IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

Rafael Pachón Alvarez
November 2021

Outline

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

Executive Summary

Summary of Methodologies

- Data
  - Collection
  - Wrangling
- EDA
  - with data visualization
  - With SQL
- Building
  - Interactive map (Folium)
  - Interactive Dashboard (Plotly)
- Classification (Predicitive Analysis

Summary of all results

- Exploratory data análisis
- Interective Analytics
- Predictive results

Introduction

- Project background and context
  - Goal: Predict if Falcon9 first stage will land successfully.
  - Reuse first stage leads to a saving (165M$ vs 62$M)
- Problems you want to find answers
  - What make the rocket successfully land

Section 1

Methodology

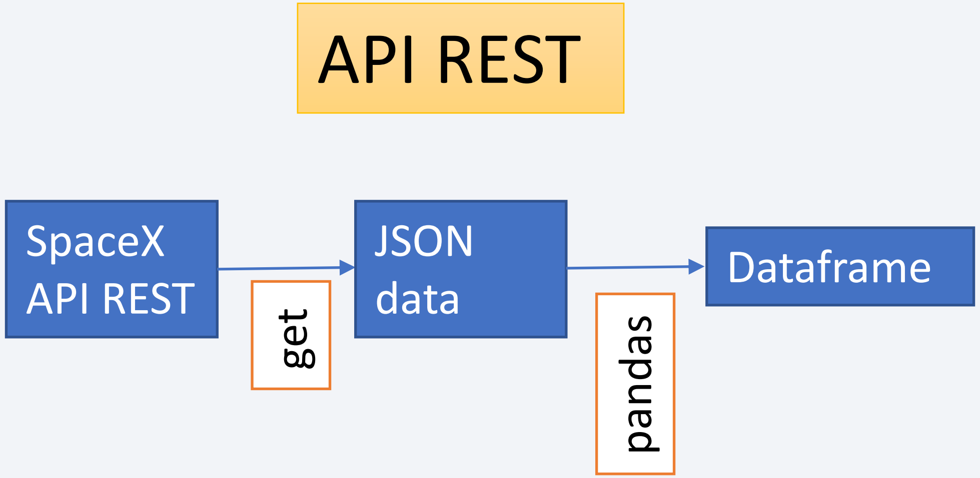
Methodology

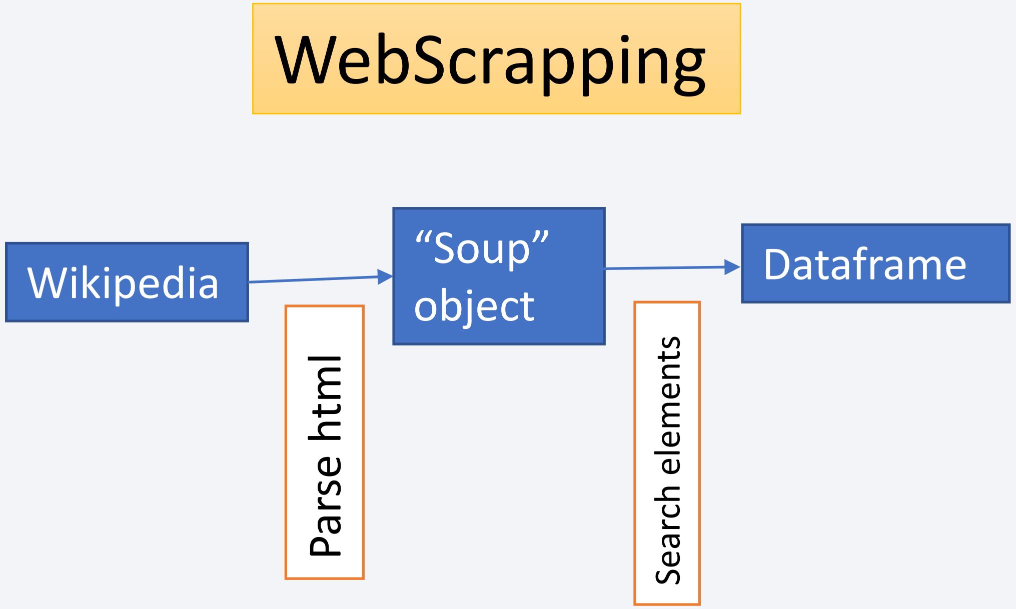
Executive Summary

- Data collection methodology:
  - SpaceX API REST
  - Data from Wikipedia (Webscrapping – Beautifullsoup)
- Perform data wrangling
  - Normalize data
  - Select relevant features
  - One hot encoding for categorical variables
- Perform exploratory data analysis (EDA) using visualization and SQL
  - Scatterplots, Barcharts, LinePlot
- Perform interactive visual analytics using Folium and Plotly Dash
- Perform predictive analysis using classification models
  - Model tunning through GridSearch

Data Collection

- Describe how data sets were collected.
  - Data from SpaceX is fetched with API REST (api.spacexdata.com/v4) [GitHub: Data collection API]
  - Data from Rocket is fetched with WebScrapping (BeautifulSoup) from Wikipedia [GitHub: Data Collection WebScrapping]





Data Collection – SpaceX API

Fetch data from API

```
spacex_url="https://api.spacexdata.com/v4/launches/past"
response = requests.get(spacex_url).json()
```

JSON to dataframe

```
response = requests.get(static_json_url).json()
data = pd.json_normalize(response)
```

Apply functions to clean data

```
getLaunchSite(data)
getPayloadData(data)
getCoreData(data)
getBoosterVersion(data)
```

Create Dictionary

```
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}
```

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

Filter data for Falcon9

```
data_falcon9 = df.loc[df['BoosterVersion']!="Falcon 1"]
```

Data Collection - Scraping

Get data from URL

```
page = requests.get(static_url)
```

Parse HTML

```
soup = BeautifulSoup(page.text, 'html.parser')
```

Get Column Names

```
column_names = []
temp = soup.find_all('th')
for x in range(len(temp)):
    try:
     name = extract_column_from_header(temp[x]
     if (name is not None and len(name) > 0):
        column_names.append(name)
    except:
     pass
```

Create Dictionary

```
launch_dict= dict.fromkeys(column_names)

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


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'] = []
launch_dict['Version Booster']=[]
launch_dict['Booster landing']=[]
launch_dict['Date']=[]
launch_dict['Time']=[]
```

Dictionary to dataframe

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

Data Wrangling

- There are some different situations where rocket booster successfully landed or not.
- To process data, it is assigned a Class with 0 for unsuccessfull landing and 1 for successfull landing

Landing	
True Ocean/False Ocean	Success/Failure landing in the Ocean
True RTLS/False RTLS	Success/Failure landing on the Ground
True ASDS/False ASDS	Success/Failure landing on a Drone Ship

