IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

Anderson Lisboa
July, 02/2024

Outline

Executive Summary

- In order to predict success satellite impulsion module land back to sand / ocean according to some characteristics such as launch place, weight, booster type…, etc some steps was necessary:
    - to gather information on databases, csv files and web scrapping.
    - Clean/discard or fill missing/ error information
    - Identify which characteristics may impact the module land using graphical and statistical methods , and discard all unnecessary characteristics
    - Finally train, test and make predictions using different algorithms types as Superised / Unsupervised ; Regression , Classification , Grouping
- As a result, we manage to identify Booster type, Launch Site and Payload Weight as the principal characteristics influences a successful module land.
- Utilizing all 83 characteristics for example: satellite orbit and witch parts was reused, we manage to obtain an successfully prediction in 83,34% of tested cases in tree different algorithms (Logistics Regression , K Nearest Neighbors and Support Vector Machines). All landed cases was correctly identified , only few failed land was wrongly identified as a success (False Positive).
- On the other Hand, Decision Tree only could predict successfully 77,78% test cases. This has either false positive and false negative results . This method should be avoid

Introduction

- New technologies in satellite launches has become a game change in the aerospacial market. Nowadays is possible to land back the first stage (propulsion) after a satellite rocket launch.
- SpaceX Falcon 9 first stage can land successfully and could represent an economy of 62 million dollars up165 million dollars on space market rocket launch depending which company, type of obit, weight, etc.
- A new company to be competitive in this market should focus his investment only on the sucessfull SpaceX Falcon 9 cases. This work is to identify witch project rockets launchs characteristics influences in the land success and predict future land success base on supervised or unsupervised algorithms.

Section 1

Methodology

Methodology

Executive Summary

- Data collection methodology: Requested data to the SpaceX API (Launchs. Rockets, Launchpads , Payload) , WebScraping wikipedia Falcon 9 table
- Perform data wrangling : Values were normalized, removed duplicate rows, secondary boosters, fill missing payload data with mean , and classify outcome in fail ad success (missing outcome was considered a fail)
- Perform exploratory data analysis (EDA) using visualization and SQL : With SQL LITE - Query using Select:
    - List of unique Launch Sites ,
    - identifying total payload mass carried by boosters launched by NASA,
    - 5 records where launch sites begin with CCA,
    - Average payload mass carried by booster version F9 v1.1,
    - date when the first successful landing outcome in ground pad was achieved,
    - boosters which have success in dronList the names of the booster_versions which have carried the maximum payload masse ship and have payload between 4000 and 6000 ,
    - List the total number of successful and failure mission outcomes
    - List Fail outcomes , month names, drone ship ,booster versions, launch_site in 2015
    - Landing outcomes between the date 2010-06-04 and 2017-03-20

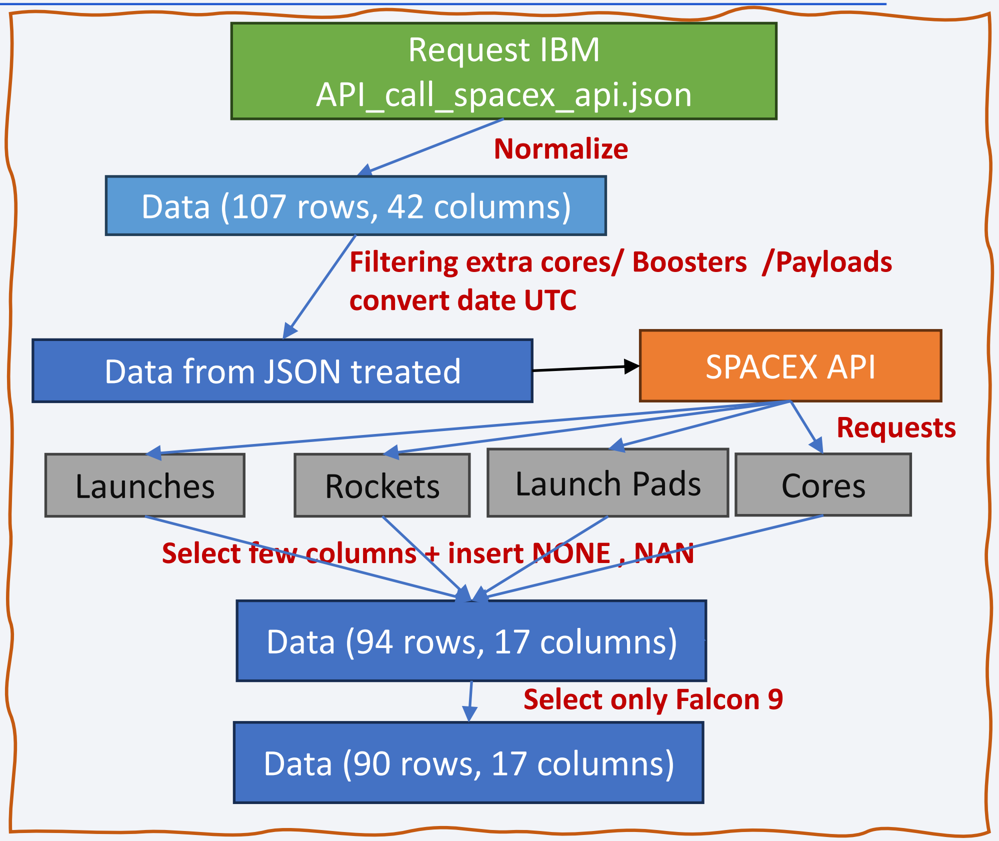
Methodology

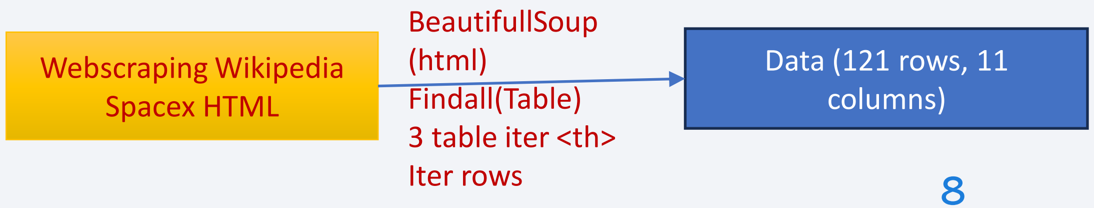
Executive Summary

- Perform interactive visual analytics using Folium and Plotly Dash
  - Cat Plot – Flight Number and Payload would affect the launch outcome
  - Cat Plot – Flight Number and Launch Site would affect the launch outcome
  - Cat Plot – Payload and Launch Site would affect the launch outcome
  - Bar Chart - relationship between success rate and orbit type
  - Cat Plot - relationship between Flight Number and Orbit type
  - Cat Plot - relationship between Payload and Orbit type
  - Line Plot – Trend of success rate by year
  - Mark all launch sites on a map , Mark the success/failed launches for each site on the map , Calculate the distances between a launch site to its proximities (road, river , airport, city, rail, school, etc), Insert circles, marks , lines, text into the map
  - Iterative Dashboard selecting Site Location and payload Range and receiving a piechart and scatter plot with the success land rate
- Perform predictive analysis using classification models
  - Predicting if the first stage will land: normalizing data, train K Nearest Neighbors, Decision Tree, Logistic Regression, Vector Machine models and evaluating model performance in a train dataset

Data Collection

- Describe how data sets were collected.
- Whith IBM Json file, we connected whith SPACEX API(https://api.spacexdata.com/v4/launches/past ; https://api.spacexdata.com/v4/rockets/; https://api.spacexdata.com/v4/launchpads/; https://api.spacexdata.com/v4/payloads/; https://api.spacexdata.com/v4/cores/)
- Selecting only important columns from API responses and inserting none in missing string type information and NAN in flotat type missing information . Now we have a dataset whith some diferentes rockets
- Filtering only Falcon 9 we have an reduced dataset with (90 rows, 17 columns
- Other font was webscraping, using BeatifullSoup we collect data from wikipedia SPACEX information (https://en.wikipedia.org/w/index.php?title=List_of_Falcon_9_and_Falcon_Heavy_launches&oldid=1027686922) , the disireble table is the tird one . Interacting throw table headers and table rowws

