IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

Orestis Bouras
10/11/2023

Outline

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

Executive Summary

The target of this capstone assignment is to predict if the SpaceX Falcon 9 first stage will land successfully. The data which was used to complete the investigation are originated from the SpaceX launch dataset while the main steps followed are summarized below:

- Data collection, wrangling and formatting
- Exploratory data analysis
- Interactive visual analytics
- Predictive analysis with several classification algorithms

Based on the investigation, it was concluded that some of the rockets' features have a strong correlation to the outcome of the launches and can guide the final result. Furthermore, after executing and comparing the accuracies of several classification algorithms, it is concluded that the decision tree algorithm performs the best on predicting if the Falcon 9 first stage will be landed successfully.

Introduction

In this capstone, we will predict if the Falcon 9 first stage will land successfully. SpaceX 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 SpaceX can reuse the first stage. Therefore, if we can determine if the first stage will land, we can determine the cost of a launch. This information can be used if an alternate company wants to bid against SpaceX for a rocket launch.

The main question to be answered from this capstone assignment is if the first stage of the rocket will land successfully. Towards getting a handle on the problem and providing a complete answer, a detailed investigation is performed; at first the available data are analyzed and interpreted and based on interactive visual analytics, a machine learning algorithm is developed to provide a prediction with a satisfying accuracy.

Section 1

Methodology

Methodology

1. Data collection methodology using:
   - SpaceX API
   - Web scraping
2. Perform data wrangling, formatting and exploratory data analysis (EDA) using:
   - Pandas and NumPy libraries
   - SQL
3. Perform interactive visual analytics:
   - Matplotlib and Seaborn libraries
   - Folium
   - Dash
4. Perform predictive analysis using classification algorithms as:
   - Logistic Regression
   - Support vector machine (SVM)
   - Decision tree
   - K-nearest neighbors (KNN)

Data Collection – SpaceX API

- The API data can be found in api.spacexdata.com/v4/rockets/
- The API provides data about several types of rocket launches that are executed by SpaceX. The current assignemnt is focused on the Falcon 9 launches, thus the data are filtered on the ‘BoosterVersion’ column.
- Several columns include missing values. For the ‘PayloadMass’ column, the missing values are replaced by the column’s mean value.
- The head of the formated dataframe is presented below:

FlightNumber	Date	BoosterVersion	PayloadMass	Orbit	LaunchSite	Outcome	Flights	GridFins	Reused	Legs	LandingPad	Block	ReusedCount	Serial	Longitude	Latitude
1	2010-06-04	Falcon 9	NaN	LEO	CCSFS SLC 40	None None	1	False	False	False	None	1.0	0	B0003	-80.577366	28.561857
2	2012-05-22	Falcon 9	525.0	LEO	CCSFS SLC 40	None None	1	False	False	False	None	1.0	0	B0005	-80.577366	28.561857
3	2013-03-01	Falcon 9	677.0	ISS	CCSFS SLC 40	None None	1	False	False	False	None	1.0	0	B0007	-80.577366	28.561857
4	2013-09-29	Falcon 9	500.0	PO	VAFB SLC 4E	False Ocean	1	False	False	False	None	1.0	0	B1003	-120.610829	34.632093
5	2013-12-03	Falcon 9	3170.0	GTO	CCSFS SLC 40	None None	1	False	False	False	None	1.0	0	B1004	-80.577366	28.561857

- The complete jupyter notebook file where the Data Collection API process is developped can be found within my GitHub repository in IBM-Data-Science-Capstone-Project/1_Data Collection API.ipynb at main · Bouraso/IBM-Data-Science-Capstone-Project (github.com)

Data Collection – Web Scraping

- The data is scraped from [List of Falcon 9 and Falcon Heavy launches - Wikipedia]
- The main dataframe is created through a BeautifulSoup object and by parsing the launch HTML tables
- The following picture presents the head of the final dataframe:

	Flight No.	Launch site	Payload	Payload mass	Orbit	Customer	Launch outcome	Version Booster	Booster landing	Date	Time
0	1	CCAFS	Dragon Spacecraft Qualification Unit	0	LEO	SpaceX	Success\n	F9 v1.0B0003.1	Failure	4 June 2010	18:45
1	2	CCAFS	Dragon	0	LEO	NASA	Success	F9 v1.0B0004.1	Failure	8 December 2010	15:43
2	3	CCAFS	Dragon	525 kg	LEO	NASA	Success	F9 v1.0B0005.1	No attempt\n	22 May 2012	07:44
3	4	CCAFS	SpaceX CRS-1	4,700 kg	LEO	NASA	Success\n	F9 v1.0B0006.1	No attempt	8 October 2012	00:35
4	5	CCAFS	SpaceX CRS-2	4,877 kg	LEO	NASA	Success\n	F9 v1.0B0007.1	No attempt\n	1 March 2013	15:10

- The complete jupyter notebook file where the Data Collection with Web Scraping process is developped can be found within my GitHub repository in IBM-Data-Science-Capstone-Project/2_Data Collection with Web Scraping.ipynb at main · Bouraso/IBM-Data-Science-Capstone-Project (github.com)

Data Wrangling

- Functions from the Pandas & NumPy libraries were used to derive useful information about the data (pictures below). For example:
  - ❖ Calculate the number of launches per LaunchSite via the “value_counts()” function
  - ❖ Similarly, calculate the number of launches per Orbit
  - ❖ An extra column “Class” was added, being equal to 0 if the Outcome was a failure or to1 if it was a success
  - ❖ Calculation of Success rate based on the mean value of “Class” and the “Outcome” column

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

CCAFS SLC 40    55
KSC LC 39A      22
VAFB SLC 4E     13
```

```
# Apply value_counts on Orbit column
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
```

```
df["Class"].mean()
df['Outcome'].value_counts()

