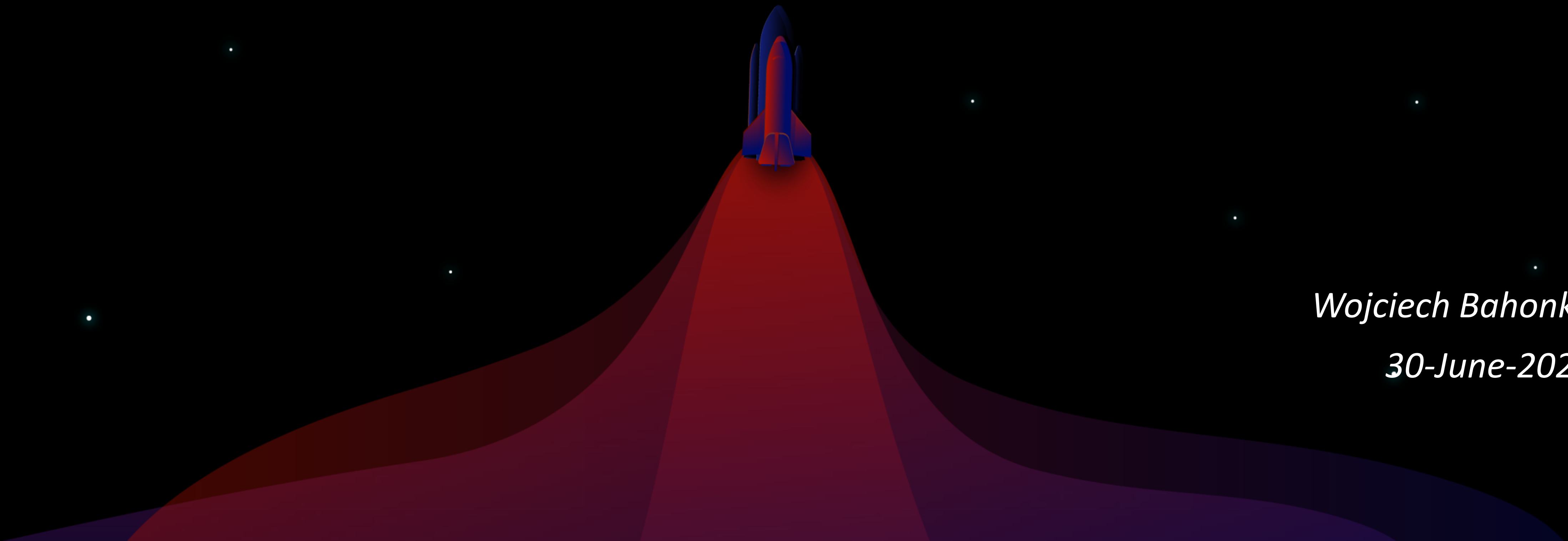
Applied Data Science Capstone



Wojciech Bahonko

30-June-2024

OUTLINE

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

EXECUTIVE SUMMARY

- **In this capstone, we will predict if the Falcon 9 first stage will land successfully using couple of methodologies below**

Main methodologies

- Data Collection thru API and using Web Scraping
- Data Wrangling
- Exploratory Data Analysis with SQL and with Data Visualization
- Interactive Visual Analytics with Folium and with Ploty Dash
- Machine Learning Prediction

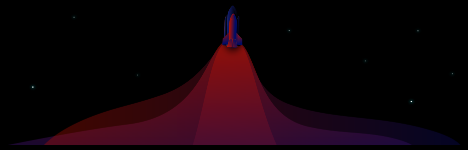
In the presentation presents the following results

- Exploratory Data Analysis results
- Interactive analytics demo in screenshots
- Predictive Analytics results

INTRODUCTION

Background and Context

- The commercial space age is here, companiesare making space travel affordable for everyone.
- Perhaps the most successful is SpaceX.
- One reason SpaceX can do this is the rocket launches are relatively inexpensive.
- 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 and also determine if SpaceX will reusethe first stage.
- This information can be used if an alternate company wants to bid against SpaceX for a rocket launch



INTRODUCTION

Main Question

that we are trying to answer is, for a given set of features about a **Falcon 9 rocket** launch which include its payload mass, orbit type, launch site etc. will the **first stage** of the rocket **land successfully**?

