IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

BRUNO ABDALLA DE SOUZA
23/03/2022

Outline

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

Executive Summary

- Summary of methodologies
  - ✓ Data collection using SpaceX and web scraping
  - ✓ Data cleaning
  - ✓ Data wrangling
  - ✓ Exploratory data analysis with SQL
  - ✓ Exploratory data analysis with visualization
  - ✓ Explore data through interactive maps and dashboard
  - ✓ Predict success/failure using machine learning
- Summary of all results
  - ✓ Exploratory data analysis using SQL, Pandas and visualization
  - ✓ Interactive maps with Folium
  - ✓ Interactive Dashboard
  - ✓ Predictive analysis

Introduction

- Project background and context

  Space X advertises Falcon 9 rocket launches on its website with a cost of 62 million dollars. Other providers cost upward of 165 million dollars each. Much of the savings is because Space X can reuse the first stage. So, if we can determine if the first stage will land, the cost of a launch can be estimated. It can be used if an alternate company wants to bid against space X for a rocket launch.

  This project presents a machine learning pipeline to predict if the first stage will be successfully landed or not.

- Problems you want to find answers

  - Which factors determine if the rocket will land successfully?

  - The interaction between features that determine the success rate of a successful landing.

  - Which operating conditions are likely to be in place to ensure a successful landing program

Section 1

Methodology

Methodology

Executive Summary

- Data collection methodology:
  - ✓ Collected using SpaceX API through http requests and web scraping from Wikipedia
- Perform data wrangling
  - ✓ class field (1 to success launch and 0 for failure). One-hot encoding to categorical features (Orbit, Launch Site, Landing Pad and Serial)
- Perform exploratory data analysis (EDA) using visualization and SQL
- Perform interactive visual analytics using Folium and Plotly Dash
- Perform predictive analysis using classification models
  - ✓ Data normalized and splited into training and test data. Models created to identify the key factors for successful rocket landing

Data Collection

- Data collected through API request and web scraping, as follows:

  ✓Space X API

  - Data collected using request (get function) using requests library in Python.
  - Data API stored in json files. It was necessary to use json_normalize (from Pandas library) and json functions (from resquests library) to transform data into pandas dataframe.
  - Data filtered just for Falcon 9 launches.
  - Data checked for missing values. Payload mass missing values was replaced by mean for payload mass.

  ✓Wikipedia page about Space X

  - Data collected using http requests to obtain the Falcon 9 Launch HTML page.
  - BeautifulSoup library in Python to collect HTML launch table and stored in a dictionary.
  - Dictionary data converted into a dataframe.

Data Collection – SpaceX API

- Data collected from get request to the SpaceX API.

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

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

Check the content of the response

```
print(response.content)
```

```
b'[{"fairings":{"reused":false,"recovery_attempt":false,"re
```

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

- Data wrangling

```
In [41]: data_falcon9.isnull().sum()

Out[41]: FlightNumber        0
         Date                0
         BoosterVersion      0
         PayloadMass         5
         Orbit               0
         LaunchSite          0
         Outcome             0
         Flights             0
         GridFins            0
         Reused              0
         Legs                0
         LandingPad         26
         Block               0
         ReusedCount         0
         Serial              0
         Longitude           0
         Latitude            0
         dtype: int64
```

```
# Calculate the mean value of PayloadMass column
mean_PayLoasMass = data_falcon9['PayloadMass'].mean

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

- Data cleaning

```
# 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 that have multi
ple payloads in a single rocket.
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 feature.
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)]
```


https://github.com/bruno-abdalla-de-souza/SpaceX/blob/master/DataCollectionFromAPI.ipynb

Data Collection - Scraping

Using BeautifulSoup library in Python, Falcon 9 launches from Wikipedia was obtained through webscraping

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

```
# use requests.get() method with the provided static_url
# assign the response to a object
response_data = requests.get(static_url)
```

Create a BeautifulSoup object from the HTML response

```
# Use BeautifulSoup() to create a BeautifulSoup object from a response text content
soup = BeautifulSoup(response_data.content, 'html5lib')
```

Print the page title to verify if the BeautifulSoup object was created properly

```
# Use soup.title attribute
soup.title
```

```
<title>List of Falcon 9 and Falcon Heavy launches - Wikipedia</title>
```

```
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']=[]
```

A loop through the table rows was made to populate launch_dict (not presented here). After this, converted to a dataframe.

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

	Flight No.	Launch site	Payload	Payload mass	Orbit	Customer	Launch outcome	Version Booster	Booster landing	Date	Time
0	1	CCSFS	Dragon Spacecraft Qualification Unit	0	LEO	SpaceX	Success\n	F9 v1.0B0003.1	Failure	4 June 2010	18:45

https://github.com/bruno-abdalla-de-souza/SpaceX/blob/master/Data_Collection_With_Webscrapping.ipynb

Data Wrangling

- Data was analyzed and manipulated using Pandas library in Python.

Number of launches on each site

```
# A''pply value_counts() on column LaunchSite
df['LaunchSite'].value_counts()
```

```
CCAFS SLC 40    55
KSC LC 39A      22
VAFB SLC 4E     13
Name: LaunchSite, dtype: int64
```

Number of launches on each orbit

```
# Apply value_counts on Orbit
df['Orbit'].value_counts()
```

```
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
```

Number of launches on each outcome

```
# landing_outcomes = values on Outcome
landing_outcomes = df['Outcome'].value_
landing_outcomes
```

```
True ASDS      41
None None      19
True RTLS      14
False ASDS      6
True Ocean      5
False Ocean     2
None ASDS       2
False RTLS      1
Name: Outcome, dtype: int64
```

Creating landing outcome as 1 or 0 for success or failure

