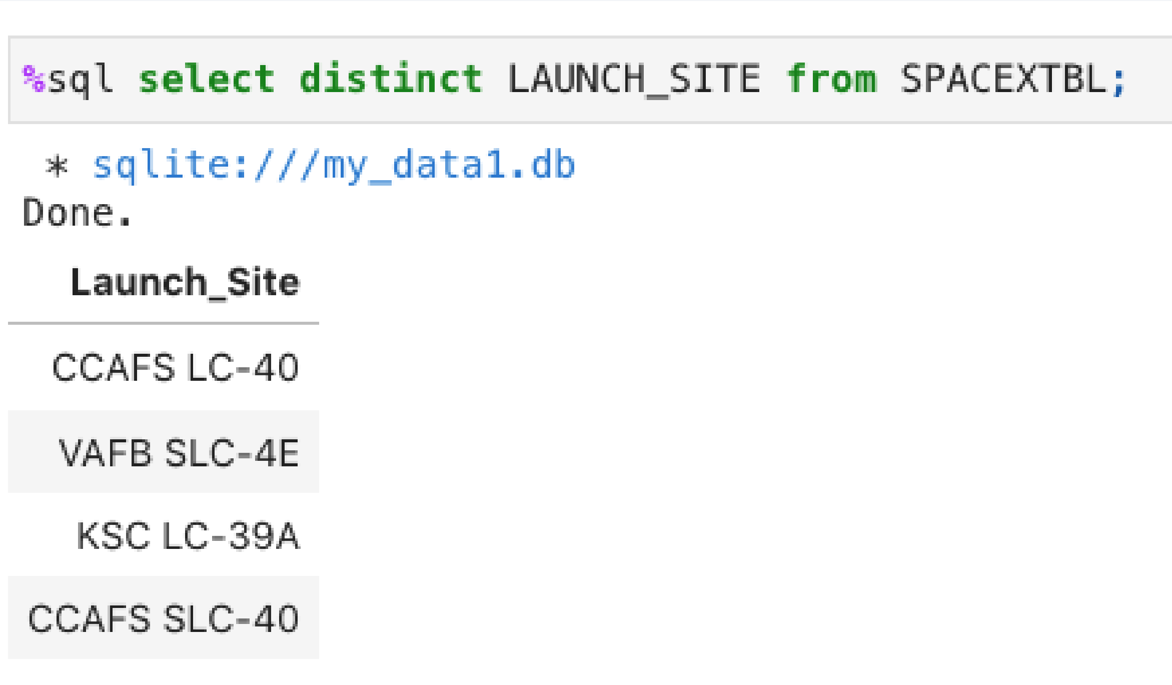
Launch Success Yearly Trend

- ***All launches were false during 2010-2013.***
- ***Average launch success kept increasing since 2013 till 2020.***



All Launch Site Names

- We can explore database by using SQL queries
- All unique site names are gotten by distinct query.
- We see that all launches took place four launch sites.



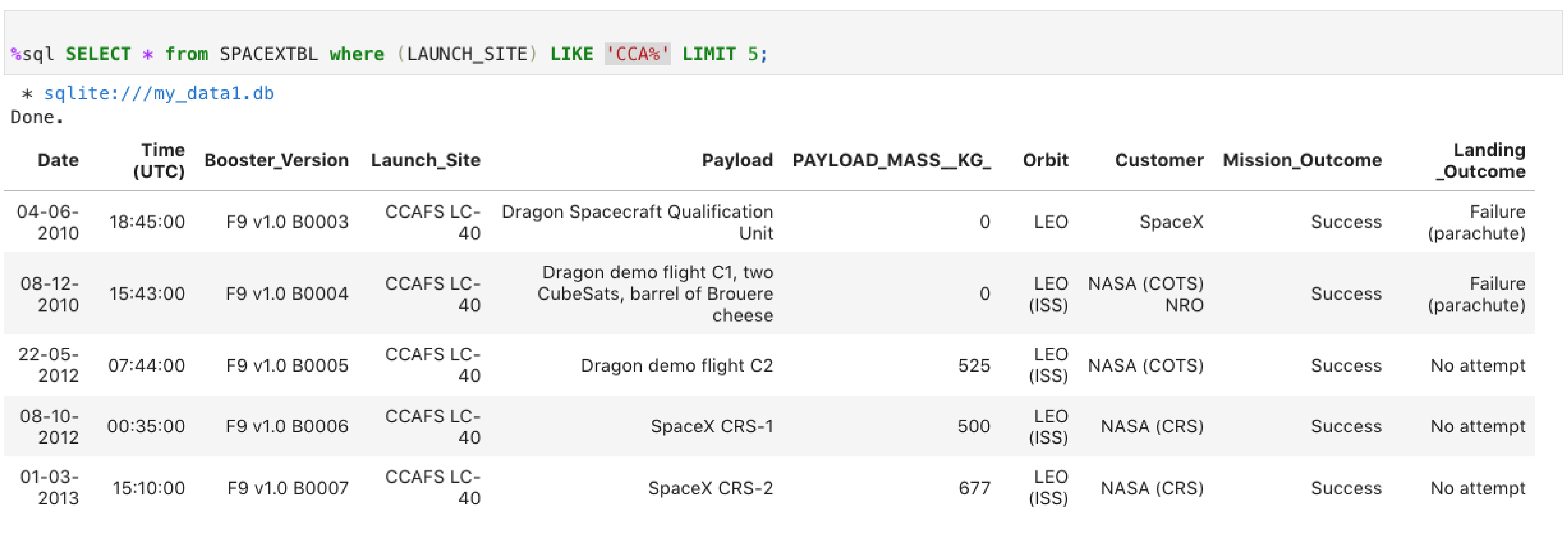
```
%sql select distinct LAUNCH_SITE from SPACEXTBL;
```

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

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

Launch Site Names Begin with 'CCA'

- To find records where launch sites begin with `CCA` we use where condition together with like condition.
- Limit 5 query gives us first five results.



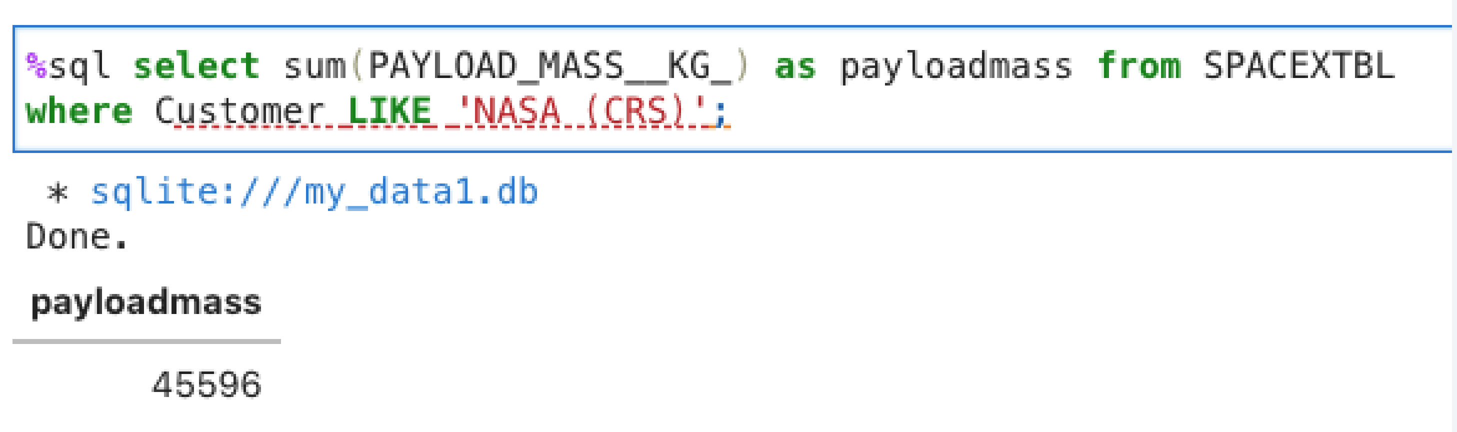
```
%sql SELECT * from SPACEXTBL where (LAUNCH_SITE) LIKE 'CCA%' LIMIT 5;
```

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

Date	Time (UTC)	Booster_Version	Launch_Site	Payload	PAYLOAD_MASS__KG_	Orbit	Customer	Mission_Outcome	Landing _Outcome
04-06-2010	18:45:00	F9 v1.0 B0003	CCAFS LC-40	Dragon Spacecraft Qualification Unit	0	LEO	SpaceX	Success	Failure (parachute)
08-12-2010	15:43:00	F9 v1.0 B0004	CCAFS LC-40	Dragon demo flight C1, two CubeSats, barrel of Brouere cheese	0	LEO (ISS)	NASA (COTS) NRO	Success	Failure (parachute)
22-05-2012	07:44:00	F9 v1.0 B0005	CCAFS LC-40	Dragon demo flight C2	525	LEO (ISS)	NASA (COTS)	Success	No attempt
08-10-2012	00:35:00	F9 v1.0 B0006	CCAFS LC-40	SpaceX CRS-1	500	LEO (ISS)	NASA (CRS)	Success	No attempt
01-03-2013	15:10:00	F9 v1.0 B0007	CCAFS LC-40	SpaceX CRS-2	677	LEO (ISS)	NASA (CRS)	Success	No attempt

Total Payload Mass

- We use sum function to calculate the total payload and we restrict results by like condition only for the boosters from NASA
- Total payload mass carried by NASA boosters is 45,596 kg.