METHODOLOGY

- Data Collection using SpaceX API
- Data Collection using Web Scraping from Wikipedia
- Data Wrangling
- Exploratory Data Analysis (EDA) with SQL
- Exploratory Data Analysis (EDA) with Visualization
- Interactive Visual Analytics with Folium
- Interactive Dashboard with Ploty Dash
- Predictive Analysis – using Machine Learning models.

METHODOLOGY
Data Collection with API

Main Steps performing data collection using SpaceX API:

- start requesting rocket launch data from SpaceX API
- decode the response content as a Json using .json() and turn it into a Pandas dataframe using .json_normalize()
- use the API again to get information about the launches using the IDs given for each launch.
- construct our dataset using the data we have obtained and combine the columns into a dictionary
- create a Pandas data frame from the dictionary.
- filter the dataframe to only include `Falcon 9` launches.
- deal with missing Values
- save results to .csv file

GitHub URL of Jupyter notebook: SpacexDataCollectionAPI

METHODOLOGY
Data Collection Web Scraping

Main Steps of performing web scraping with BeautifulSoup to collect Falcon 9 historical launch records from a Wikipedia page

- request the Falcon9 Launch Wiki page from its URL
- create a `BeautifulSoup` object from the HTML `response`
- extract all column/variable names from the HTML table header
- create an empty dictionary with keys from the extracted column names
- parsing launch of HTML tables
- create dataframe from dictrionary
- export results to .csv file

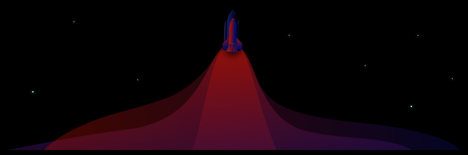
GitHub URL of Jupyter notebook: [WebScraping]

METHODOLOGY
Data Wrangling

Main Steps of Data Wrangling

- calculate the number of launches on each site
    - each launch aims to an dedicated orbit, and here are some common orbit types
- calculate the number and occurrence of each orbit
- calculate the number and occurence of mission outcome of the orbits
- create a landing outcome label from Outcome column
- Export results to .csv file

GitHub URL of Jupyter notebook: DataWrangling



METHODOLOGY
EDA with SQL

Performed the following SQL queries

- display the names of the unique launch sites in the space mission.
- display five 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 succesful 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.
- list the records which will display the month names, failure landing_outcomes in drone ship ,booster versions, launch_site for the months 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.

GitHub URL of Jupyter notebook: ExploratoryDataAnlysisSQL

METHODOLOGY
EDA with Visualization

Using data from API and Wiki – we perform futher exploratory data analysis to visualize

- the relationship between Flight Number and Launch Site
- the relationship between Payload and Launch Site
- relationship between success rate of each orbit type
- the relationship between FlightNumber and Orbit type
- the launch success yearly trend

GitHub URL of Jupyter notebook: [ExploratoryDataAnalysisVisualization]()

METHODOLOGY
Folium

In this part discovered many interesting insights related to the launch sites' location using folium, in a very interactive way.

- first mark the launch site locations and their close proximities on an interactive map.
- then, we can explore the map with those markers and try to discover any patterns from them.
- finally, we should be able to explain how to choose an optimal launch site.

GitHub URL of Jupyter notebook: [Folium]

METHODOLOGY
Ploty Dash

In this part visual part build a dashboard using Ploty Dash on detailed launch records.

Using the dashboard we want to answer the following questions:

- Which site has the largest successful launches?
- Which site has the highest launch success rate?
- Which payload range(s) has the highest launch success rate?
- Which payload range(s) has the lowest launch success rate?
- Which F9 Booster version (v1.0, v1.1, FT, B4, B5, etc.) has the highest
- launch success rate?

GitHub URL of Jupyter notebook: PlotyDash

METHODOLOGY
Predictive Analysis

We will build a machine learning pipeline to predict if the first stage of the Falcon 9 lands successfully.