```
# landing_class = 0 if bad_outcome
# landing_class = 1 otherwise
landing_class = []
landing_class = ~df['Outcome'].isin(bad_outcomes).astype(int)+2
```

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

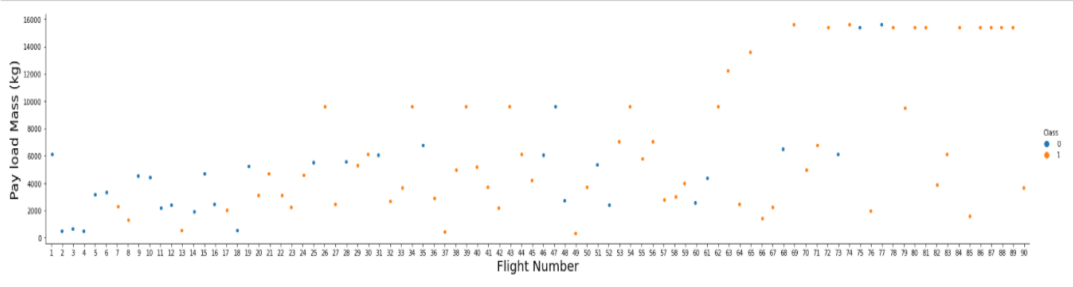
	Class
0	0
1	0
2	0
3	0
4	0
5	0
6	1
7	1

https://github.com/bruno-abdalla-de-souza/SpaceX/blob/master/DataWrangling.ipynb

EDA with Data Visualization

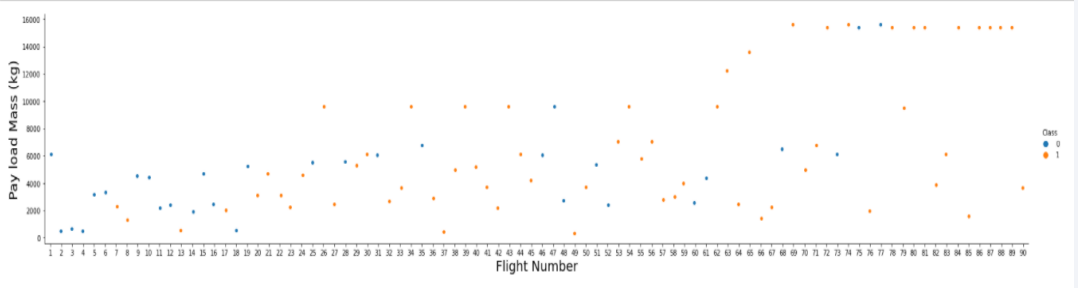
Variables were plotted trying to understand the influence of key variables, such as flight number, launch site, payload mass and orbit in the success rate of a launch

Payload mass x Flight number



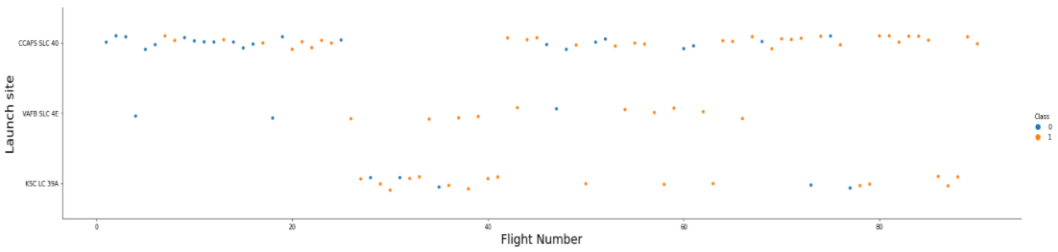
As flight number increases, payload mass increases and also the success rate

Launch site x Payload mass



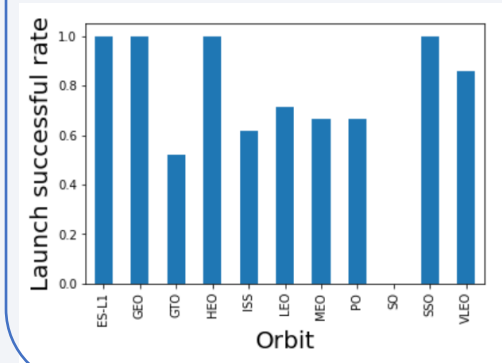
As payload mass increases, success rate increases. For VAFB-SLC launch site there are no rockets launched for heavy payload mass (>= 10000)

Launch site x Flight number



As flight number increases, the success rate for all launch sites has increased

Launch success rate x Orbit



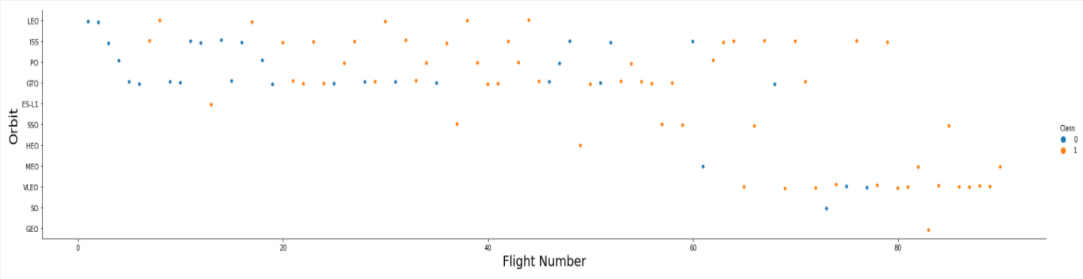
ES-L1, GEO, HEO and SSO have the highest success rate

https://github.com/bruno-abdalla-de-souza/SpaceX/blob/master/ExploratoryDataAnalysisWithVisualization.ipynb

EDA with Data Visualization

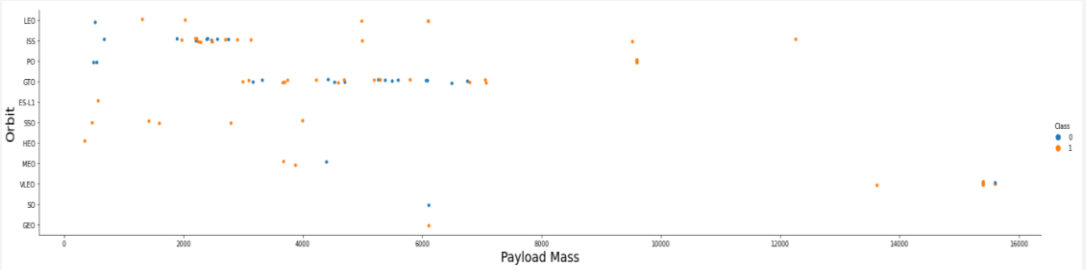
Variables were plotted trying to understand the influence of key variables, such as flight number, launch site, payload mass and orbit in the success rate of a launch

Orbit x Flight number



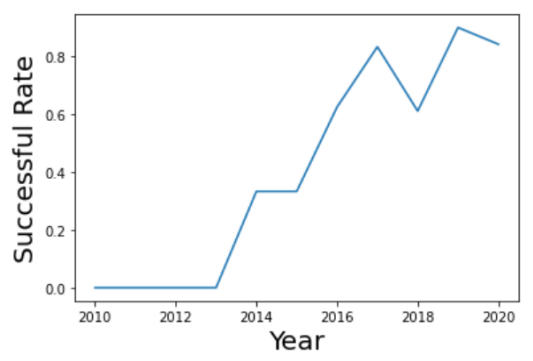
For LEO/ISS/VLEO, success rates seems to be related to the flight numbers

Orbit x Payload mass



With heavy payloads the successful landing or positive landing rate are more for Polar/LEO/ISS. For GTO is not possible to see a relation.

Launch success rate x Orbit



Since 2013, sucess rate kept increasing until 2020

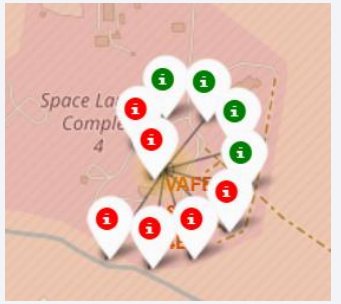
https://github.com/bruno-abdalla-de-souza/SpaceX/blob/master/ExploratoryDataAnalysisWithVisualization.ipynb

EDA with SQL

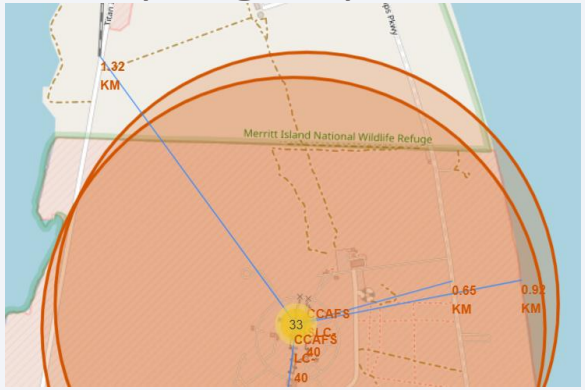
- SpaceX dataset (csv file) was loaded into a DB2 database and queried data using the jupyter notebook.
- The queries provided some insights from the data, such as:
  - Unique launch sites in the space mission.
  - Total payload mass launched by NASA (CRS)
  - Average payload mass for booster version F9 v1.1
  - List of dates with sucessful landing
  - Number of successful and failure mission outcomes
  - Number of failed landing outcomes in drone ship, their booster version and launch site names

https://github.com/bruno-abdalla-de-souza/SpaceX/blob/master/ExploratoryDataAnalysisWithSQL.ipynb

Build an Interactive Map with Folium



- Launch sites marked with built objects, such as markers, circles and lines.
  - Success or failure launches were indicated for each site using folium map.
- Launch outcomes, success or failure, were sorted to 1 and 0, respectively.
  - Marker clusters were included at launch sites.
- Calculated the distances between a launch site and its proximities.
  - Nearest coastlines, railways, highways and cities.



https://github.com/bruno-abdalla-de-souza/SpaceX/blob/master/DataVisualizationWithFolium.ipynb

Build a Dashboard with Plotly Dash

- An interactive dashboard with Plotly Dash was built using a dedicated Cloud IDE.
- Data spacex_launch_dash.csv was loaded into a Pandas dataframe in Python.
- A dropdown menu was created to interact with a pie chart showing successful rate for each launch site.
- Through a payload slide bar to select a payload mass range was used to select a payload range, a scatterplot shows the mission outcome for each booster version.

https://github.com/bruno-abdalla-de-souza/SpaceX/blob/master/PlotlyDashboard

Predictive Analysis (Classification)

- Data loaded through Pandas dataframe read_csv function.
- Data transformed to array using Numpy library.
- Data was normalized using preprocessing library with StandardScaler function.
- Data splited into training and testing set using train_test_split library (80% for training and 20% for testing_)
- 4 classification algorithms were fitted (Logistic Regression, Support Vector Machine, k-Nearest Neighbours and Decision Tree model). Their parameters were tuned using GridSearchCV.
- The confusion matrix was plotted for all models, presenting the same results for all of them.
- Best model was chosen based on the highest accuracy: Decision Tree Model.

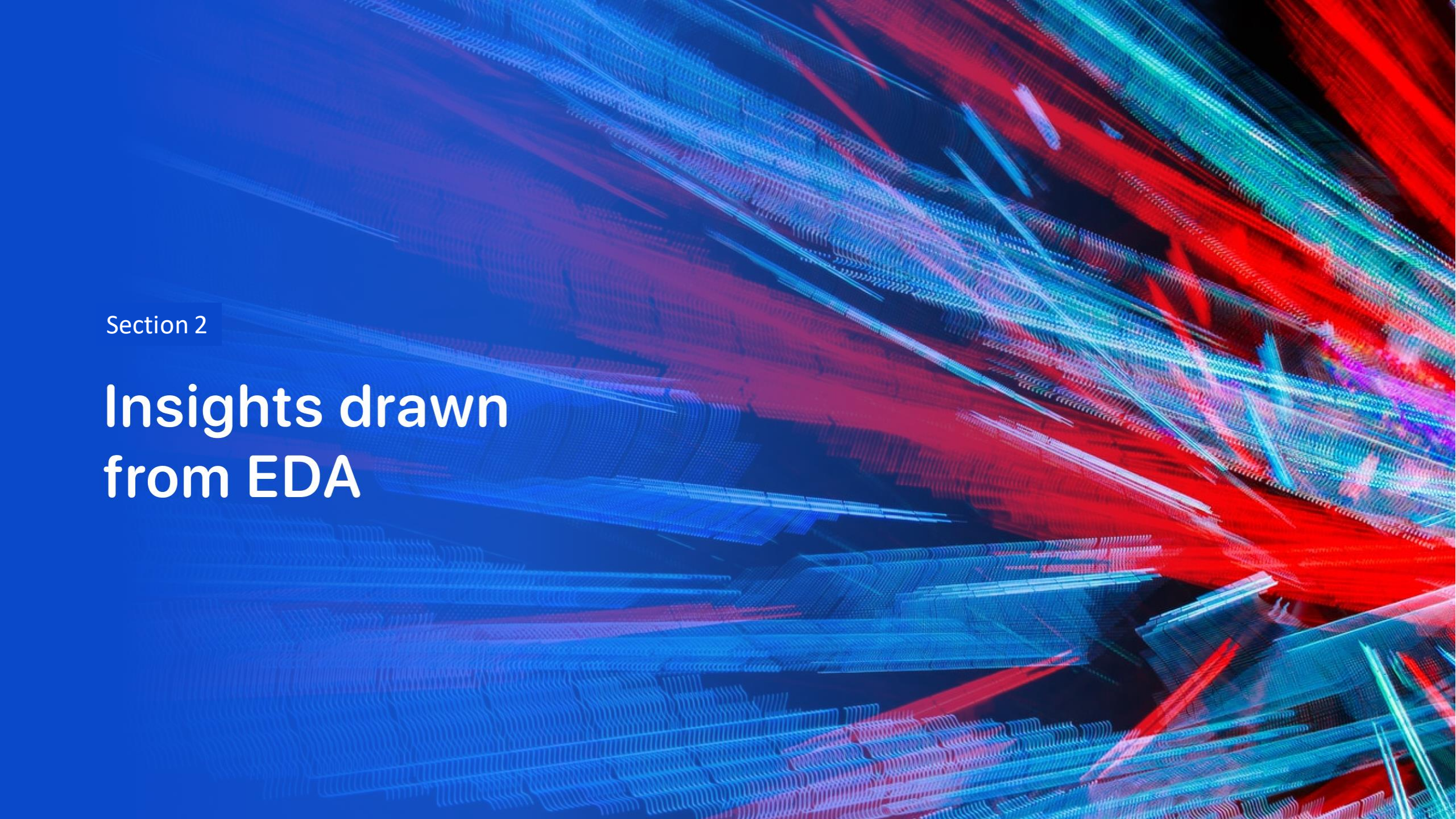
