Winning Space Race with Data Science

Josu Mendibe
June 2024

Outline

Executive Summary

- Subject matter

This is a Capstone Project presentation based on the outcomes of all tasks in previous modules of the IBM Data Science course.

- Summary of methodologies

Data Collection: API / Web Scraping

Data Wrangling

Exploratory Data Analysis (EDA): SQL / Data Visualization

Interactive Visual Analytics with Folium / Plotly Dash Dashboard

Predictive analysis using classification models

- Summary of all results

Exploratory Data Analysis (EDA) results

Interactive analytics in screenshots

Predictive Analytics result

Introduction

- Project background and context

The Project Scenario considers a Database with Space rocket launches of the SpaceX company.

In order to economize the launches advertised to clients, the company try to reuse part of the rockets, the first stage, so that once finished the task of pushing the load through the primary stage, the rocket, can come back and land without failure.

The data referred to the launches includes different attributes for each launch, such as Flight Number, Date, Booster version (specifically for this study is the Falcon 9 Rocket), Payload mass Orbit, Launch Site or Outcome.

- Problems you want to find answers

Because it is so expensive to send any mass (kilograms) into the space, the data from the tests will help us to predict whether the launch will be successful or not, based in the different attributes.

Section 1

Methodology

Methodology

Executive Summary

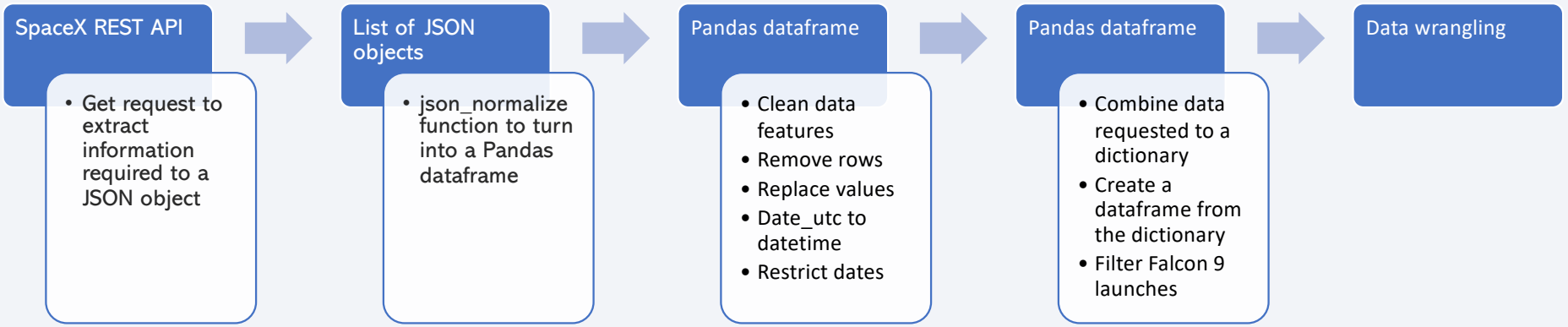
- Data collection methodology:
  - Data gathered from an API, specifically the SpaceX REST API
  - Data using the web scraping method from Wiki HTML pages.
- Perform data wrangling

JSON object collected from API and web scrapping from HTML turned into a Pandas data frame. Removed rows, columns, applied filters, dealing with missing values and data.

- Perform exploratory data analysis (EDA) using visualization and SQL
- Perform interactive visual analytics using Folium and Plotly Dash
- Perform predictive analysis using classification models

Using Machine Learning Prediction to find the method that performs best: Logistic Regression, Support Vector Machine (SVM), Decision Tree Classifier and K Nearest Neighbors.

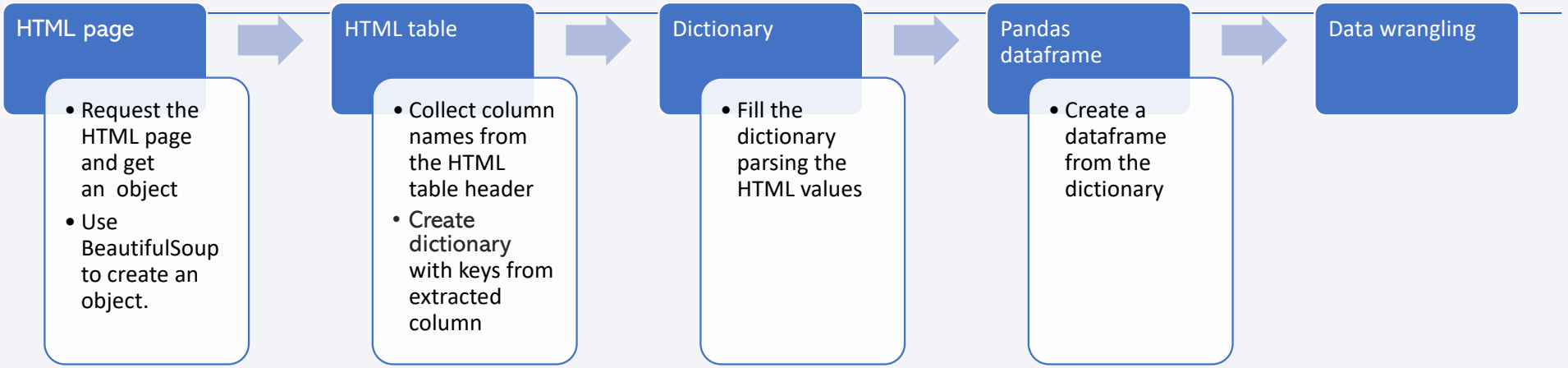
Data Collection – SpaceX API



- From the SpaceX REST API we make the get request to the SpaceX API to get the data.
- The response will be in the form of a JSON, specifically a list of JSON objects, where each one represent a launch.
- We use the json_normalize function to "normalize" the structured json data into a flat table and then to a dataframe.
- We clean some data, keeping only the features we want, removing some rows, replacing values in the list, converting data to datetime and restricting dates of the launches.
- We combine the data needed from the requested stored in lists to a dictionary, and then we create a new Pandas dataframe from the dictionary.
- Finally, we will keep only the Falcon 9 launches.

Link to GitHub URL: https://github.com/MaxMinox/Capstone_Project/blob/b709fc1709b2ead8368d3e3b88cc14cd3bd1129f/Notebooks/01%20jupyter-labs-spacex-data-collection-api.ipynb

Data Collection - Scraping



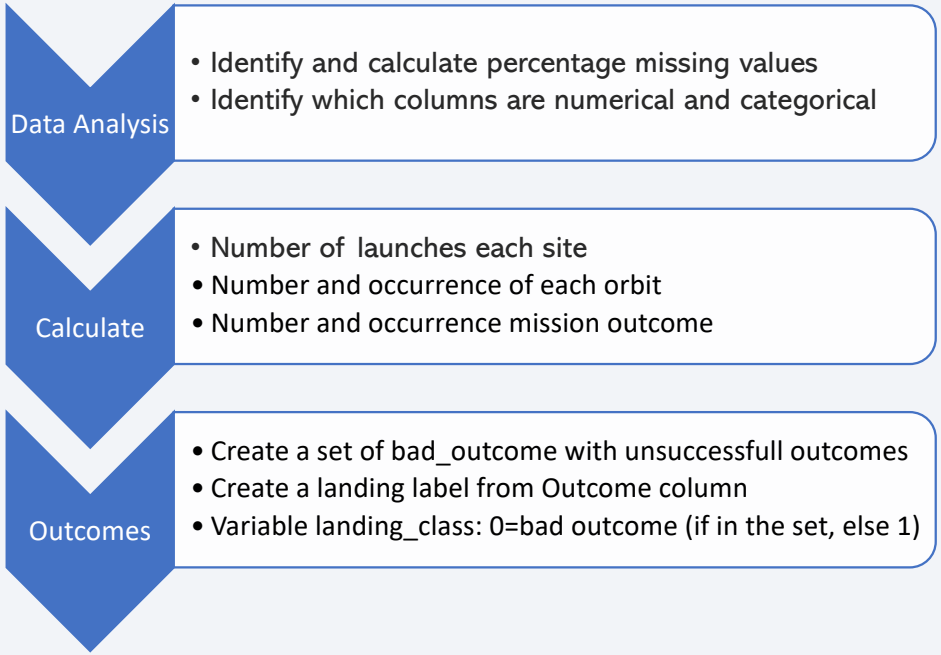
- We perform HTTP GET method to request the Falcon9 Launch HTML page, and assign the response to an object.
- We use BeautifulSoup to create an object
- We collect all relevant column names from the HTML table header
- We create an empty dictionary with keys from the extracted column names
- We fill up the dictionary with launch records extracted from table rows.
- After we filled in the parsed launch record values into the dictionary, we create a dataframe from it.