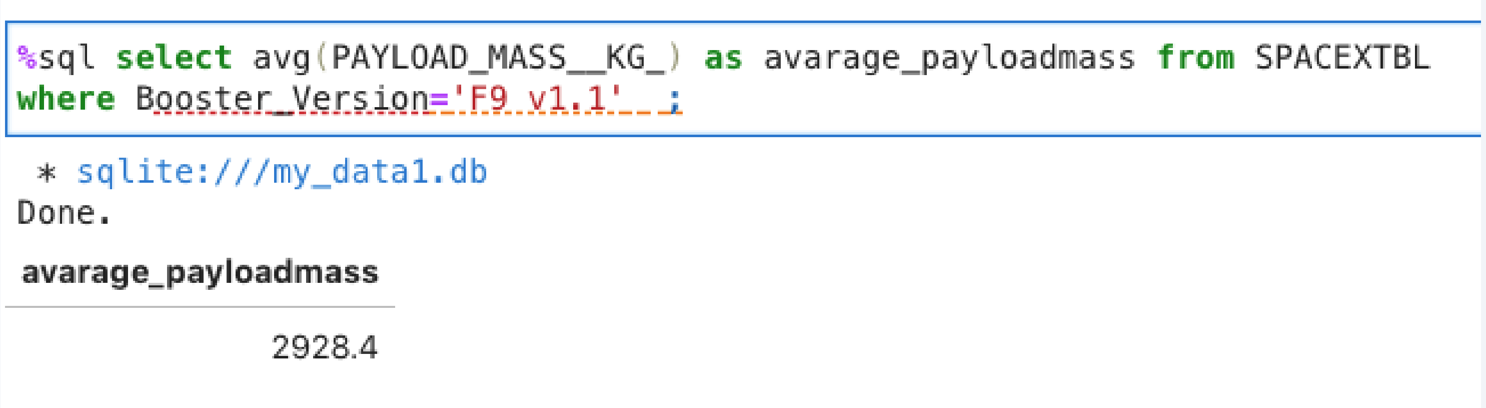


```
%sql select sum(PAYLOAD_MASS__KG_) as payloadmass from SPACEXTBL
where Customer LIKE 'NASA (CRS)';
```

```
 * sqlite:///my_data1.db
Done.
payloadmass
      45596
```

Average Payload Mass by F9 v1.1

- The avg function gives us the average payload mass and again we restrict it by like query to calculate average payload mass carried by booster version F9 v1.1
- The result is 2928.4 kg.



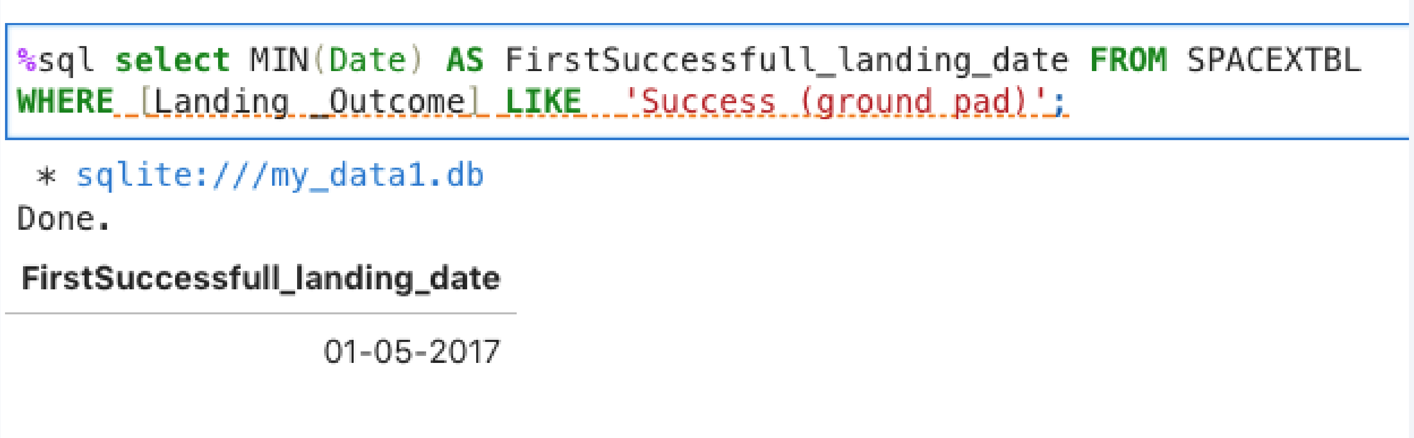
```
%sql select avg(PAYLOAD_MASS__KG_) as avarage_payloadmass from SPACEXTBL
where Booster_Version='F9 v1.1'  ;
```

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

avarage_payloadmass
2928.4

First Successful Ground Landing Date

- Find the dates of the first successful landing outcome on ground pad took place on the date 01-05-2017.
- We get the result by selecting minimum date on the data with the condition of "Success (ground pad)" in Landing_Outcome column.



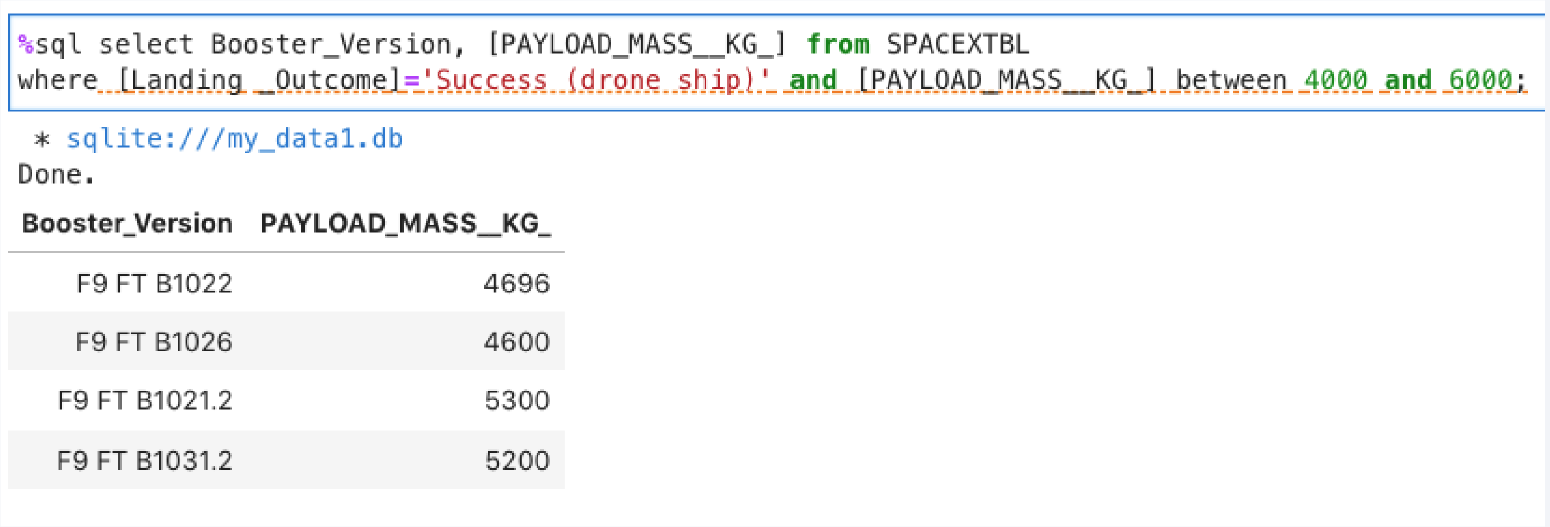
```
%sql select MIN(Date) AS FirstSuccessfull_landing_date FROM SPACEXTBL
WHERE [Landing _Outcome] LIKE  'Success (ground pad)';
```

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

FirstSuccessfull_landing_date
01-05-2017

Successful Drone Ship Landing with Payload between 4000 and 6000

- We find out from previous plots that lower payload masses have more success rate. The SQL query below gives us list the names of boosters which have successfully landed on drone ship and had payload mass greater than 4000 but less than 6000.
- We see that F9 FT boosters are in this range.