EDA on dataset Overview

Calculate number of launches at each site

Calculate number of launches at each orbit

Calculate Landing Outcome

Calculate Landing Outcome

Orbit types

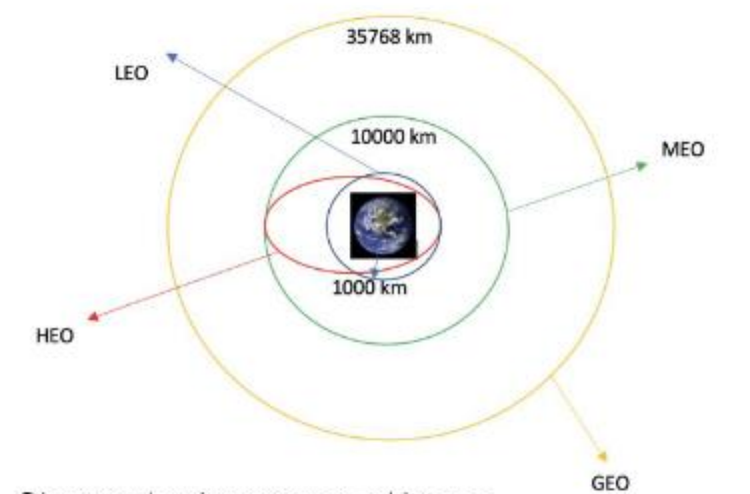


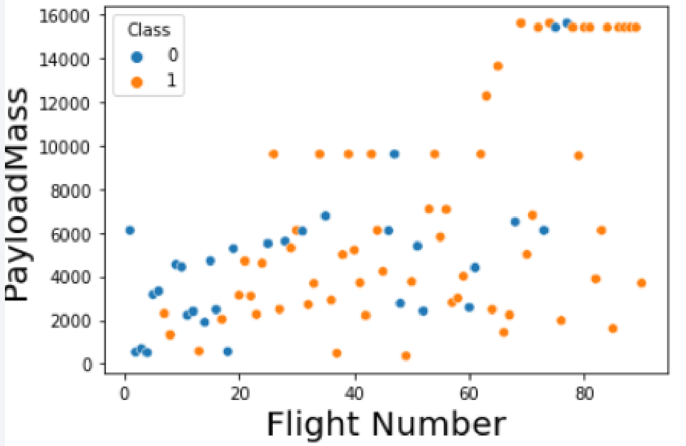
Diagram showing common orbit types SpaceX uses

EDA with Data Visualization

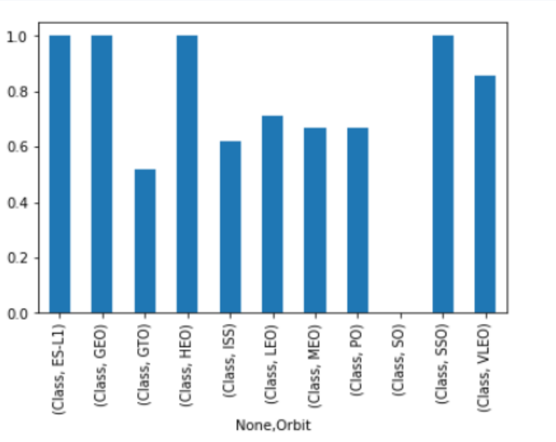
Several plots has been added to the EDA such as

- Scatter Plots
- BarCharts
- Line Plot

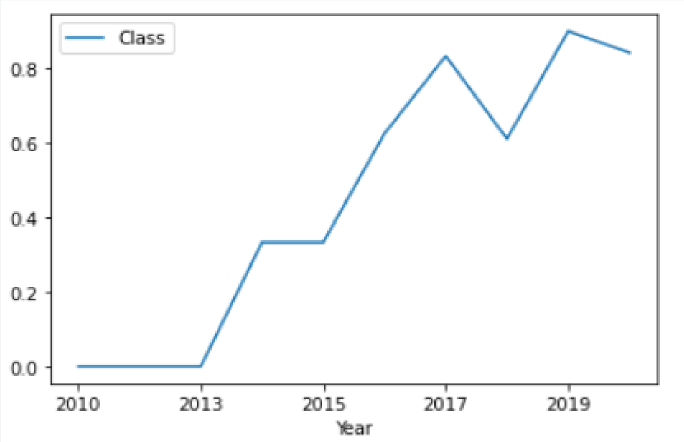
ScatterPlot



BarChart



LineCharts



As a comment:

- Landing are better since 2013 and with number of flights (know-how)
- With heavy payloads the successful landing rate is increasing and LEO and ISS Orbits(see notebook for details)

[Github. EDA with Python]

EDA with SQL

- As an example of EDA with SQL, the following queries have been performed:
  - *Display the names of the unique launch sites in the space mission*
  - *Display 5 records where launch sites begin with the string 'CCA'¶*
  - *Display the total payload mass carried by boosters launched by NASA (CRS)¶*
  - *Display average payload mass carried by booster version F9 v1.1*
  - *List the date when the first successful 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. Use a subquery*
  - *List the failed landing_outcomes in drone ship, their booster versions, and launch site names for in year 2015*
  - *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*

[Github: EDA with SQL]

Build an Interactive Map with Folium

- Interactive map with launch data. With lat/lon of each launch it is placed a circle.
- Each launch is coloured in Green (Success, class=1) or Red (Failure, class=0)
- With Haversine’s formula, it is calculated distance from launch site to other references:
  - Coast
  - Highway
  - City
- Github: Interactive Map with Folium

Build a Dashboard with Plotly Dash

- It is shown in a dashboard a pie chart with the proportion of Success/Failure depending on launching site.
  - **Goal:** Check if the launching site influences the success ratio
- It is also shown a scatterplot with the Success/Failure depending on the Payload weight. It is also possible to filter Paiload weight interval. Data is presented parametrize by booster type
  - **Goal**: Evaluate if the weight of the payload and booster type has influence in success ratio
- Github: Interactive dashboard

Predictive Analysis (Classification)