Link to GitHub URL: https://github.com/MaxMinox/Capstone_Project/blob/b709fc1709b2ead8368d3e3b88cc14cd3bd1129f/Notebooks/02%20jupyter-labs-webscraping.ipynb

Data Wrangling

- Exploratory Data Analysis (EDA) will help us find some patterns in the data and determine what would be the label for training supervised models.
- In the data set, there are several different cases where the booster did not land successfully. We will convert those outcomes into Training Labels with the following meaning:

  1 The booster successfully landed
  0 Unsuccessful.



Link to GitHub URL: https://github.com/MaxMinox/Capstone_Project/blob/b709fc1709b2ead8368d3e3b88cc14cd3bd1129f/Notebooks/03%20labs-jupyter-spacex-Data%20wrangling.ipynb

EDA with Data Visualization

We perform Exploratory Data Analysis (EDA) with data visualization using Pandas and Matplotlib.

- Combining multiple features will determine what attributes are correlated with successful landings.
- In the charts we plot out one variable vs. another, and overlay the outcome of the launch, so that we can see in the chart patterns or any statistical relationship.
- Variables used: Landing outcome (0 = Unsuccessful, 1 = Successful), Flight Number, Payload Mass, Orbit Type, Flight Number, Year.

Summary of plots:

Flight Number vs. Launch Site

Payload vs. Launch Site

Success Rate vs. Orbit Type

Flight Number vs. Orbit Type

Payload vs. Orbit Type

Launch Success Yearly Trend

Link to GitHub URL:
https://github.com/MaxMinox/Capstone_Project/blob/b709fc1709b2ead8368d3e3b88cc14cd3bd1129f/Notebooks/05%20edadataviz.ipynb

EDA with SQL

Summary of the SQL queries performed in the Jupyter notebook:

- All Launch Site Names
- Launch Site Names Begin with “CCA”
- Total Payload Mass
- Average Load Mass by F9 v 1.1
- First Successful Ground Landing Date
- Successful Drone Ship Landing with Payload between 4000 and 6000
- Total Number of Successful and Failure Mission Outcomes
- Booster Carried Maximum Payload
- 2015 Launch Records
- Rank Landing Outcomes Between 2010-06-04 and 2017-03-20

Link to GitHub URL:
https://github.com/MaxMinox/Capstone_Project/blob/b709fc1709b2ead8368d3e3b88cc14cd3bd1129f/Notebooks/04%20jupyter-labs-eda-sql-coursera_sqllite.ipynb

Build an Interactive Map with Folium

- First, we create a Folium Map object, where we will add another objects with different geographic coordinates.

Folium is a library that helps to create interactive maps and is useful for visualizing geospatial data

- The initial center location will be the NASA Johnson Space Center at Houston, Texas.

An initial specific location helps to center the view of the plot

- We add a circle for each launch site in data frame launch_sites with folium.Circle and folium.Marker

This objects will help to explore the map by zoom-in/out the marked areas, and we can see the proximities to other areas, such as the Equator line, the coast, towns, airports, roads and train railroads, etc.

- We mark the success/failed launches for each site on the map

With the launch outcomes for each site, we can see which sites have high success rates (green = success, red = failure)

- We create a MarkerCluster object for each site

As each site has many launch records (same coordinate), we create as many marker clusters as launches for each site

- We create a line from a Launch site to the coast or other point of interest, and we measure it and add this distance to the chart

Exploring and analysing the proximities of launch sites is important to be aware of the risks.

Link to GitHub URL:
https://github.com/MaxMinox/Capstone_Project/blob/b709fc1709b2ead8368d3e3b88cc14cd3bd1129f/Notebooks/06%20lab_jupyter_launch_site_location.ipynb

Build a Dashboard with Plotly Dash

Dash help us building web applications. With Plotly we can build a dashboard using Python.

The idea of this Dashboard is to see if the relationship between the Launch Site and the Payload Mass has any dependence in the Outcome of the mission

- First, we add a dropdown list to enable Launch Site selection
  In this list, we can select All the Launch Sites, or any of the other 4 Sites.

- We add a pie chart to show the total successful launches count for all sites
  This plot can help us see the success ratio for all sites or one particular site as well

- Then we add a slider to select payload range
  The slider will provide us the possibility to choose different combinations for payload mass

- Finally, we add a scatter chart to show the correlation between payload and launch success
  In this plot we can check how different payload mass may impact success rates for each launch site

Link to GitHub URL:
https://github.com/MaxMinox/Capstone_Project/blob/b709fc1709b2ead8368d3e3b88cc14cd3bd1129f/Notebooks/07%20spacex_dash_app.py

Predictive Analysis (Classification)

- We process and standardize properly the data.
    1. We use the Class column (success/unsuccessful outcome) to the variable 'Y' creating a Numpy array.
    2. We standardize the dataframe into the variable 'X' with Scikit Learn: preprocessing.StandardScaler() and fit_transform().

- We use the function train_test_split to split the data X and Y into training and test data

- To compute the probability of the success, we will use machine learning models with 4 different Scikit-Learn tools for predictive data analysis:
    1. We will create a logistic regression object with LogisticRegression()
    2. We will create a support vector machine object with SVC()
    3. We will create a decision tree classifier object with DecisionTreeClassifier()
    4. We will create a k nearest neighbors object with KNeighborsClassifier()

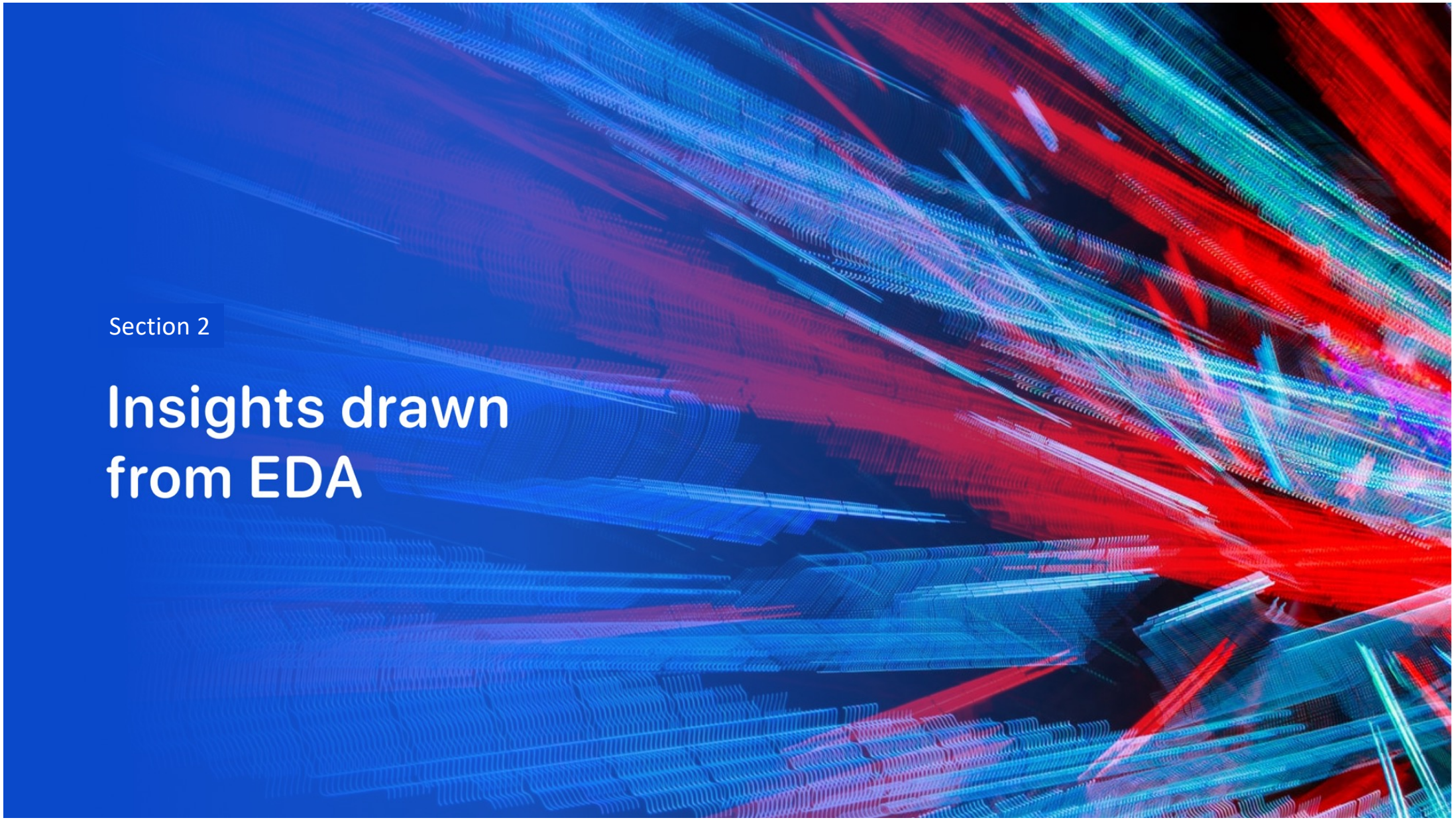
- For each model:
    1. We will display the best parameters using the data attribute best_params_ and the accuracy on the validation data using the data attribute best_score_
    2. Then we calculate the accuracy on the test data using the method score.
    3. We will plot the confusion matrix for each model