True ASDS      41
None None      19
True RTLS      14
False ASDS      6
True Ocean      5
False Ocean     2
None ASDS       2
False RTLS      1
```

- The complete jupyter notebook file where the Data Wrangling process is developped can be found within my GitHub repository in [IBM-Data-Science-Capstone-Project/3_Data Wrangling.ipynb at main · Bouraso/IBM-Data-Science-Capstone-Project (github.com)]

EDA with SQL

- In this step, several SQL queries were used to answer important questions about the data. A summarized list of the main SQL queries and the relevant information is presented below:
  - ❖ Remove the blanks from the table
  - ❖ Display the names of each unique launch site
  - ❖ Display 5 records where launch sites begin with the string 'CCA'
  - ❖ Display the total payload mass carried by boosters launched by NASA (CRS)
  - ❖ Display the average payload mass carried by booster F9 v1.1
  - ❖ List the date when the first successful landing outcome in ground pad was achieved
  - ❖ List the boosters with success in drone ship and with payload mass between 4000 and 6000
  - ❖ List the total number of successful and failure mission outcomes
  - ❖ List the booster versions which have the maximum payload mass
  - ❖ List the records (specified columns) in year 2015
  - ❖ Rank the count of landing outcomes between a specified date period
- The complete jupyter notebook file where the EDA with SQL process is developped can be found within my GitHub repository in [IBM-Data-Science-Capstone-Project/4_EDA with SQL.ipynb at main · Bouraso/IBM-Data-Science-Capstone-Project (github.com)]

EDA with Data Visualization

- In this step, functions from Matplotlib and Seaborn libraries were used to visualize the data through different kinds of plots (scatterplot, bar-plot, line chart) and gain insights on the relationships between the features of the data.
- The plots are summarized below:
  - ❖ Relationship between Flight Number and Payload Mass with a scatterplot
  - ❖ Relationship between Flight Number and Launch Site with a scatterplot
  - ❖ Relationship between Payload Mass and Launch Site with a scatterplot
  - ❖ Success Rate per Orbit Type with a bar-plot
  - ❖ Relationship between Flight Number and Orbit Type site with a scatterplot
  - ❖ Relationship between Payload Mass and Orbit Type site with a scatterplot
  - ❖ Average Success Rate per Year with a line chart
- The complete jupyter notebook file where the EDA with Data Visualization process is developped can be found within my GitHub repository in IBM-Data-Science-Capstone-Project/5_EDA with Visualization.ipynb at main · Bouraso/IBM-Data-Science-Capstone-Project (github.com)

Build an Interactive Map with Folium

- In this step, functions from the Folium library were used to visualize the data through interactive maps.
- The Folium library is used to:
  - ❖ Mark all launch sites on a map
  - ❖ Mark the succeeded and failed launches for each site on the map
  - ❖ Mark the distances between a launch site to its proximites such as the nearest city, railway or highway

- The complete jupyter notebook file where the Interactive Maps with Folium are developped can be found within my GitHub repository in IBM-Data-Science-Capstone-Project/6_Interactive Visual Analytics with Folium.ipynb at main · Bouraso/IBM-Data-Science-Capstone-Project (github.com)

Build a Dashboard with Plotly Dash

- In this step, functions from the Dash library were used to generate an interactive dashboard where the user can easily and quickly scroll on a ‘Launch Site’ dropdown menu and a ‘Payload Mass’ range slider to further filter and investigate the data.

- Using a pie chart and a scatterplot, the interactive dashboard shows:

  - ❖ The total success launches for each launch site
  - ❖ The total success launches for the launch site with the highest success rate
  - ❖ The correlation between the payload mass and the mission outcome (success or failure) for each launch site with different payload selected in the range slider

- The complete python script where the Dashboard with Plotly Dash is developped can be found within my GitHub repository in IBM-Data-Science-Capstone-Project/7_Build an Interactive Dashboard with Ploty Dash.py at main · Bouraso/IBM-Data-Science-Capstone-Project (github.com)

Predictive Analysis (Classification)

- In this step, functions from the Scikit-learn library are used to create and compare the accuracies of several machine learning models.
- The machine learning prediction analysis includes the following steps:
  - ❖ Standarizing the data
  - ❖ Splitting the data into train & test data
  - ❖ Creating the machine learning models with the following algorithms:
    1. Logistic Regression
    2. Support Vector Machine (SVM)
    3. Decision tree
    4. K-nearest neighbors (KNN)
  - ❖ Fit the models on the training set
  - ❖ Find the best combination of hyperparameters for each model
  - ❖ Evaluate the models based on their accuracy scores and confusion matrix
- The complete jupyter notebook file where the Predictive Analysis is developped can be found within my GitHub repository in IBM-Data-Science-Capstone-Project/8_Machine Learning Prediction.ipynb at main · Bouraso/IBM-Data-Science-Capstone-Project (github.com)

Section 2

Insights drawn from EDA

Results

- The results are split on the upcoming slides into the following Sections:
  - Section2:
    - SQL (EDA with SQL)
    - Visualization: Matplotlib and Seaborn
  - Section3:
    - Folium
  - Section4:
    - Dash
  - Section5:
    - Predictive Analysis

Results SQL: All Launch Site Names

- The names of the unique launch sites in the space mission:

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

- The query uses the 'DISTINCT' function to identify the unique sites:

```
%sql select distinct Launch_Site from SPACEXTABLE
```

Results SQL: Launch Site Names Begin with 'CCA'

- The 5 records where launch sites begin with ‘CCA’ are:

Date	Time (UTC)	Booster_Version	Launch_Site	Payload	PAYLOAD_MASS__KG_	Orbit	Customer	Mission_Outcome	Landing_Outcome
6/4/2010	18:45:00	F9 v1.0 B0003	CCAFS LC-40	Dragon Spacecraft Qualification Unit	0	LEO	SpaceX	Success	Failure (parachute)
12/8/2010	15:43:00	F9 v1.0 B0004	CCAFS LC-40	Dragon demo flight C1, two CubeSats, barrel of Brouere cheese	0	LEO (ISS)	NASA (COTS) NRO	Success	Failure (parachute)
22/05/2012	7:44:00	F9 v1.0 B0005	CCAFS LC-40	Dragon demo flight C2	525	LEO (ISS)	NASA (COTS)	Success	No attempt
10/8/2012	0:35:00	F9 v1.0 B0006	CCAFS LC-40	SpaceX CRS-1	500	LEO (ISS)	NASA (CRS)	Success	No attempt
3/1/2013	15:10:00	F9 v1.0 B0007	CCAFS LC-40	SpaceX CRS-2	677	LEO (ISS)	NASA (CRS)	Success	No attempt

- The query uses the ‘LIKE’ function to specify which characters should the name contain and ‘LIMIT’ to show only the first 5 records:

```
%sql SELECT * FROM SPACEXTABLE WHERE Launch_Site LIKE 'CCA%' LIMIT 5
```

Results SQL: Total Payload Mass

- The total payload mass carried by the boosters launched by NASA (CRS) is:

sum(PAYLOAD_MASS__KG_)
45596

- The query uses the 'SUM' aggregation function to summarize the payload mass and the 'WHERE' to specify the wanted customer:

```
%sql SELECT sum(PAYLOAD_MASS__KG_) FROM SPACEXTABLE where Customer="NASA (CRS)"
```

Results SQL: Average Payload Mass by F9 v1.1

- The average payload mass carried by the booster version F9 v1.1 is:

AVG(PAYLOAD_MASS__KG_)
2928.4

- The query uses the 'AVG' aggregation function to calculate the average of the payload mass and the 'WHERE' to specify the wanted booster version:

```
%sql SELECT AVG(PAYLOAD_MASS__KG_) FROM SPACEXTABLE where Booster_Version="F9 v1.1"
```

Results SQL: First Successful Ground Landing Date

- The first successful ground landing date is:

min(Date)
1/8/2018

- The query uses the 'MIN' aggregation function to return the first date and the 'WHERE' to specify the wanted landing outcome:

```
%sql select min(Date) from SPACEXTABLE where Landing_Outcome="Success (ground pad)"
```

Results SQL: Successful Drone Ship Landing with Payload between 4000 and 6000

- The names of the boosters that have a successful drone ship landing and a payload mass between 4000 and 6000kg are:

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

- The query uses the 'DISTINCT' function to identify the unique boosters, the 'WHERE' to specify the wanted landing outcome and the 'AND / BETWEEN' to limit the payload mass:

```
%sql select distinct Booster_Version from SPACEXTABLE where Landing_Outcome="Success (drone ship)" and PAYLOAD_MASS__KG_ between 4000 and 6000
```

Results SQL: Total Number of Successful and Failure Mission Outcomes

- The total number of successful and failure missions outcomes is:

number_of_success_outcomes	number_of_failure_outcomes
100	1

- The query uses the ‘COUNT’ aggregation function to count the rows and the ‘GROUP BY’ function to group the data based on the specified feature:

```
%sql select Mission_Outcome, count(*) from SPACEXTABLE GROUP BY Mission_Outcome
```

Results SQL: Boosters Carried Maximum Payload

- The names of the boosters that carried the maximum payload mass are:

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

- The query uses the 'MAX' aggregation function to find the maximum payload mass within a subquery:

```
%sql select Booster_Version from SPACEXTABLE where PAYLOAD_MASS__KG_ = (SELECT MAX(PAYLOAD_MASS__KG_) FROM SPACEXTABLE)
```

Results SQL: 2015 Launch Records

- The failed landing outcomes in drone ship, their booster version and the launch site are:

DATE	booster_version	launch_site
2015-01-10	F9 v1.1 B1012	CCAFS LC-40
2015-04-14	F9 v1.1 B1015	CCAFS LC-40

- The query uses the ‘FORMAT’ function to change the date format to month and year-specific and the ‘WHERE’ function for further filtering:

```
%sql select FORMAT(Date, 'MMMM') AS month_name,Landing_Outcome,Booster_Version, Launch_Site FROM SPACEXTABLE WHERE year(Date)=2015 AND Landing_Outcome = 'Failure (drone ship)'
```

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

- The ranking of the landing outcomes for the specified date period is:

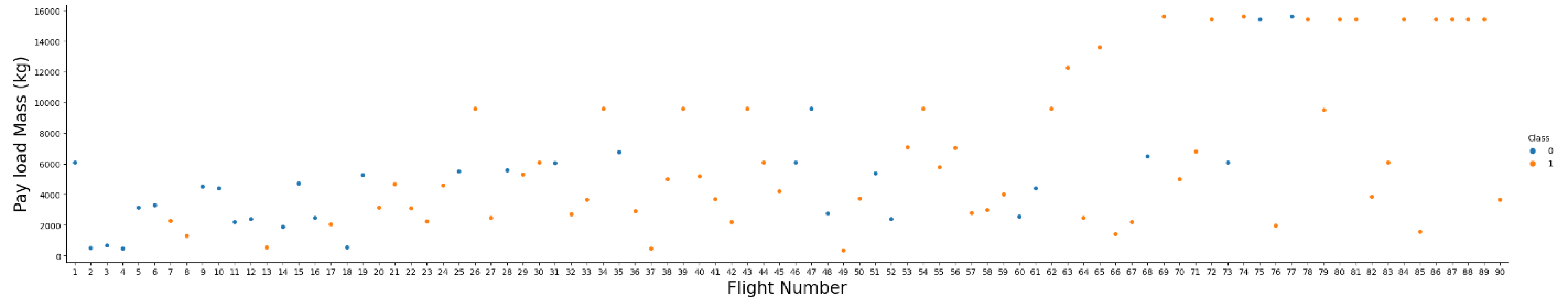
Landing_Outcome	count
Success	15
No attempt	6
Success (ground pad)	5
Success (drone ship)	5
Failure (drone ship)	4
Failure	3
Controlled (ocean)	3
Failure (parachute)	1

- The query uses is counting the existent rows per landing outcome after filtering on the specified dates:

```
%sql select Landing_Outcome, count(*) as count from SPACEXTABLE where Date >= '04/06/2010' AND Date <= '20/03/2017' group by Landing_Outcome order by count desc
```

Visualization: Flight Number vs. Payload

- The relationship between the Flight Number and the Payload Mass is displayed below:

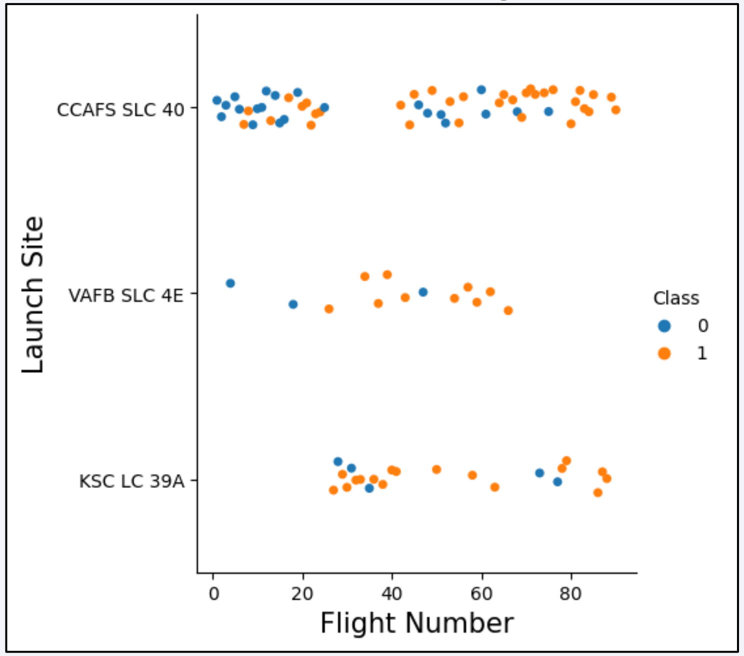