Prepare data → Create Models → Evaluate Models → Tune Model → Select Best Model

Prepare data
- Assign Class 0/1
- Normalize
- Split train/test data

Create Models
- Logistic Regression
- SVM
- Decission Tree
- Knn

Evaluate Models
- Score (test set)
- Confusion Matrix

Tune Model
- Grid Search

Select Best Model

Github: Predictive Model

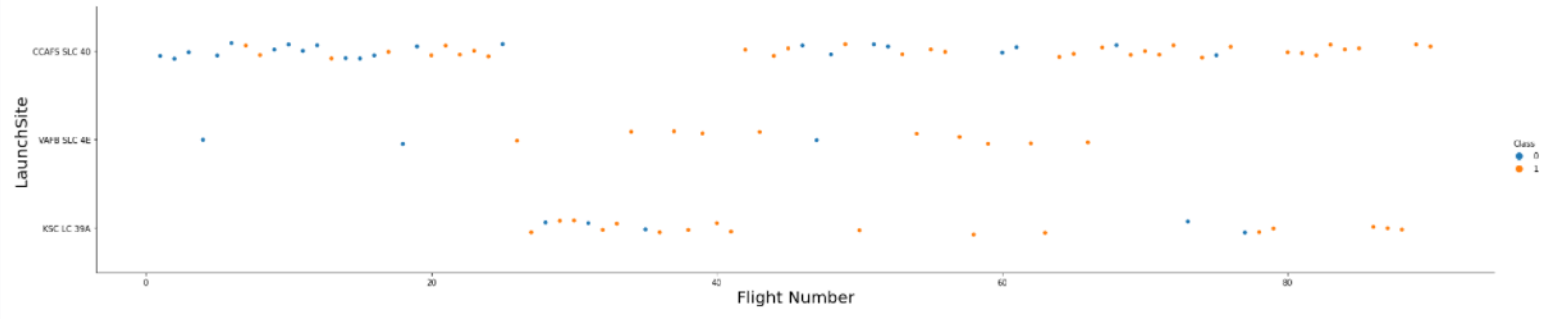
Results

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

Section 2

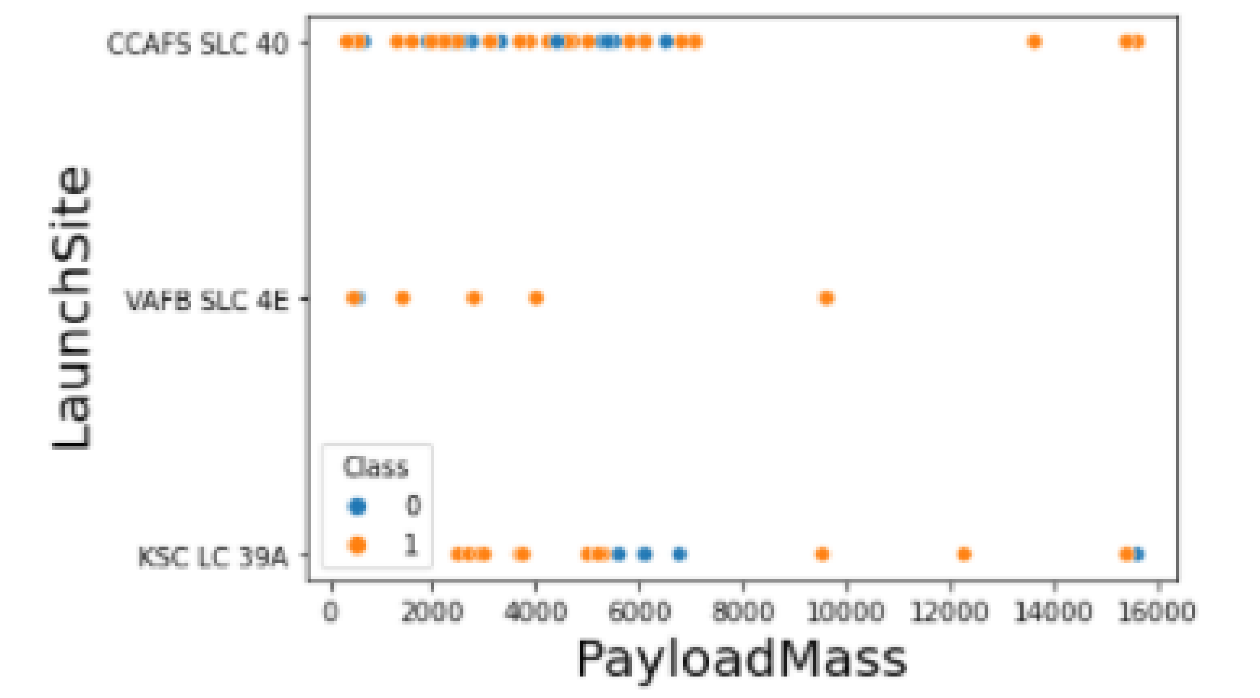
Insights drawn from EDA

Flight Number vs. Launch Site



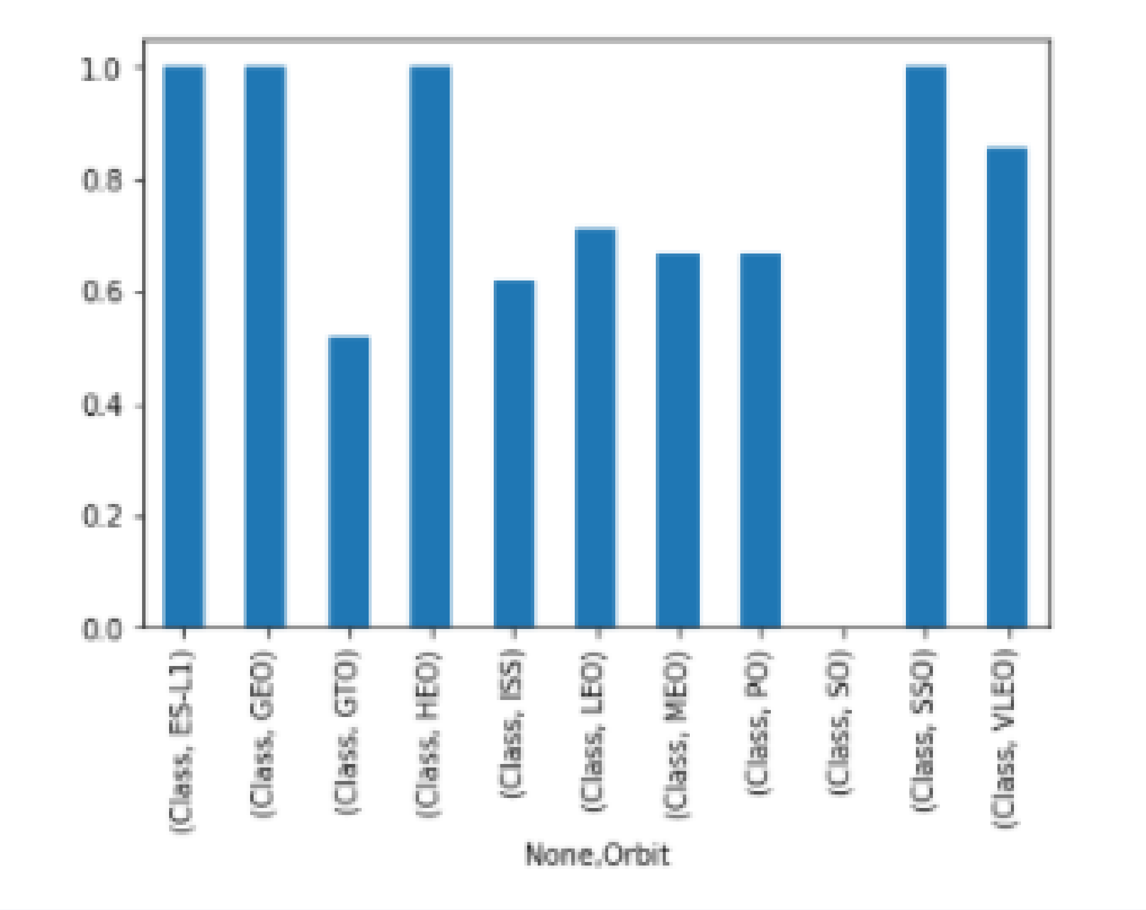
As number of flight increases, also increases success rate → Experience and know-how gained

Payload vs. Launch Site



- As payload is between 3000 and 5000Kg, results are better in CCAFS SLC 40 but it is not that clear in the other
- No posible to take a decission with this visualization.