https://github.com/bruno-abdalla-de-souza/SpaceX/blob/master/MachineLearningPrediction.ipynb

Results

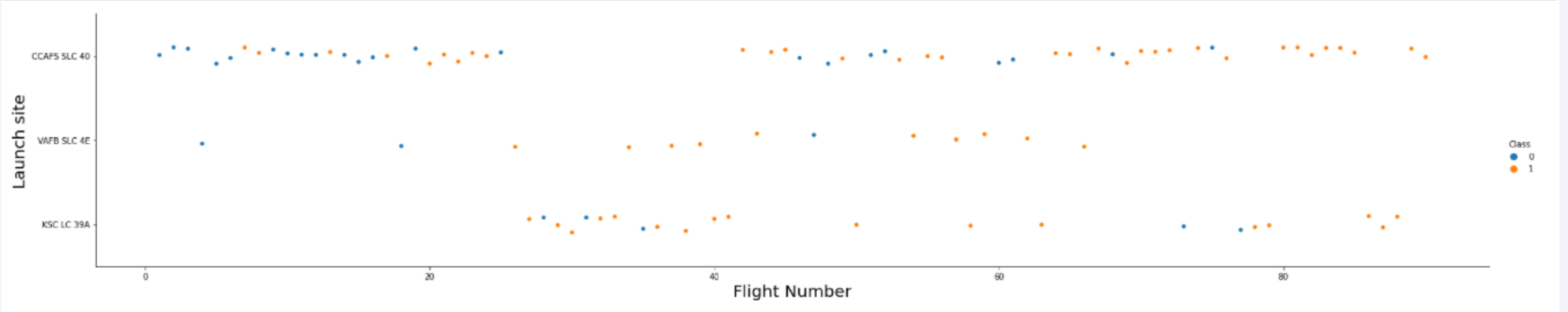
- Exploratory data analysis results
- Interactive analytics demo in screenshots
- Predictive analysis results

Section 2

Insights drawn from EDA

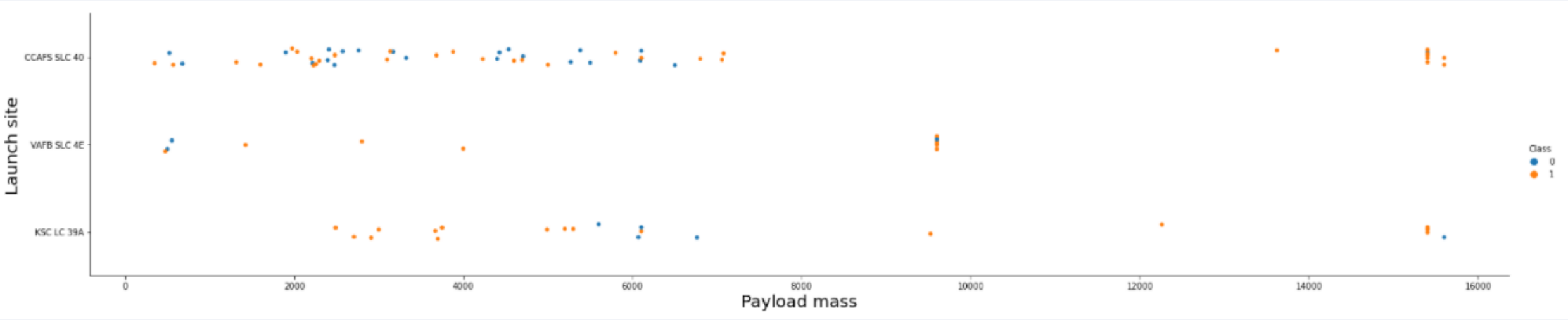


Flight Number vs. Launch Site



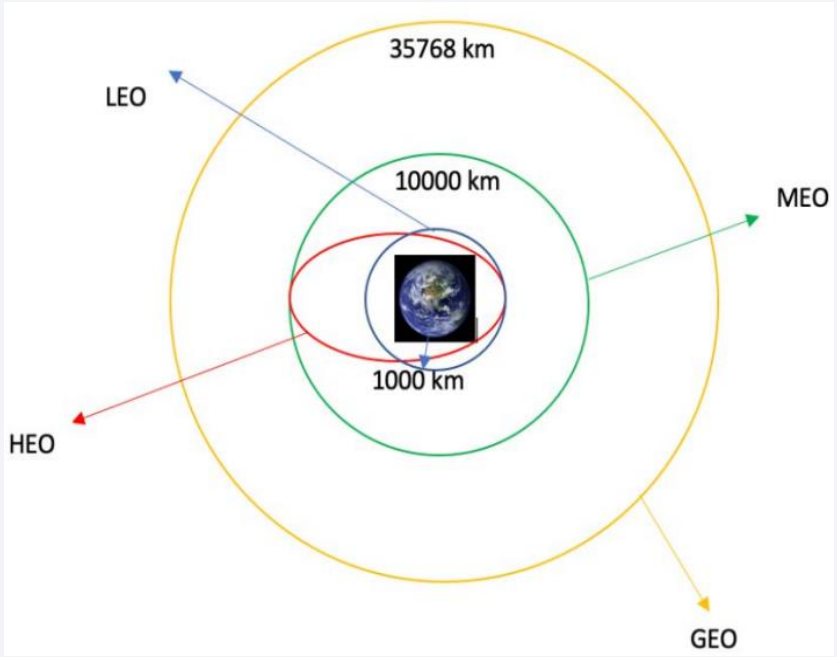
- As flight number increases, the success rate seems to increase.
- CCAFS SLC 40 is the most used site. It was the first launch and the first to achieve a successful launch.
- VAFB SLC 4E is the least used for launching, but it has a high success launch rate.

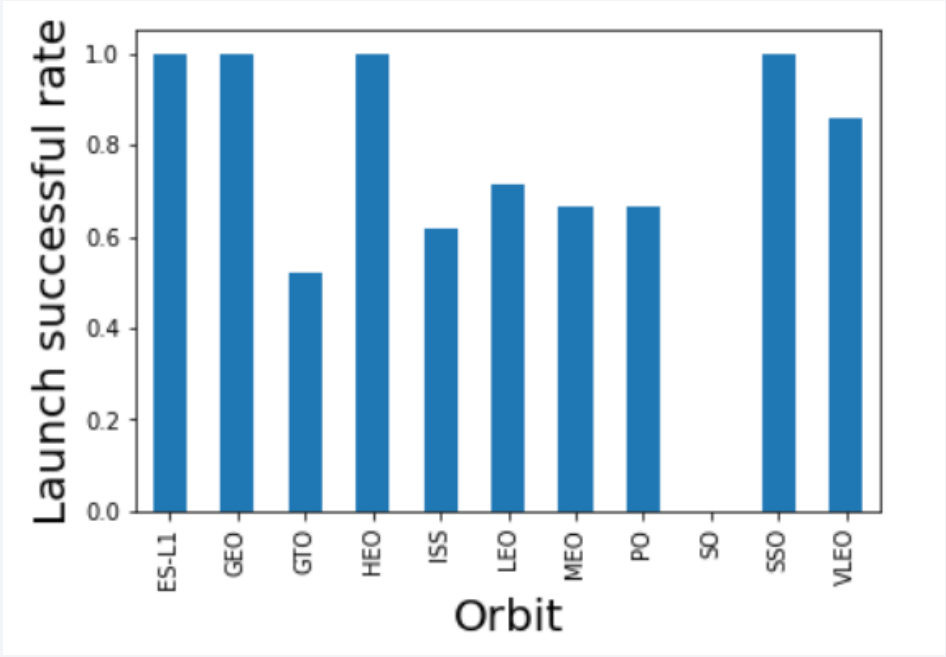
Payload vs. Launch Site



- CCAFS SLC 40 doesn't has a good success rate for payload less than to 8000 kg, but it has an excellent success rate for major payload (greater than 10000 kg). It is also the most used site for heavy loads.
- VAFB SLC 4E is the least used for launching, but it has a high success launch rate.
- KSC LC 39A is used for light, medium and heavy payloads.

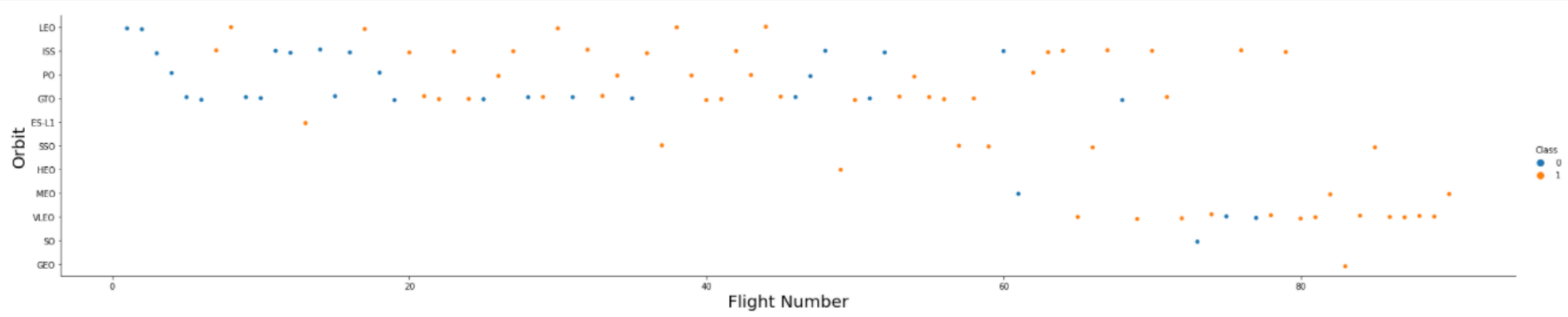
Success Rate vs. Orbit Type





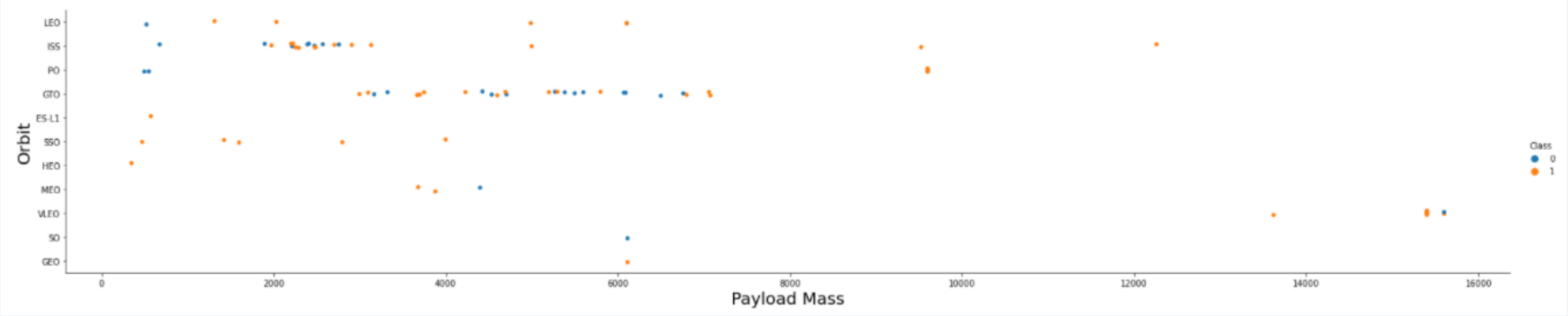
- ES-L1, GEO, HEO and SSO have the major success rate for launching.
- GTO, ISS, LEO, MEO and PO has the worst performance considering success rate.
- SO orbit was not used so far.

Flight Number vs. Orbit Type



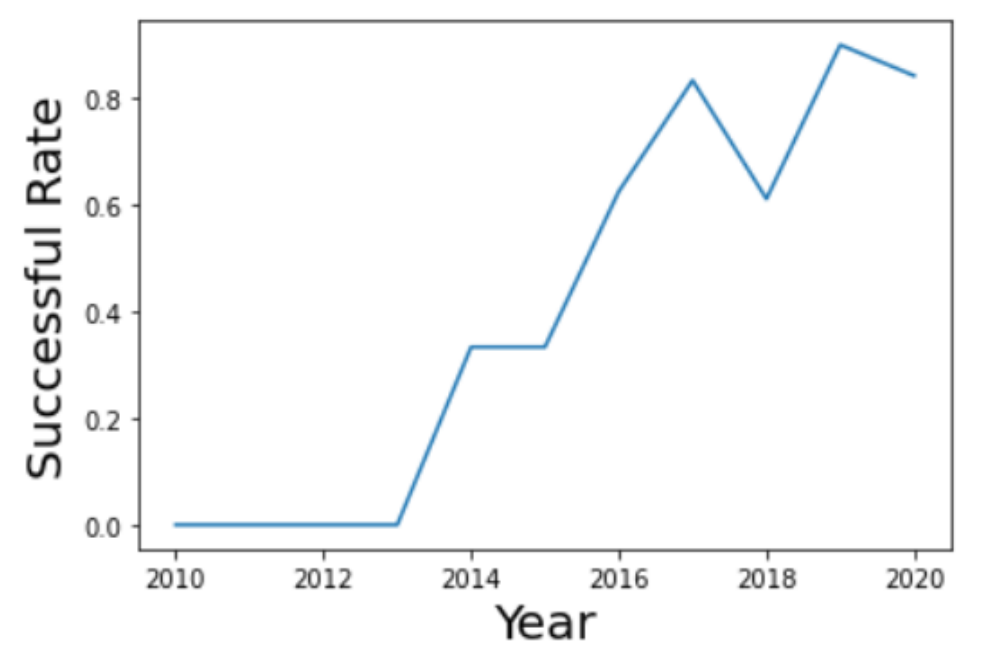
- Started with lower and transfer orbits (LEO, ISS, PO and GTO). After gaining knowledge with the previous failures, Space X targeted higher orbits with a good success rate.
- First successful mission was on a flight to ISS and the second one to LEO.
- For all orbits or at least most of them, as flight number increases, success rate also increases.

Payload vs. Orbit Type



- Most launches were with loads below 8000 kg.
- VLEO was used only for heavy payloads, greater than 13000 kg.
- LEO, ISS, SSO, HEO, MEO, SO and GEO was used with payloads less than around 6000 kg.
- GTO success rate not seems to be related to payload.

Launch Success Yearly Trend



- All flights until 2013 were unsuccessful.
- There was a continuous increase in successful landings from 2014 to 2017.
- 2018 showed a decrease in success rate, returning to increase in 2019.

All Launch Site Names

SQL QUERY