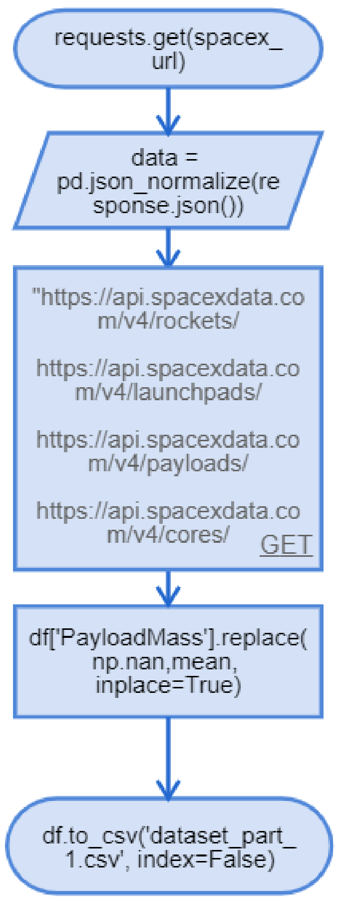


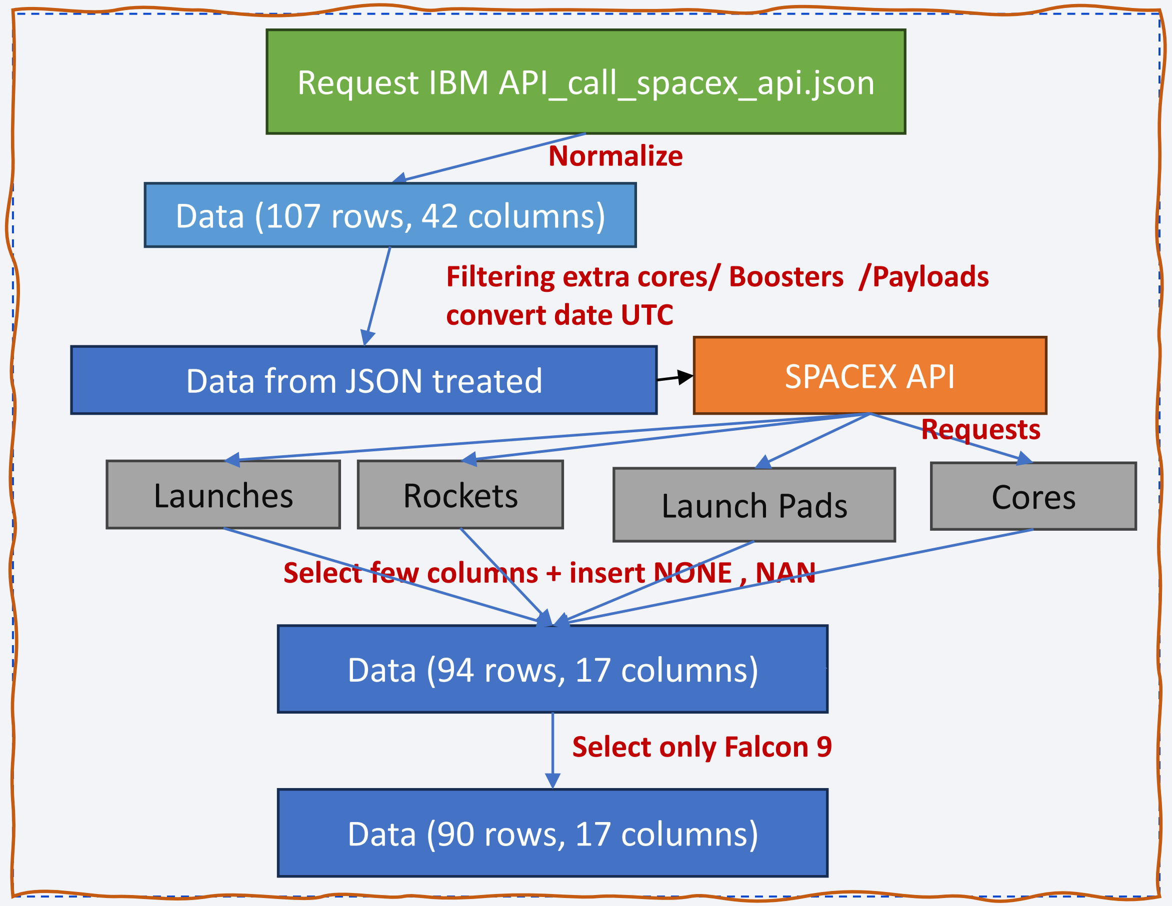


Data Collection – SpaceX API

- GITHUB – Jupter file Link:
- https://github.com/Mazolli47/CourseraIBM/blob/main/BACKUP/01_jupyter-labs-spacex-data-collection-api.ipynb

```python
# Takes the dataset and uses the cores column to call the API and append the data to the lists
def getCoreData(data):
    for core in data['cores']:
            if core['core'] != None:
                response = requests.get("https://api.spacexdata.com/v4/cores/"+core['core']).json()
                Block.append(response['block'])
                ReusedCount.append(response['reuse_count'])
                Serial.append(response['serial'])
            else:
                Block.append(None)
                ReusedCount.append(None)
                Serial.append(None)
            Outcome.append(str(core['landing_success'])+' '+str(core['landing_type']))
            Flights.append(core['flight'])
            GridFins.append(core['gridfins'])
            Reused.append(core['reused'])
            Legs.append(core['legs'])
            LandingPad.append(core['landpad'])
```





Data Collection - Scraping

- GITHUB –Jupter file Link:
- https://github.com/Mazolli47/CourseraIBM/blob/main/BACKUP/02_jupyter-labs-webscraping.ipynb

```
static_url = "https://en.wikipedia.org/w/index.php?title=List_of_Falc
```

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

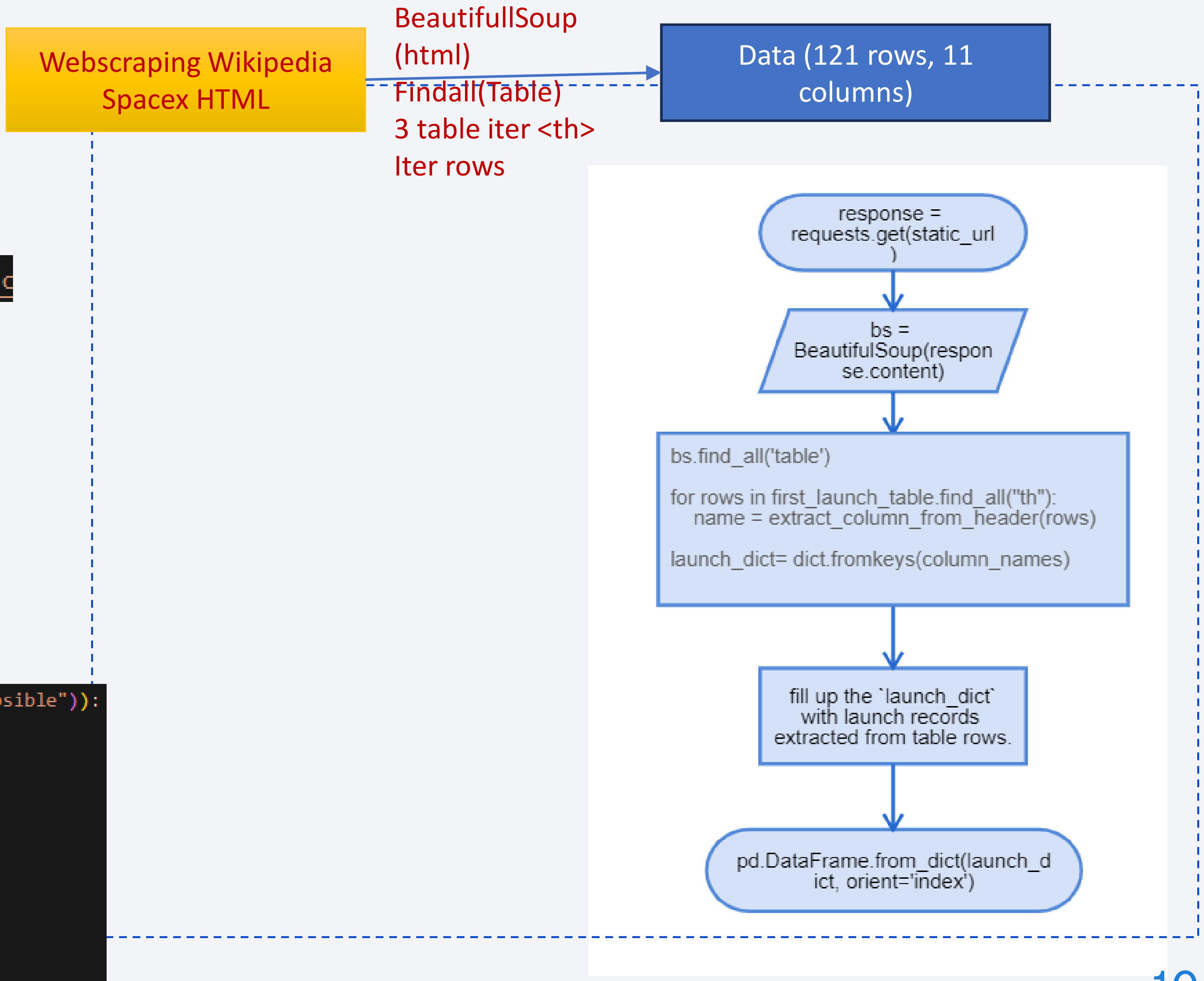
```
soup = BeautifulSoup(data, 'html')
```

```
html_tables=soup.find_all('table')
```

```
first_launch_table = html_tables[2]
```

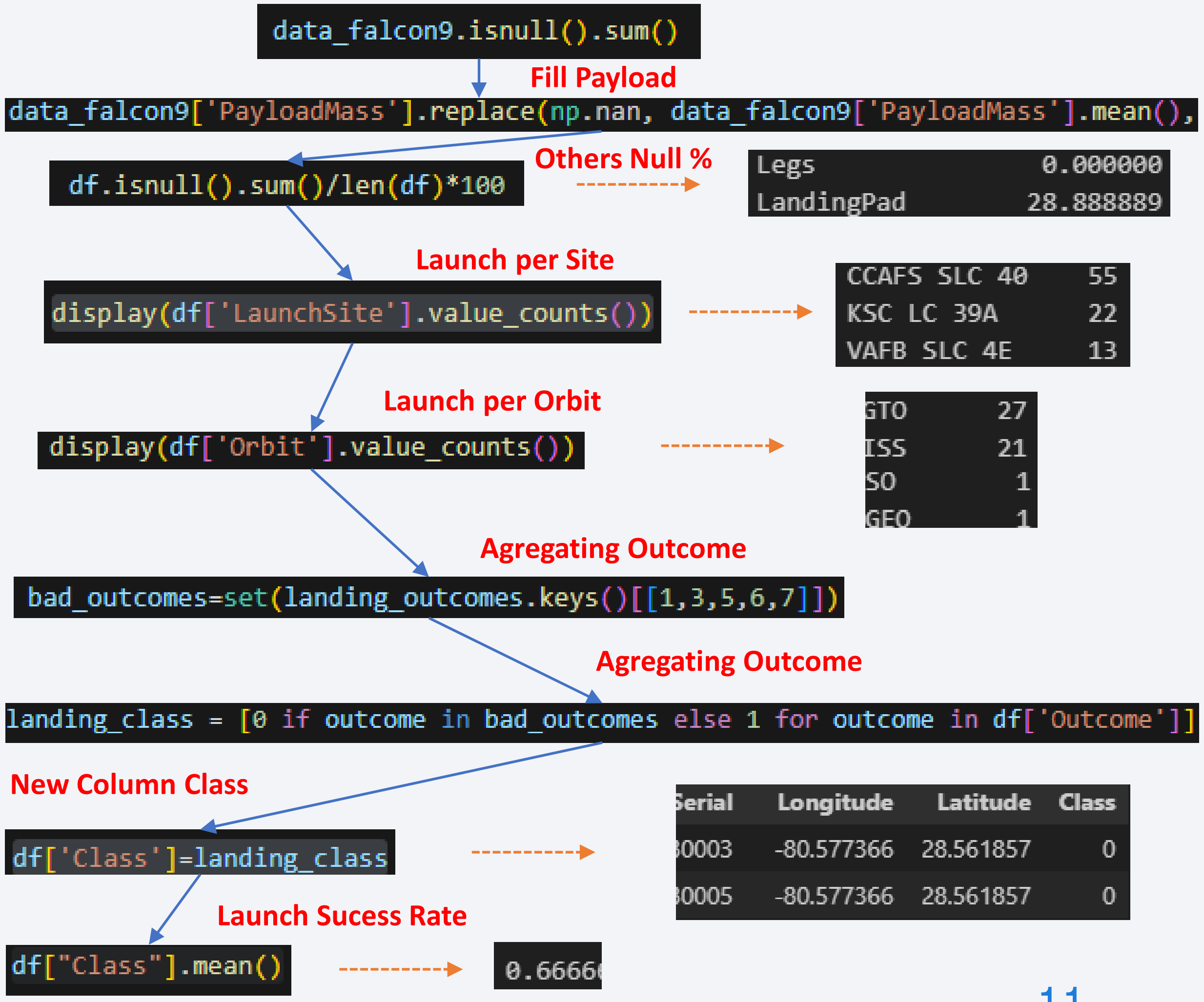
```
for row in first_launch_table.find_all('th'):
        col_name = extract_column_from_header(row)
        if col_name is not None and len(col_name)>0 :
            column_names.append(col_name)
```

```
for table_number,table in enumerate(soup.find_all('table',"wikitable plainrowheaders collapsible")):
   # get table row
    for rows in table.find_all("tr"):
        #check to see if first table heading is as number corresponding to launch a number
        if rows.th:
            if rows.th.string:
                flight_number=rows.th.string.strip()
                flag=flight_number.isdigit()
        else:
            flag=False
        #get table element
        row=rows.find_all('td')
```