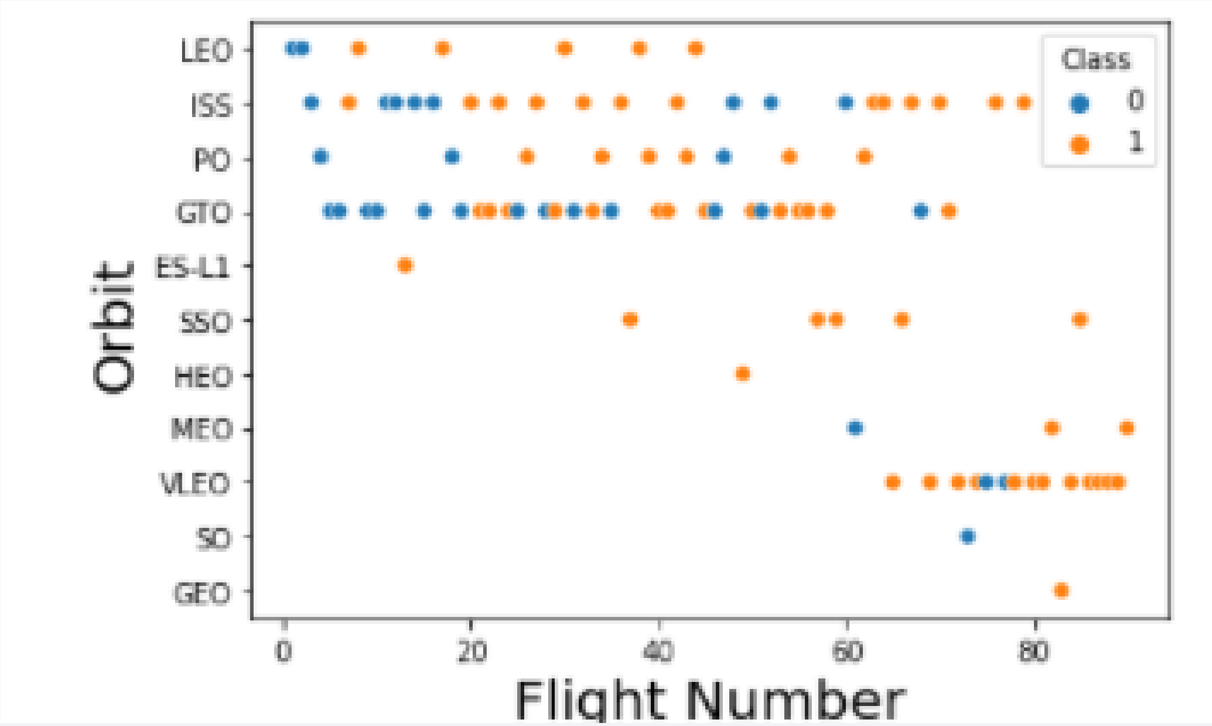
Success Rate vs. Orbit Type



- Best results for:
  - ES-L1
  - GEO
  - HEO
  - SSO

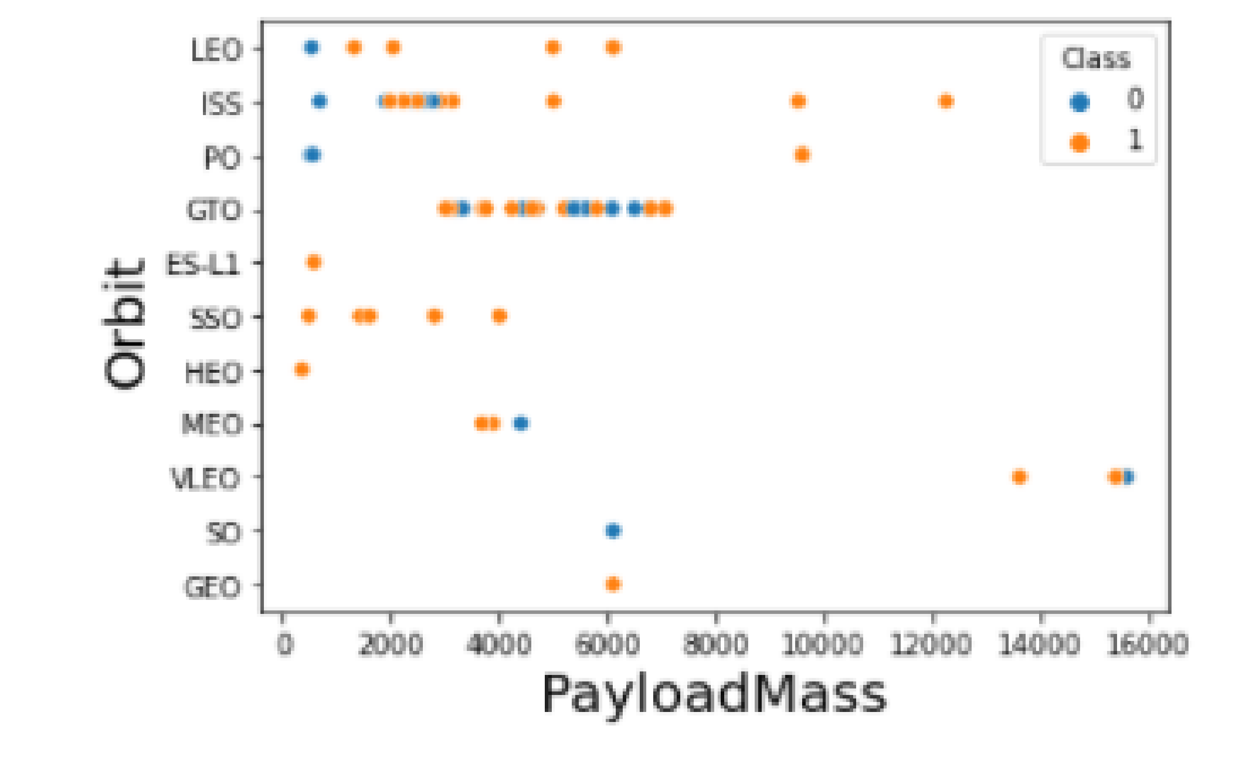
Flight Number vs. Orbit Type

- There is a trend as results are better as number of flights increases (know-how/experience)
- It is happening in all Orbits but GTO

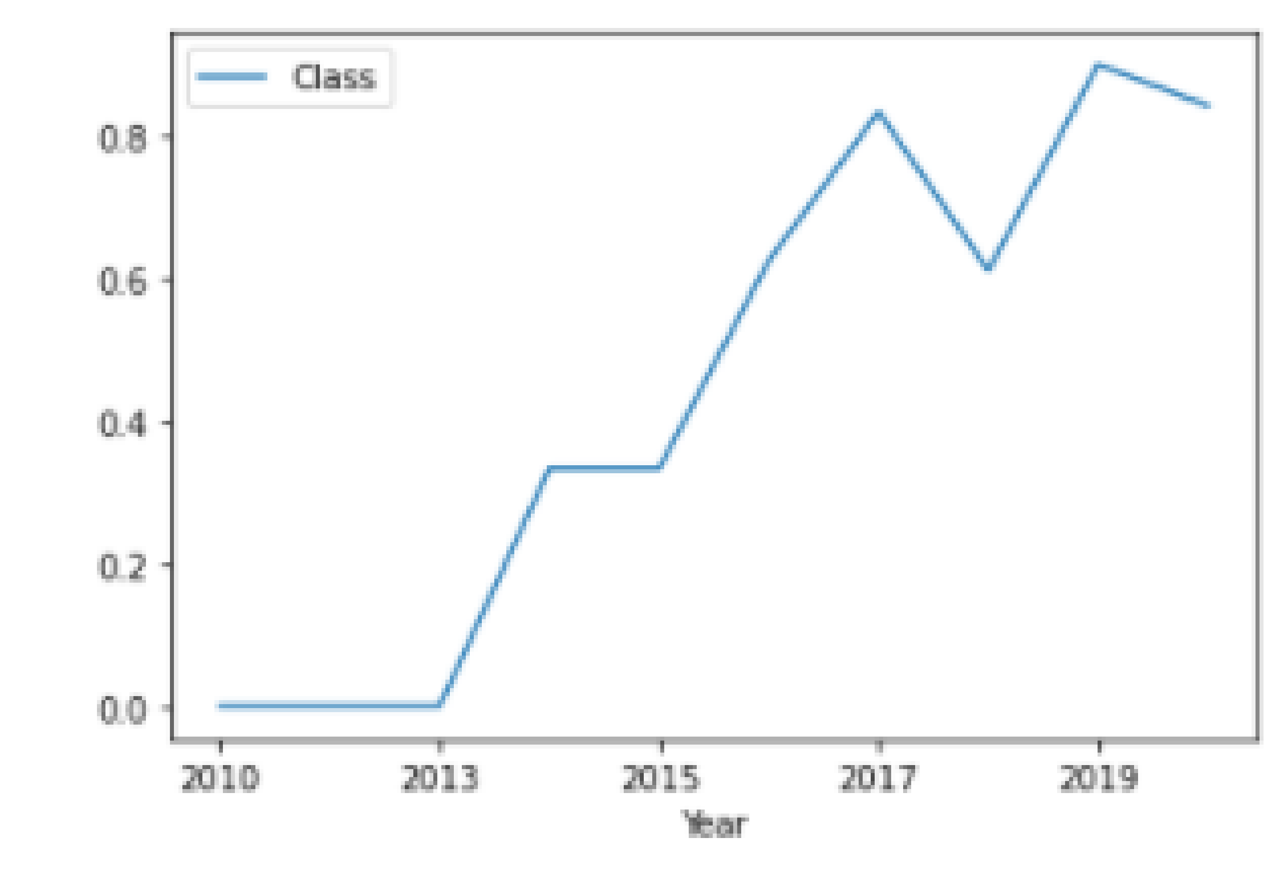


Payload vs. Orbit Type

- For the majority of the orbits but GTO, success rate increases if Payload Mass is between 3000 and 5000Kg for the majority of the samples
- GTO orbit is the most frequent
- There are not many samples of high payload to evaluate.