This will include:

- Preprocessing, allowing usto standardize our data, and
- Train_test_split, allowing us to split our data into training and testing data,
- we will train the model and perform Grid Search, allowing us to find the hyperparameters that allow a given algorithm to perform best.
- using the best hyperparameter values, we will determine the model with the best accuracy using the training data.
- will test Logistic Regression,Support Vector machines, Decision Tree Classifier, and K-nearest neighbors.
- we will output the confusion matrix.

GitHub URL of Jupyter notebook: [MachineLearningPrediction]

RESULTS

All Launch Site Names

There are **four unique Launch Sites** used by SpaceX for Falcon 9 rocket

```
[11]: %sql select distinct launch_site from spacextable
```

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

[11]:

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

RESULTS

Launch Sites Begin 'CCA'

We found at **least five launch sites** begin with the string '**CCA**'

[12]:
```
%sql select * from spacextable where launch_site like 'CCA%' limit 5
```

* sqlite:///my_data1.db
Done.

[12]:

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

RESULTS

Total Payload By NASA(CRS)

```
[13]: %sql select sum(payload_mass__kg_) from spacextable where customer = 'NASA (CRS)'

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

[13]:

sum(payload_mass__kg_)
45596

Total payload mass carried by boosters launched by **NASA (CRS)** is **45 596 kg**

RESULTS

Average Payload By F9 v1.1

```
[14]: %sql select avg(payload_mass__kg_) as average from SPACEXTBL where booster_version like 'F9 v1.1%'

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

[14]:

average
2534.6666666666665

Average payload mass carried by booster version **F9 v1.1** is **2 534 kg**

RESULTS

First Successful Ground Landing Date

```
[17]: %sql select min(date) as date from SPACEXTBL where mission_outcome like 'Success'

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

[17]:

date
2010-06-04

Date when the **first succesful landing** outcome in **ground pad** was achieved is **4th of June 2010**

RESULTS

Successful Drone Ship Landing

```
[22]: %sql select booster_version from spacextable where (mission_outcome like 'Success') and (payload_mass__kg_ between 4000 and 6000) and (landing_outcome like 'Success (drone ship)')
 * sqlite:///my_data1.db
Done.
```

[22]:

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

F9 FT booster versions have **success** in **drone ship** and have **payload mass greater** than **4000 but less** than **6000 kg**

RESULTS

Total Numbers Of Mission Outcomes

```
[23]: %sql select mission_outcome, count(*) as count from spacextable group by mission_outcome order by mission_outcome
```

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

[23]:

Mission_Outcome	count
Failure (in flight)	1
Success	98
Success	1
Success (payload status unclear)	1

Successful mission outcome was **100** while **failure** just **1**

RESULTS

Boosters Carried Maximum Payload Mass

```
[24]: %sql select booster_version from spacextable where payload_mass__kg_ = (select max(payload_mass__kg_) from SPACEXTBL)
 * sqlite:///my_data1.db
Done.
```

[24]:

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

F9 B5 booster versions have carried the **maxium payload mass**

RESULTS

2015 Launch Sites

```
[26]: %sql select substr(date,6,2) as month, landing_outcome, booster_version, launch_site from spacextable where substr(date,0,5) ='2015' and landing_outcome like 'Failure (drone ship)'

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

month_name	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

In 2015 failure landing_outcomes in drone ship was happened in **January** and **April** by **F9 v1.1** booster **in CCAFS LC 40** launch site

RESULTS

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

```
[27]: %sql select landing_outcome, count(*) as count from spacextable where date >= '2010-06-04' and date <= '2017-03-20' group by landing_outcome order by 2 desc
 * sqlite:///my_data1.db
Done.
```

[27]:

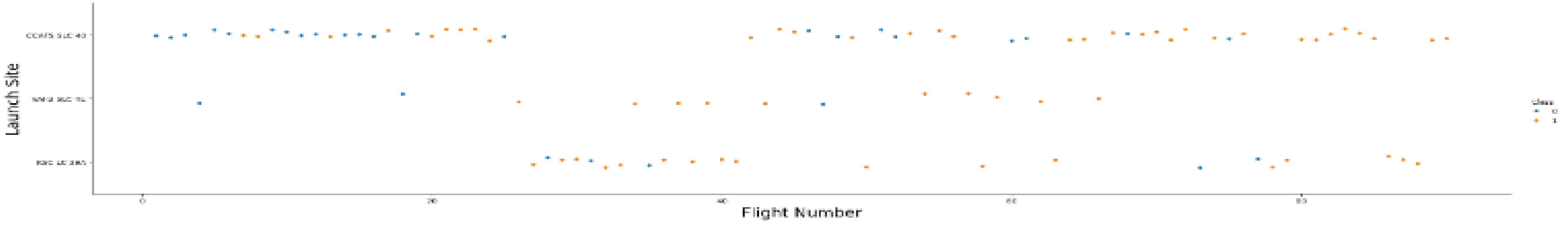
Landing_Outcome	count
No attempt	10
Success (drone ship)	5
Failure (drone ship)	5
Success (ground pad)	3
Controlled (ocean)	3
Uncontrolled (ocean)	2
Failure (parachute)	2
Precluded (drone ship)	1

In selected period – the **SpaceX** had the **same successful as failure landing outcomes** in drone ship but **the most not taken landings with total 10.**

RESULTS

Flight Number vs Launch Sites

[6]:
```
### TASK 1: Visualize the relationship between Flight Number and Launch Site
sns.catplot(x='FlightNumber', y='LaunchSite', hue='Class', data=df, aspect=5)
plt.xlabel('Flight Number', fontsize=20)
plt.ylabel('Launch Site', fontsize=20)
plt.show()
```



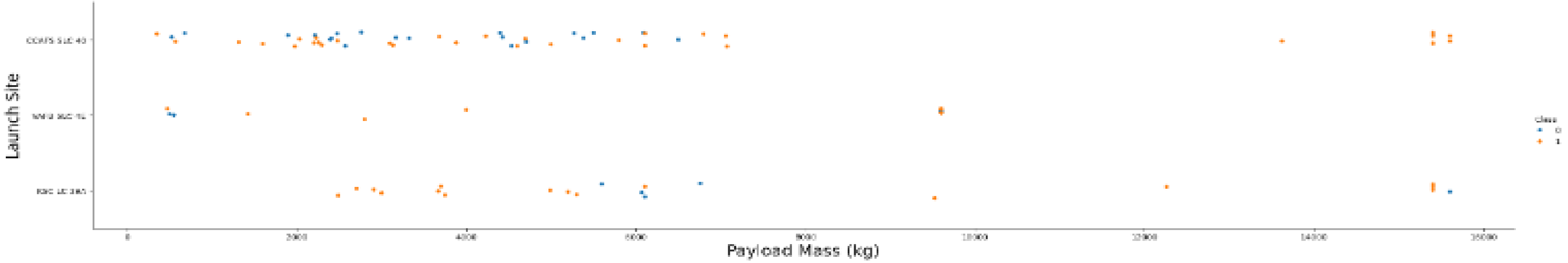
The CCAFS SLC 40 has the highest flight number whilst VAFB SLC 40 has the fewest flight number

RESULTS

Payload Mass vs Launch Sites

[7]:

```
### TASK 2: Visualize the relationship between Payload and Launch Site
sns.catplot(x='PayloadMass', y='LaunchSite', hue='Class', data=df, aspect = 5)
plt.xlabel('Payload Mass (kg)',fontsize=20)
plt.ylabel('Launch Site',fontsize=20)
plt.show()
```



For every launch site higher payload mass, the higher success rate.
More of the launches with payload mass greater than 7 000 kg were successful.