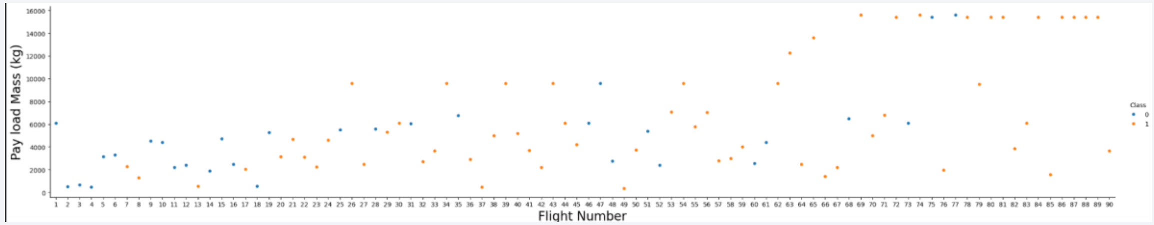
Data Wrangling

- We will perform some Exploratory Data Analysis (EDA) to find some patterns in the data and determine how will be training supervised models (labels selection)
- First replaced missing payload mass with the mean value , after observing column values types and verifying the existence of null values
- Calculate the number of launches on each site, CCAFS SLC 40 has the majority of launch/land (55) on the Other Hand VAFB SLC 4E has the lowest launch count (13)
- Calculate the number and occurrence of each orbit, GTO orbit has the highest launch count (27) but GEO, SO, HEO, ES-L1 a had only one launch each one (cannot perform a statistical inference in those four orbit)
- Aggregating 7 different types of outcomes in only Success or Fail to land (new column "Class"), After determining the mean landing success rate of 66%
- GITHUB –Jupter file Link:
- https://github.com/Mazolli47/CourseraIBM/blob/main/BACKUP/03_labs-jupyter-spacex-Data%20wrangling.ipynb

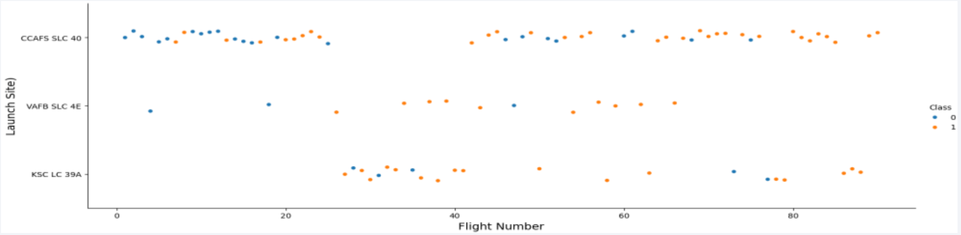


EDA with Data Visualization

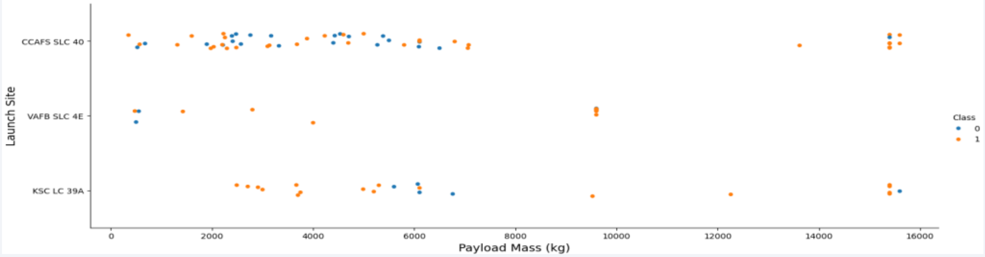
Cat Plot– Flight Number and Payload would affect the launch outcome. Payload increased in last launches, and success rates increased too

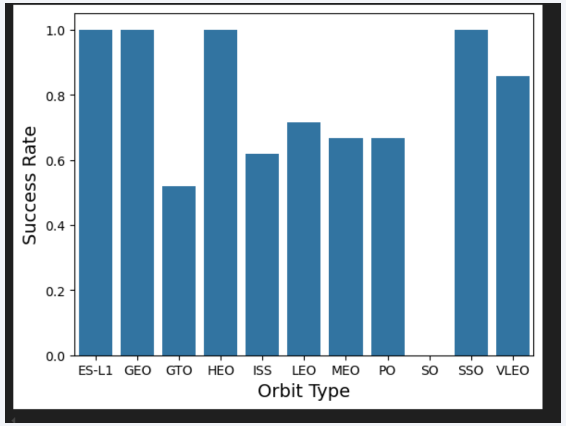


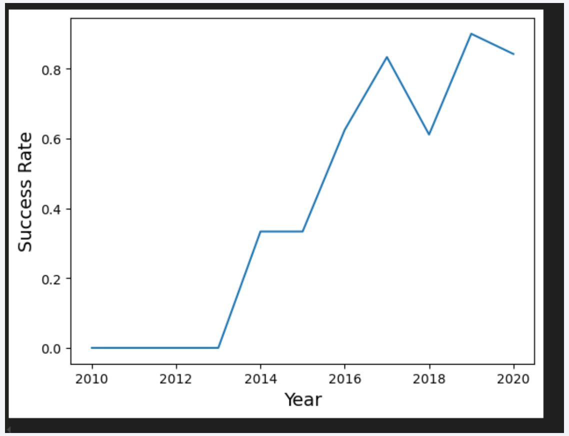
Cat Plot – Flight Number and Launch Site would affect the launch outcome, CCAFS LC-40 has the majority rocket launch and good success rates.



Cat Plot – Payload and Launch Site would affect the launch outcome. Low information for payload > 1000







Reason to use CAT Plot: Identify relationship between 2 variables and differentiate data by one category

Reason to use Bar Chart – Discrete numbers of ítens, or mean, count, max by category

Reason to use Line plot – A continuos data along the years

- GitHub Link https://github.com/Mazolli47/CourseraIBM/blob/main/BACKUP/05_edadataviz.ipynb

EDA with SQL

GitHub Link
https://github.com/Mazolli47/CourseraIBM/blob/main/BACKUP/04_jupyter-labs-eda-sql-coursera_sqllite.ipynb

- SQL queries you performed:
  - List of unique Launch Sites ,
```
Select distinct(LAUNCH_SITE) From SPACEXTBL
```
  - identifying total payload mass carried by boosters launched by NASA,
```
Select sum(PAYLOAD_MASS__KG_) AS 'Total Payload Mass (KG)' From SPACEXTBL Where CUSTOMER LIKE '%NASA (CRS)%'
```
  - 5 records where launch sites begin with CCA,
```
Select * From SPACEXTBL Where LAUNCH_SITE LIKE 'CCA%' LIMIT 5
```
  - Average payload mass carried by booster version F9 v1.1,
```
Select avg(PAYLOAD_MASS__KG_) AS 'Average Payload Mass (KG)' From SPACEXTBL Where BOOSTER_VERSION LIKE '%F9 v1.1%'
```
  - Date when the first successful landing outcome in ground pad was achieved,
```
Select min(DATE) From SPACEXTBL AS 'First Ground PAD Landing Date' Where Landing_Outcome LIKE '%Success% (ground pad)%'
```
  - boosters which have success in drone List the names of the booster_versions which have carried the maximum payload masse ship and have payload between 4000 and 6000 ,
```
Select distinct BOOSTER_VERSION From SPACEXTBL Where Landing_Outcome LIKE '%Success% (drone ship)%' and PAYLOAD_MASS__KG_ between 4000 AND 6000
```
  - List the total number of successful and failure mission outcomes
```
Select Mission_Outcome AS 'Mission Outcome', count(*) AS 'TOTAL' From SPACEXTBL  group by Mission_Outcome
```
  - List Fail outcomes , month names, drone ship ,booster versions, launch_site in 2015
```
%sql SELECT Landing_Outcome AS 'Landing Outcome', BOOSTER_VERSION AS 'Booster Version', LAUNCH_SITE AS 'Launch Site',  substr(Date, 6,2) AS 'Month',
'2015' AS 'Year' from SPACEXTBL where Landing_Outcome LIKE'%Failure% (drone ship)%' and substr(Date,0,5)='2015'
```
  - Landing outcomes between the date 2010-06-04 and 2017-03-20
```
%sql SELECT LANDING_OUTCOME AS 'LANDING OUTCOME' , count(LANDING_OUTCOME) AS 'Quantity' FROM SPACEXTBL
WHERE DATE BETWEEN '2010-06-04' AND  '2017-03-20'GROUP BY LANDING_OUTCOME  ORDER BY 'Quantity' DESC;
```

Build an Interactive Map with Folium

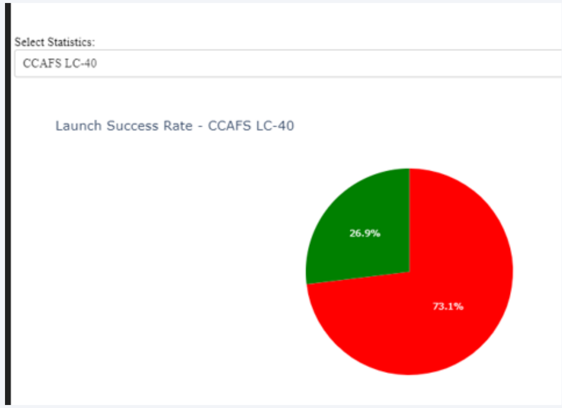
- Marked all launch sites visually on a map with latitude, longitude coordinates circled
- Marked the success/failed launches for each site, with Green and Red markers on the map
- Calculated the distance from the launch site to various land markers.
- Lines are drawn on the map to measure distance to landmarks

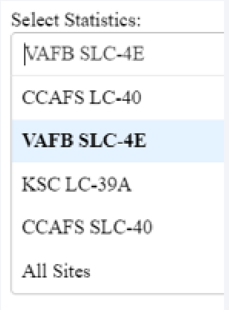
- This helps to answer below questions easily:
- *Are launch sites in close proximity to railways?*
- *Are launch sites in close proximity to highways?*
- *Are launch sites in close proximity to coastline?*
- *Do launch sites keep certain distance away from cities?*