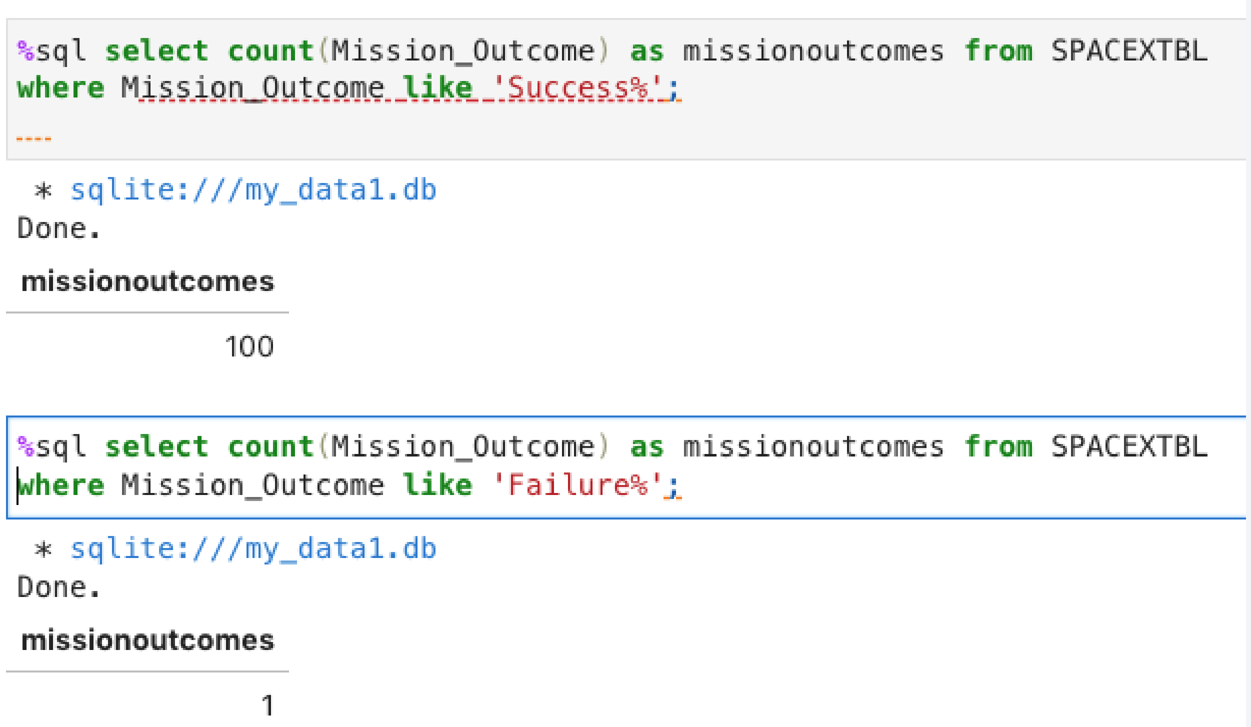
```
%sql select Booster_Version, [PAYLOAD_MASS__KG_] from SPACEXTBL
where [Landing _Outcome]='Success (drone ship)' and [PAYLOAD_MASS__KG_] between 4000 and 6000;
```

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

Booster_Version	PAYLOAD_MASS__KG_
F9 FT B1022	4696
F9 FT B1026	4600
F9 FT B1021.2	5300
F9 FT B1031.2	5200

Total Number of Successful and Failure Mission Outcomes

- The total number of successful outcome is 100 and failure mission outcome is 1.
- We select 'Success' and 'Failure' outcomes in 'Mission Outcomes' column and get sum of them.



```
%sql select count(Mission_Outcome) as missionoutcomes from SPACEXTBL
where Mission_Outcome like 'Success%';
```

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

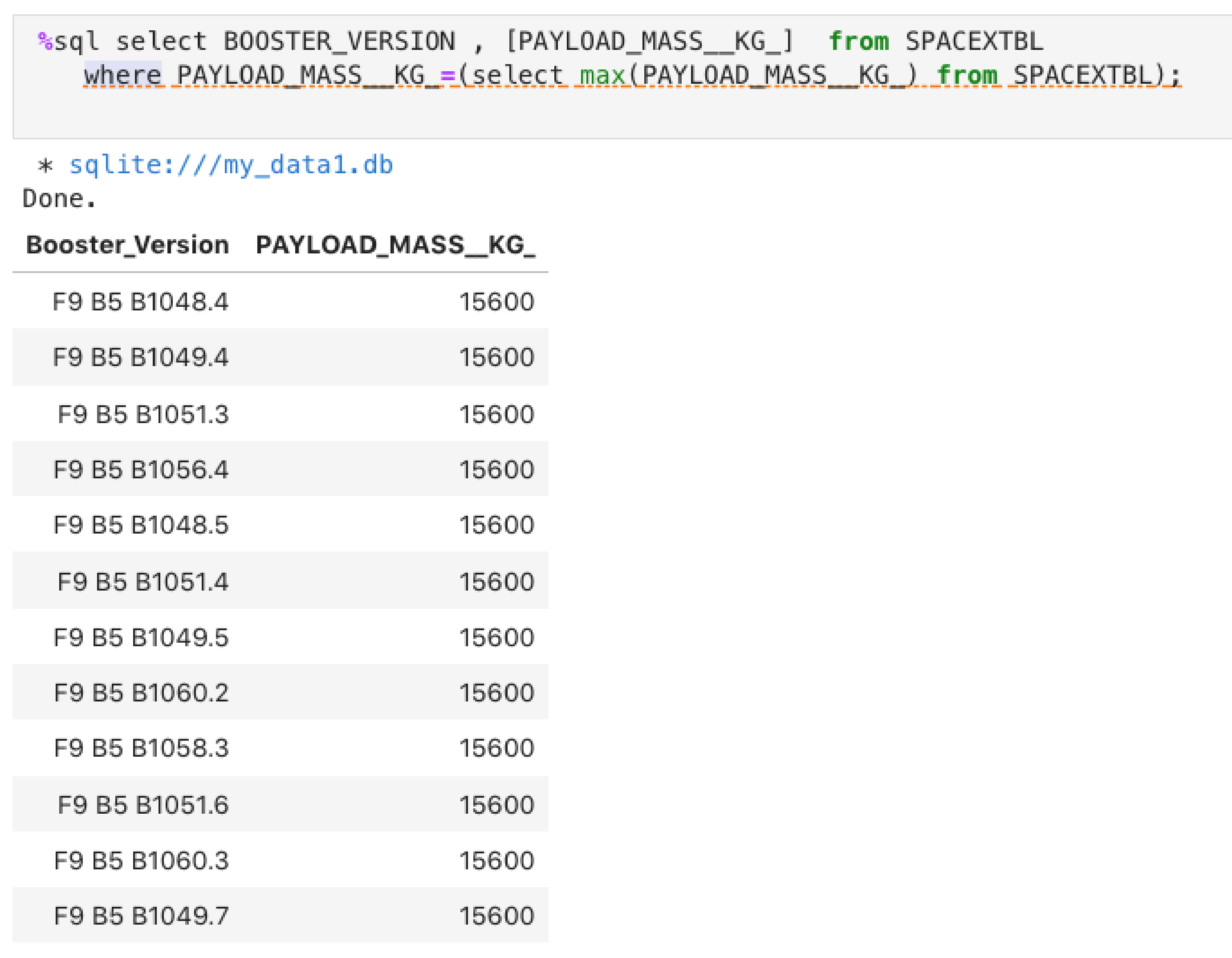
missionoutcomes
100

```
%sql select count(Mission_Outcome) as missionoutcomes from SPACEXTBL
where Mission_Outcome like 'Failure%';
```

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

missionoutcomes
1

Boosters Carried Maximum Payload



```
%sql select BOOSTER_VERSION , [PAYLOAD_MASS__KG_]  from SPACEXTBL
   where PAYLOAD_MASS__KG_=(select max(PAYLOAD_MASS__KG_) from SPACEXTBL);
```

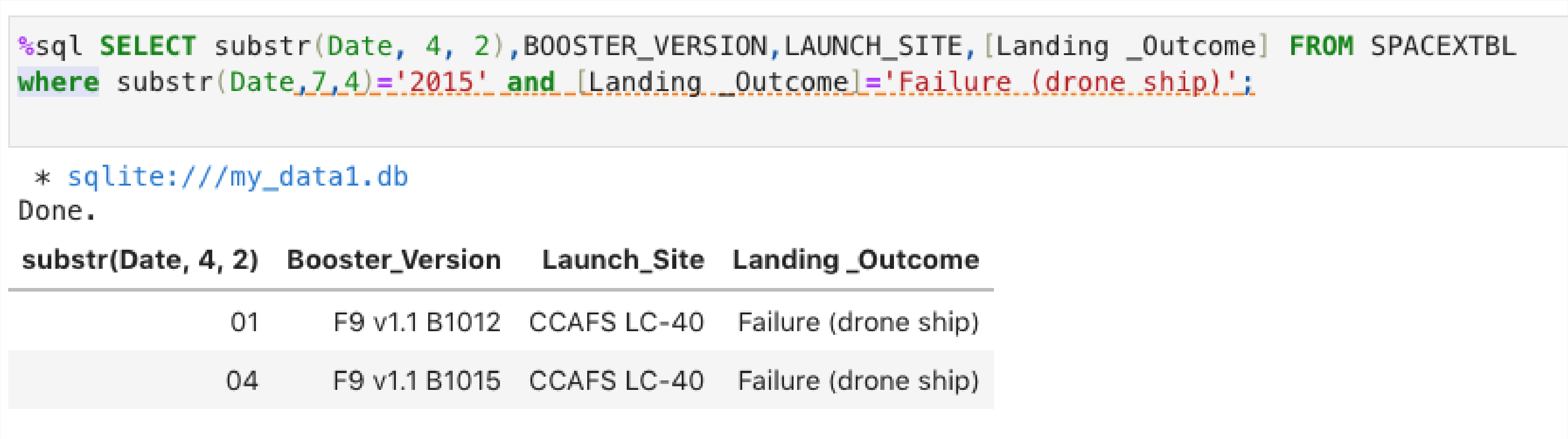
```
 * sqlite:///my_data1.db
Done.
```

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

- We can also explore the list the names of the booster which have carried the maximum payload mass.
- T max query gives the maximum number in the column which is equal to 15600 kg in our data.
- It seems that there are several boosters having this payload mass.
- Then we separate them by where PAYLOAD_MASS__KG=(select max(PAYLOAD_MASS__KG) from SPACEXTBL) query.