RESULTS

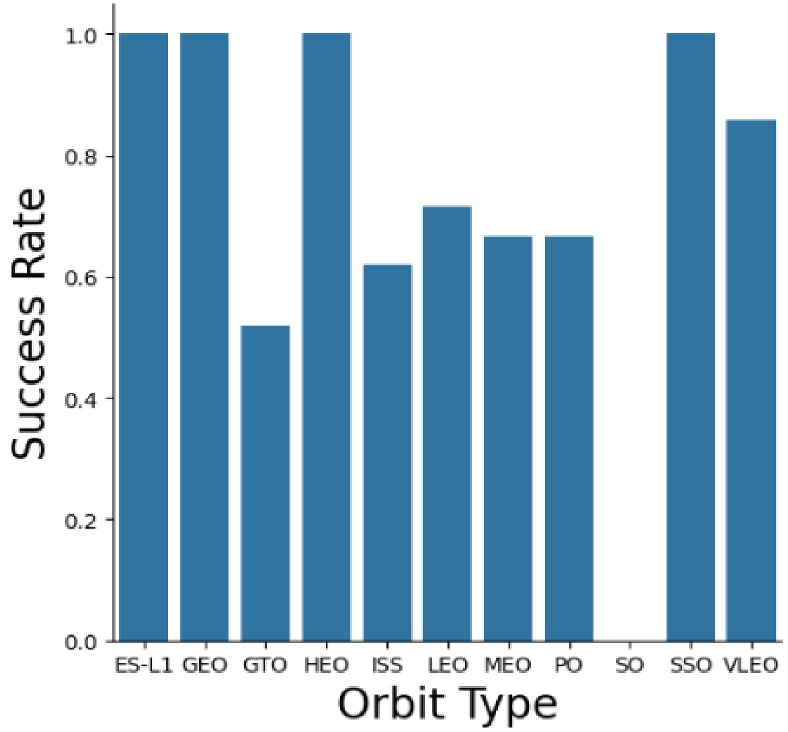
Success Rate vs Orbit Type

There are **four orbits** with **100% success rate:**

- **ES-L1**
- **GEO**
- **HEO**
- **SSO**

But only **SO orbit** is with **0% success rate**

```
[8]: ### TASK  3: Visualize the relationship between success rate of each orbit type
sns.catplot(x= 'Orbit', y = 'Class', data = df.groupby('Orbit')['Class'].mean().reset_index(), kind = 'bar')
plt.xlabel('Orbit Type',fontsize=20)
plt.ylabel('Success Rate',fontsize=20)
plt.show()
```

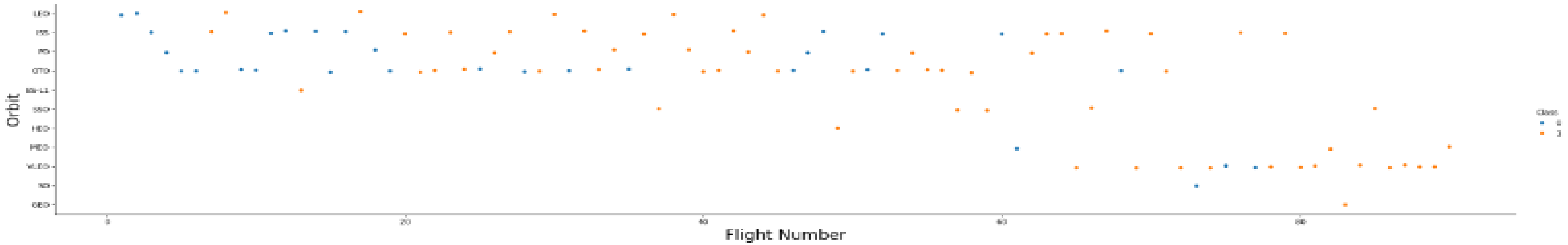


RESULTS

Flight Number vs Orbit Type

[9]:

```python
### TASK  4: Visualize the relationship between FlightNumber and Orbit type
sns.catplot(x = 'FlightNumber', y = 'Orbit', hue = 'Class', data = df, aspect = 5)
plt.xlabel('Flight Number', fontsize = 20)
plt.ylabel('Orbit', fontsize = 20)
plt.show()
```