- Finally, we will find the method which performs best.

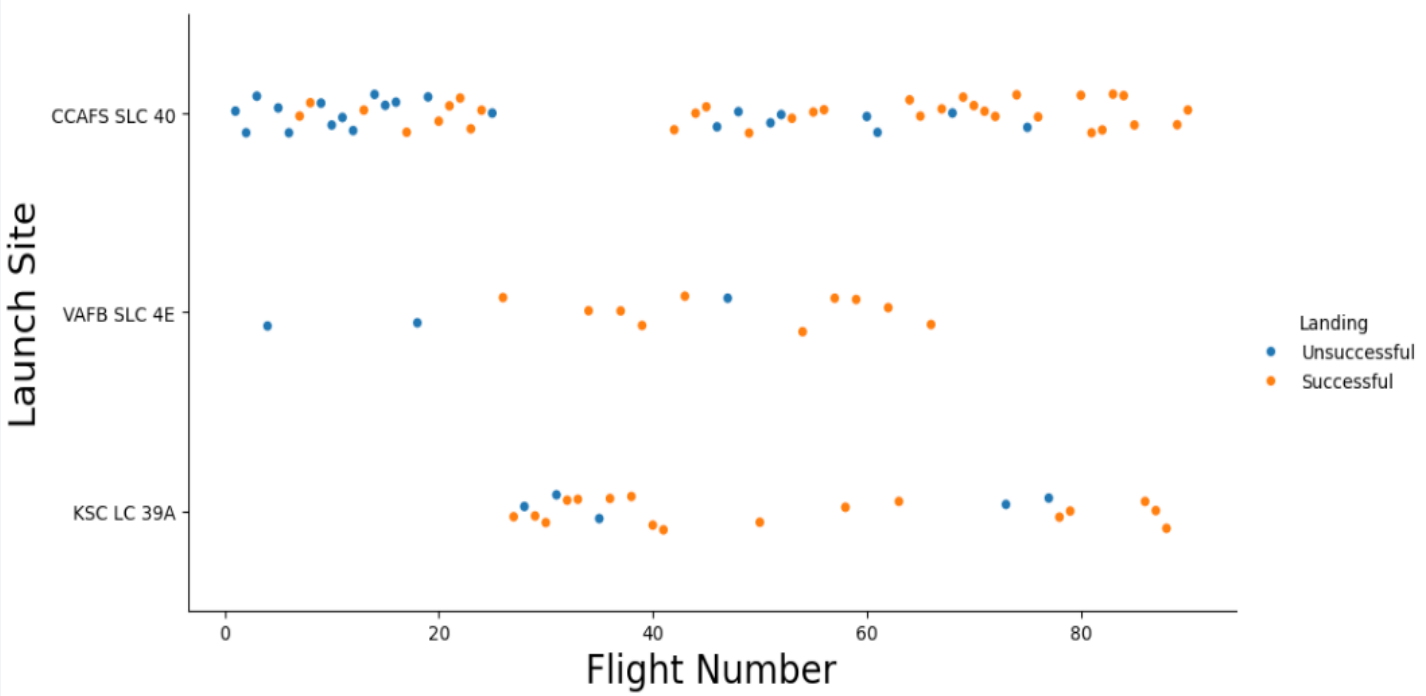
Link to GitHub URL:
https://github.com/MaxMinox/Capstone_Project/blob/b709fc1709b2ead8368d3e3b88cc14cd3bd1129f/Notebooks/08%20SpaceX_Machine%20Learning%20Prediction_Part_5.ipynb

Results

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


Section 2
Insights drawn from EDA

Flight Number vs. Launch Site

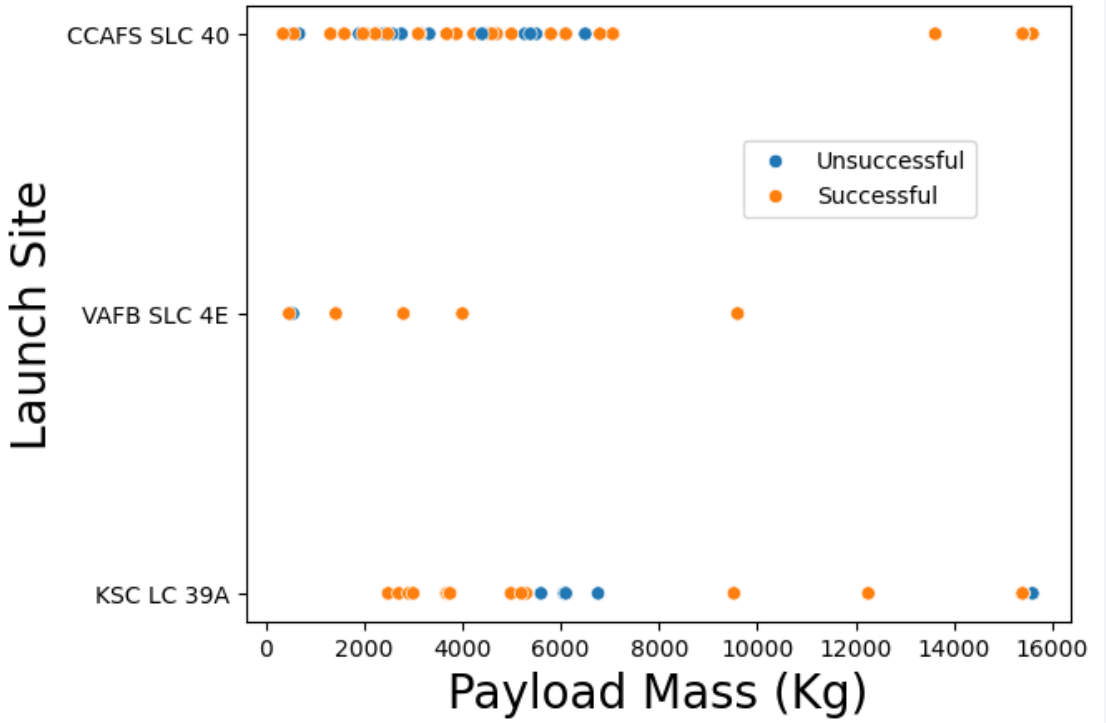


Success rate increases with Flight Number

```
df['Landing']=df['Class']
df['Landing'].replace(0, 'Unsuccessful', inplace=True)
df['Landing'].replace(1, 'Successful', inplace=True)
```

```
### TASK 1: Visualize the relationship between Flight Number and Launch Site
sns.catplot(y="LaunchSite", x="FlightNumber", hue="Landing", data=df, aspect = 2)
plt.xlabel("Flight Number",fontsize=20)
plt.ylabel("Launch Site",fontsize=20)
plt.show()
```