2015 Launch Records

- We can get the list of the failed landing_outcomes in drone ship, their booster versions and launch site names for in year 2015.
- We use the date condition for 2015 and equal condition for landing outcome as failure (drone ship).



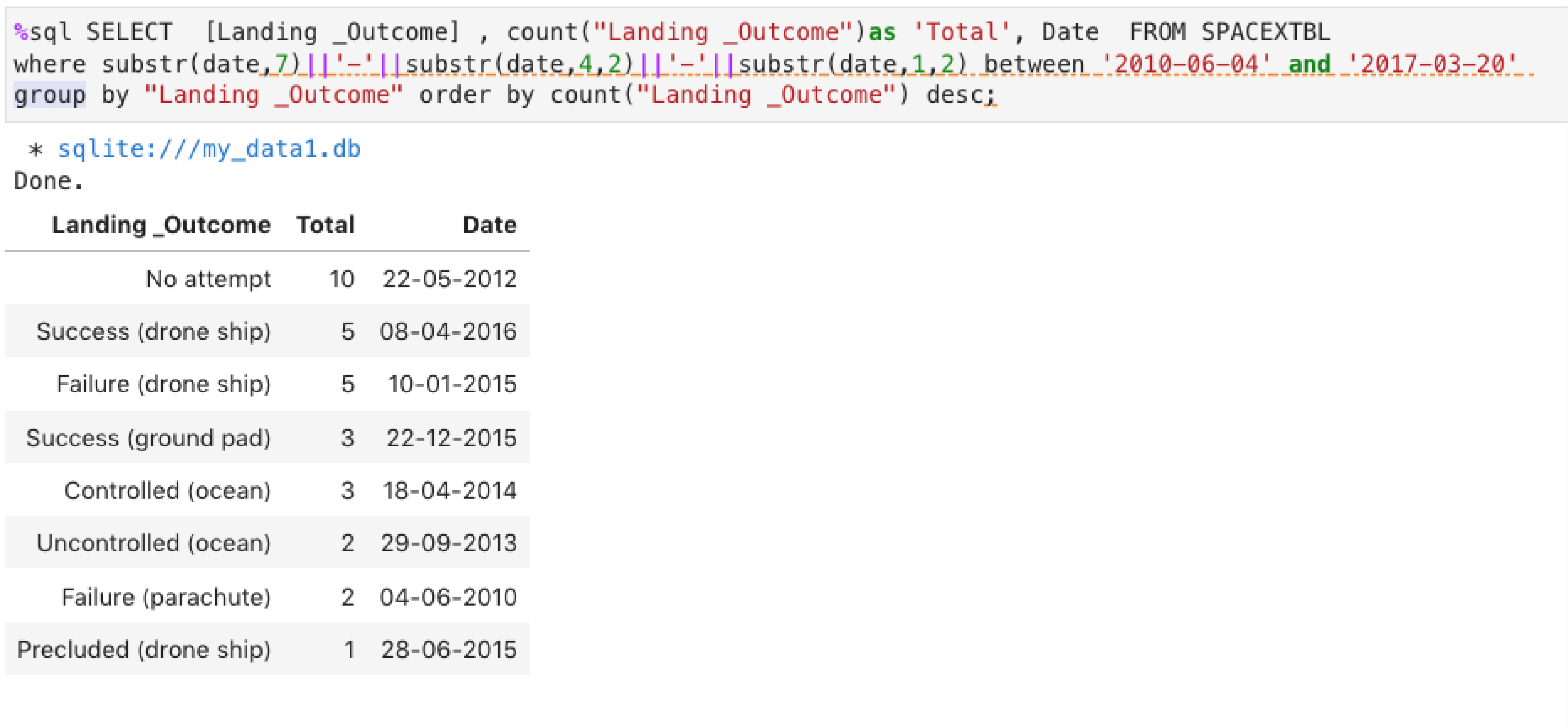
```
%sql SELECT substr(Date, 4, 2),BOOSTER_VERSION,LAUNCH_SITE,[Landing _Outcome] FROM SPACEXTBL
where substr(Date,7,4)='2015' and [Landing _Outcome]='Failure (drone ship)';
```

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

substr(Date, 4, 2)	Booster_Version	Launch_Site	Landing _Outcome
01	F9 v1.1 B1012	CCAFS LC-40	Failure (drone ship)
04	F9 v1.1 B1015	CCAFS LC-40	Failure (drone ship)

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

- The image on the left shows the rank landing outcomes between the date 2010-06-04 and 2017-03-20, in descending order.
- The success rate during these years is 32%.
- We get this table using by date condition, grouping by Landing Outcomes and counting total outcomes.



```
%sql SELECT  [Landing _Outcome] , count("Landing _Outcome")as 'Total', Date  FROM SPACEXTBL
where substr(date,7)||'-'||substr(date,4,2)||'-'||substr(date,1,2) between '2010-06-04' and '2017-03-20'
group by "Landing _Outcome" order by count("Landing _Outcome") desc;
```

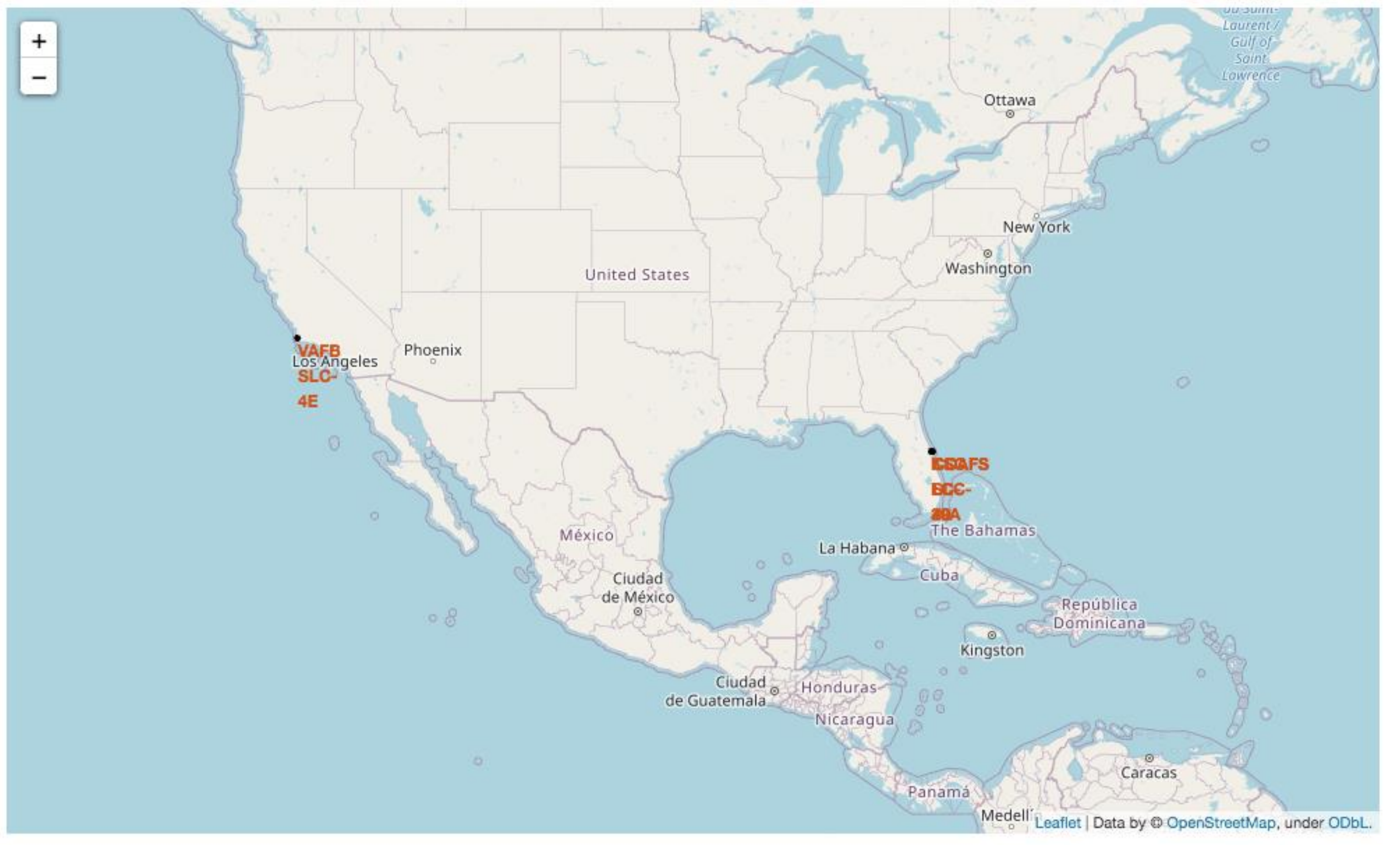
```
 * sqlite:///my_data1.db
Done.
```

Landing _Outcome	Total	Date
No attempt	10	22-05-2012
Success (drone ship)	5	08-04-2016
Failure (drone ship)	5	10-01-2015
Success (ground pad)	3	22-12-2015
Controlled (ocean)	3	18-04-2014
Uncontrolled (ocean)	2	29-09-2013
Failure (parachute)	2	04-06-2010
Precluded (drone ship)	1	28-06-2015