Launch Success Yearly Trend



- There is a clear trend of increasing success as year is after 2013.
- Team is increasing experience an know-how

All Launch Site Names

Task 1

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

In [13]:
```
%sql select distinct launch_site from SPACEXTBL
```

```
 * ibm_db_sa://njk72964:***@764264db-9824-4b7c-82df-40d1b13897c2.bs2io90l08kqb1od8lcg.databases.appdomain.cloud:32536/bludb
Done.
```

Out[13]:

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

In SQL query it has been added “distinct” to get unique results from column *launch_site*

Launch Site Names Begin with 'CCA'

Task 2

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

```
In [20]: %sql select * from SPACEXTBL where launch_site like 'CCA%' limit 5
```

```
 * ibm_db_sa://njk72964:***@764264db-9824-4b7c-82df-40d1b13897c2.bs2io90l08kqb1od8lcg.databases.appdomain.cloud:32536/bludb
Done.
```

Out[20]:

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

- It has been added “like ‘CCA%’” for the query to fetch only records that match the pattern that *launch_site* begings with CCA
- It has also been added “limit 5” to show only five records

Total Payload Mass

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

```
In [16]: %sql select sum(PAYLOAD_MASS__KG_) as sum_payload from SPACEXTBL where customer='NASA (CRS)'

 * ibm_db_sa://njk72964:***@764264db-9824-4b7c-82df-40d1b13897c2.bs2io90l08kqb1od8lcg.databases.appdomain.cloud:32536/bludb
Done.

Out[16]:
```

sum_payload
45596

- In the query, it has been added the built-in function “sum” and then rename the result with the operator “as”
- Condition is met for those records where *customer* equials “NASA (CRS)”

Average Payload Mass by F9 v1.1

Task 4

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

In [18]:
```
%sql select avg(PAYLOAD_MASS__KG_) as avg_payload from SPACEXTBL where booster_version='F9 v1.1'
```

```
 * ibm_db_sa://njk72964:***@764264db-9824-4b7c-82df-40d1b13897c2.bs2io90l08kqb1od8lcg.databases.appdomain.cloud:32536/bludb
Done.
```

Out[18]:

avg_payload
2928

- Similar to the previous, but this time it has been added the built-in operator “avg” to calculate mean value of the columna. As before, the result has been renamed to “avg_payload”

First Successful Ground Landing Date

List the date when the first successful landing outcome in ground pad was acheived.

Hint:Use min function

In [21]:
```
%sql select min(date) as min_date from SPACEXTBL where Landing__Outcome='Success (ground pad)'
```

```
 * ibm_db_sa://njk72964:***@764264db-9824-4b7c-82df-40d1b13897c2.bs2io90l08kqb1od8lcg.databases.appdomain.cloud:32536/bludb
Done.
```

Out[21]:

min_date
2015-12-22

- It has been added the built-in function “min” to get the first date for all records that successfully landed on the ground

Successful Drone Ship Landing with Payload between 4000 and 6000

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

```
In [22]: %sql select distinct booster_version from SPACEXTBL where landing__outcome='Success (drone ship)' and payload_mass__kg_ >=4000 and payload_mass__kg_<=6000
```

```
 * ibm_db_sa://njk72964:***@764264db-9824-4b7c-82df-40d1b13897c2.bs2io90l08kqb1od8lcg.databases.appdomain.cloud:32536/bludb
Done.
```

Out[22]:

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

- It has been added “distinct” to show only unique values
- In the condition it has been specified to fetch records that successfully landed on a drone and the payload weight were between 4000 and 6000Kg

Total Number of Successful and Failure Mission Outcomes

List the total number of successful and failure mission outcomes

In [25]:
```
%sql select mission_outcome, count(*) as Number from SPACEXTBL group by mission_outcome
```

```
 * ibm_db_sa://njk72964:***@764264db-9824-4b7c-82df-40d1b13897c2.bs2io90l08kqb1od8lcg.databases.appdomain.cloud:32536/bludb
Done.
```

Out[25]:

mission_outcome	number
Failure (in flight)	1
Success	99
Success (payload status unclear)	1

- In this query, it has been added the operator “count(*)” to count similar records that met the condition
- For count to work, it is needed to group records with “group_by” and then detail the name of the columns to perform the grouping

Boosters Carried Maximum Payload

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

In [27]:
```
%sql select distinct booster_version from spacextbl where payload_mass__kg_=(select max(payload_mass__kg_) from spacextbl)
```

```
 * ibm_db_sa://njk72964:***@764264db-9824-4b7c-82df-40d1b13897c2.bs2io90l08kqb1od8lcg.databases.appdomain.cloud:32536/bludb
Done.
```

Out[27]:

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

- In this query it has been added the operator “distinct” to fetch unique results
- To calculate in advance the maximum payload, it is used a subquery to be evaluated in the condition.

2015 Launch Records

List the failed landing_outcomes in drone ship, their booster versions, and launch site names for in year 2015

```
In [22]: %sql select Landing__Outcome, Booster_Version, Launch_Site from SPACEXTBL where landing__outcome='Failure (drone ship)' and year(date)=2015
```

```
 * ibm_db_sa://njk72964:***@764264db-9824-4b7c-82df-40d1b13897c2.bs2io90l08kqb1od8lcg.databases.appdomain.cloud:32536/bludb
Done.
```

Out[22]:

landing__outcome	booster_version	launch_site
Failure (drone ship)	F9 v1.1 B1012	CCAFS LC-40
Failure (drone ship)	F9 v1.1 B1015	CCAFS LC-40

- For this query, several columns has been fetched
- For the condition, it is needed to match that the landind over a drone ship has failed but only for year 2015
- To get year from date, it has been used built-in function “year”. To achieve this result, it is needed that *date* were on date format (i.e. not string)

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

Rank the count of landing outcomes (such as Failure (drone ship) or Success (ground pad)) between the date 2010-06-04 and 2017-03-20, in descending order

```
In [31]: %sql select Landing__Outcome, count(*) as Count from SPACEXTBL where date between'2010-06-04' and '2017-03-20' group by Landing__Outcome order by count(*) desc

 * ibm_db_sa://njk72964:***@764264db-9824-4b7c-82df-40d1b13897c2.bs2io90l08kqb1od8lcg.databases.appdomain.cloud:32536/bludb
Done.
```

Out[31]:

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

- In this query, it is needed to perform “count(*)” operator and also “group by”
- In the condition, operator “where” has been added to limit the results to the detailed dates
- As it is needed to show results in descending order, it has been added an “order by” instruction with the name of the column to sort and then “desc” so as results were produced in descending order