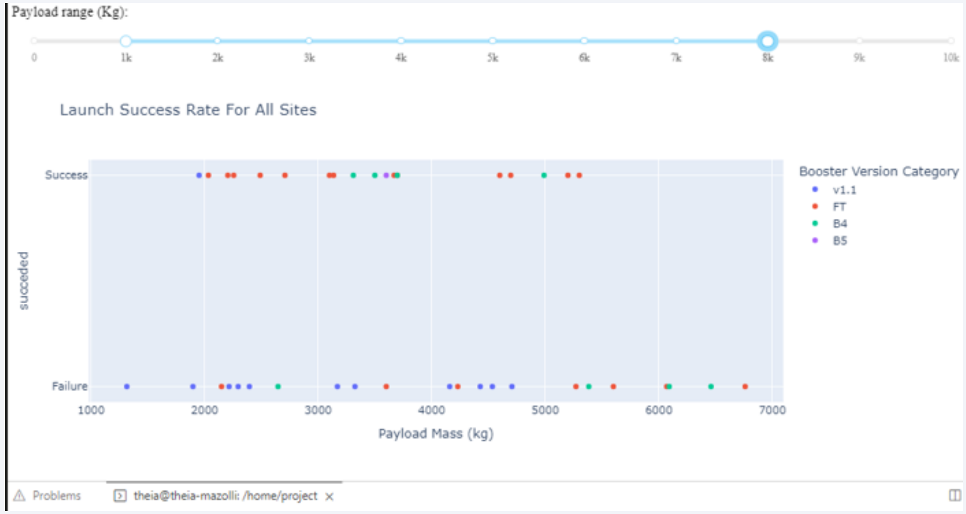
GitHub LINK https://github.com/Mazolli47/CourseraIBM/blob/main/BACKUP/06_lab_jupyter_launch_site_location.ipynb

Build a Dashboard with Plotly Dash

- Plotly is a python wrapper on the JavaScript library 'leaflet'. It enables us to interact with our data visualizations and host it as a website
- Dash has a full HTML objects interaction witch allow to:
- Plot a Pie Chart: That shows number of launches from each launch site as well as
- number of successful and failed launches from those sites
- Callback function for `site dropdown` as input, `success pie chart` as output
- Callback function for `site dropdown` and `payload slider` as inputs,and `success payload scatter chart` as output
- Scatter Graph: Relationship between the success of a launch (Outcome)
- and Payload (in kg) for different versions of boosters







- GitHub LINK https://github.com/Mazolli47/CourseralBM/blob/main/BACKUP/07_spacex_dash_app%20preenchendo_Final.py

Predictive Analysis (Classification)

- Predicting if the first stage will land: normalizing data, train K Nearest Neighbors, Decision Tree, Logistic Regression, Vector Machine models and evaluating model performance in a train dataset

Building Model

- Load dataset into NumPy and Pandas and Transform data.
- Clean / Fill missing data
- Split data into Training and Test datasets
- Standardize/Normalize all the parameters

Evaluating Model

- Check accuracy for each model
- Get tuned hyperparameters for diferent algorithms
- Analise each Confusion matrix for false negative or false positive

Improvimg Model

-Use feature engineering and model tuning

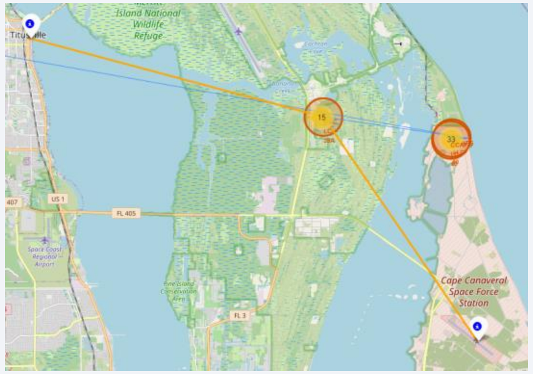
Finding the best model

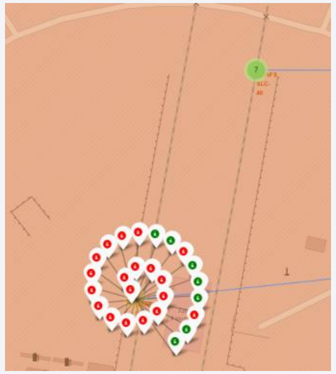
• comparing:
- weighted accurance
- F1 score
- Recall
-Jackard index
- Log Loss (statistical models)

GitHub LINK
https://github.com/Mazolli47/CourseraIBM/blob/main/BACKUP/08_SpaceX_Machine%20Learning%20Prediction_Part_5.ipynb

Results

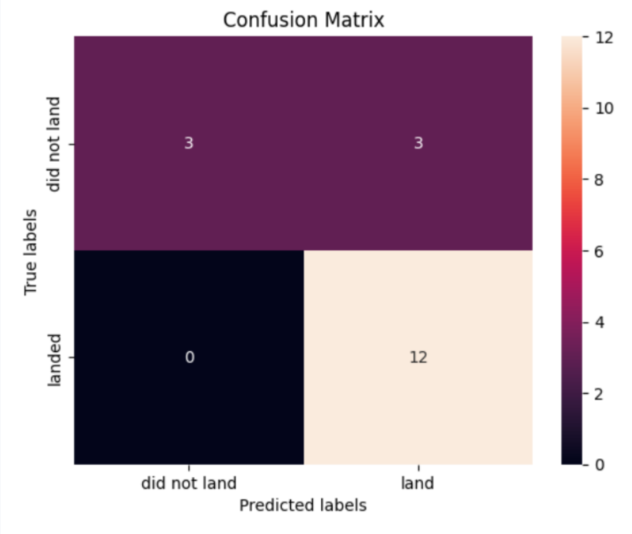
- Exploratory data analysis results – Site Launch and Payload weight had the most influence in launch succes;.
- Interactive analytics demo in screenshots – Sites distances to locations and land classification





Predictive analysis results – KNN , Logistic Regression and SVM had the same good performance. Accuracy 83.3%, Jaccard 0.8, F1 score 0.85, and same Confusion Matrix

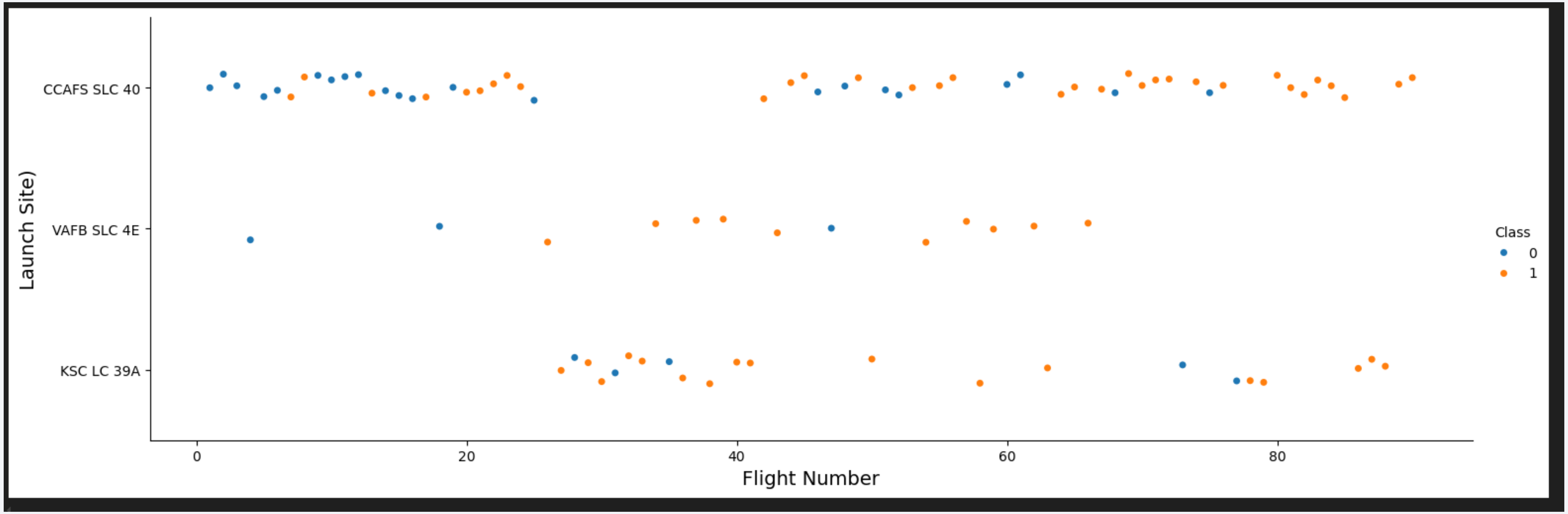
Evaluation Method	Accuracy Score	F1 Score	Jaccard Index	Log Loss
Model				
KNN	0.8333333333333334	0.8518518518518519	0.8	NaN
Logistic Regression	0.8333333333333334	0.8518518518518519	0.5	0.4786666968559154
SVM	0.8333333333333334	0.8518518518518519	0.8	NaN
TREE	0.8333333333333334	0.8518518518518519	0.8	NaN



Section 2

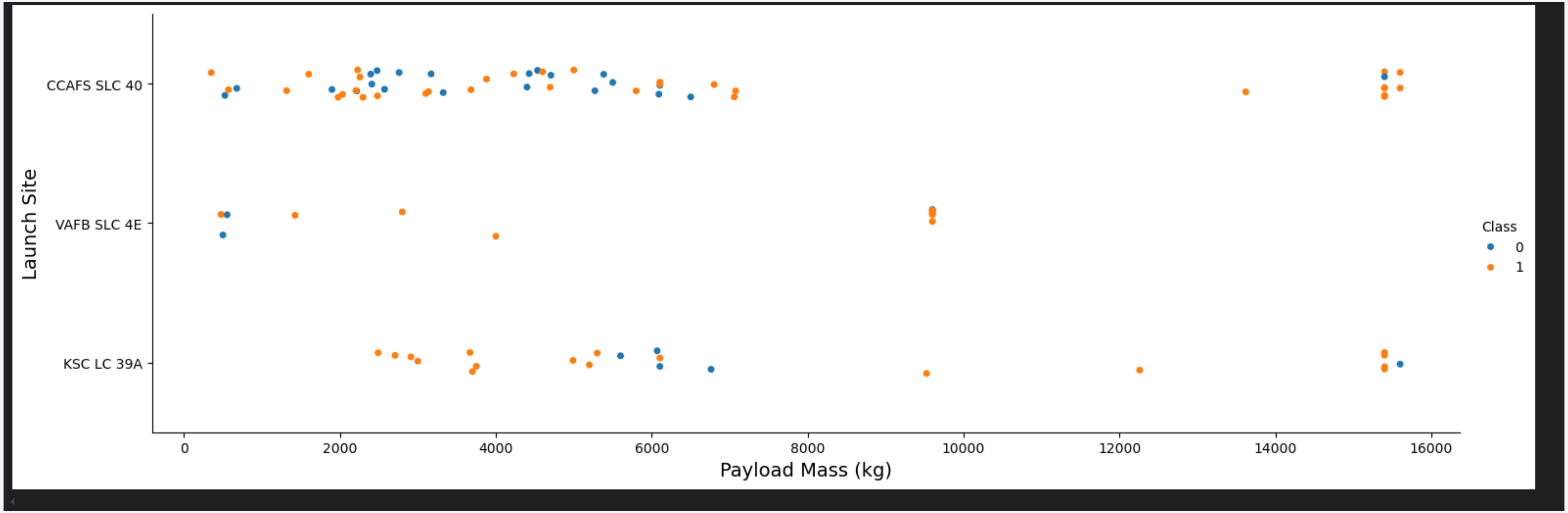
Insights drawn from EDA

Flight Number vs. Launch Site



CCAFS SLC 40 Site has the majority of launch, as consequence, has the majority of success and fails. On the beginning (lower flight numbers) the success rate was very low but lately the success rate is very high. Success rate increases in newer launch