Section 3

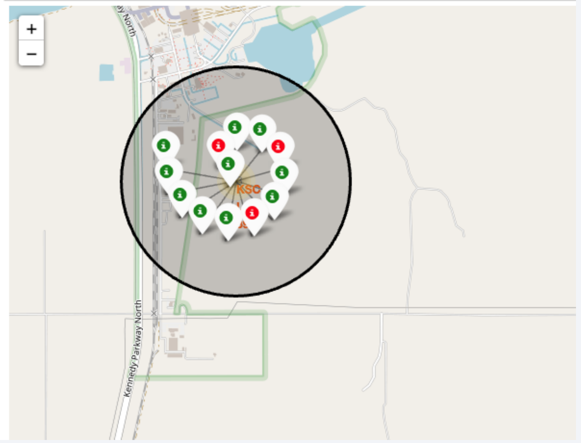
Launch Sites Proximities Analysis

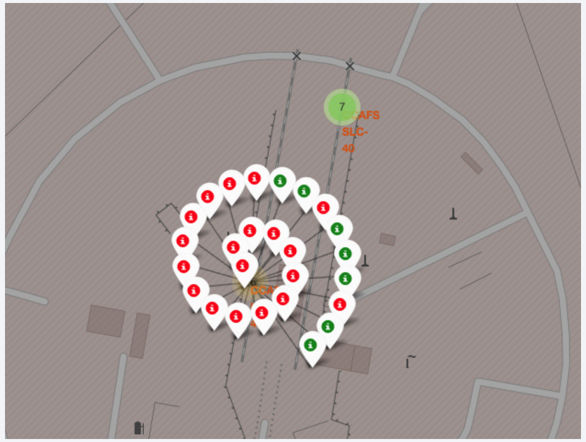
All Launch Sites on the Map

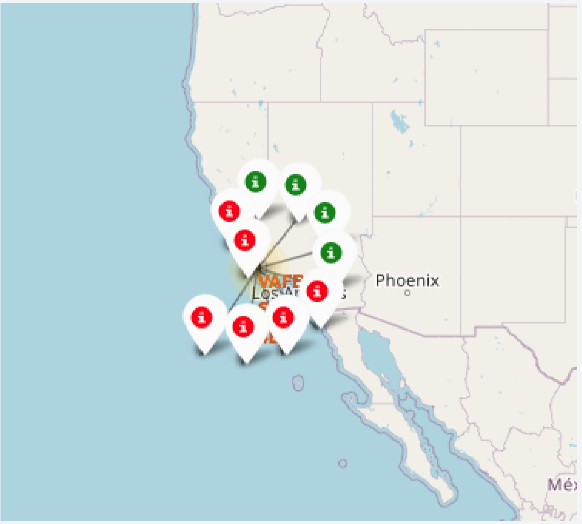


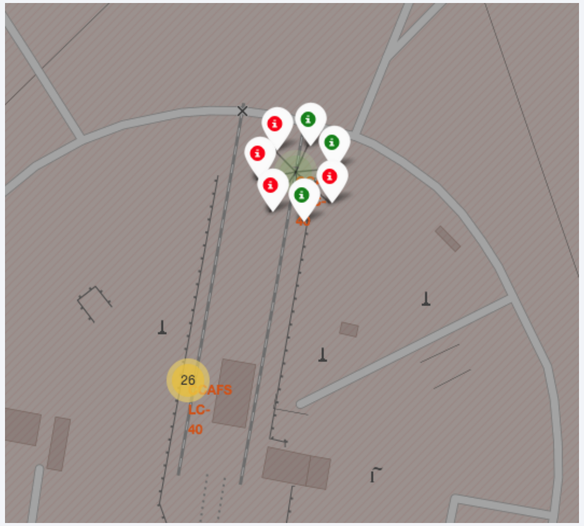
- All launch sites are in proximity to the Equator line.
- All launch sites are in very close proximity to the coast.
- They are mostly away from cities to prevent accidents in failed launch situations.