```
sns.catplot(y="PayloadMass", x="FlightNumber", hue="Class", data=df, aspect = 5)
plt.xlabel("Flight Number",fontsize=20)
plt.ylabel("Pay load Mass (kg)",fontsize=20)
plt.show()
```

- Insights:
  - ✓ As the Flight Number increases, it is more likely to have a successful landing

Visualization: Flight Number vs. Launch Site

- The relationship between the Flight Number and the Launch Site is displayed below:



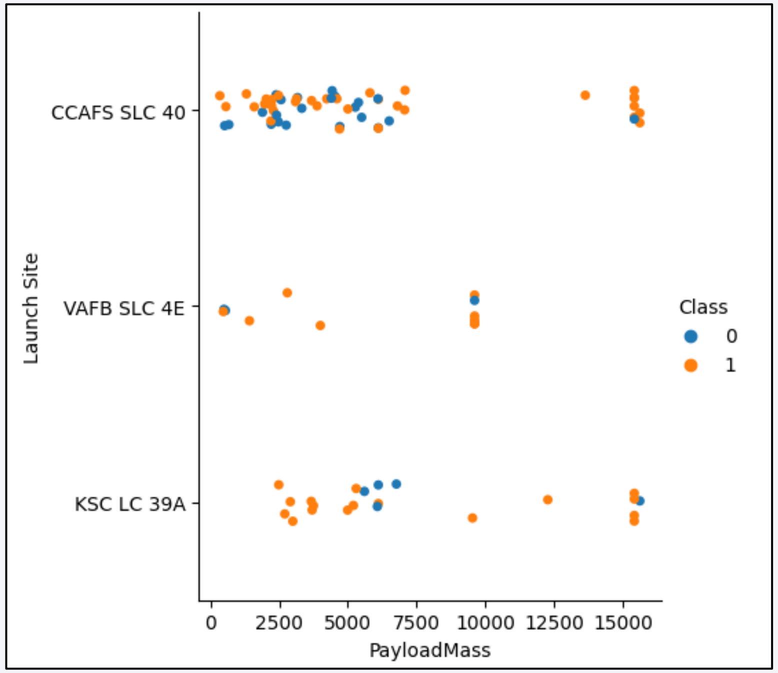

```
sns.catplot(y="LaunchSite",x="FlightNumber",hue="Class", data=df, aspect = 1)
plt.ylabel("Launch Site",fontsize=15)
plt.xlabel("Flight Number",fontsize=15)
plt.show()
```

- Insights:

  ✓Each Launch Site has a different Success Rate

Visualization: Payload vs. Launch Site

- The relationship between the Payload Mass and the Launch Site is displayed below:



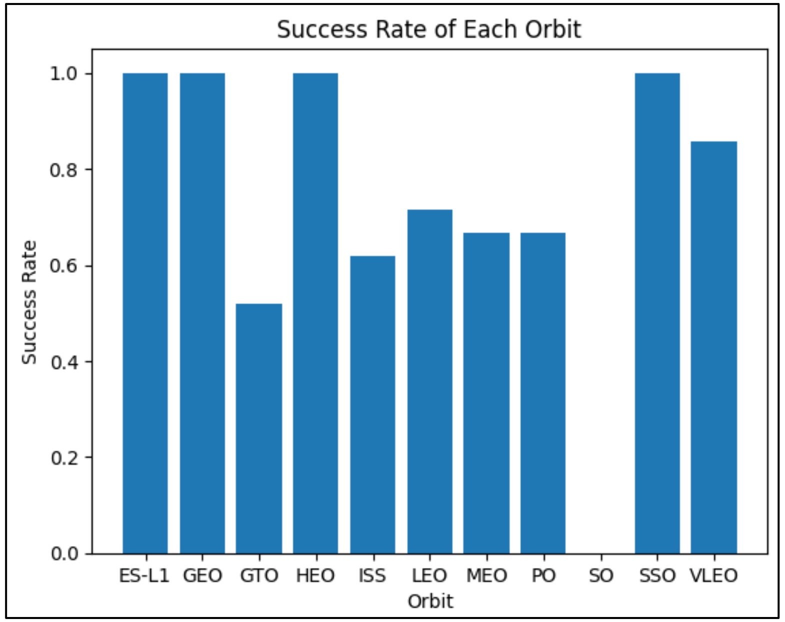

```
sns.catplot(data=df, x='PayloadMass', y='LaunchSite', hue='Class')
plt.xlabel('PayloadMass')
plt.ylabel('Launch Site')
```

- Insights:

  ✓The VAFB SLC site has no launches for heavy payload mass (>10000kg)

Visualization: Success Rate vs. Orbit Type

- The Success Rate per Orbit Type is displayed below:

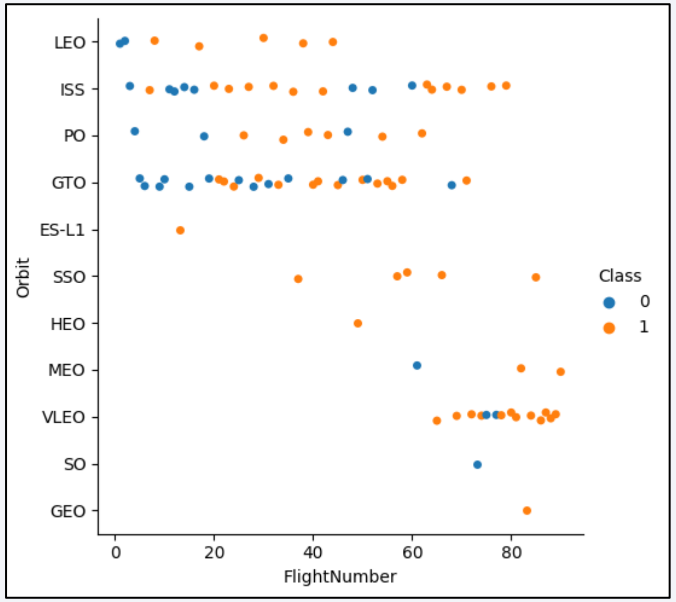



```python
orbit_success_rate = df.groupby('Orbit')['Class'].mean()
plt.bar(orbit_success_rate.index, orbit_success_rate.values)
plt.xlabel('Orbit')
plt.ylabel('Success Rate')
plt.title('Success Rate of Each Orbit')
plt.show()
```

- Insights:
  - ✓The Orbit GTO has the lowest success rate
  - ✓Several Orbits have 100% success rate (ES-L1, GEO, HEO, SS)

Visualization: Flight Number vs. Orbit Type

- The relationship between the Flight Number and the Orbit Type is displayed below:

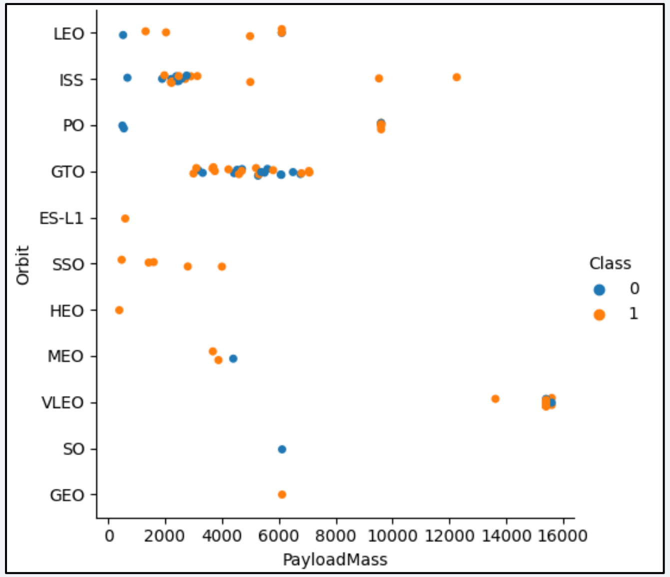