Payload vs. Launch Site

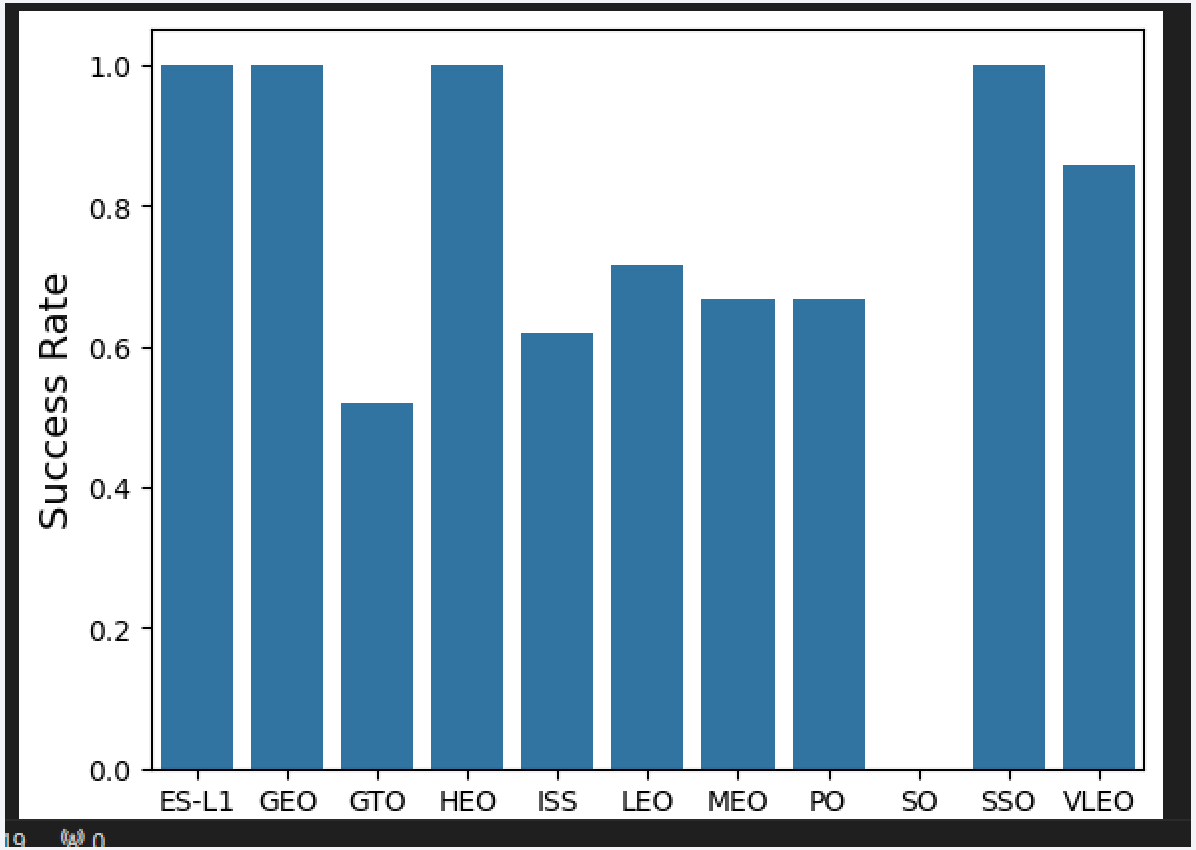


- VAFB SLC 4E Site dos not launches rocket with payload above 10.000, most fails are below 8000 and on CCAFS SLC 40 SITE

Success Rate vs. Orbit Type

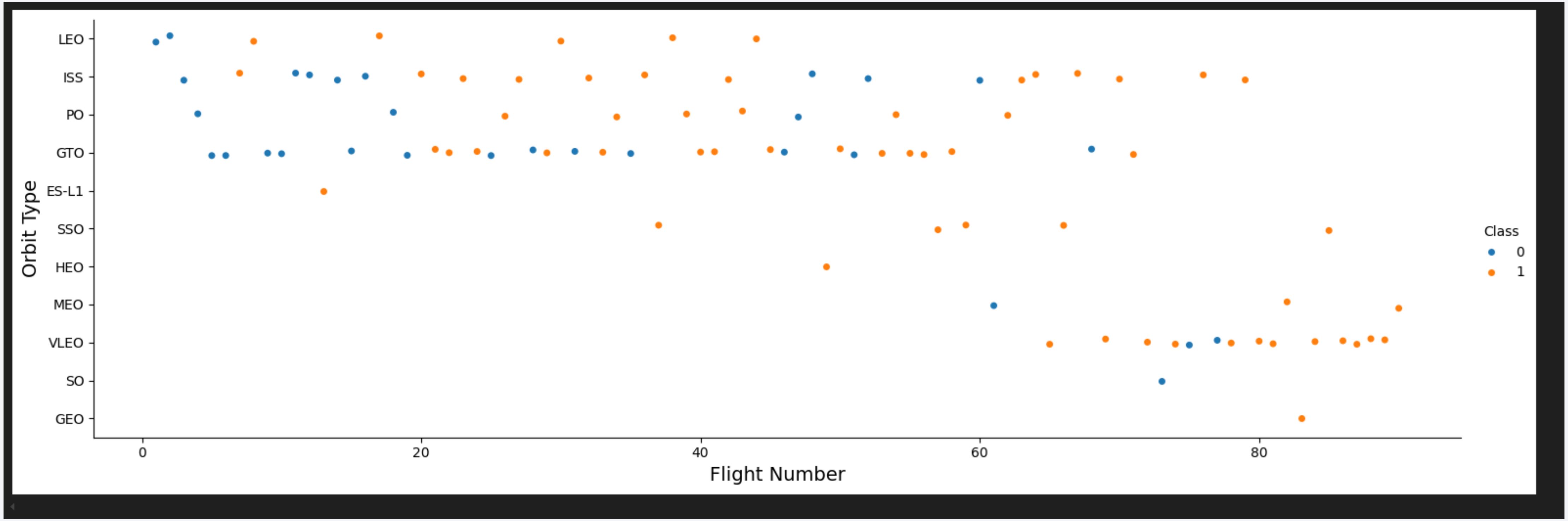
- Not considering Satellite Orbit GEO, SO, HEO, ES-L1 with only one launch each, SSO and VLEO has the highest Success Rate above 80%

```
GTO       27
ISS       21
VLEO      14
PO         9
LEO        7
SSO        5
MEO        3
ES-L1      1
HEO        1
SO         1
GEO        1
Name: Orbit, dtype: int64
```



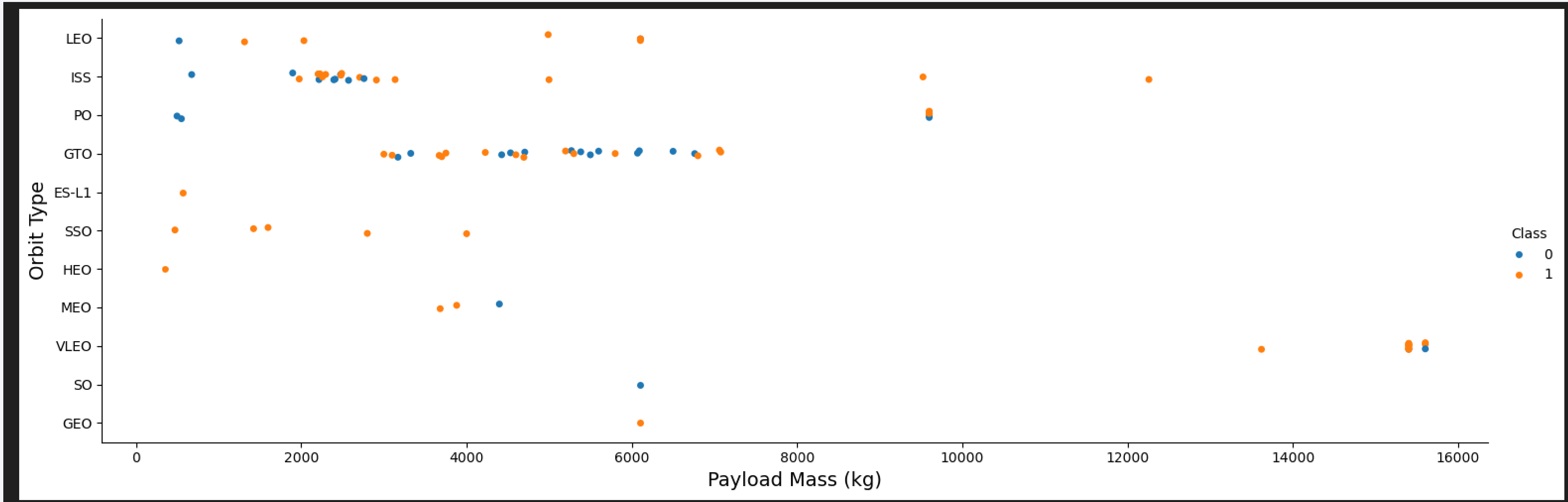
Best Rates

Flight Number vs. Orbit Type




```
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 vs. Orbit Type

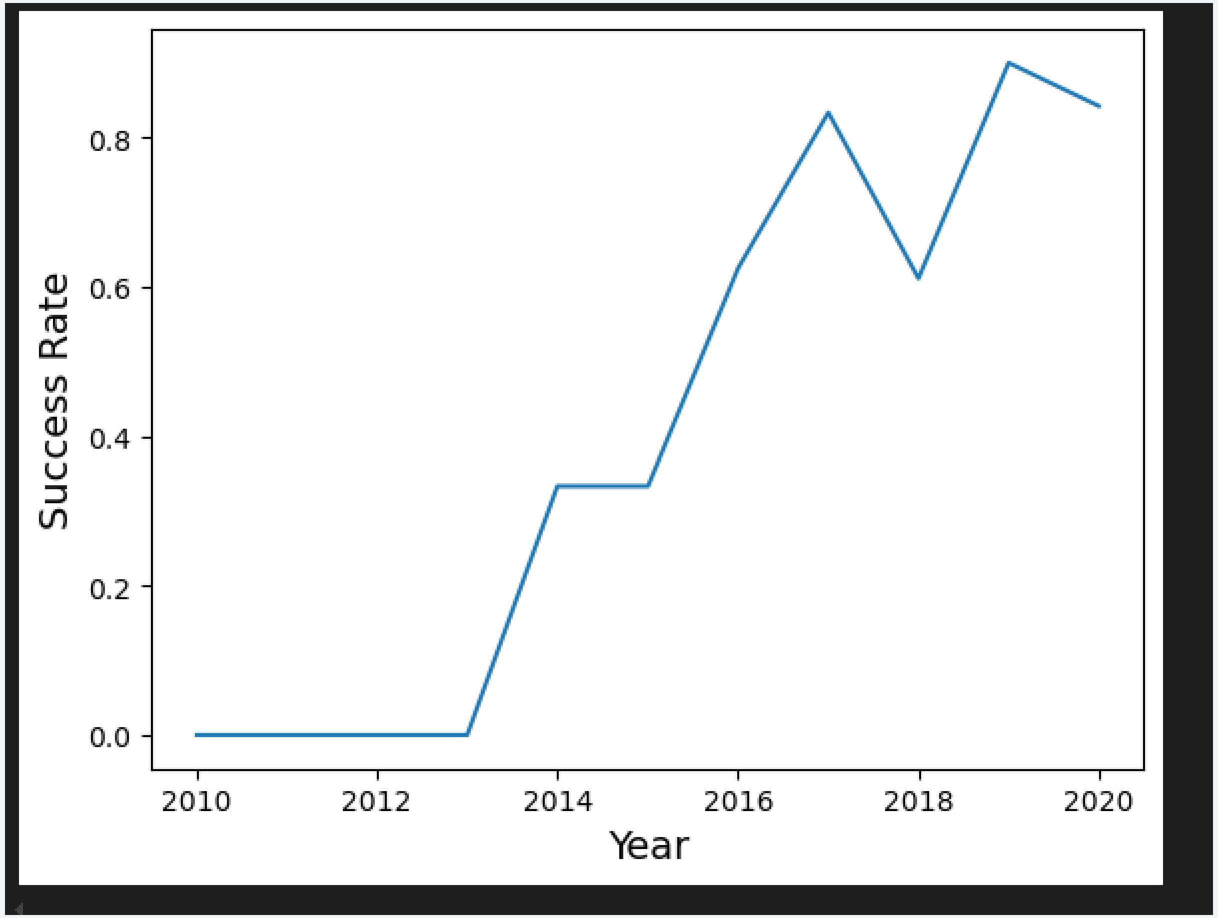