Success/Fail Indicators on Map



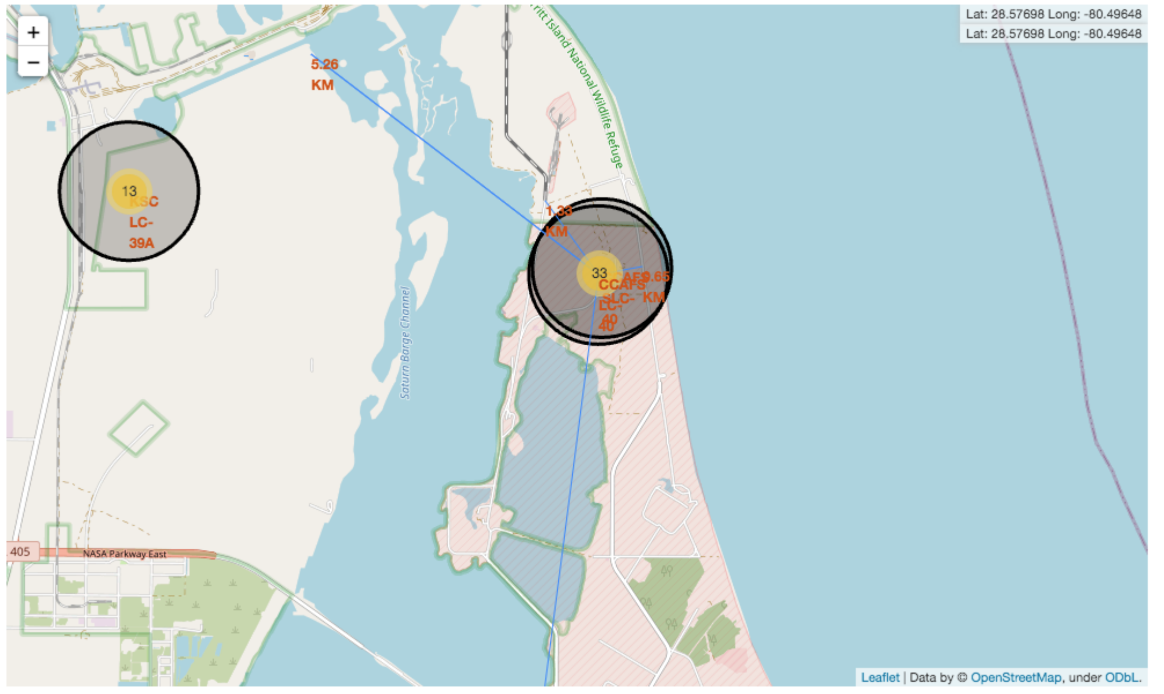


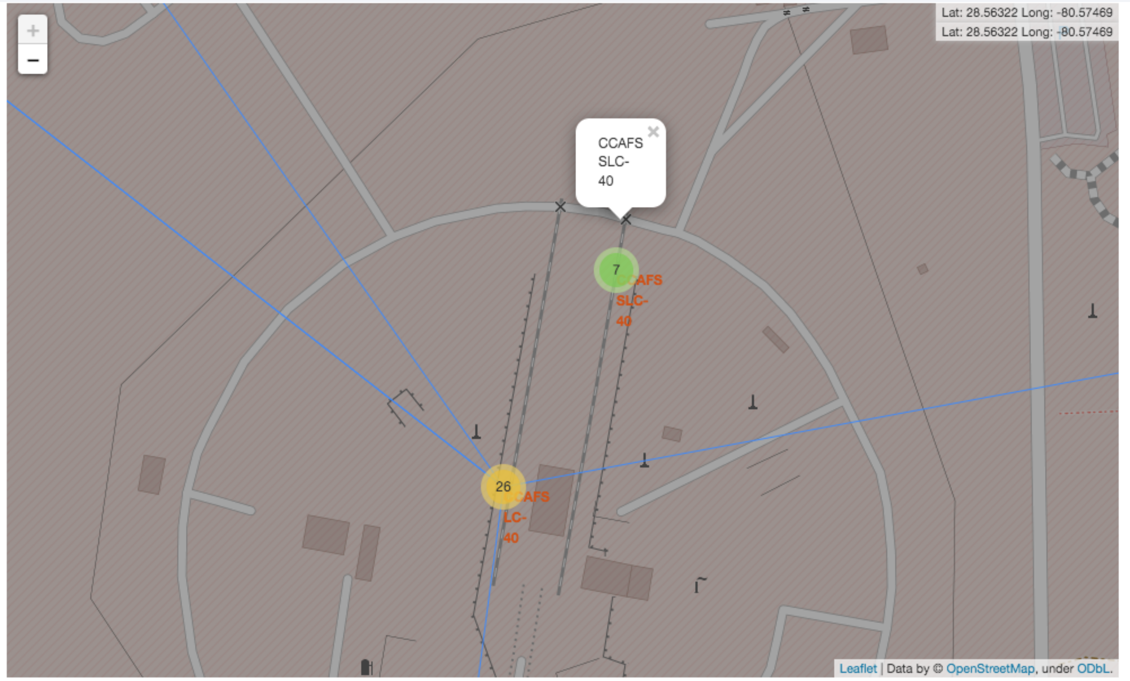




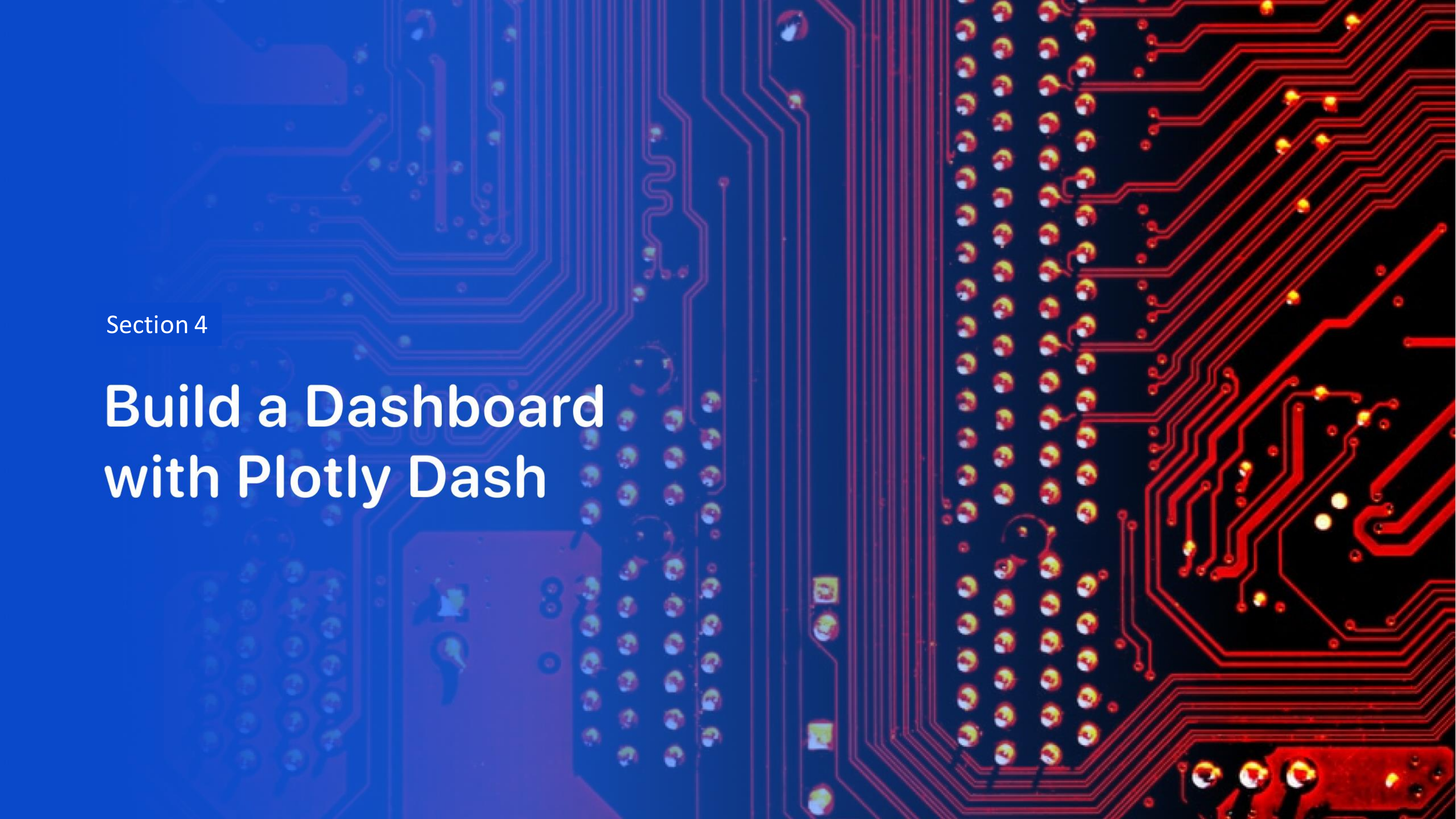
- By using Folium map, we can indicates labels [0,1] by colors.
- Green markers represent successful launches while red markers do fails.
- KSC LG 39 A had the most successful launches from all sites.

Launch Sites to Its Proximities


Lat: 28.57698 Long: -80.49648
Lat: 28.57698 Long: -80.49648
+
−
5.26 KM
13
KSC LC-39A
1.33 KM
33
CCAFS SLC-40
AFS 0.65 KM
LC-40
itt Island National Wildlife Refuge
Saturn Barge Channel
405
NASA Parkway East
Leaflet | Data by © OpenStreetMap, under ODbL.


Lat: 28.56322 Long: -80.57469
Lat: 28.56322 Long: -80.57469
+
−
CCAFS SLC-40
7
AFS SLC-40
26
AFS LC-40
Leaflet | Data by © OpenStreetMap, under ODbL.

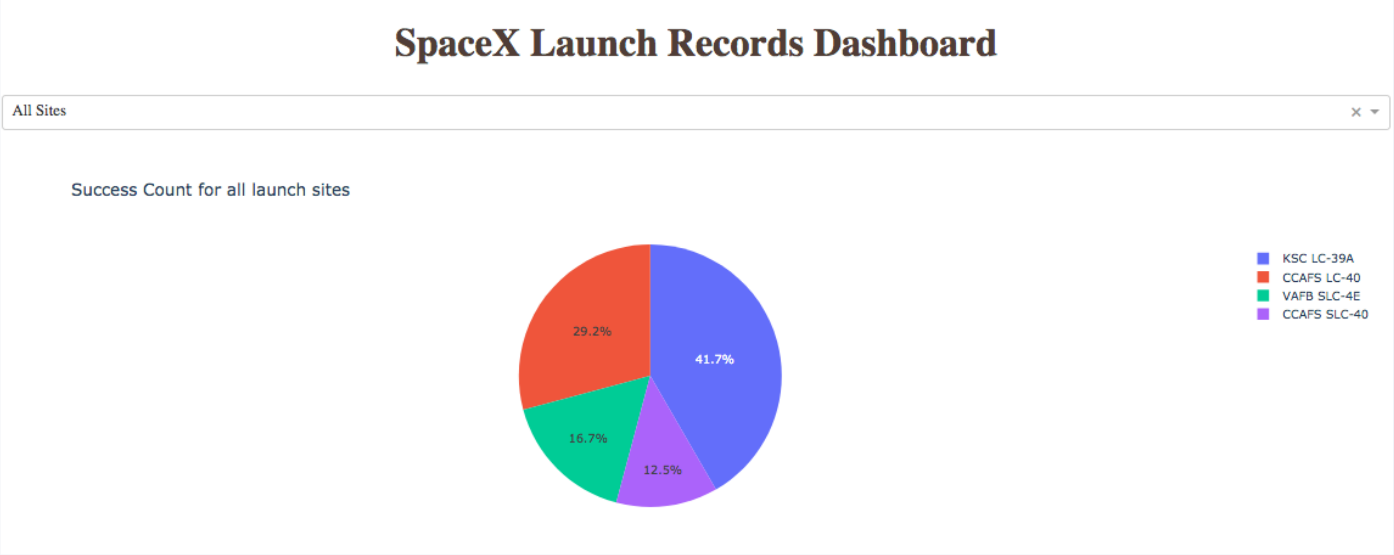
- The map on the left shows the distance of a chosen proximity.
- The map on the right shows distances of closest highway, city and railroad line.