Payload vs. Launch Site

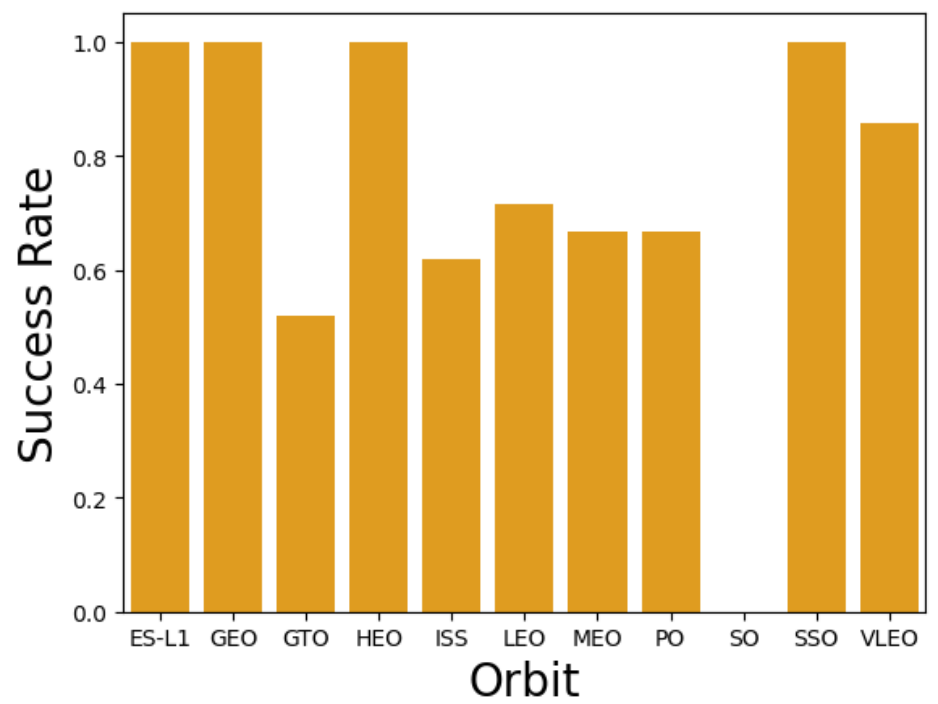


Success rate increases with Payload Mass.

VAFB-SLC has no rockets launched for Payload > 10000.

```
# Plot a scatter point chart with x axis to be Pay Load Mass (kg) and y axis to be the launch site, and hue to be the class value
sns.scatterplot(y="LaunchSite", x="PayloadMass", hue='Landing', data=df)
plt.xlabel("Payload Mass (Kg)",fontsize=20)
plt.ylabel("Launch Site",fontsize=20)
plt.legend(loc='upper right', bbox_to_anchor=(0.9, 0.8), ncol=1)
plt.show()
```

Success Rate vs. Orbit Type

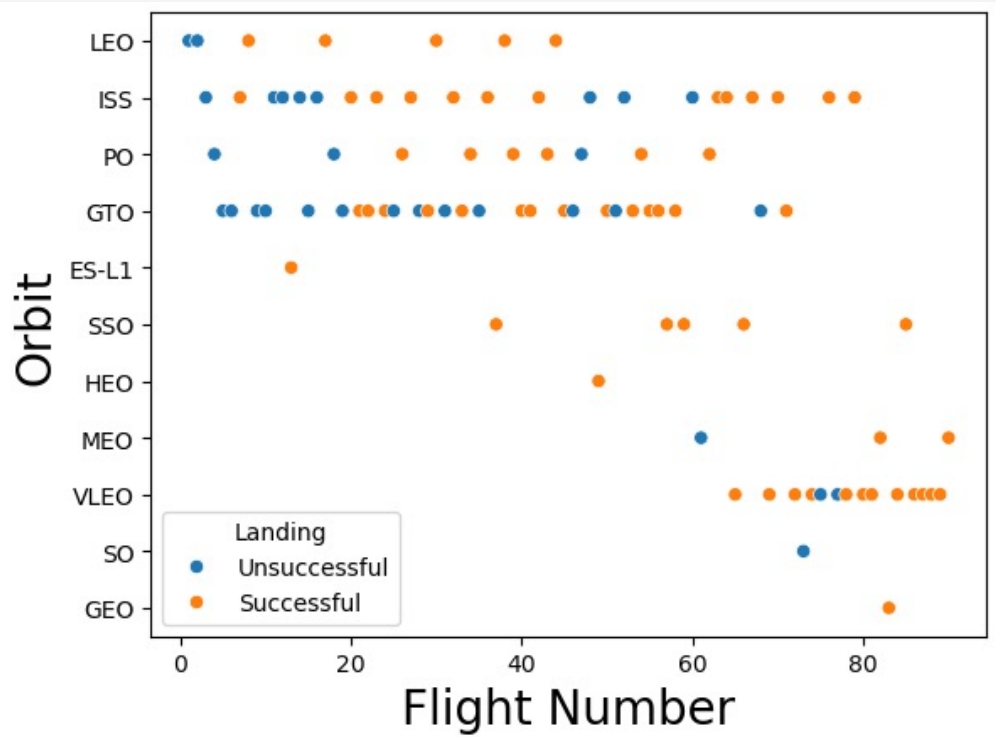


The orbits with the most successful rate are:
ES-L1, GEO, HEO and SSO.

SO orbit has a complete unsuccessful rate.

```
# HINT use groupby method on Orbit column and get the mean of Class column
df_orbit = df.groupby(df['Orbit'], as_index=False).agg({"Class": "mean"})
sns.barplot(y="Class", x="Orbit", data=df_orbit, color='orange')
plt.xlabel("Orbit", fontsize=20)
plt.ylabel("Success Rate", fontsize=20)
plt.show()
```

Flight Number vs. Orbit Type



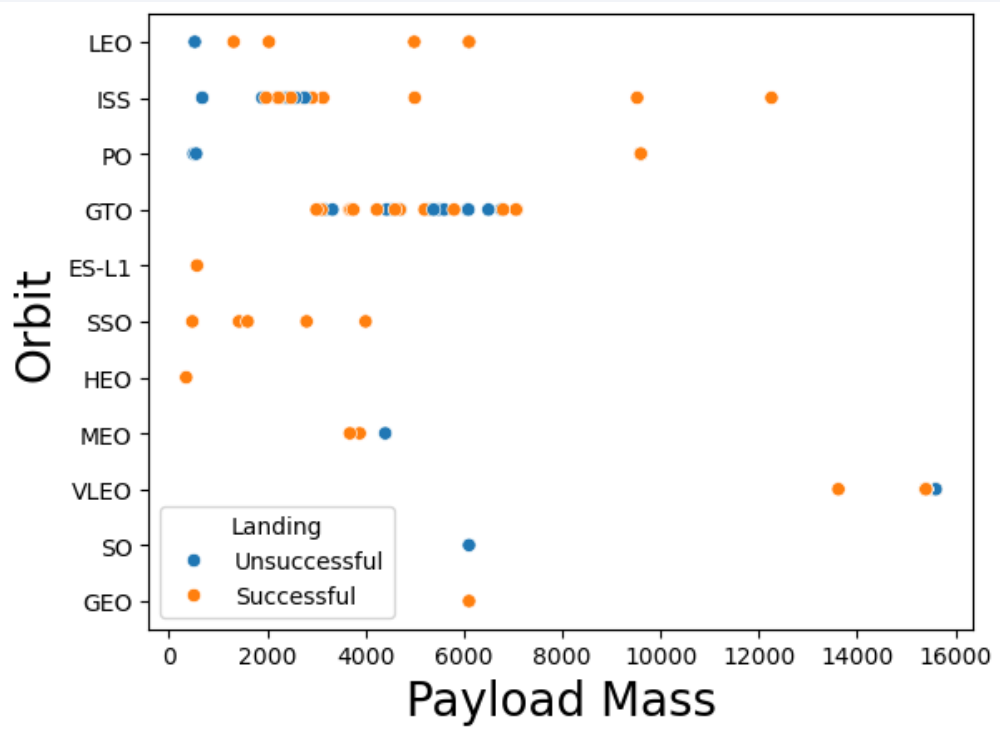

```
# Plot a scatter point chart with x axis to be FlightNumber and y axis to be the Orbit, and hue to be the class value
sns.scatterplot(y="Orbit", x="FlightNumber", hue="Landing", data=df)
plt.xlabel("Flight Number", fontsize=20)
plt.ylabel("Orbit", fontsize=20)
plt.show()
```

The success rate in Orbits increases with the Flight Number, more detectable in LEO orbit.