```
%%sql
    SELECT DISTINCT
        LAUNCH_SITE
    FROM
        SPACEXTBL
```

RESULT

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

- Query to select the distinct launch sites from the table SPACEXTBL.
- The keyword DISTINCT was used to obtain the distinct launch sites, without allow repetition.

Launch Site Names Begin with 'CCA'

SQL QUERY

```
%%sql
    SELECT
        *
    FROM
        SPACEXTBL
    WHERE
        LAUNCH_SITE LIKE 'CCA%'
    LIMIT 5
```

RESULT

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	07:44:00	F9 v1.0 B0005	CCAFS LC-40	Dragon demo flight C2	525	LEO (ISS)	NASA (COTS)	Success	No attempt
2012-10-08	00:35:00	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

- Query to select 5 records from table SPACEXTBL, where launch sites begin with string 'CCA'.
- The keyword LIKE with the wildcard was used to allow any combination after the CCA string.

Total Payload Mass

SQL QUERY

```
%%sql
    SELECT
        SUM(PAYLOAD_MASS__KG_)
    FROM
        SPACEXTBL
    WHERE
        CUSTOMER LIKE 'NASA%'
```

RESULT

1
99980

- Query to select the total payload mass sum from table SPACEXTBL for NASA.
- The total payload was obtained through the SUM of PAYLOAD_MASS__KG_ field.
- The keyword LIKE with the wildcard was used to allow any combination after the NASA string.

Average Payload Mass by F9 v1.1

SQL QUERY

```
%%sql
    SELECT
        AVG(PAYLOAD_MASS__KG_)
    FROM
        SPACEXTBL
    WHERE
        BOOSTER_VERSION LIKE 'F9 v1.1%'
```

RESULT

1
2534

- Query to select the payload average from table SPACEXTBL for F9 v1.1 booster version.
- The average payload was obtained through the AVG of PAYLOAD_MASS__KG_ field.
- The keyword LIKE with the wildcard was used to allow any combination after the 'F9 v1.1' string.

First Successful Ground Landing Date

SQL QUERY

```
%%sql
    SELECT
        MIN(DATE) AS FIRST_DATE
    FROM
        SPACEXTBL
    WHERE
        UPPER(LANDING__OUTCOME) LIKE '%SUCCESS (GROUND PAD)%'
```

RESULT

first_date
2015-12-22

- Query to select the first date with successful landing outcome in ground pad.
- The minimum date was obtained through the MIN of DATE field.
- The keyword LIKE with the wildcard was used to allow any combination after the 'SUCCESS (GROUND PAD)' UPPERCASE string.

Successful Drone Ship Landing with Payload between 4000 and 6000

SQL QUERY

```
%%sql
    SELECT DISTINCT
        BOOSTER_VERSION
    FROM
        SPACEXTBL
    WHERE
        UPPER(LANDING__OUTCOME) LIKE '%SUCCESS%DRONE%' AND
        PAYLOAD_MASS__KG_ BETWEEN 4001 AND 5999
```

RESULT

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

- Query to select a booster version list from table SPACEXTBL, where landing outcome was a successful drone ship and payload between 4000 and 6000 kg.
- The keyword LIKE with the wildcard was used to allow any combination that has SUCCESS and after has DRONE. It was used the UPPER function in order to have no concern about letter cases in thwe words Success and drone.

Total Number of Successful and Failure Mission Outcomes

SQL QUERY

```
%%sql
    SELECT
        MISSION_OUTCOME, COUNT(*) AS NUMBER
    FROM
        SPACEXTBL
    GROUP BY
        MISSION_OUTCOME
```

RESULT

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

- Query to select the mission outcome and the total number of mission outcomes.
- Used the function COUNT to obtain the number of mission outcomes.
- The keyword GROUP BY group the unique mission outcomes, allowing to calculate the number of each one.

Boosters Carried Maximum Payload

SQL QUERY

```
%%sql
    SELECT DISTINCT
        BOOSTER_VERSION, PAYLOAD_MASS__KG_
    FROM
        SPACEXTBL
    WHERE
        PAYLOAD_MASS__KG_ =
            (SELECT
                 MAX(PAYLOAD_MASS__KG_)
            FROM
                SPACEXTBL)
```

RESULT

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

- Query to select booster version with maximum payload. It was necessary to use a subquery to calculate the maximum payload, using it as a condition in WHERE clause.

2015 Launch Records

SQL QUERY

```
%%sql
    SELECT
        BOOSTER_VERSION, LAUNCH_SITE
    FROM
        SPACEXTBL
    WHERE
        UPPER(LANDING__OUTCOME) LIKE '%FAILURE%DRONE%' AND
        YEAR(DATE) = 2015
```

RESULT

booster_version	launch_site
F9 v1.1 B1012	CCAFS LC-40
F9 v1.1 B1015	CCAFS LC-40

- Query to select booster version and launch sites with drone failure landing outcome in 2015.
- The keyword LIKE with the wildcard was used to allow any combination that has FAILURE and after has DRONE. It was used the UPPER function in order to have no concern about letter cases in the words Failure and drone.
- It was used the function YEAR in DATE field to establish the 2015 year in WHERE clause.

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

SQL QUERY