Section 4

Build a Dashboard with Plotly Dash

SpaceX Launch Records for All Sites

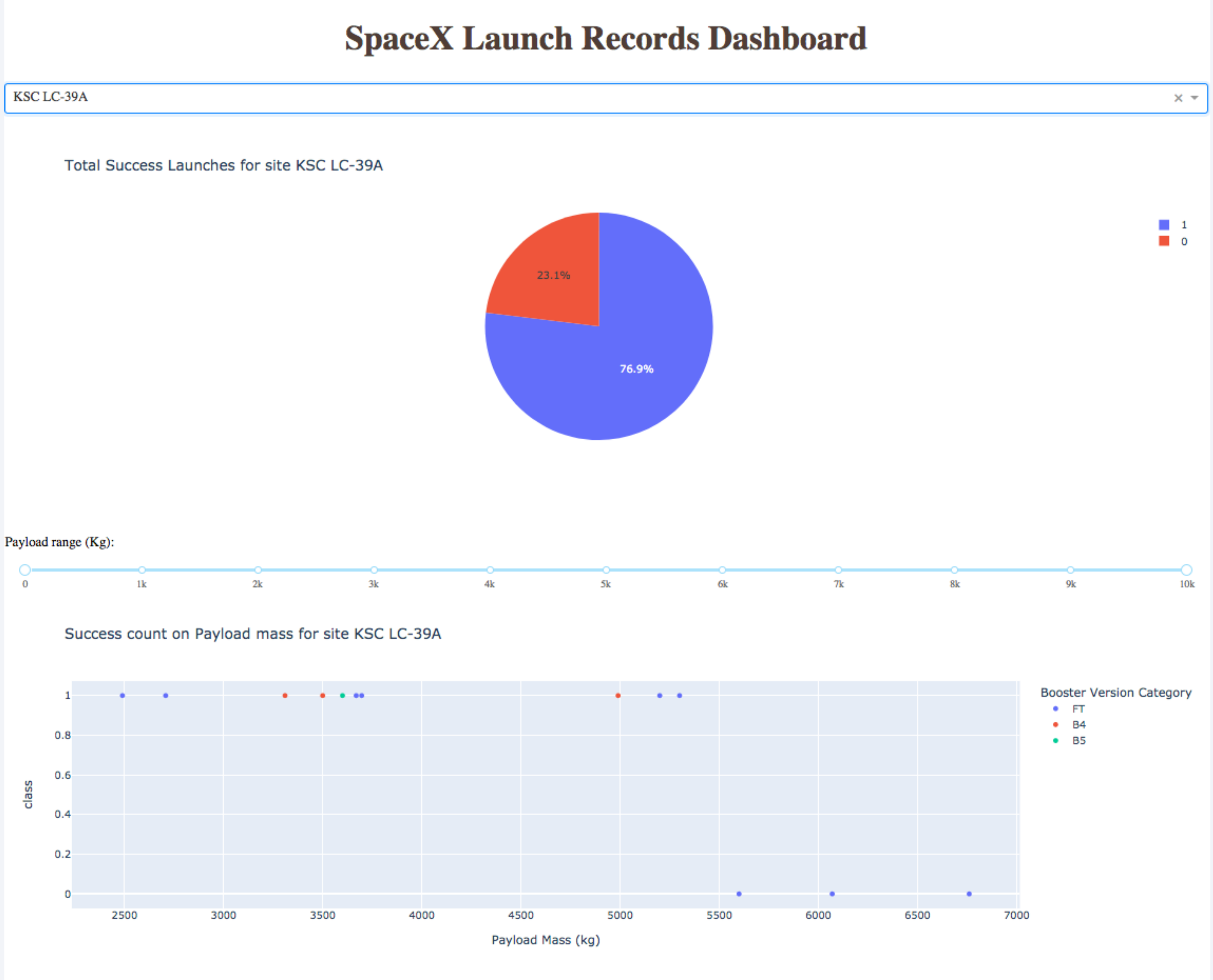


SpaceX Launch Records Dashboard

All Sites

- The most successful site is KSC LC 39A with 41.7%.
- Second is CCAFS LC-40 with 29.2%.
- VAFB SLC-4E and CCAFS SLC-40 follow them with 16.7% and 12.5% rates respectively.

Launch Site With Highest Score



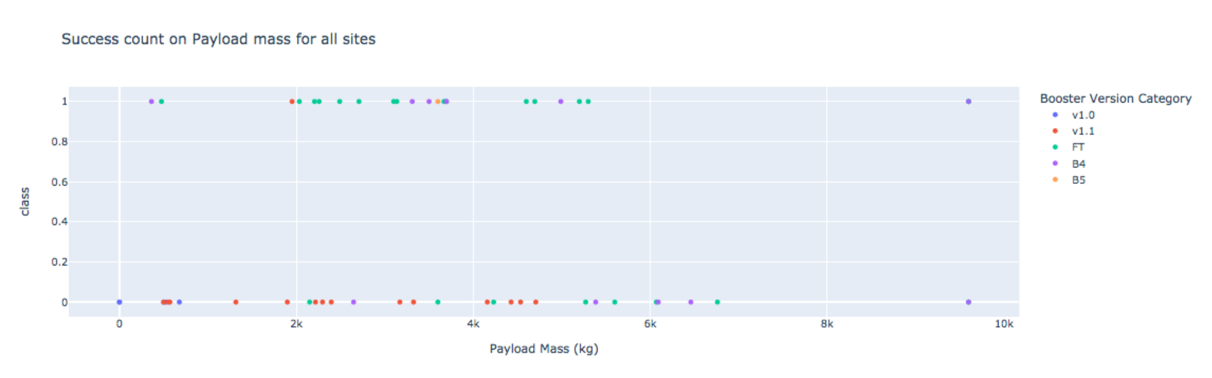
SpaceX Launch Records Dashboard

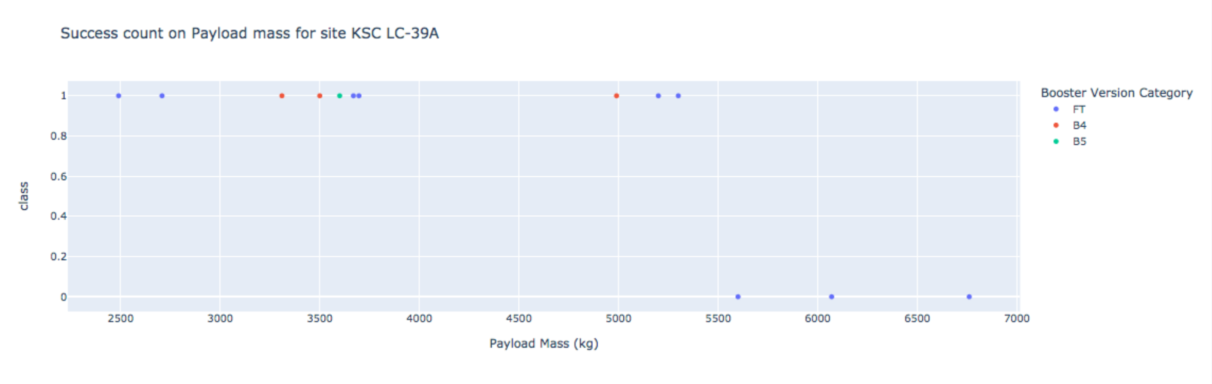
KSC LC-39A

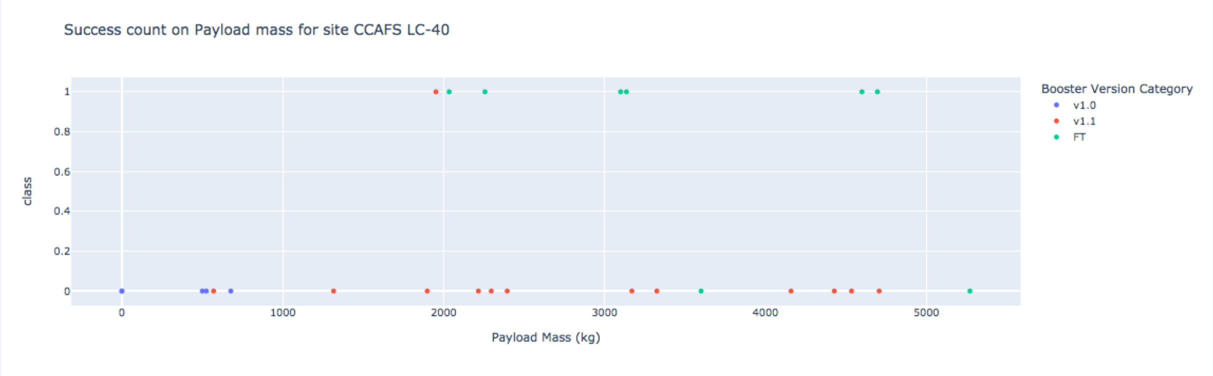
Payload range (Kg):

- KSC LC 39A is the most successful launch site amongst others with ~76.9%.
- Payload mass less than 5k has 100% success rate and more than 5k has 0%.
- Booster version FT is more successful.