Orbits LEO, ISS, PO, GTO and VLEO have more data.

Orbits ES-L1, HEO, SO and GEO have few data.

Payload vs. Orbit Type

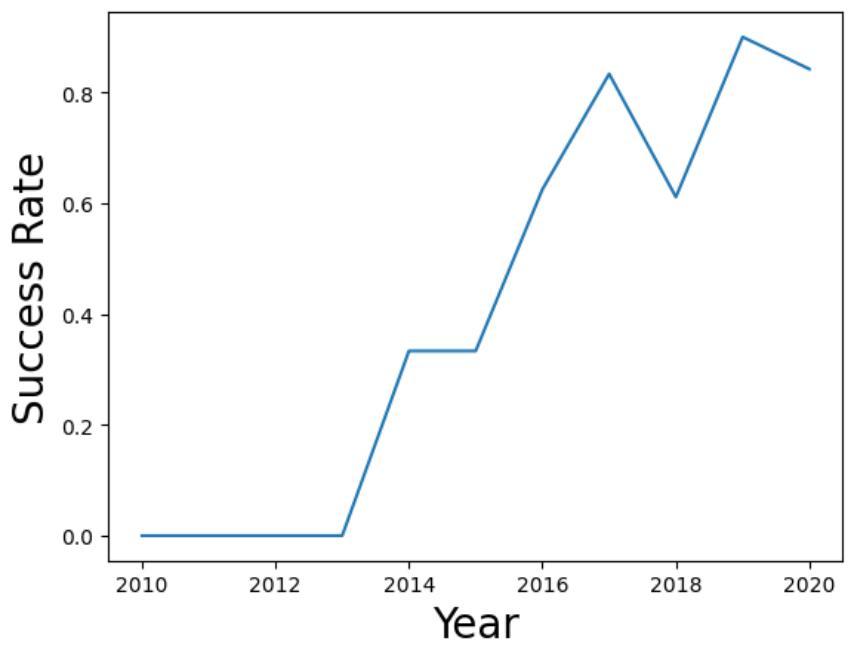


Heavy payloads affects increasing the success rate in orbits like LEO, ISS and PO.

Orbits GTO, MEO and VLEO do not seem to be impacted by payload mass.

```
# Plot a scatter point chart with x axis to be Payload and y axis to be the Orbit, and hue to be the class value
sns.scatterplot(y="Orbit", x="PayloadMass", hue="Landing", data=df)
plt.xlabel("Payload Mass", fontsize=20)
plt.ylabel("Orbit", fontsize=20)
plt.show()
```

Launch Success Yearly Trend



The success rate keep increasing since 2013

```
# A function to Extract years from the date
df['Year'] = pd.DatetimeIndex(df["Date"]).year.astype(int)
df_y = df.groupby(df['Year'], as_index=False).agg({"Class": "mean"})

sns.lineplot(y="Class", x="Year", data=df_y)
plt.xlabel("Year", fontsize=20)
plt.ylabel("Success Rate", fontsize=20)
plt.show()
```

All Launch Site Names

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

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

```
%sql select distinct Launch_Site from SPACEXTABLE
 * sqlite:///my_data1.db
Done.
```

Launch Site Names Begin with 'CCA'

List with the first 5 rows of the Launch Sites whose names begin with ‘CCA’

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

```
%sql select * from SPACEXTABLE where Launch_Site like 'CCA%' limit 5
```

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

Total Payload Mass

sum(PAYLOAD_MASS__KG_)
45596

Total payload mass carried by boosters launched by NASA (CRS)

```
%sql select sum(PAYLOAD_MASS__KG_) from SPACEXTABLE where Customer='NASA (CRS)'
```

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

Average Payload Mass by F9 v1.1

avg(PAYLOAD_MASS__KG_)
2928.4

Total payload mass carried by booster version F9 v1.1

```
%sql select avg(PAYLOAD_MASS__KG_) from SPACEXTABLE where Booster_Version like 'F9 v1.1'
```

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

First Successful Ground Landing Date

min(DATE)
2015-12-22

Date when the first successful landing outcome in ground pad was achieved.

```
%sql select min(DATE) from SPACEXTABLE where Landing_Outcome='Success (ground pad)'
 * sqlite:///my_data1.db
Done.
```

Successful Drone Ship Landing with Payload between 4000 and 6000

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

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

```
%sql select Booster_Version from SPACEXTABLE where Landing_Outcome='Success (drone ship)' and PAYLOAD_MASS__KG_ between 4000 and 6000
 * sqlite:///my_data1.db
Done.
```

Total Number of Successful and Failure Mission Outcomes

Failure/Success	count()
F	1
S	100

Total number of successful and failure mission outcomes.

```
%sql select substr(Mission_Outcome, 1, 1) as 'Failure/Success', count() from SPACEXTABLE group by 1
 * sqlite:///my_data1.db
Done.
```

Boosters Carried Maximum Payload

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

Names of the booster_versions which have carried the maximum payload mass.

```
%sql select distinct Booster_Version, PAYLOAD_MASS__KG_  from SPACEXTABLE where PAYLOAD_MASS__KG_ = (select max(PAYLOAD_MASS__KG_) from SPACEXTABLE)
```

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

2015 Launch Records

Records which display the month names, failure landing_outcomes in drone ship ,booster versions and launch_site for the months in year 2015.

Month	Landing_Outcome	Booster_Version	Launch_Site
01	Failure (drone ship)	F9 v1.1 B1012	CCAFS LC-40
04	Failure (drone ship)	F9 v1.1 B1015	CCAFS LC-40

```
%sql select substr(Date, 6, 2) as Month, Landing_Outcome, Booster_Version, Launch_Site from SPACEXTABLE where Landing_Outcome='Failure (drone ship)' and Date like '%2015%'
 * sqlite:///my_data1.db
Done.
```

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

Landing_Outcome	count()
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

Ranking of the count of landing outcomes between the date 2010-06-04 and 2017-03-20, in descending order.

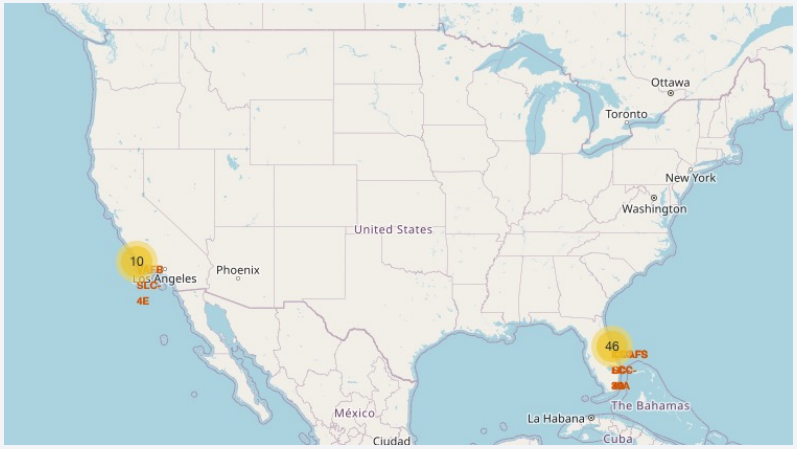
```
%sql select distinct Landing_Outcome, count() from SPACEXTABLE where Date between '2010-06-04' and '2017-03-20' group by Landing_Outcome order by 2 desc
 * sqlite:///my_data1.db
Done.
```


Section 3
Launch Sites
Proximities Analysis

All Launch Sites on a Map

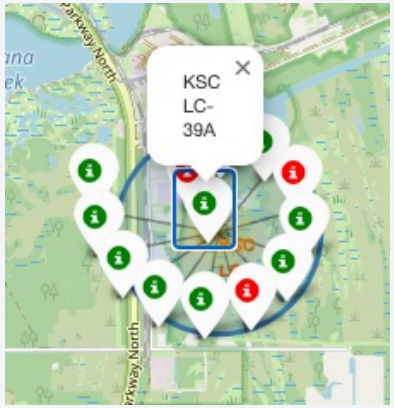
In the chart we can see all the Launch Sites are in EE.UU.