```
%%sql
    SELECT
        LANDING__OUTCOME, COUNT(*) AS COUNT_LANDING_OUTCOME
    FROM
        SPACEXTBL
    WHERE
        DATE BETWEEN '2010-06-04' AND '2017-03-20'
    GROUP BY
        LANDING__OUTCOME
    ORDER BY
        COUNT_LANDING_OUTCOME DESC
```

RESULT

landing__outcome	count_landing_outcome
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

- Query to rank the count of landing outcomes between the date 2010-06-04 and 2017-03-20, in descending order.
- Keyword GROUP BY to group LANDING-OUTCOME field.
- WHERE clause to filter the date between 2014-06-04 and 2017-03-20.
- ORDER BY clause to sort COUNT_LANDING_OUTCOME in descending order.

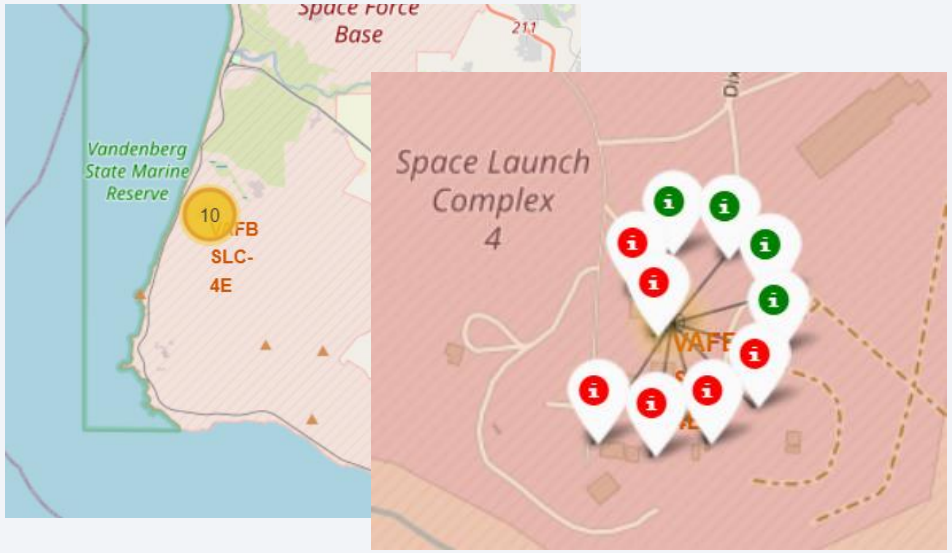

Section 3
Launch Sites
Proximities Analysis

SpaceX launch sites

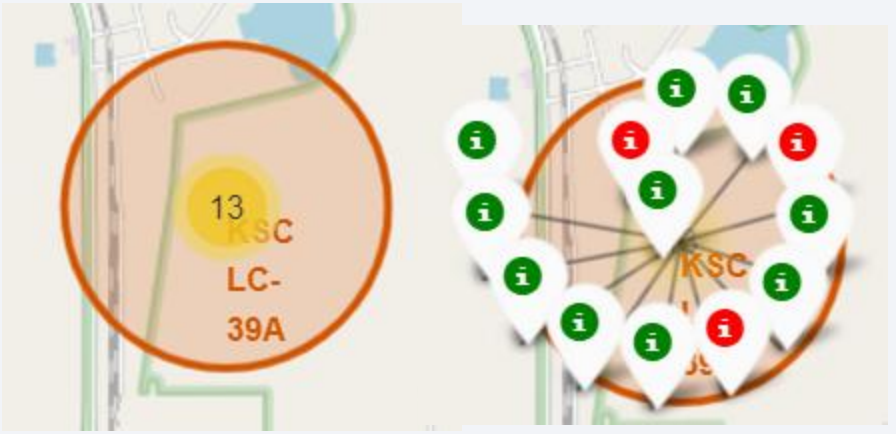
We can notice that Space X launch sites are near to the coast and Equator line. It's near from the Equator line to move faster than others points of the globe and it's near to the coast to avoid populated areas during launching.

Position of success and failure launches position on map

- VAFB SLC-4E

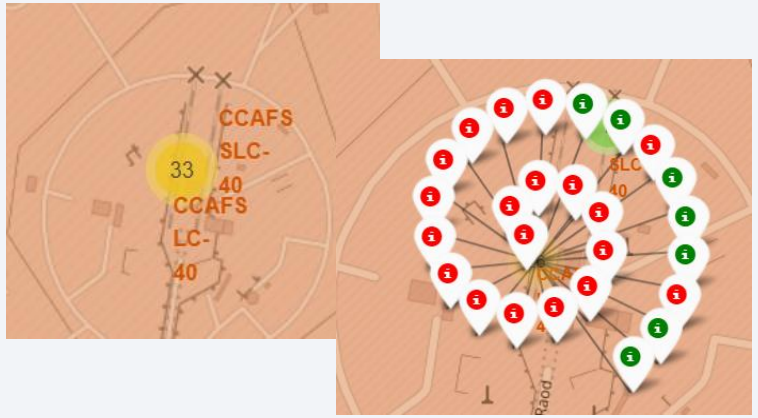


- SKSC LC-39A



SKSC LC-39A has a good success rate, but VAFB SLC-4E and CCAFS are the most used for launches

- CCAFS



Legend

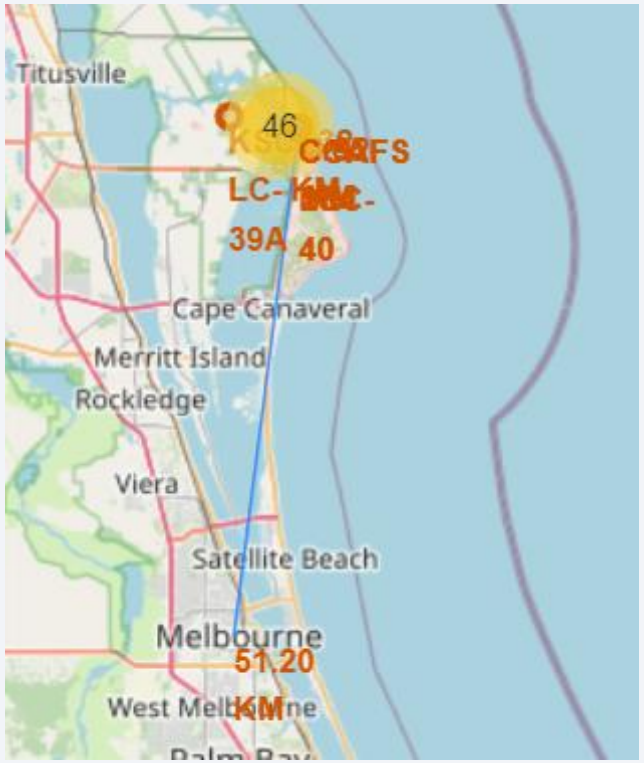
Green: success launch

Red: failure launch

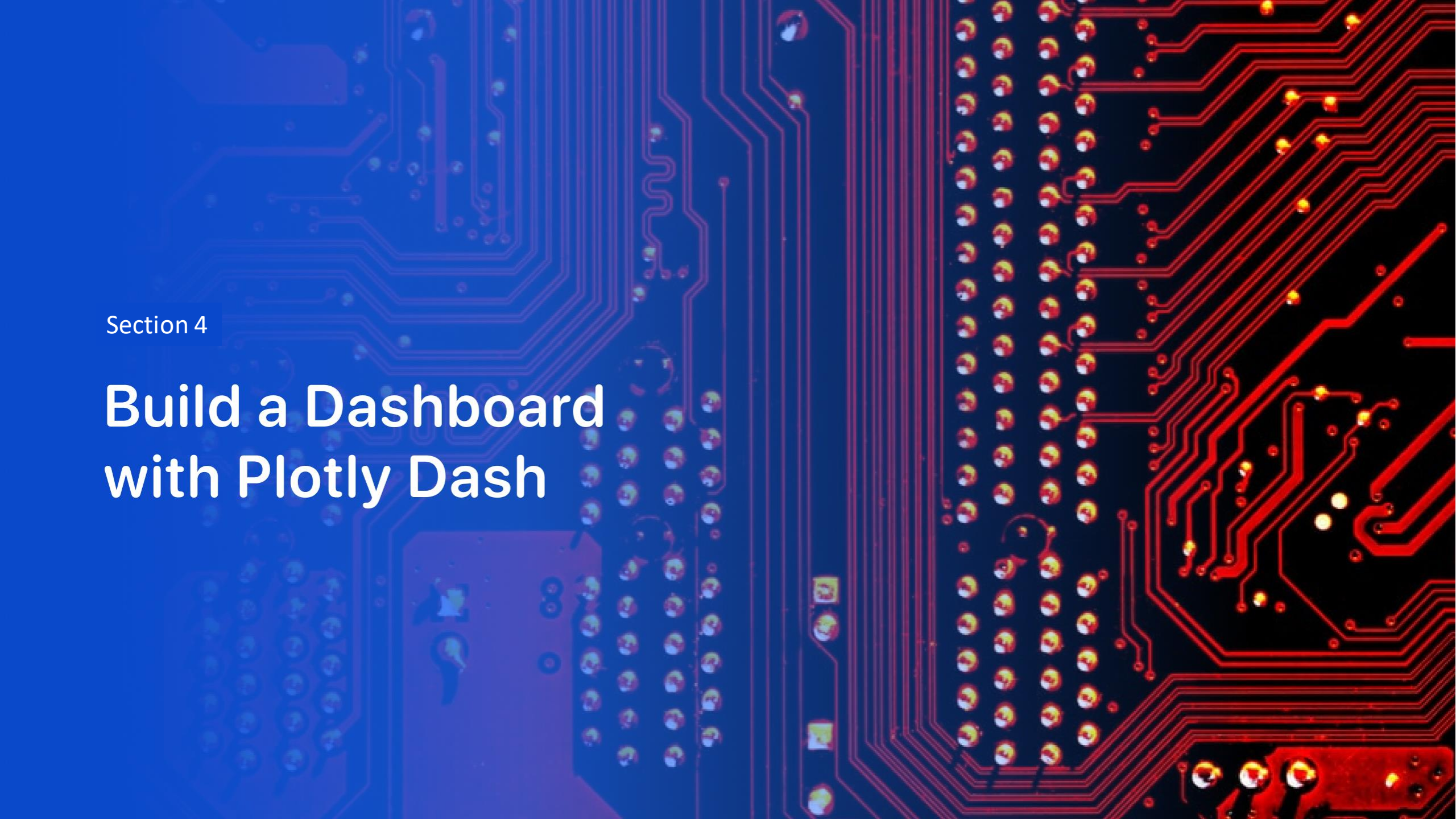
Distance to coast, highway, railway and city

Distance to railway (1.32 km), highway (0.65 km) and coast (0.92 km)

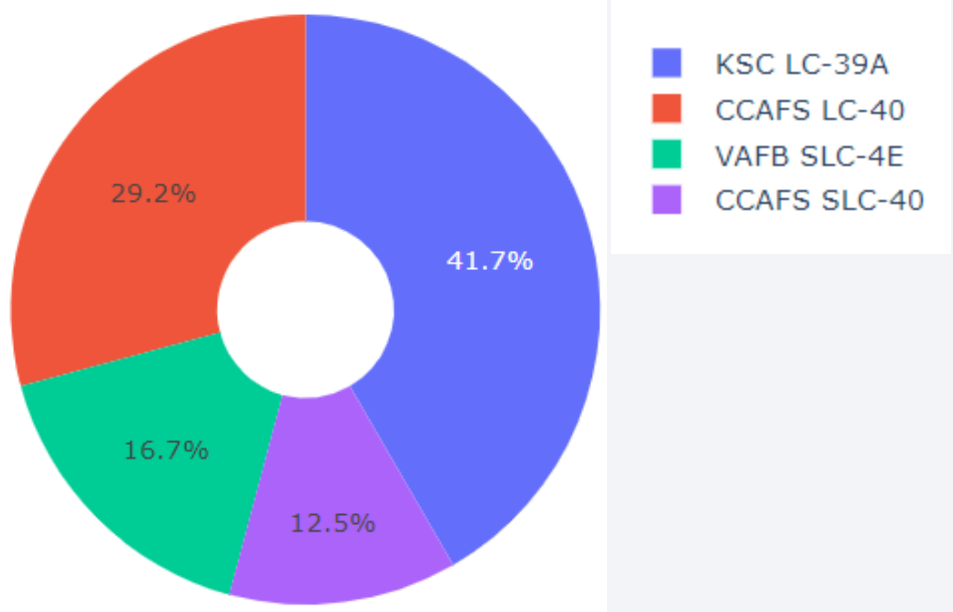
Distance to Melbourne (51.2 km)



Far from populated area (city) in case of failure, avoiding accidents. Near to highways, railways and coast in order to the logistics of receiving materials, goods, etc