```
sns.catplot(data=df, x='FlightNumber', y='Orbit', hue='Class')
plt.xlabel('FlightNumber')
plt.ylabel('Orbit')
```

- Insights:
  - ✓The LEO Orbit has a success rate related to the number of flights
  - ✓For the GTO Orbit, no relationship was detected

Visualization: Payload vs. Orbit Type

- The relationship between the Orbit Type and the Payload Mass is displayed below:

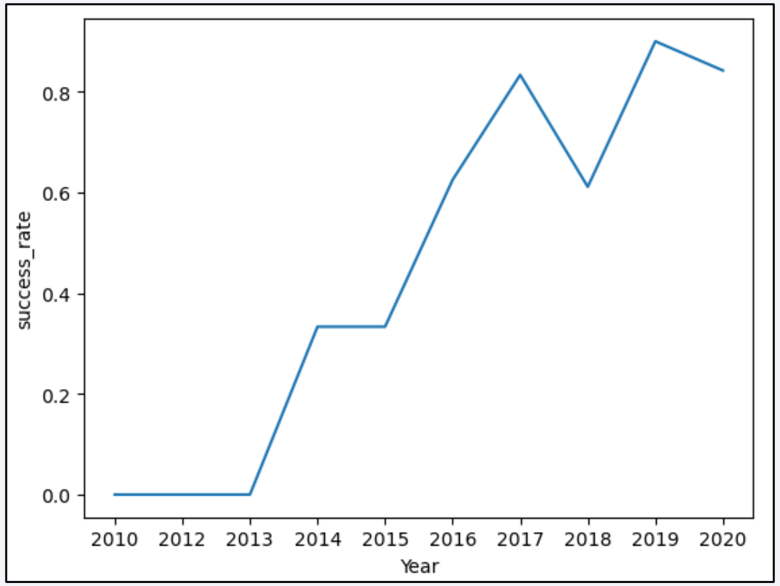



```
sns.catplot(data=df, x='PayloadMass', y='Orbit', hue='Class')
plt.xlabel('PayloadMass')
plt.ylabel('Orbit')
```

- Insights:
  - ✓With heavy payloads ISS and PO present a high success rate
  - ✓No relationship detected for the GTO

Visualization: Launch Success Yearly Trend

- The Success Rate per Year is displayed below:




```
success_rate = df.groupby('Date')['Class'].mean()
plt.plot(success_rate.index, success_rate)
plt.xlabel('Year')
plt.ylabel('success_rate')
```

- Insights:

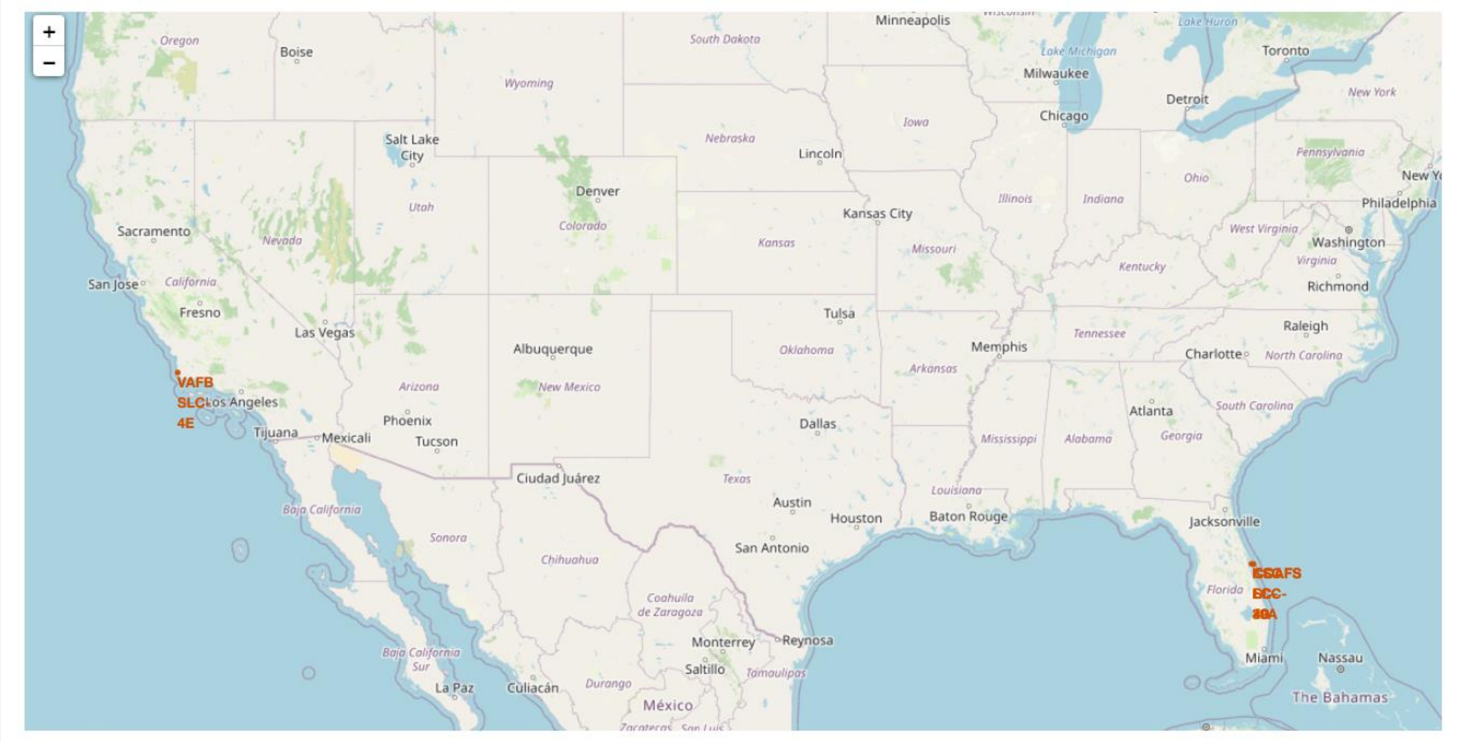
  ✓The success rate since 2013 kept increasing with a small dr in 2018

Section 3

Launch Sites Proximities Analysis

Folium: Launch Sites on Map

- All the Launch Sites are depicted with a Circle and a Marker on the map below:

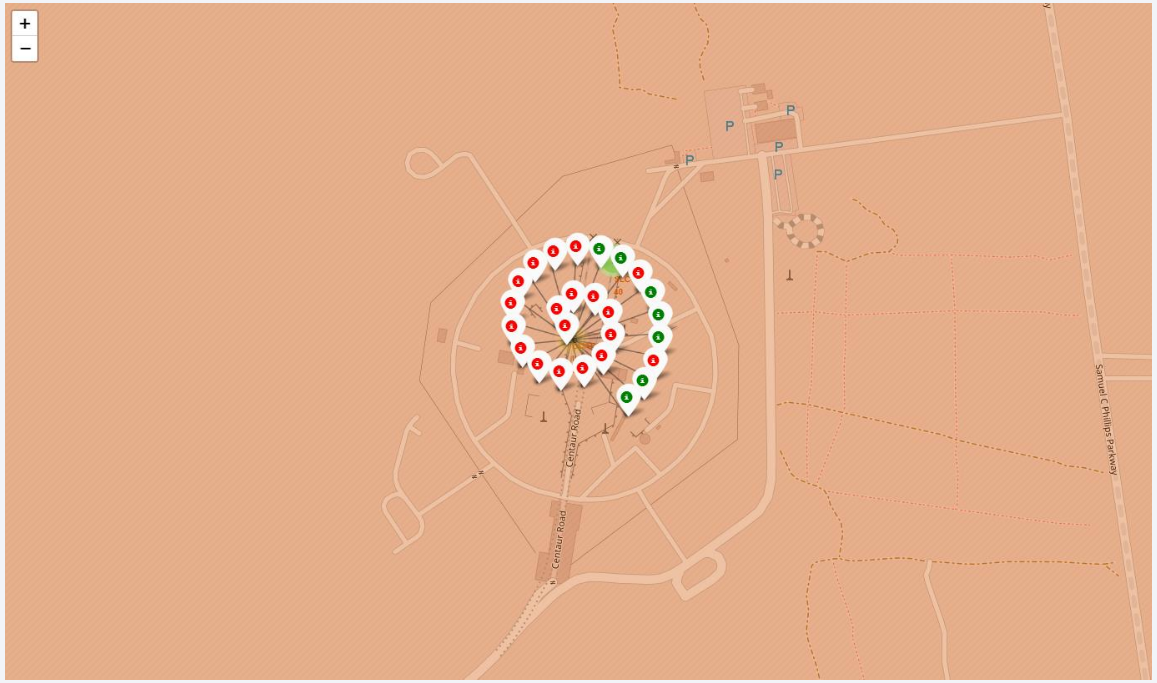


- Insights:

  ✓All launch sites seem to be in close proximity to the coast

Folium: Succeeded & Failed Launches on Site

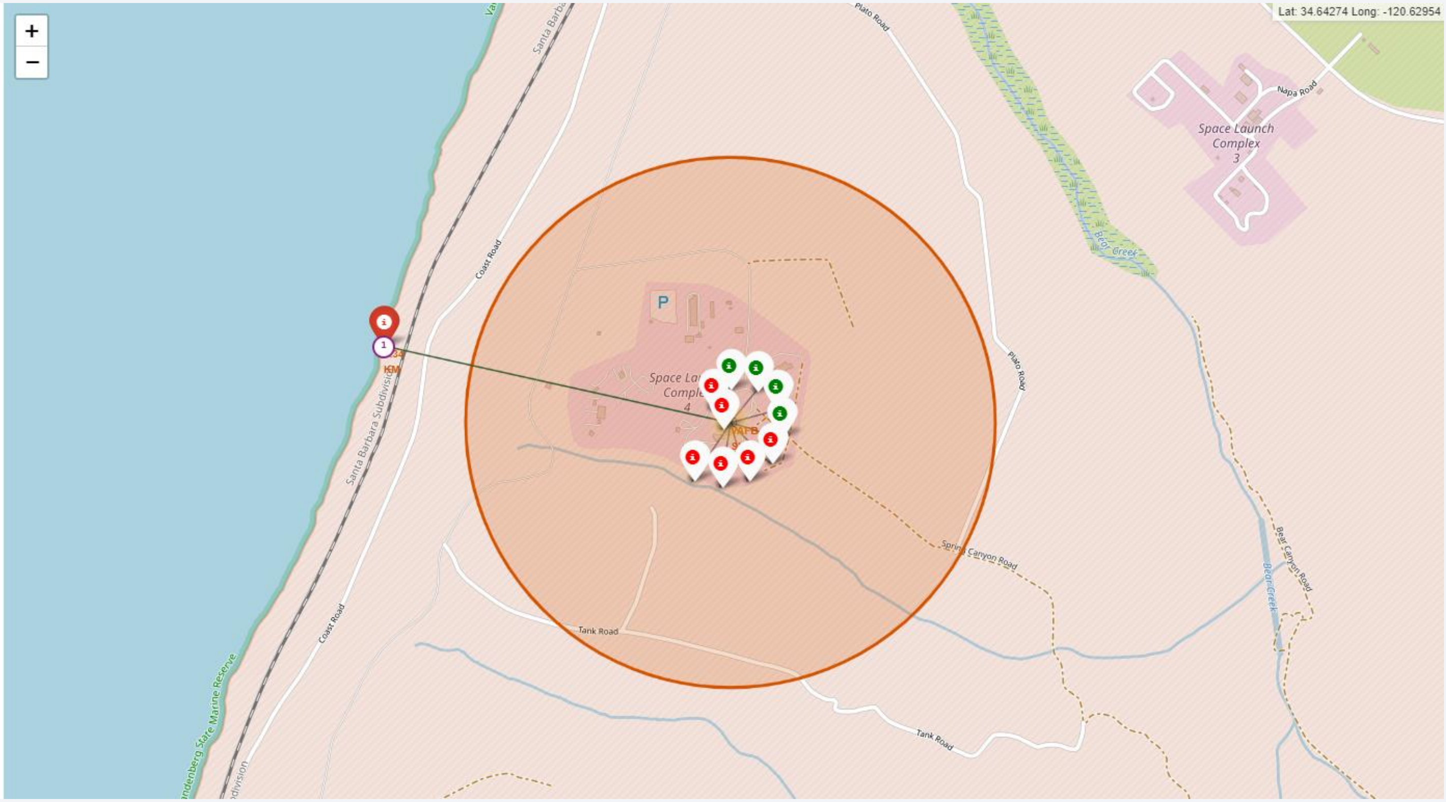
- The succeeded and failed Launches for each site on Map → zooming in one of the launch sites, the green (successful launch) and the red (failed launch) tags can be viewed:



- Insights:
  - ✓By zooming on each launch site, the success rate of each site can be easily estimated from the colored tags

Folium: Distance between Site and its proximities

- The distances between a Launch Site to its proximities such as the nearest city, railway or highway → the picture below shoes the distance between the VAFB SLC-4E launch site and the nearest coastline (1.34km) :

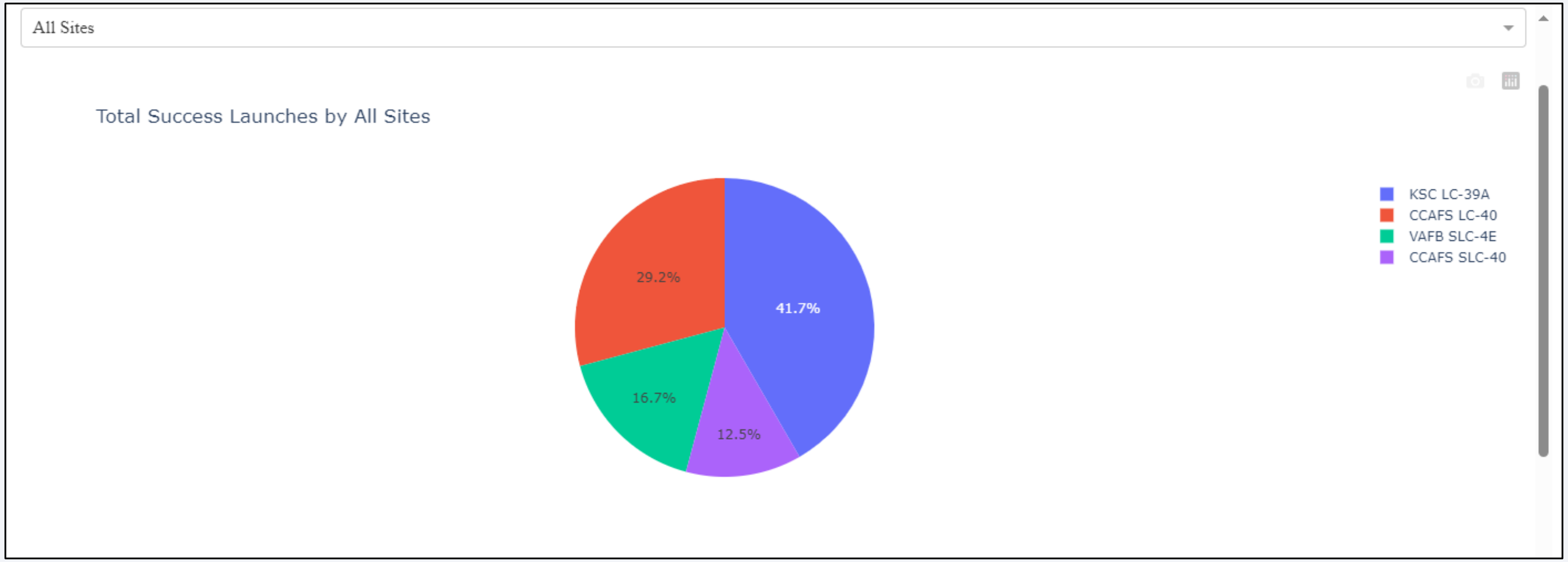


Section 4

Build a Dashboard with Plotly Dash

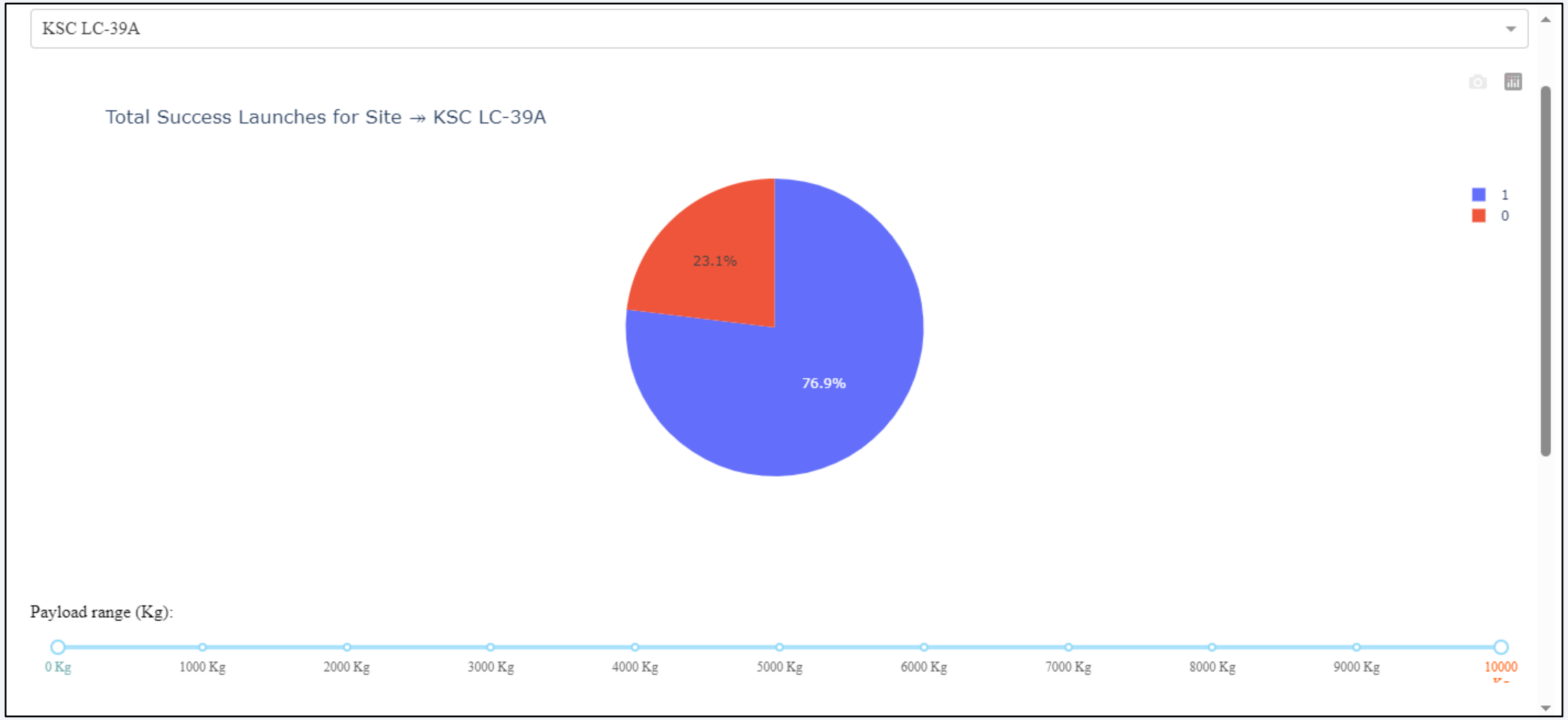
Dash: Total Success Launches by All Sites