Payload vs. Launch Site Scatter Plots


Success count on Payload mass for all sites


Success count on Payload mass for site KSC LC-39A


Success count on Payload mass for site CCAFS LC-40

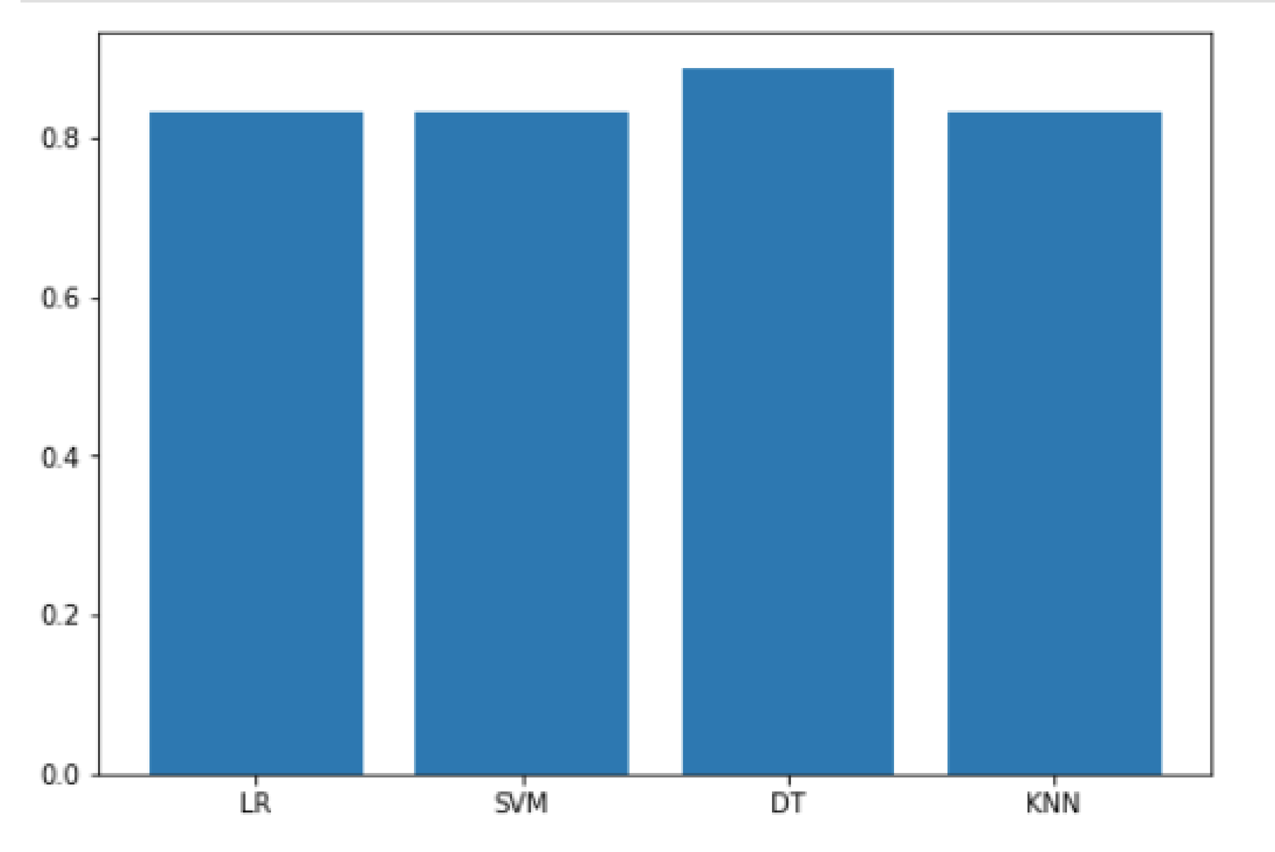
- Although most of the successful launches are <5k, it is not distinguishable for all sites because, most of the failures are <5k too.
- The most successful booster version is B4.
- At the KSC LC-39A site, launches with payload mass under 5k is all successful.
- For the site CCAFS LC-40, booster version FT has 75% success rate.

Section 5

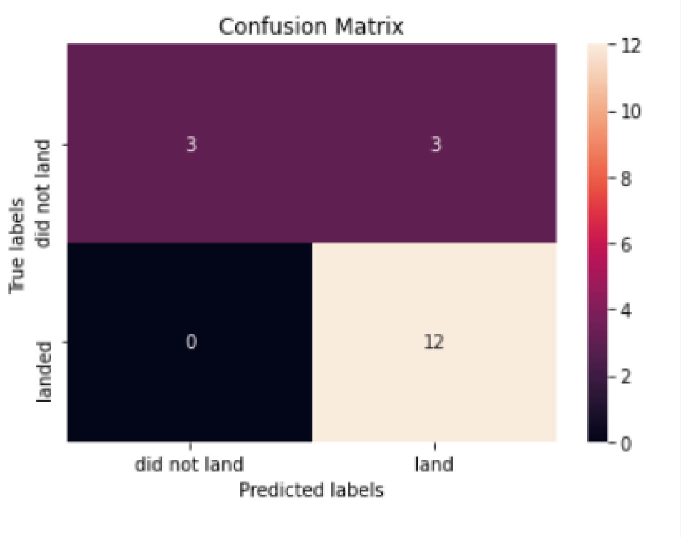
Predictive Analysis (Classification)

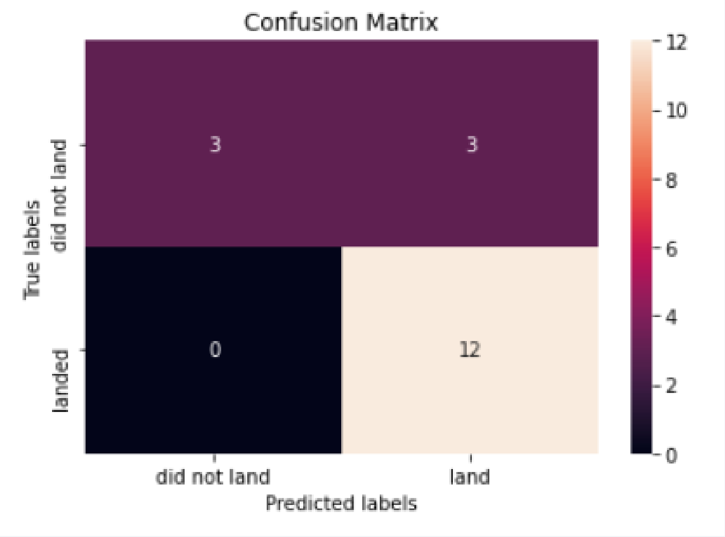
Classification Accuracy

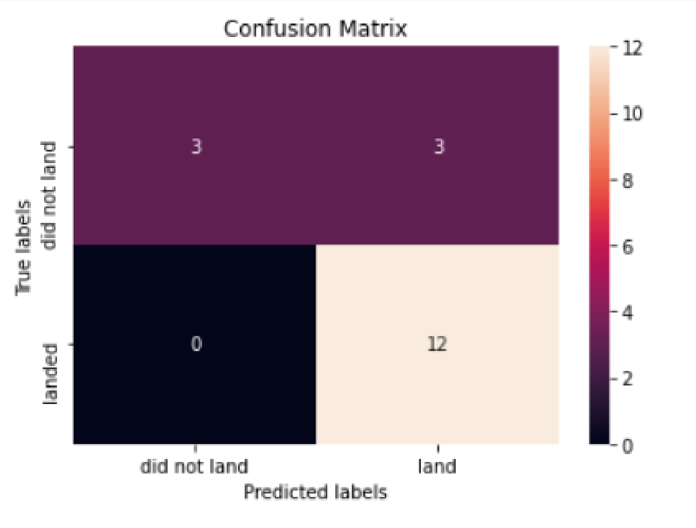
- We use Logistic Regression, Support Vector Machine, Decision Tree and K-Nearest Neighbour models for classification.
- The Decision Tree model give us the best accuracy with 0.8888
- Other models are all 0.8333

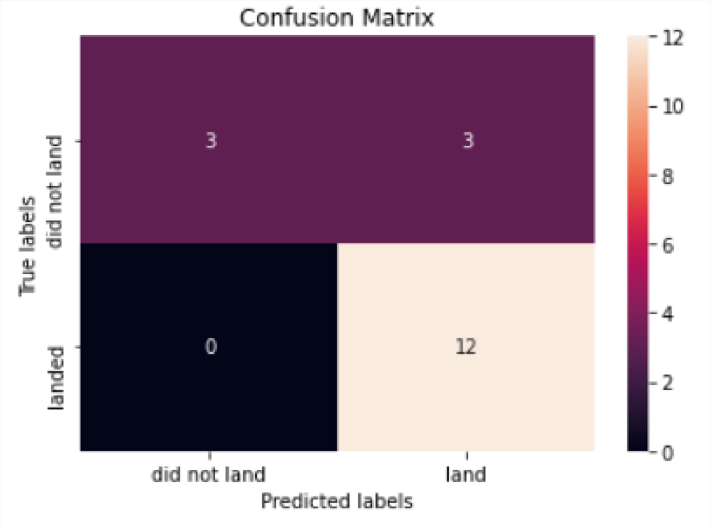


Confusion Matrix









- *For all four models we have same confusion matrices.*
- *Each cell in the matrix indicates True/False Negative/Positive results as below.*

True Negative	False Positive
False Negative	True Positive

- "**true positive**" for correctly predicted landing values: 12
- "**false positive**" for incorrectly predicted landing values: 3
- "**true negative**" for correctly predicted not-landing values :3
- "**false negative**" for incorrectly predicted not-landing values : 3

Conclusions

- The success rate of Falcon 9 SpaceX launches increased in time.
- Low weighted payloads perform better than the heavier payloads.
- KSC LC 39A has the most successful launches in all other sites.
- Orbit GEO, HEO, SSO, ES-L1 have the best success rate.
- Booster Version BT has striking amount of success.
- The best classification model for this dataset is Decision Tree with the accuracy of 0.88.

Appendix

- All Python code snippets can be found on https://github.com/GulcinBU/GulcinBU

Thank you!