```
With heavy payloads the successful landing or positive landing rate are more for Polar,LEO and ISS.

However for GTO we cannot distinguish this well as both positive landing rate and negative landing(unsuccessful mission) are both there here.
```

Launch Success Yearly Trend

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



All Launch Site Names

```
%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

- Selecting only the Launch Site name without repetitions from the table SPACEXTBL inside my_data1.db
- Result: Only 4 Locations in the database

Launch Site Names Begin with 'CCA'

```
%sql Select * From SPACEXTBL Where LAUNCH_SITE LIKE "CCA%" LIMIT 5

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

- Like function to utilize REGEXP , in this case beginning with CCA; LIMIT to show the first 5 Rows
- Result : Only Showing CCAFS LC-40 in the first 5 rows , CCAFS SLC-40 is not showing and may appear in higher rows number

Date	Time (UTC)	Booster_Version	Launch_Site	Payload	PAYLOAD_MASS__KG_	Orbit	Customer	Mission_Outcome	Landing_Outcome
2010-06-04	18:45:00	F9 v1.0 B0003	CCAFS LC-40	Dragon Spacecraft Qualification Unit	0	LEO	SpaceX	Success	Failure (parachute)
2010-12-08	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)
2012-05-22	7:44:00	F9 v1.0 B0005	CCAFS LC-40	Dragon demo flight C2	525	LEO (ISS)	NASA (COTS)	Success	No attempt
2012-10-08	0:35:00	F9 v1.0 B0006	CCAFS LC-40	SpaceX CRS-1	500	LEO (ISS)	NASA (CRS)	Success	No attempt
2013-03-01	15:10:00	F9 v1.0 B0007	CCAFS LC-40	SpaceX CRS-2	677	LEO (ISS)	NASA (CRS)	Success	No attempt

Total Payload Mass

```
%sql Select sum(PAYLOAD_MASS__KG_) AS "Total Payload Mass (KG)" From SPACEXTBL Where CUSTOMER LIKE '%NASA (CRS)%'
```

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

Total Payload Mass (KG)
48213

- **Total payload mass carried by boosters launched by NASA (CRS).**
- Funcion SUM() to sum payload column only where customer name has “NASA (CRS)” in the beginning , middle or end. Renaming sum result as Total Payload Mass(KG)

Average Payload Mass by F9 v1.1

```
%sql Select avg(PAYLOAD_MASS__KG_) AS "Average Payload Mass (KG)" From SPACEXTBL Where BOOSTER_VERSION LIKE '%F9 v1.1%'
```

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

Average Payload Mass (KG)
2534.6666666666665

- Calculate the average payload mass carried by booster version F9 v1.1
- Funcion AVG() to calculate the mean payload only where booster version name has “F9 v1.1” in the beginning , middle or end.
- Renaming AVG result as Average Payload Mass(KG)

First Successful Ground Landing Date

```
%sql Select min(DATE) From SPACEXTBL AS "First Ground PAD Landing Date" Where Landing_Outcome LIKE '%Success% (ground pad)%'
```

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

min(DATE)
2015-12-22

- Find the dates of the first successful landing outcome with the internal MIN() function

Successful Drone Ship Landing with Payload between 4000 and 6000

```
%sql Select distinct BOOSTER_VERSION From SPACEXTBL Where Landing_Outcome LIKE "%Success% (drone ship)%" and PAYLOAD_MASS__KG_ between 4000 AND 6000

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

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

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

- List the names of boosters which have successfully landed on drone ship and had payload mass greater than 4000 but less than 6000
- Using database internal Functions Distinct (no repetitions), LIKE and Between

Total Number of Successful and Failure Mission Outcomes

```
%sql Select Mission_Outcome AS "Mission Outcome", count(*) AS "TOTAL" From SPACEXTBL  group by Mission_Outcome
```

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

Mission Outcome	TOTAL
Failure (in flight)	1
Success	98
Success	1
Success (payload status unclear)	1

- Calculate the total number of successful and failure mission outcomes
- Here we use count() to count columns grouping by Mission Outcome. Note we have 100 success (different writing forms in the database) and 1 failure

Boosters Carried Maximum Payload

```
%sql Select distinct BOOSTER_VERSION From SPACEXTBL Where PAYLOAD_MASS__KG_ = (select max(PAYLOAD_MASS__KG_) from SPACEXTBL)
```

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

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

Subquery to calculate maximum payload and a principal query to select whithout repetition booster version whitcp payload is the maximum

2015 Launch Records

```
%sql SELECT Landing_Outcome AS "Landing Outcome", BOOSTER_VERSION AS "Booster Version", LAUNCH_SITE AS "Launch Site",  substr(Date, 6,2) AS "Month", "2015" AS "Year"
from SPACEXTBL where Landing_Outcome LIKE'%Failure% (drone ship)%' and substr(Date,0,5)='2015'
```

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

Landing Outcome	Booster Version	Launch Site	Month	Year
Failure (drone ship)	F9 v1.1 B1012	CCAFS LC-40	01	2015
Failure (drone ship)	F9 v1.1 B1015	CCAFS LC-40	04	2015

- List the failed landing_outcomes in drone ship, their booster versions, and launch site names for in year 2015
- Diferently in others databases management system , SQL LITE dos no have date, month, year interna functions. In this cas was necessary to use SUBSTR () to split date columns in substrings
- LIKE internal function was used to find two pieces of word “Failure” in the beginning or middle of the Landing Outcome and “ (drone ship)” in the middle (after Failure) or in the end , trying to catch typo errore

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

```
%sql SELECT LANDING_OUTCOME AS "LANDING OUTCOME" , count(LANDING_OUTCOME) AS "Quantity" FROM SPACEXTBL
WHERE DATE BETWEEN '2010-06-04' AND  '2017-03-20'GROUP BY LANDING_OUTCOME  ORDER BY "Quantity" DESC;
```

[66]

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

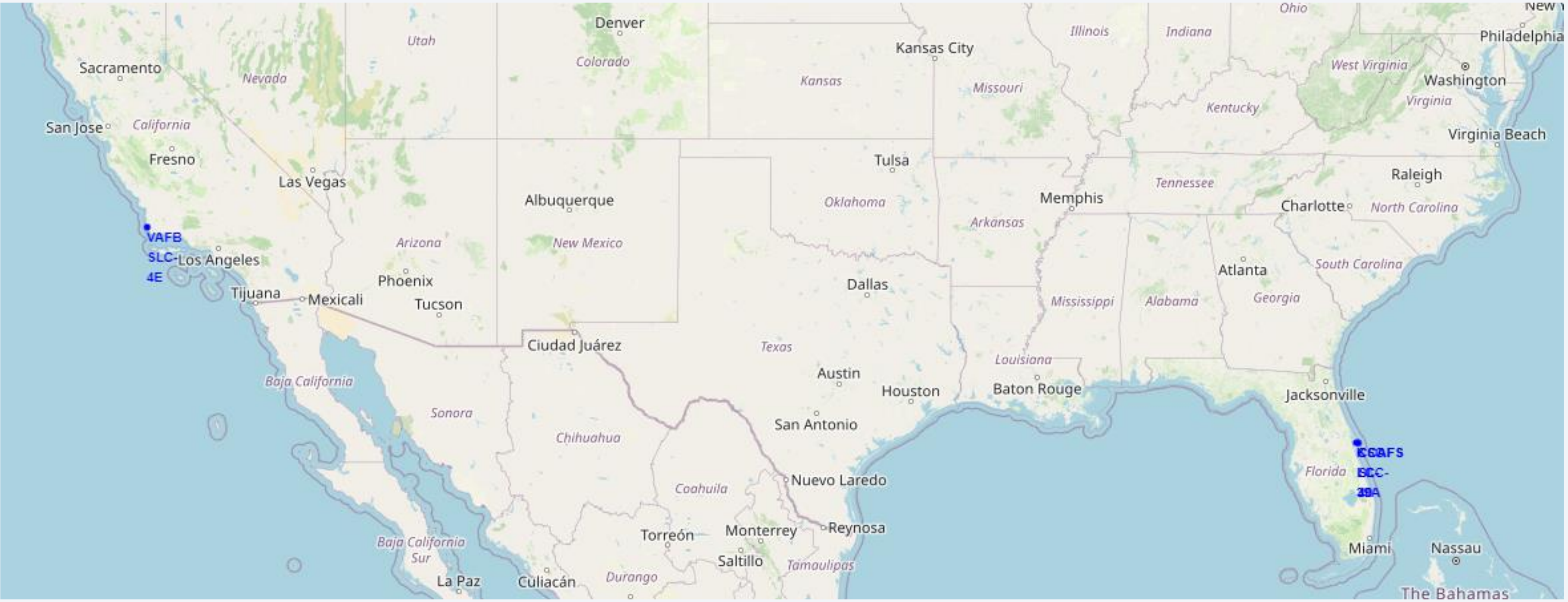
LANDING OUTCOME	Quantity
No attempt	10
Success (drone ship)	5
Failure (drone ship)	5
Success (ground pad)	3
Controlled (ocean)	3
Uncontrolled (ocean)	2
Failure (parachute)	2
Precluded (drone ship)	1

- 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
- Using Internal function count() ,Order By, Between and ordering the result set (Order By) by the Quantity column in a descendent way (DESC)