- The picture below represents the upper part of the dashboard where a pie chart displays the total success launches as a percentage for each Launch Site
- The user has the possibility to select a single Launch Site with the dropdown arrow:



Dash: Launch Site with the highest Success Rate

- The picture below represents the total Success Launches for the Site KSC LC-39A which is the site with the highest Success Rate based on the pie chart of the previous slide:



Dash: Payload vs. Launch Outcome for All Sites

- The two pictures below show the Payload Mass vs. Outcome scatterplot for all the Sites for two different Payload Mass ranges:

<u>2000 to 6000kg</u>

<u>6000 to 10000kg</u>

✓The FT booster seems to have the highest success rate while the v1.1 has the lowest one

✓Significantly lower number of launches with a heavier payload

✓Significantly higher percentage of failed launches for the range of 6000-10000kg

Section 5

Predictive Analysis (Classification)

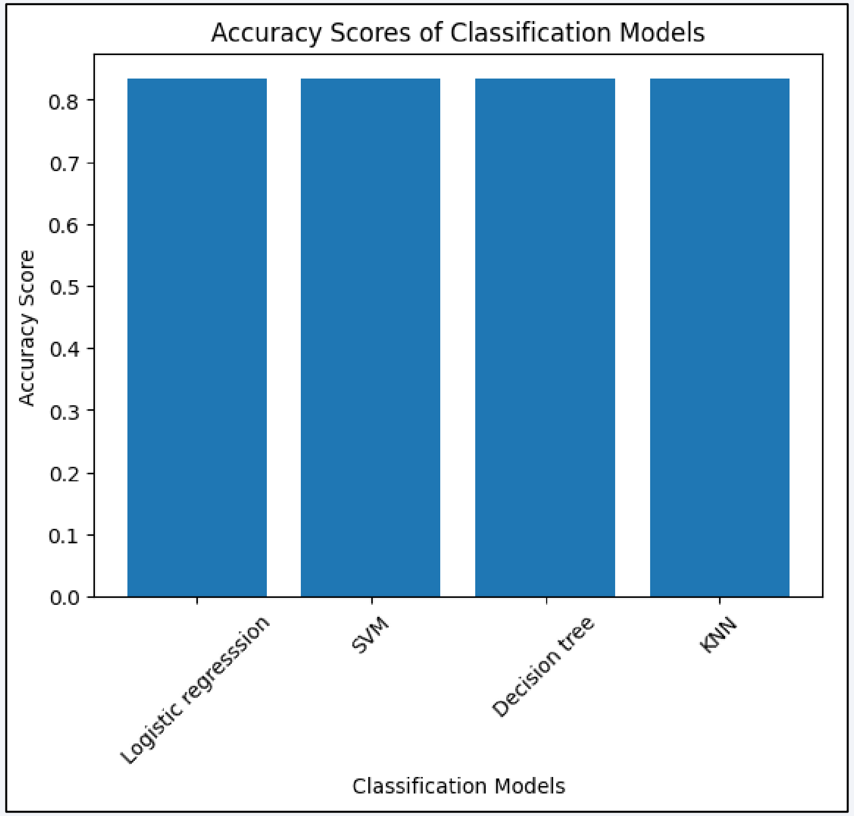
Classification Accuracy

- On the left bar-plot, the Accuracy for each one of the 4 machines learning algorithms is displayed.
- The Accuracy was calculated based on the method “score” on the test dataset.

✓The Accuracy is equal for all the 4 algorithms.

✓Therefore, the GridSearchCV best scored are compared to find the best performing algorithm. (table at the bottom)

✓The Decision Tree algorithm has the highest best score