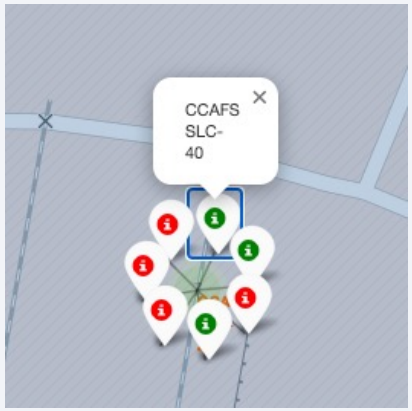
Success and failed launches for each site on the map

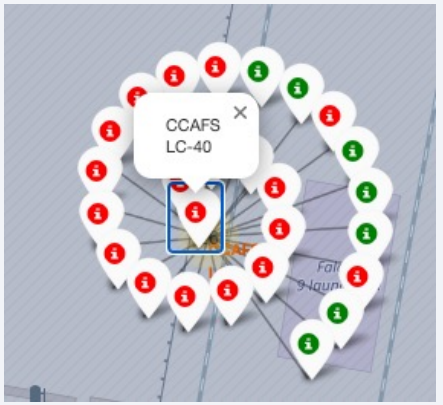


When we zoom in the map and click on the launch sites, we can see the number of Flights of each one, in green with success, in red unsuccessful launches.

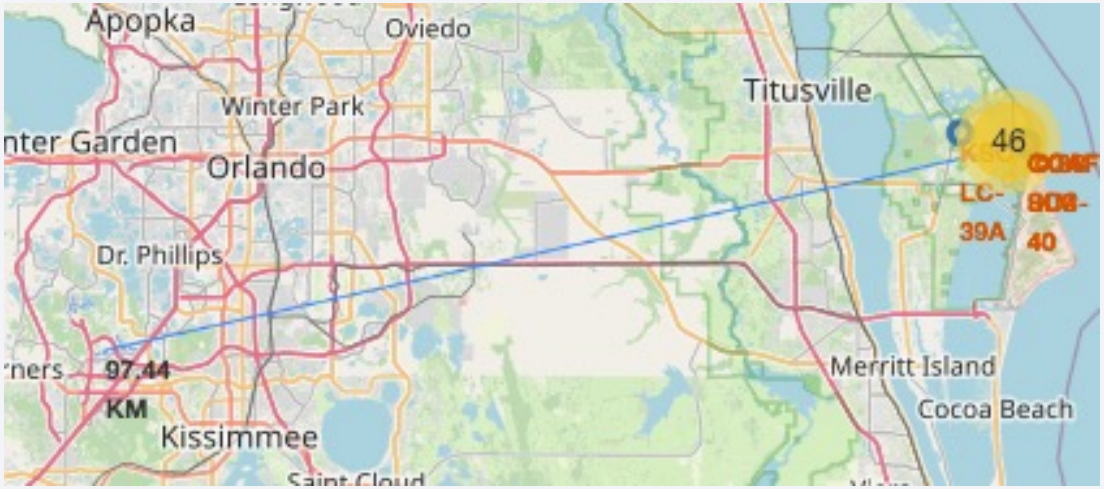
We also can see that KSC LC-39 site has more successful launches than the others sites.