Section 3

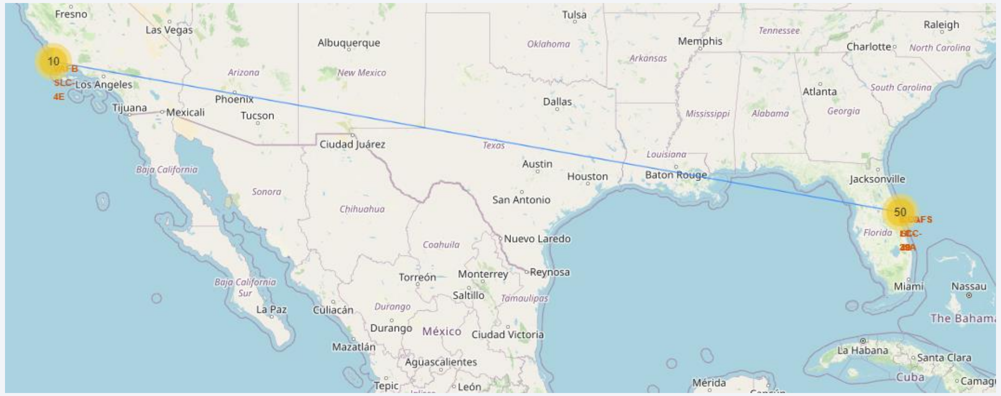
Launch Sites Proximities Analysis

Launch Sites On A Map

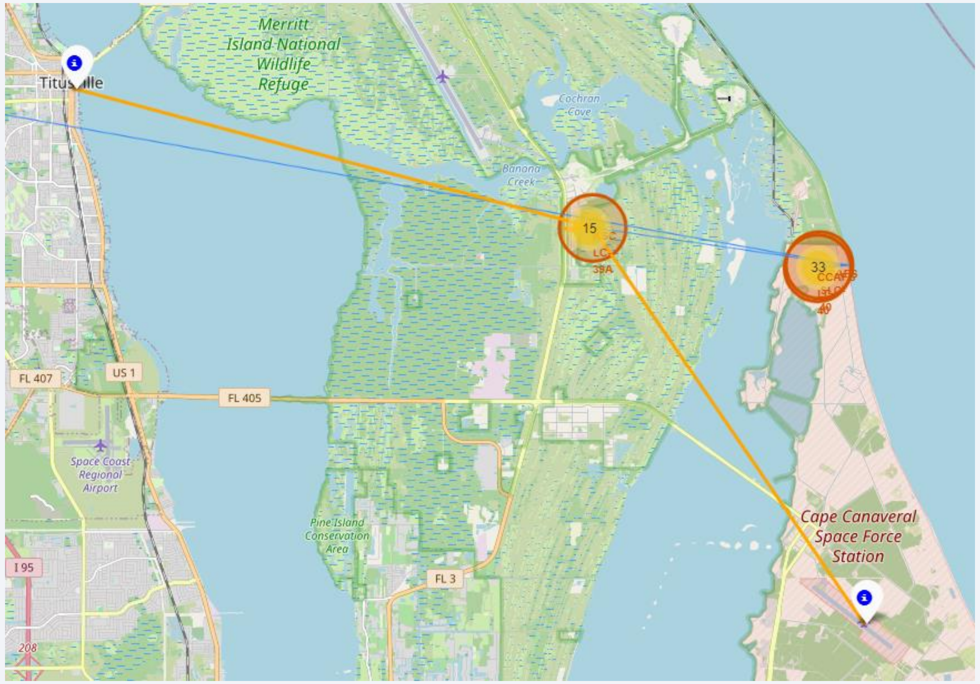


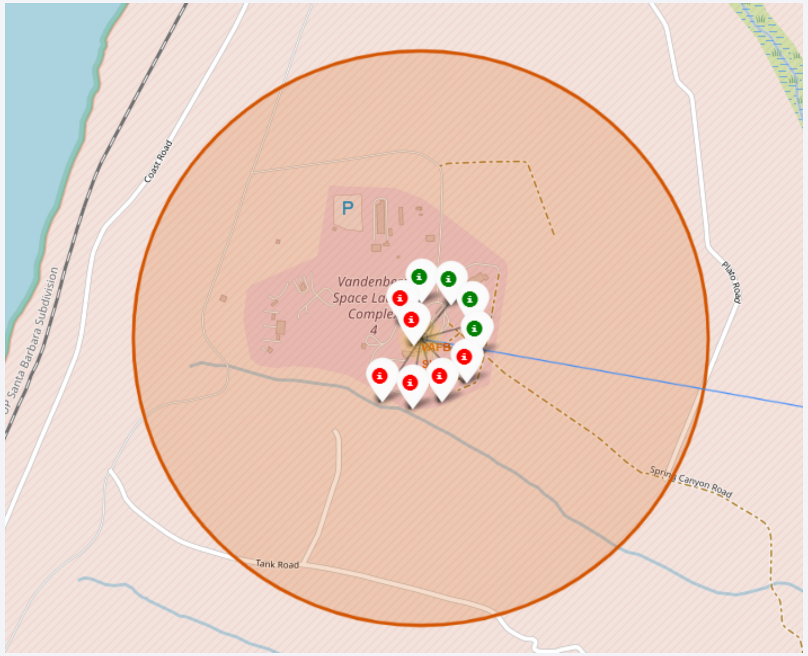
“SpaceX launch sites are in– Florida on the USA east coast and California on the USA west coast”

Sites locations, Distance And Landed Outcomes On A Map



- *Green colored are the successful landed and red colored are the failed ones”*
- *Lines represent calculated distances Site- Site (blue lines) and Site – Interest Locations (Orange Lines)*
- *Circle numbers represent Total Land outcomes in a region*





PÚBLICA

Launch Site Proximities

- Coast Proximity

- Road / Rail Way Proximity

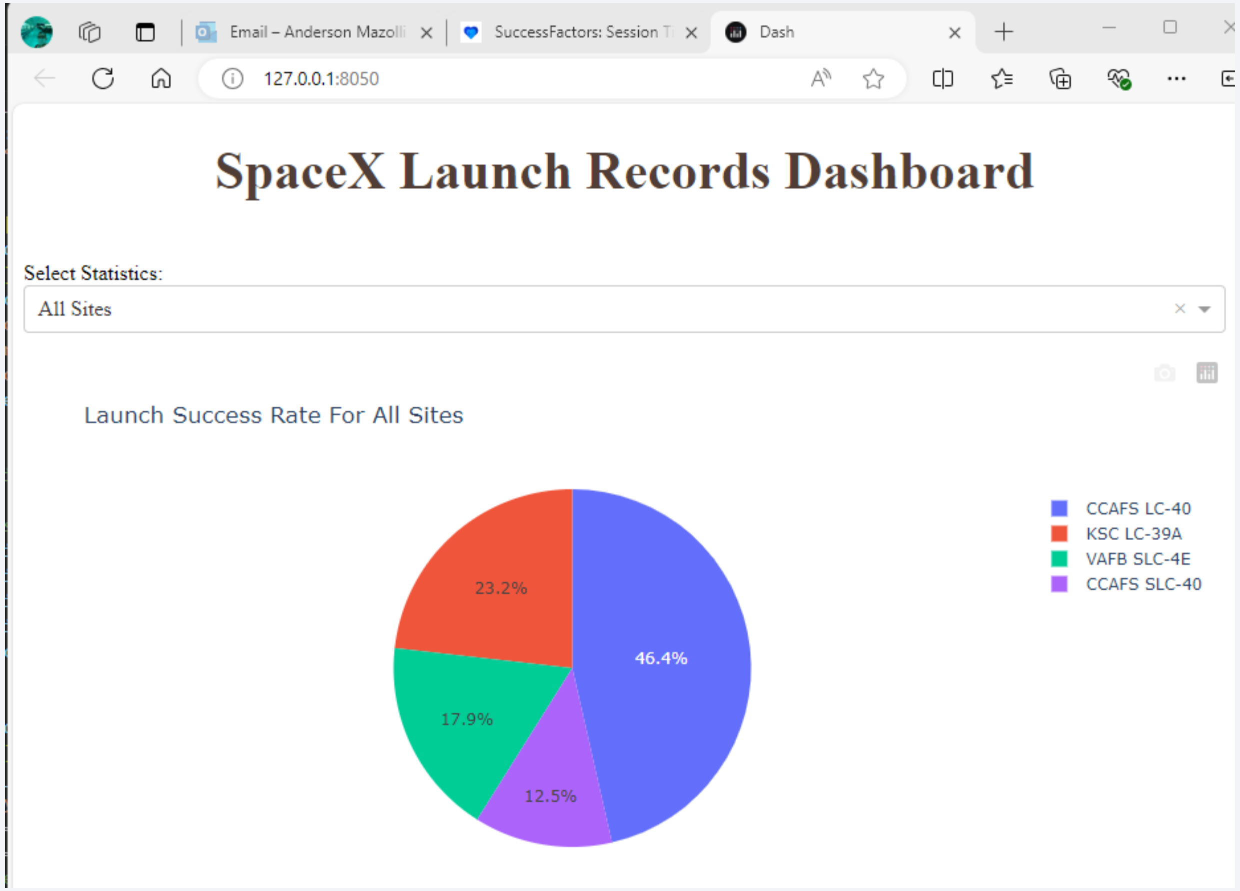
Here, we can observe distance of launch sites from east coast, highways, key road, railway line visualized"