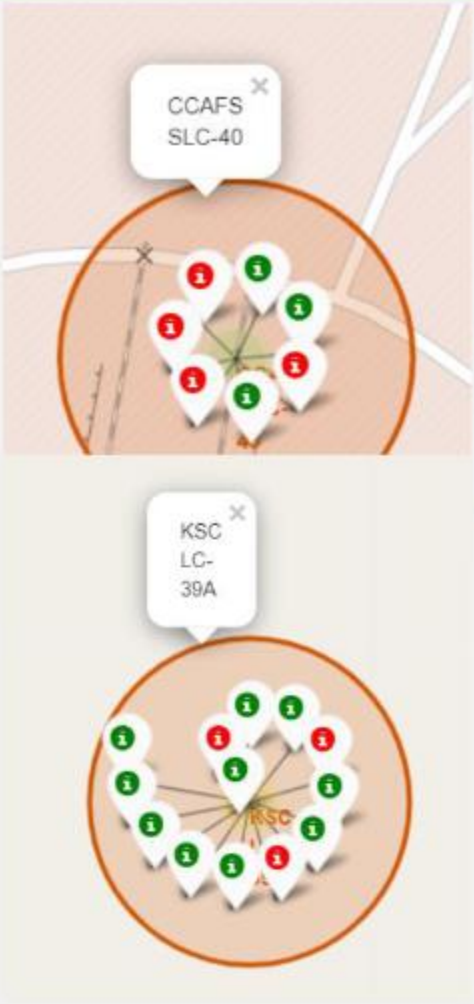
Section 4

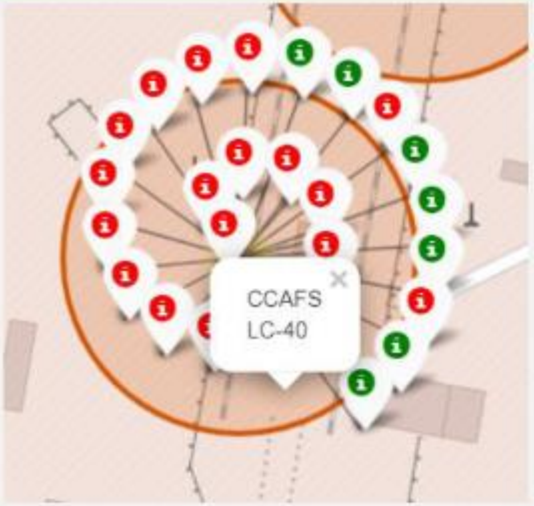
Launch Sites Proximities Analysis

Launch site locations

As seen, all launches has been performed in Florida and California

Coloured Markers





Florida Launch Sites

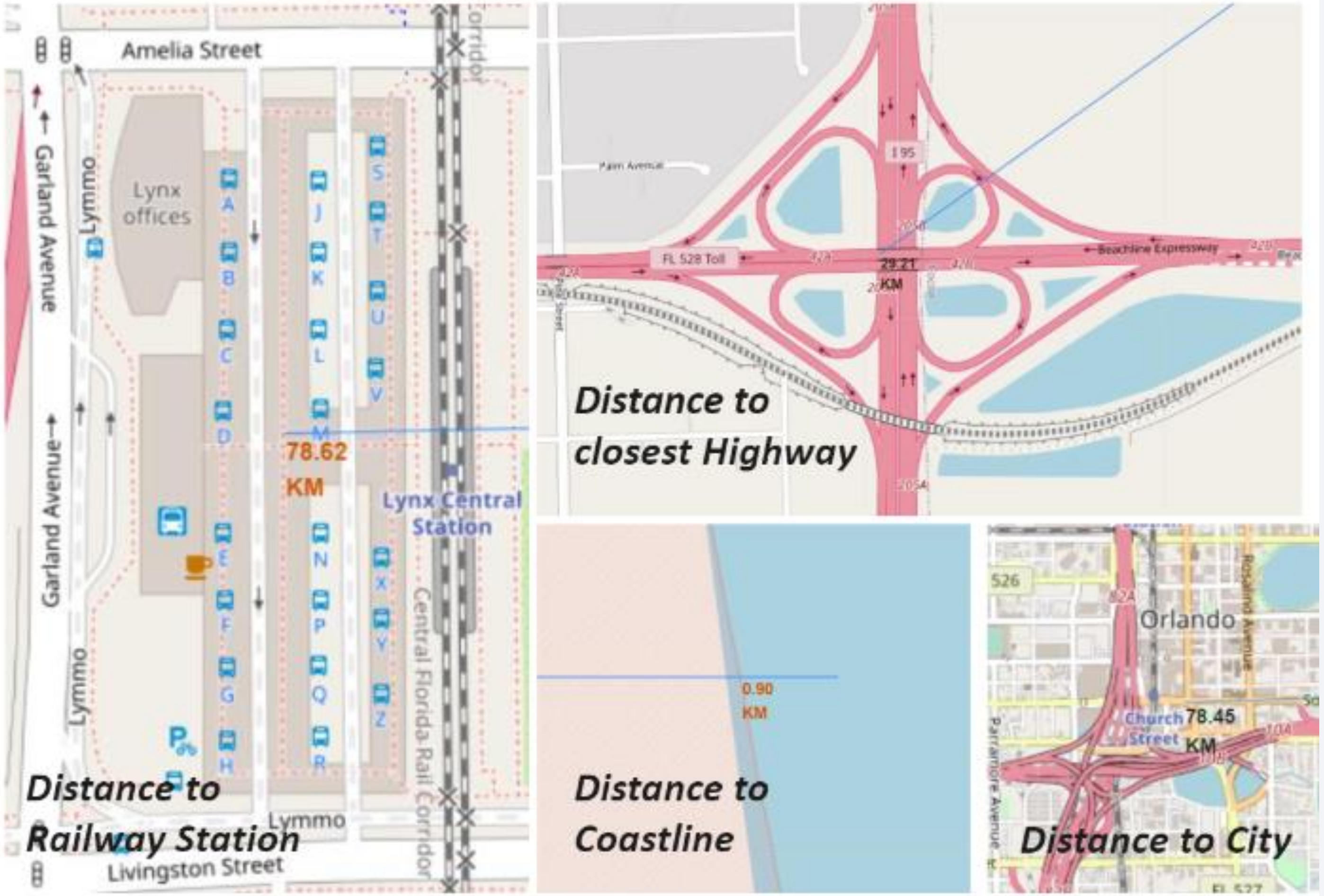
Green Marker shows successful Launches and Red Marker shows Failures