Section 4
Build a Dashboard with Plotly Dash

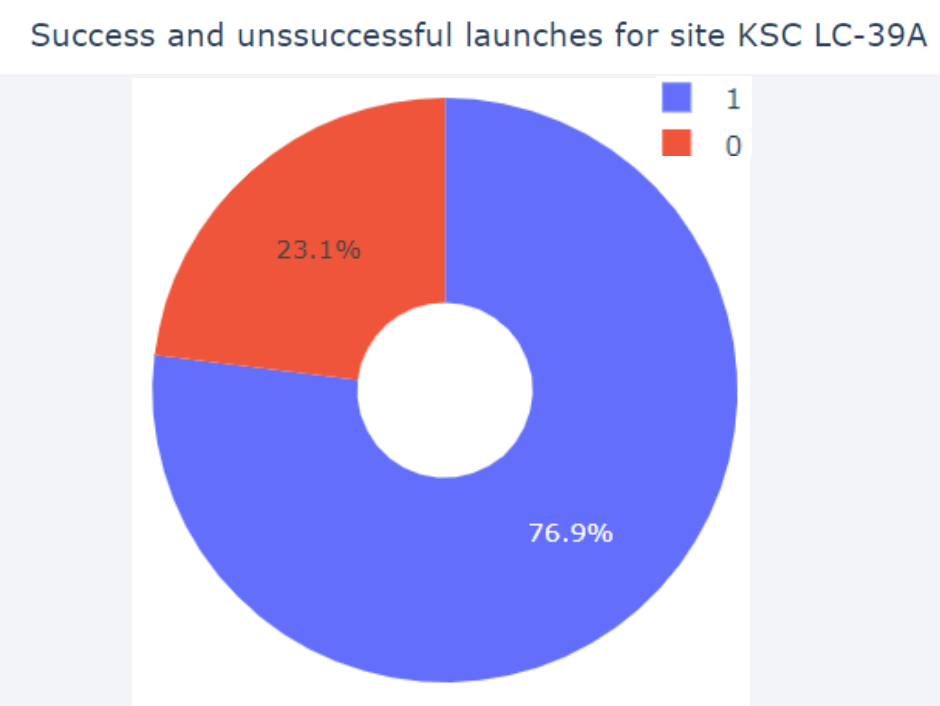
Success launch rate for all sites



- KSC LC 39A has the highest success rate considering all sites, 41,7%, followed by CCAFS LC-40 with 29,2%.

- CCAFS SLC-40 presents 12,5% success rate, the lowest.

Launch site location as a key factor



- KSC LC 39A presents a 76,9% success rate, i.e, for every 100 launches, almost 77 are successful landed.

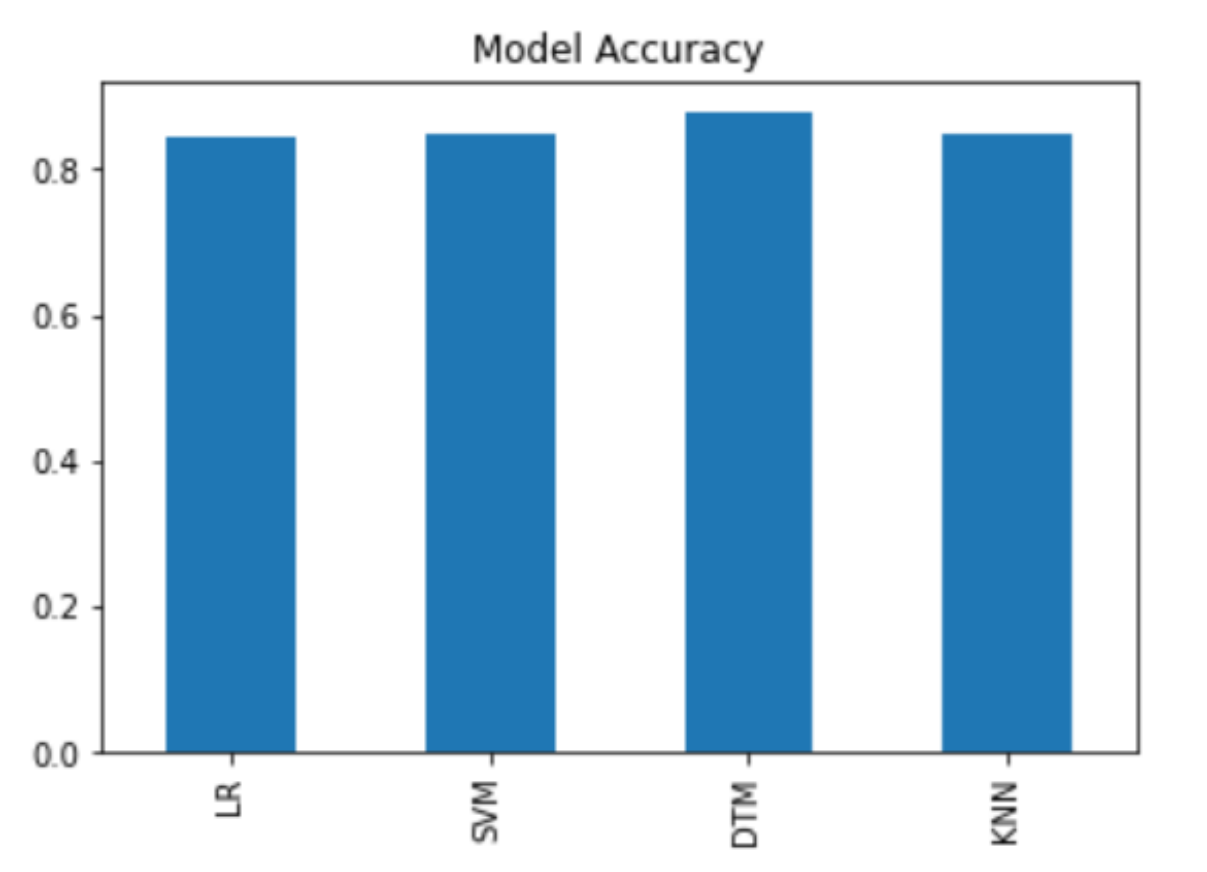
Payload is also a key factor to the success rate

- The majority launches are made with payloads lower or equal than 5000 kg.
- When payload is higher than 6000 kg, the success rate decreases.
- Payloads between 2000 and 4000 kg has a greater chance of success.

Section 5

Predictive Analysis (Classification)

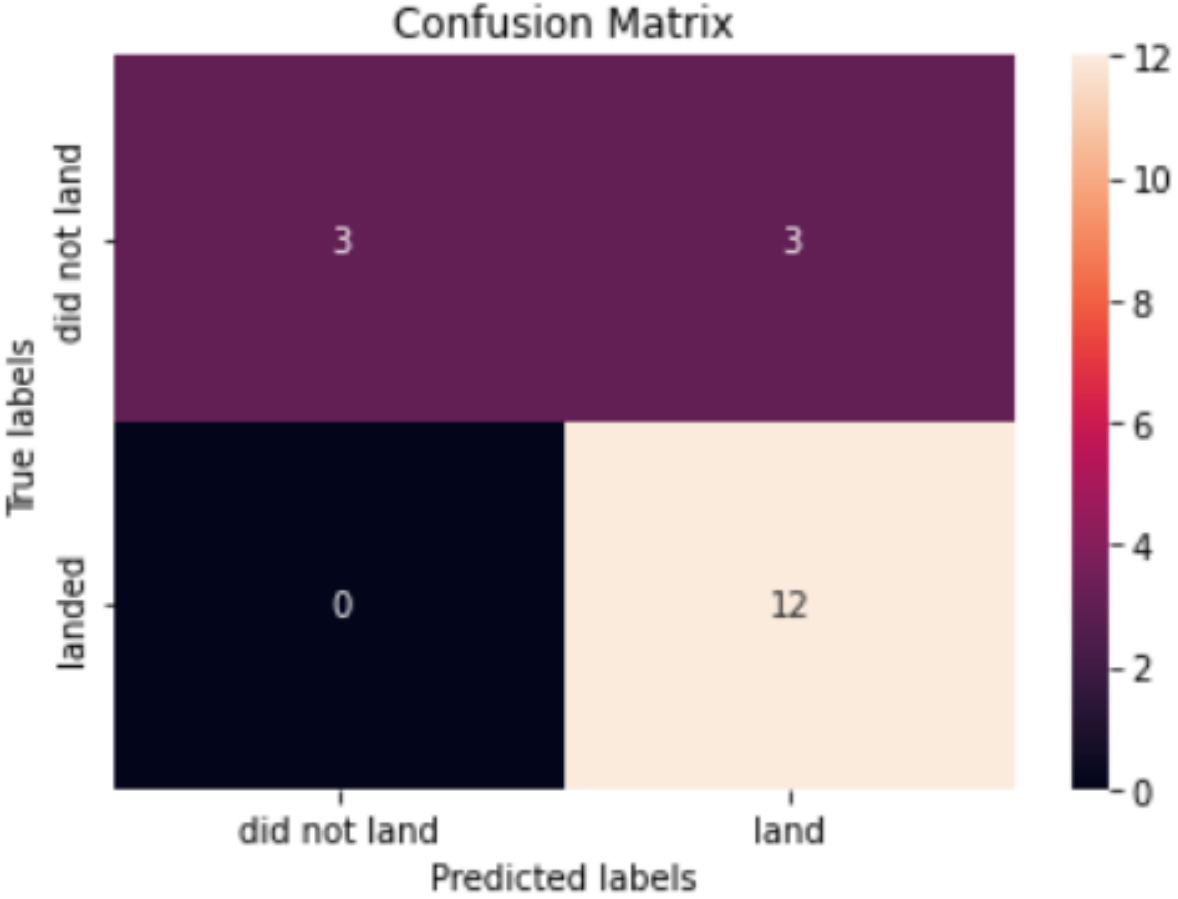
Classification Accuracy



- The chart shows the accuracy of each classification model, as follows:
  - Logistics Regression: 84,64%
  - Support Vector Machine: 84,82%
  - Decision Tree Model: 87,67%
  - K-Nearest Neighbour: 84,82%

- Decision Tree Model was the model with the highest accuracy. The others three models have similar accuracies.

Confusion Matrix for Decision Tree Model



- Confusion matrix for all classification models were the same.
- It has classified wrongly three successful missions, but in fact they were failed missions.

Conclusions

- As flight number increases, the success rate seems to increase.
- CCAFS SLC 40 is the most used site. It was the first launch and the first to achieve a successful launch.