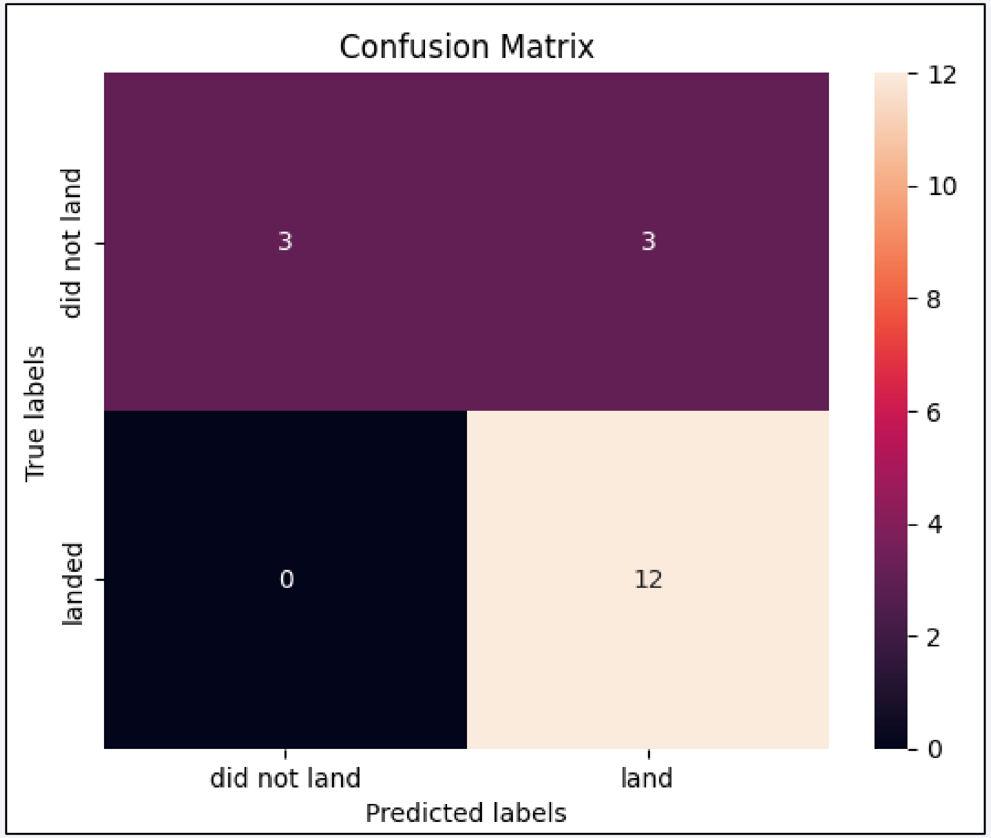
	Best scores
Logistic regresssion	0.846429
SVM	0.848214
Decision tree	0.862500
KNN	0.848214

Confusion Matrix

- On the left visual, the Confusion Matrix for the decision tree model is displayed.

✓Examining the Confusion Matrix, it seems that the specific model can fairly distinguish the different classes

✓The major problem on the prediction comes from the false positives (top-right square=3), meaning that the model predicts a successful landing while the true outcome was a failure



Conclusions

- The main goal of this capstone assignment was to predict if the first stage of a given Falcon 9 launch will land successfully.
- A dedicated dataset was thoroughly processed and analyzed through data collection, wrangling, formatting and visualization in order to identify possible patterns and correlations between the data's features
- Several features like Payload Mass and Orbit Type can strongly affect the launching outcome in a certain way and depending the Launch Site.
- Important to note that the Total Success rate is increasing from 2013 onwards.
- Four classification machine learning algorithms were employed and trained to be able to predict the Outcome based on the historical Flight data.
- The predictive model produced by the Decision Tree algorithm performed the best. The major problem on that model comes from the false positive predictions

Appendix

Thank you!