California Launch Site

37

Working with distances

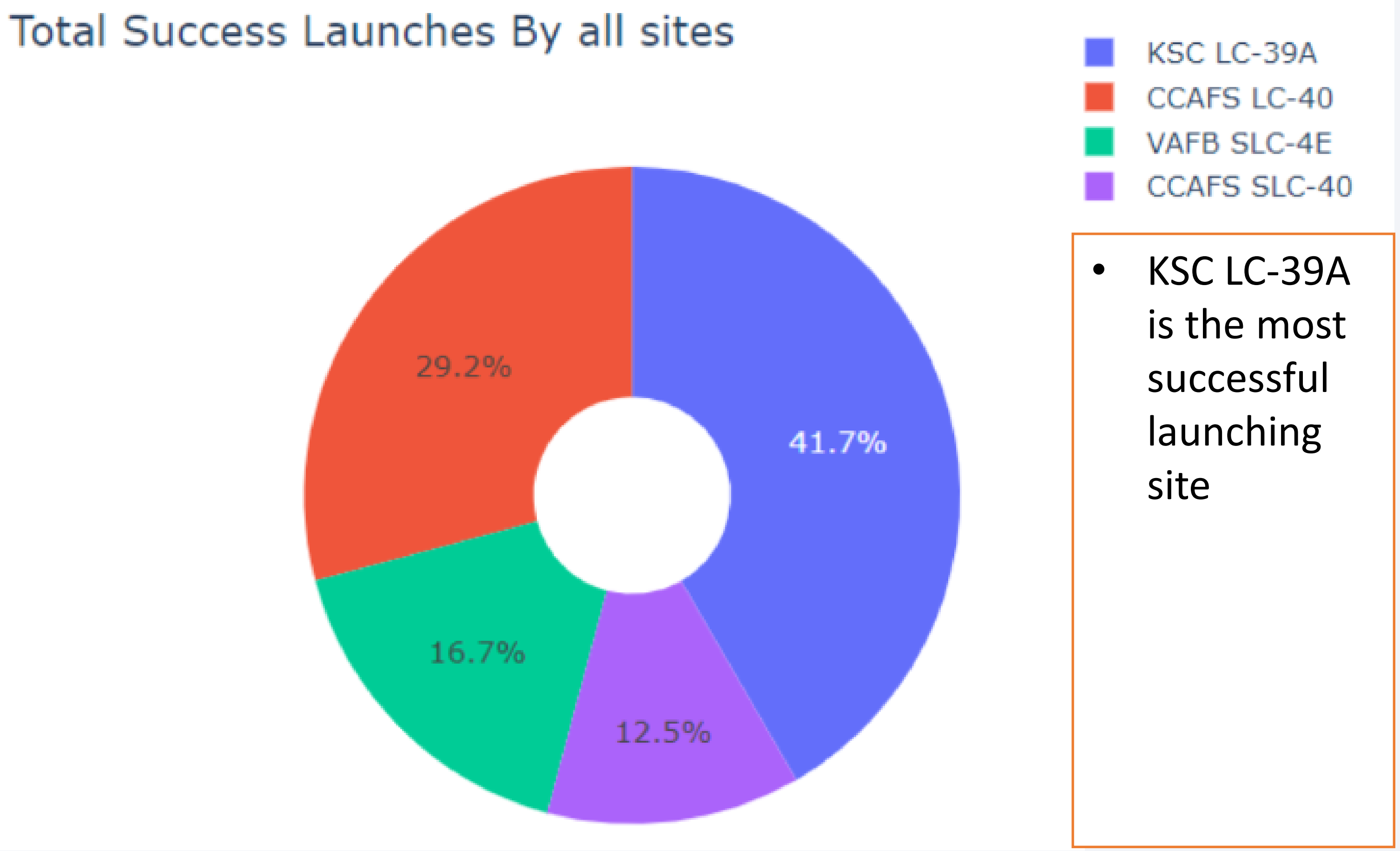
- Are launch sites in close proximity to railways? No
- Are launch sites in close proximity to highways? No
- Are launch sites in close proximity to coastline? Yes
- Do launch sites keep certain distance away from cities? Yes