- VAFB SLC 4E is the least used for launching, but it has a high success launch rate.
- CCAFS SLC 40 doesn't has a high success rate for payload less than to 8000 kg, but it has an excellent success rate for major payload (greater than 10000 kg). It is also the most used site for heavy loads.
- ES-L1, GEO, HEO and SSO orbits have the major success rate for launching. GTO, ISS, LEO, MEO and PO has the worst performance considering success rate. SO orbit was not used so far.

Conclusions

- Space X started with lower and transfer orbits (LEO, ISS, PO and GTO). Later Space X targeted higher orbits with a good success rate. First successful mission was on a flight to ISS. For all orbits or at least most of them, as flight number increases, success rate also increases.
- All flights until 2013 were unsuccessful. Continuous increase in successful landings from 2014 to 2017. 2018 showed a decrease in success rate, returning to increase in 2019.
- Space X launch sites are near to the coast and Equator line. It's near from the Equator line to move faster than others points of the globe and it's near to the coast to avoid populated areas during launching.
- Space X launch sites are far from populated area, avoiding accidents in crowded areas. They are near to highways, railways and coast in order to the logistics of receiving materials, goods, etc.
- KSC LC 39A has the highest success rate considering all sites, 41,7%, followed by CCAFS LC-40 with 29,2%.
- Decision Tree Model, a machine learning classification model, achieved the highest accuracy. The others three models have similar accuracies.

Appendix

- All code was developed using Python language and it was on Github

PythonCode_For_SPACEX_Project

Thank you!