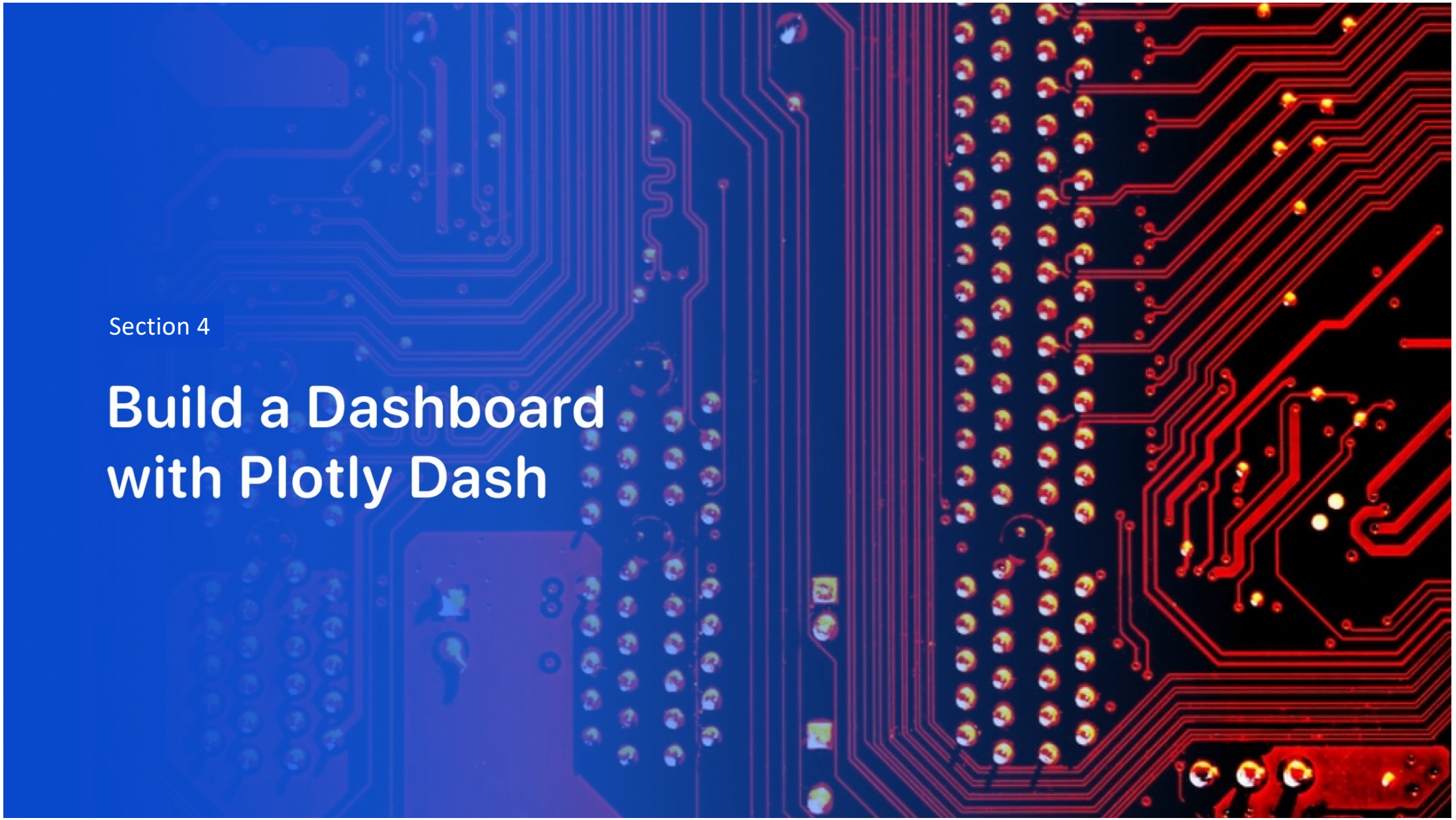


Distances between a launch site to its proximities



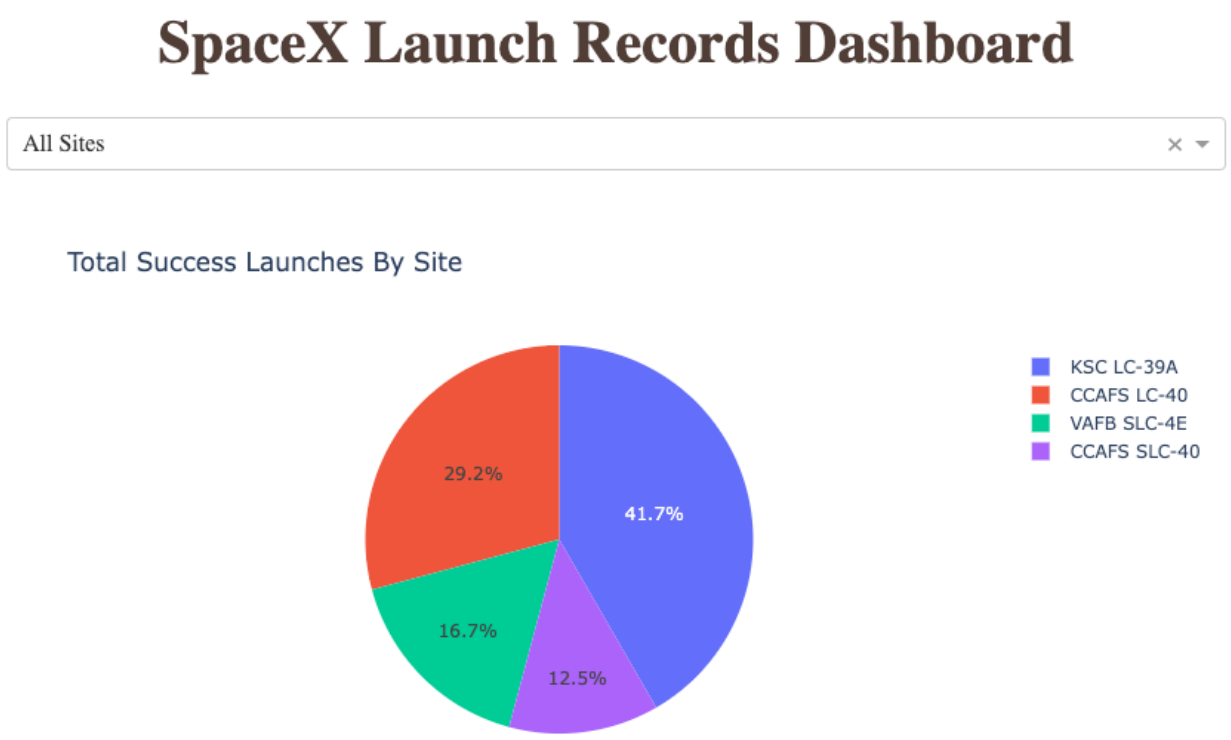
When we measure the distance from a site to an especific point on the map, we can estimate the risk in case of an accident.

In the example we measured the distance from CCAFS LC-40 to Disneyland: 97.44 Km., so we could estimate that the risk is low.


Section 4
Build a Dashboard with Plotly Dash

SpaceX Launch Records Dashboard

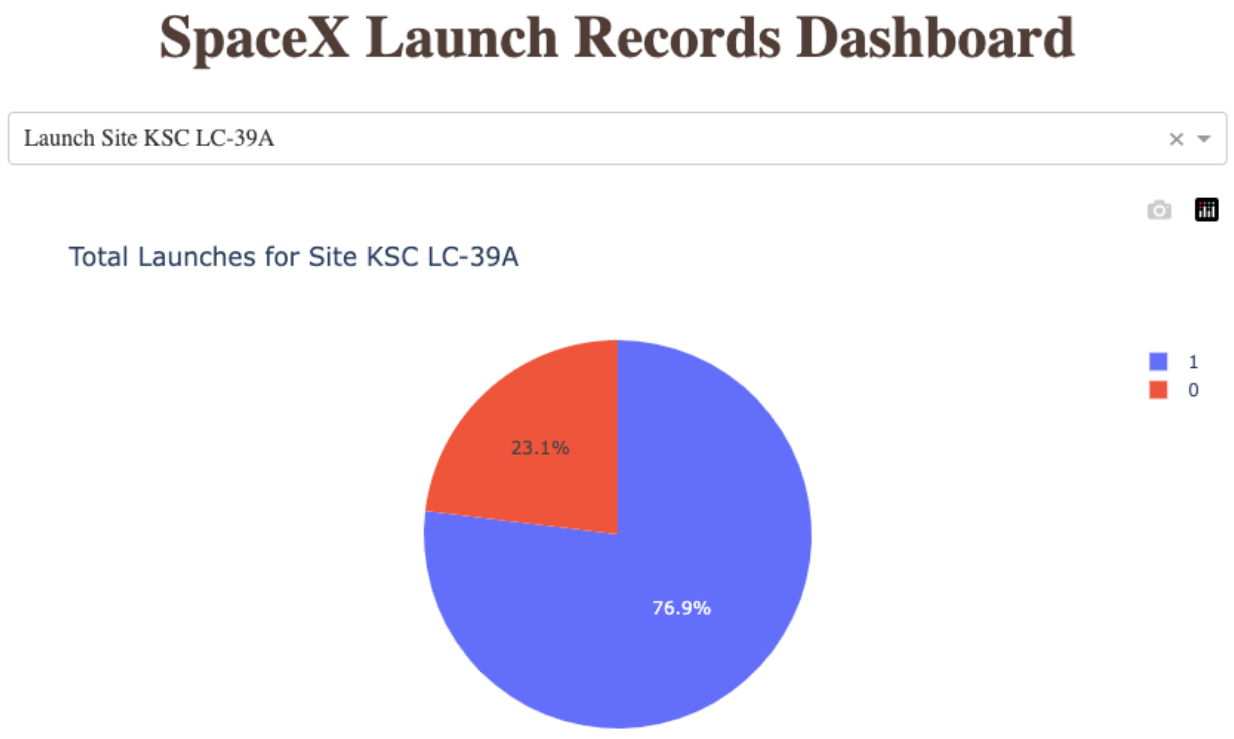
Total success launches by site



In this plot we can see that for All Sites, the best rate of launches is for KSC LC-39A site.

SpaceX Launch Records Dashboard

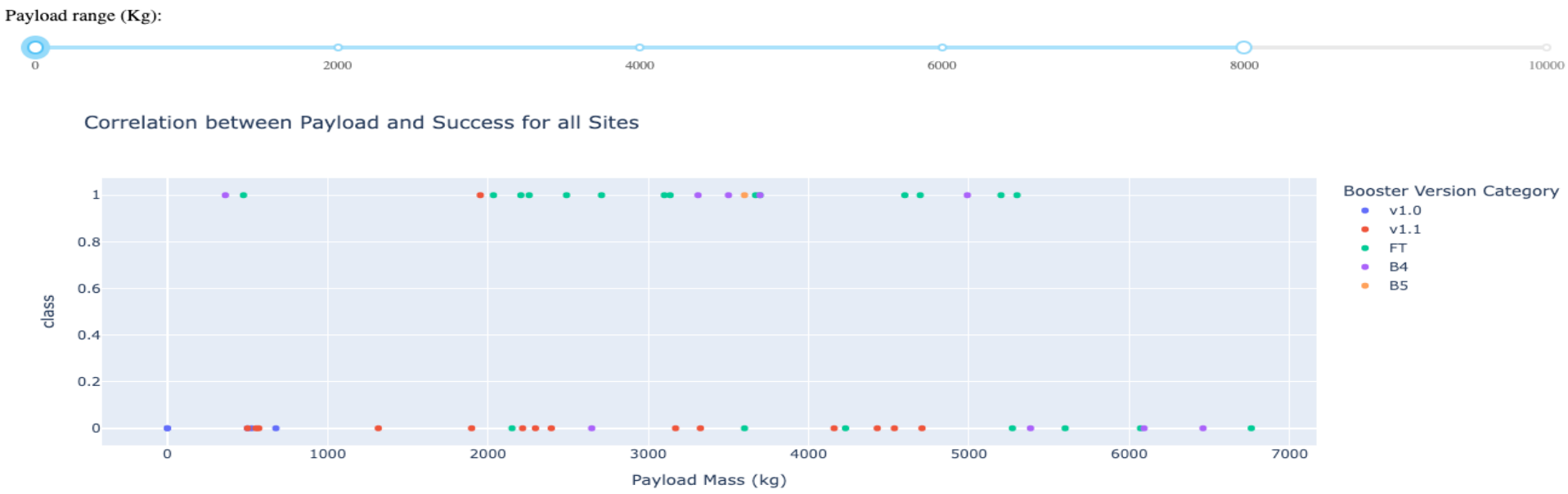
Launch site with highest launch success ratio



In this plot we can see for site KSC LC-39A the success launches rate is 76,9%.