Section 5

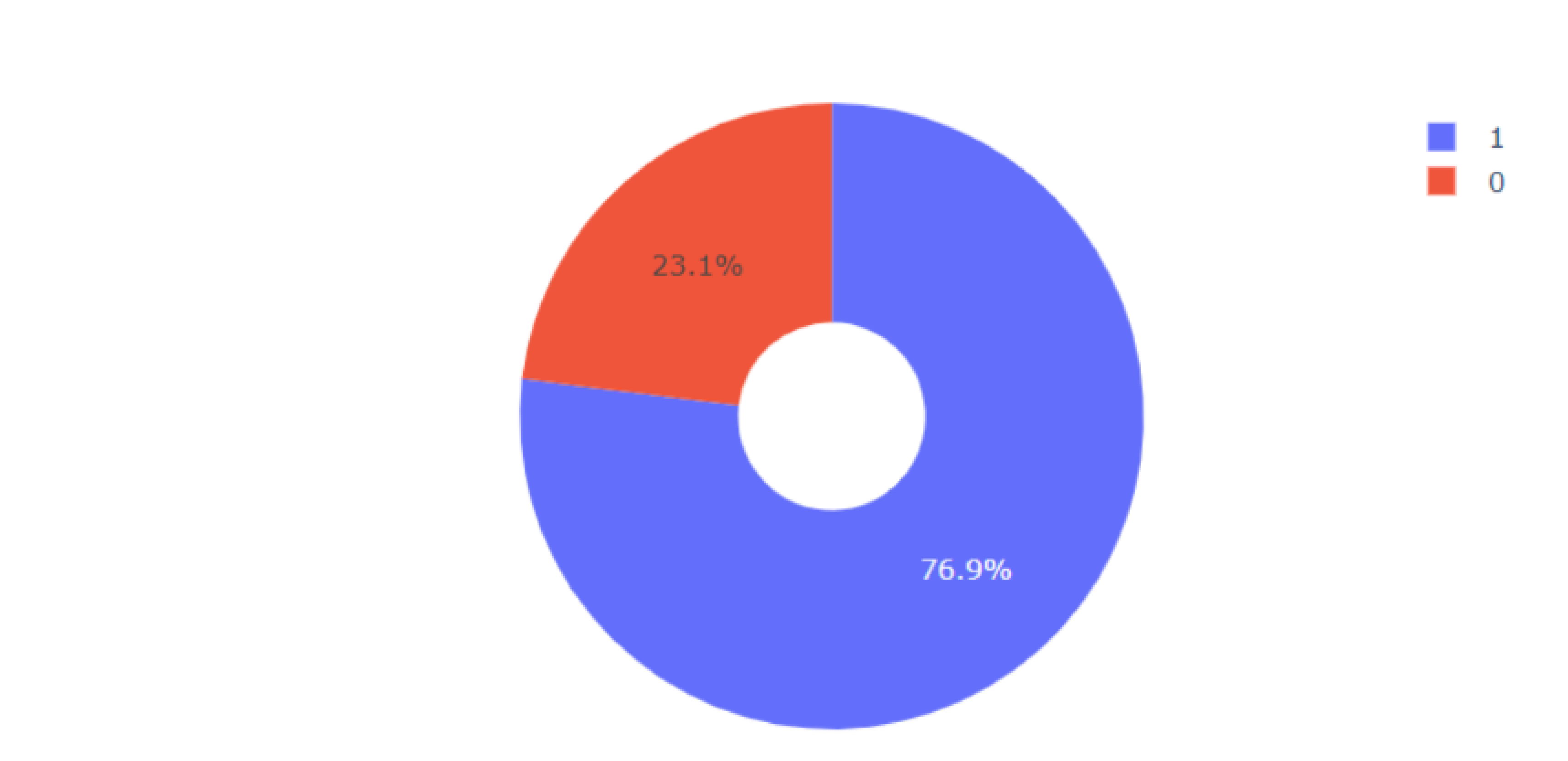
Build a Dashboard with Plotly Dash

Success rate per launching site



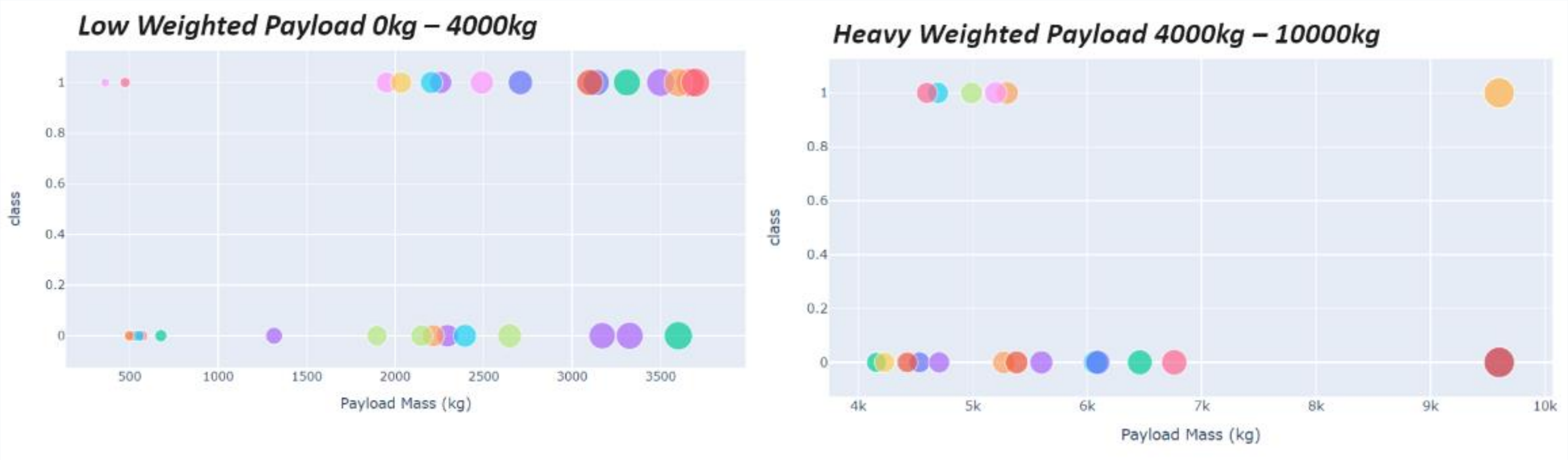
- KSC LC-39A is the most successful launching site

Detailed results for KSC LC-39A



Almost 3 out of 4 launches from KSC LC-39A is successfully

Success rate vs payload weight

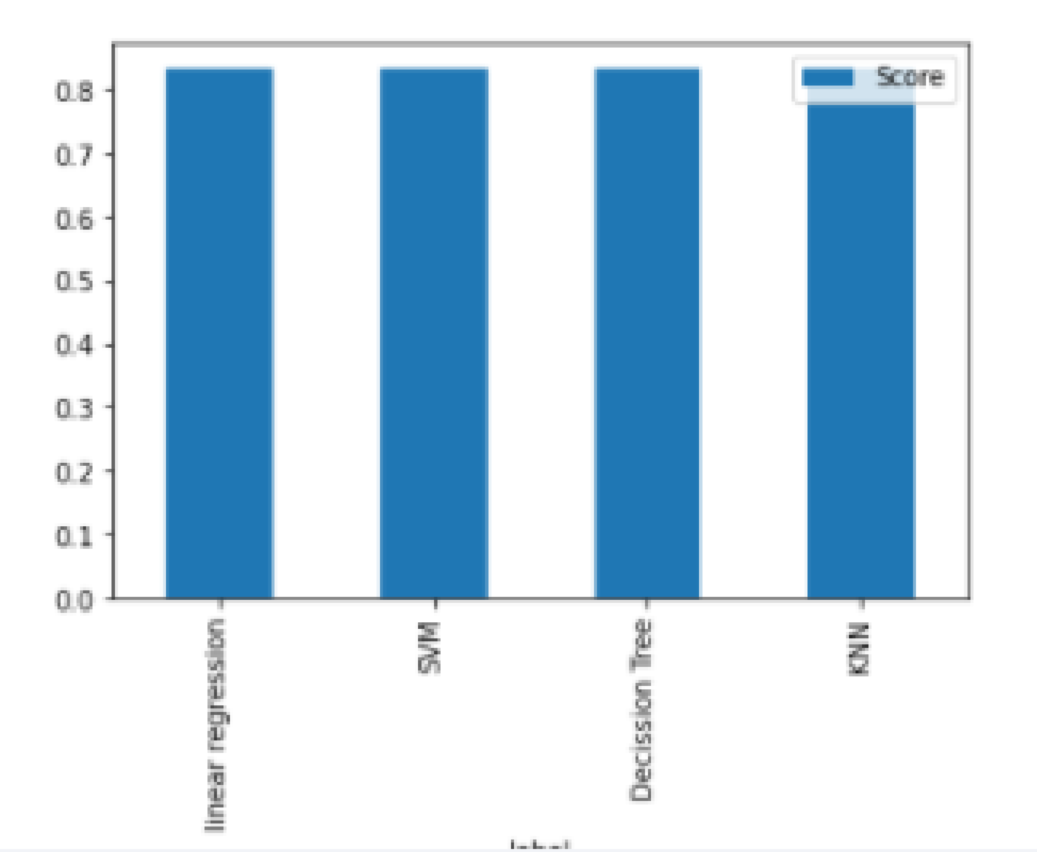


As seen payloads from 2000 to 5000Kg has better results than other

Section 6

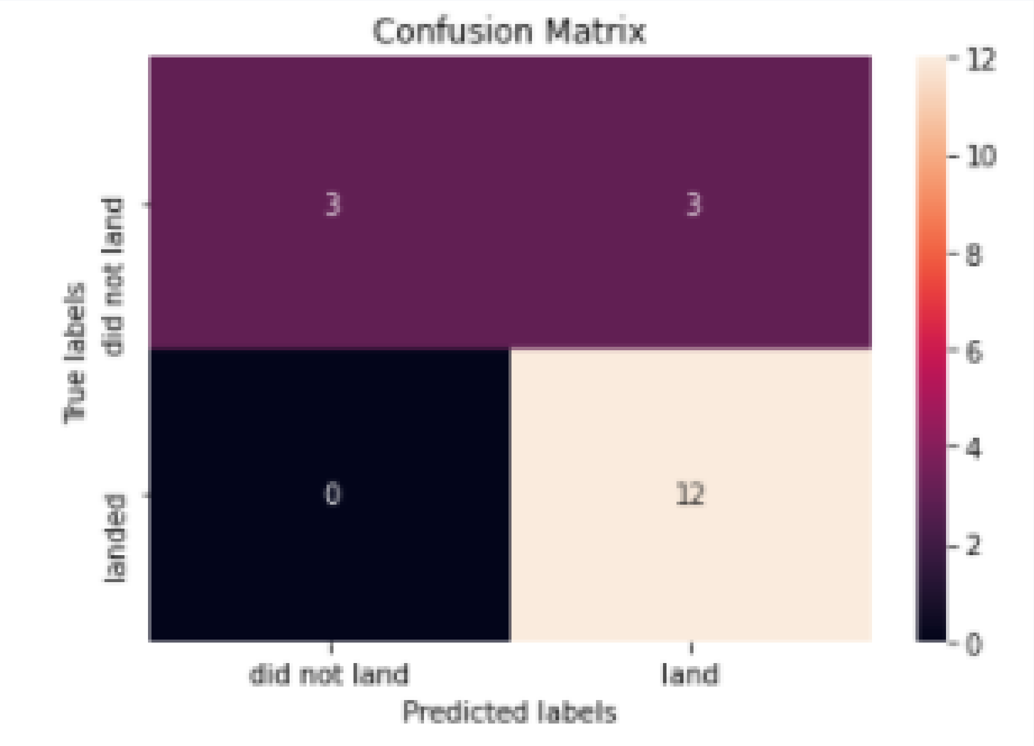
Predictive Analysis (Classification)

Classification Accuracy



- All models have the same Score and confusión Matrix.

Confusion Matrix



- Confusion matrix is the same for all models.
- Main problem are False Positives (launches that did not land but the models predict they land)

Conclusions

- Success landing rate is increasing with time and with number of launches. This means that the team is increasing experience and know-how
- Success landing is associated to payload weigh between 3000 and 5000Kg
- KSC L39A (Florida) is the most successful launching site
- GTO orbit is the most frequent but the one with the major failure rate. For the other, there are not enough samples to get a conclusion

Appendix

- Haversine Formula
- SpaceX
- Folium
- Plotly
- To check notebooks and results, please visit my Github My Github

Thank you!