Section 4

Build a Dashboard with Plotly Dash

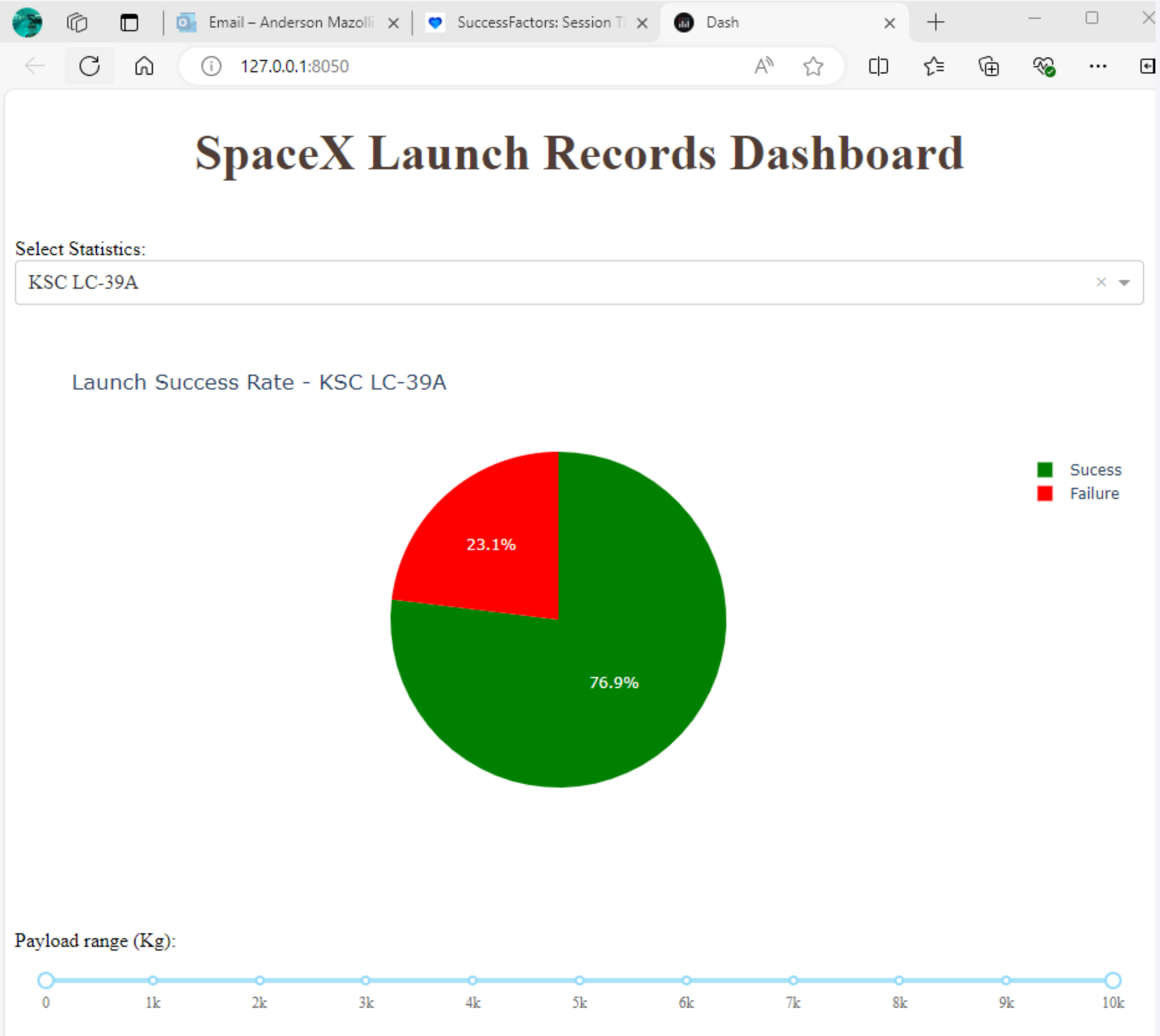
Dashboard – Launch Success Rate for All Sites

- CAFS LC-40 has the majority of Success launch in the total success launch , 46,4%.
- As noted before , CAFS LC-40 has the majority of total launch in the dataset , as a consequence it has the highest number of sucess



Most Successful Launch Site

- Considering only launch per Site, KSC LC-39A has the Highest success rate of 76,9 %



<Dashboard Screenshot 3>

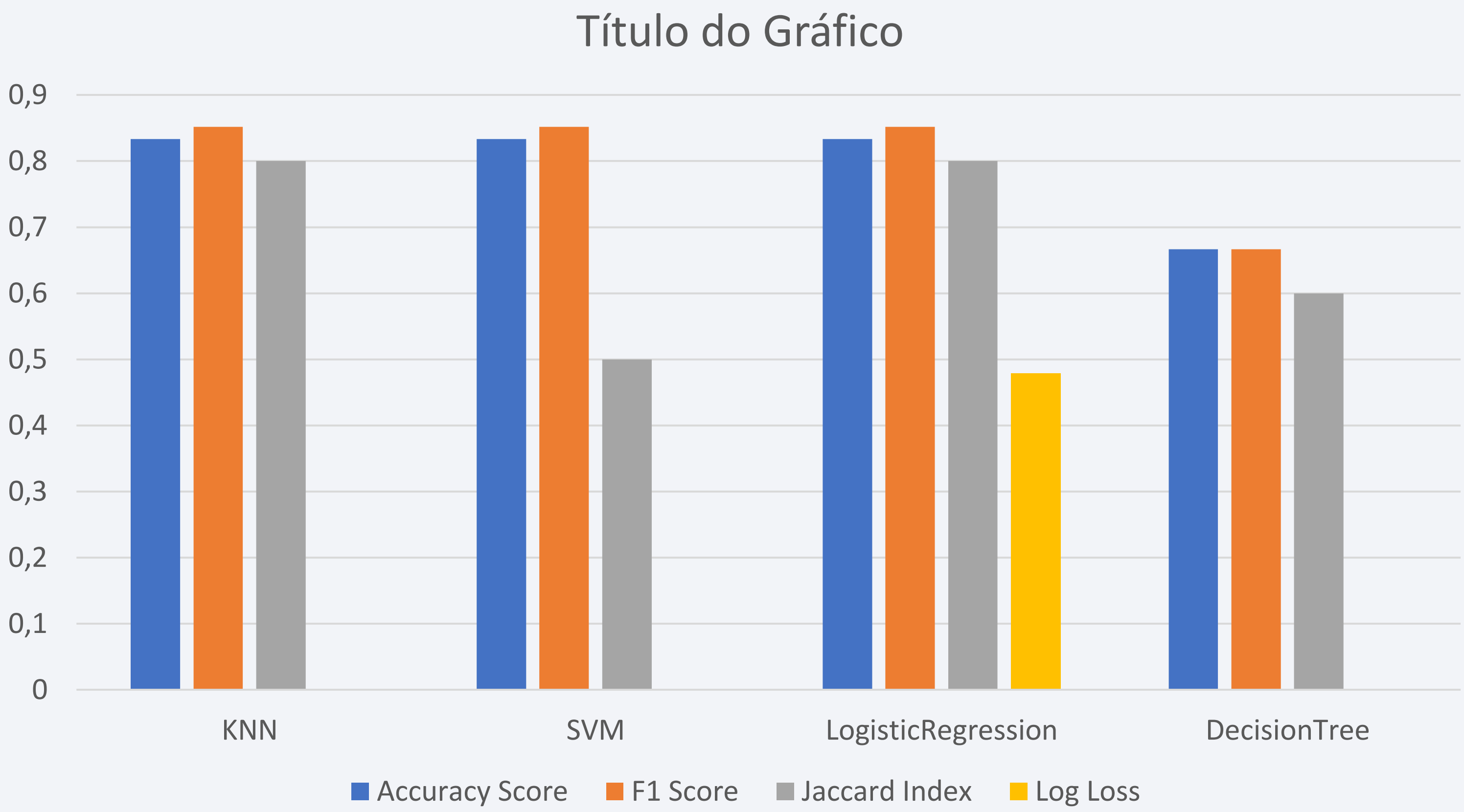
- Rocket with FT booster has more success
- Most Success occurs when payload is between 2000 and 5000
- Most Failure occurs when payload is between 4000 and 7000
- Insuficient data above 8000

Section 5

Predictive Analysis (Classification)

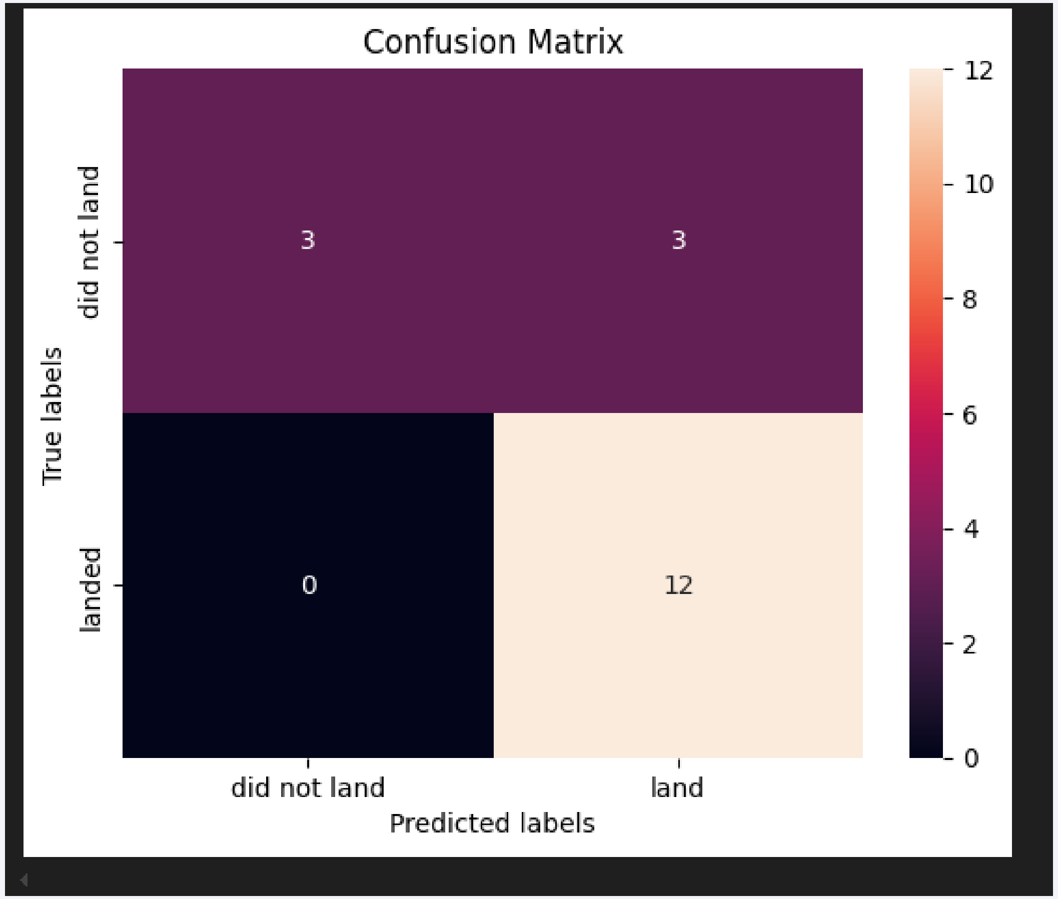
Classification Accuracy

- *KNN, Support Vector & Logistic Regression Methods have high accuracy, SVM achieve a lower Jackard Index in some test datasets so if possible choose KNN or Logistic Regretion .*
- *Logistic Regretion allows Statitics inference as LOG LOSS*
- *Decision Tree in this casem is not as good as the other ones*



Confusion Matrix

- *KNN,Logistic Regression & SVM have predict correctly 15 of 18 tested launches.*
- *The 3 model predict correctly all 12 sucessul launches whith none false negative*
- *All the error is in false positive area , 3 fails are predicted as a sucess*



Conclusions

- ✓ KNN, Logistic Regression and SVM are the best classifier models for this dataset
- ✓ CAFS LC-40 Site lauched the majority Falcon 9 rockets, a lot of fails in the beginning and a higher success rate in the recently launchs
- ✓ KSC LC-39A Site has the Highest success rate of 76,9 %
- ✓ FT Booster have the highest launch success rates
- ✓ Most Success occurs when payload is lower (between 2000 and 5000)
- ✓ The SpaceX launches have been continuously getting better from year 2013 to 2020

Appendix

- Include any relevant assets like Python code snippets, SQL queries, charts, Notebook outputs, or data sets that you may have created during this project

Thank you!