SpaceX Launch Records Dashboard

Payload vs. Launch Outcome scatter plot for all sites



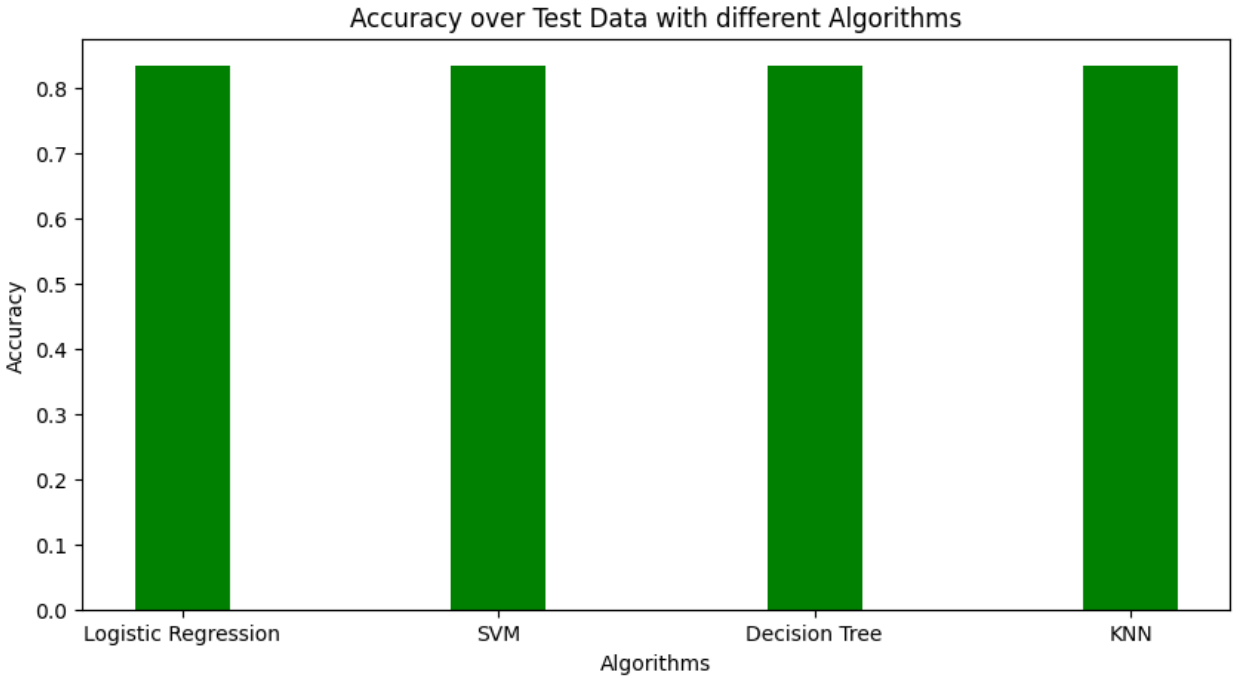
In this plot, we can find that the FT Booster Version has the highest launch success rate

It also looks like the optimal payload mass for this FT Booster is in the range between 500 Kg and 5500 Kg

Section 5

Predictive Analysis (Classification)

Classification Accuracy

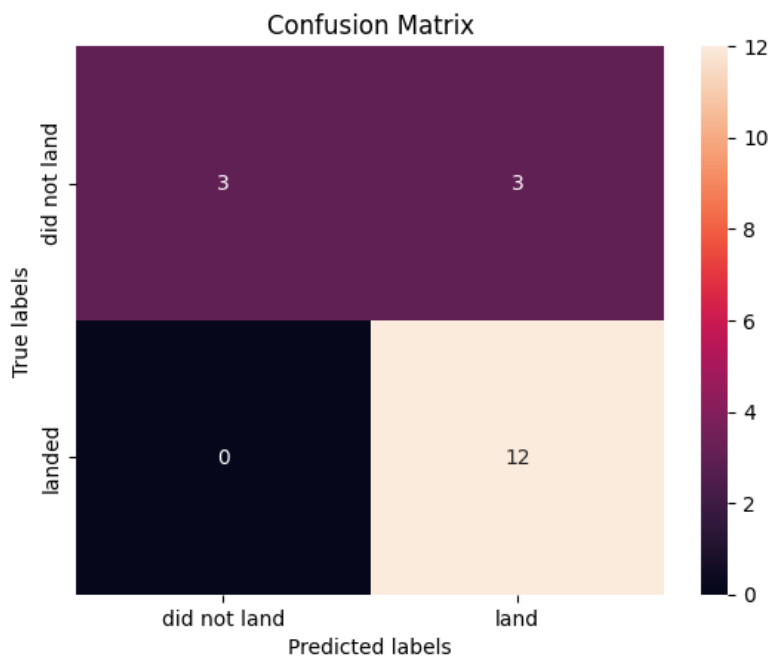


	Accuracy
Logistic Regression	0.833333
SVM	0.833333
Decision Tree	0.833333
KNN	0.833333

All models show the same accuracy score with the Test Data:

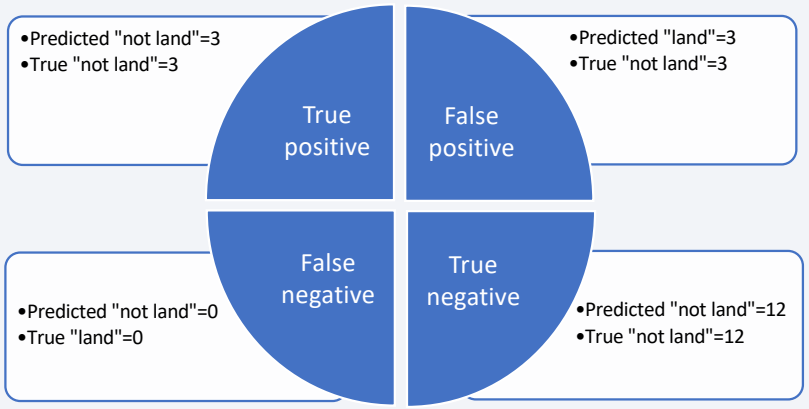
That is: 0,833 = 83,33%

Confusion Matrix



The confusion matrix visualizes and summarizes the performance of a classification algorithm.

The 4 models show the same Confusion Matrix Plot predicting the results.



Interpreting results we can see the problem is in the false positives

Conclusions

- Success increases with Flight numbers
- Success increases with higher payload mass
- The orbits with the most successful rate are ES-L1, GEO, HEO and SSO
- Success keep increasing since 2013 until 2020
- KSC LC-39A has the best launch success rate and more successful launches of any site
- FT Booster Version has the highest launch success rate
- All 4 machine learning models have the same accuracy of 83,33%, predicting a high success rate.
- In spite of this accuracy, all the models produce the same significant number of False Positive cases when predicting successful launches .

Appendix

- Link to GitHub files:

Jupyter Notebooks and Code:

https://github.com/MaxMinox/Capstone_Project/tree/b709fc1709b2ead8368d3e3b88cc14cd3bd1129f/Notebooks

Datasets

https://github.com/MaxMinox/Capstone_Project/tree/b709fc1709b2ead8368d3e3b88cc14cd3bd1129f/Data

Thank you!