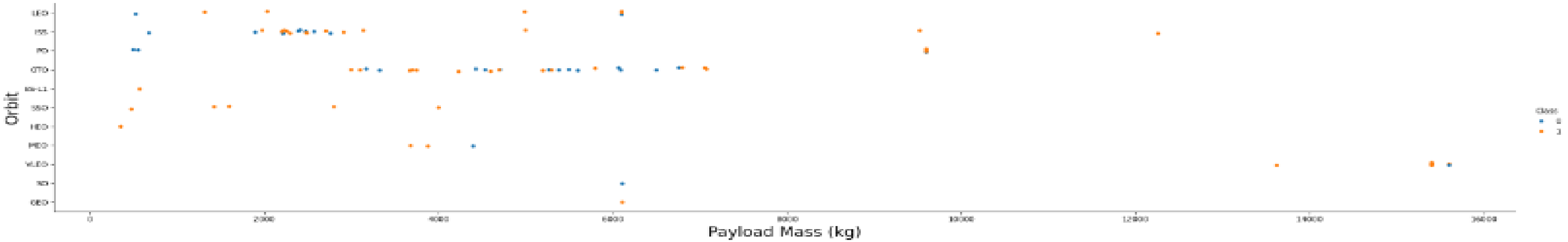


Success rate have **improved** for **mostly orbits** where **more flights** were happened – especially for **LEO** and **VLEO** orbits whilst **GEO** orbit **no relationship** between **success rate** and **flight number**

RESULTS

Payload Mass vs Orbit Type

[10]:
```python
### TASK  5: Visualize the relationship between Payload and Orbit type
sns.catplot(x = 'PayloadMass', y = 'Orbit', hue = 'Class', data = df, aspect = 5)
plt.xlabel('Payload Mass (kg)', fontsize = 20)
plt.ylabel('Orbit', fontsize = 20)
plt.show()
```

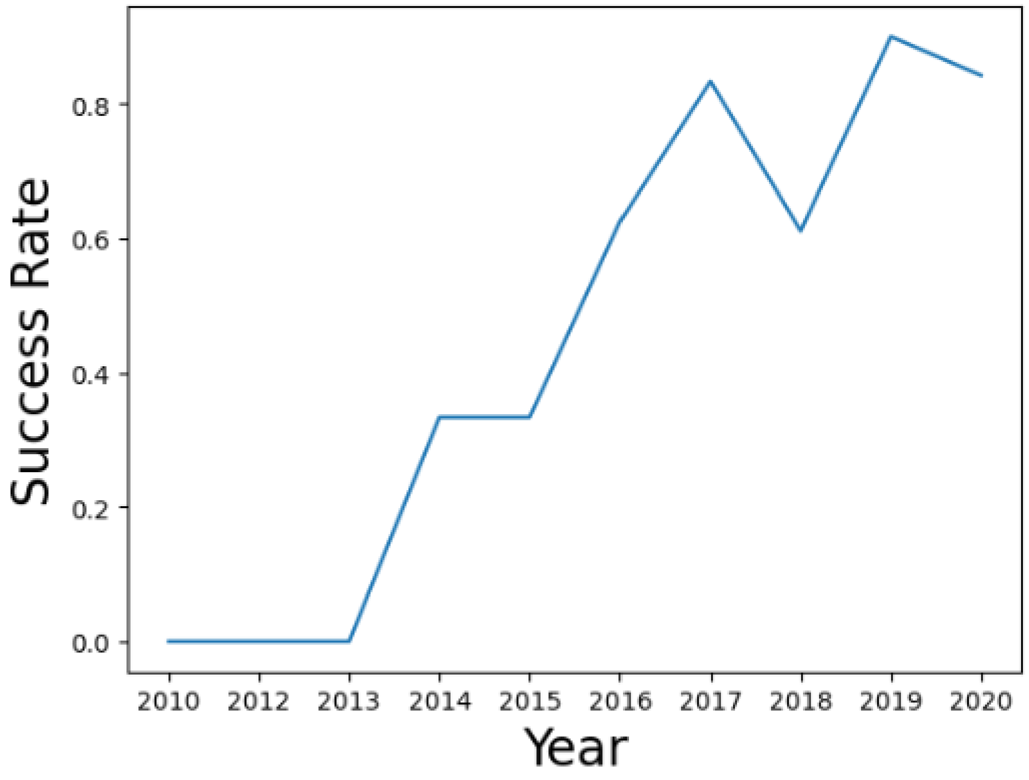


Success rate have **improved** for **orbit LEO, ISS and PO** with **greater payload mass** but **negative impact** on **SSO** and **HEO orbits**.

RESULTS

Launch Sites Yearly Trend

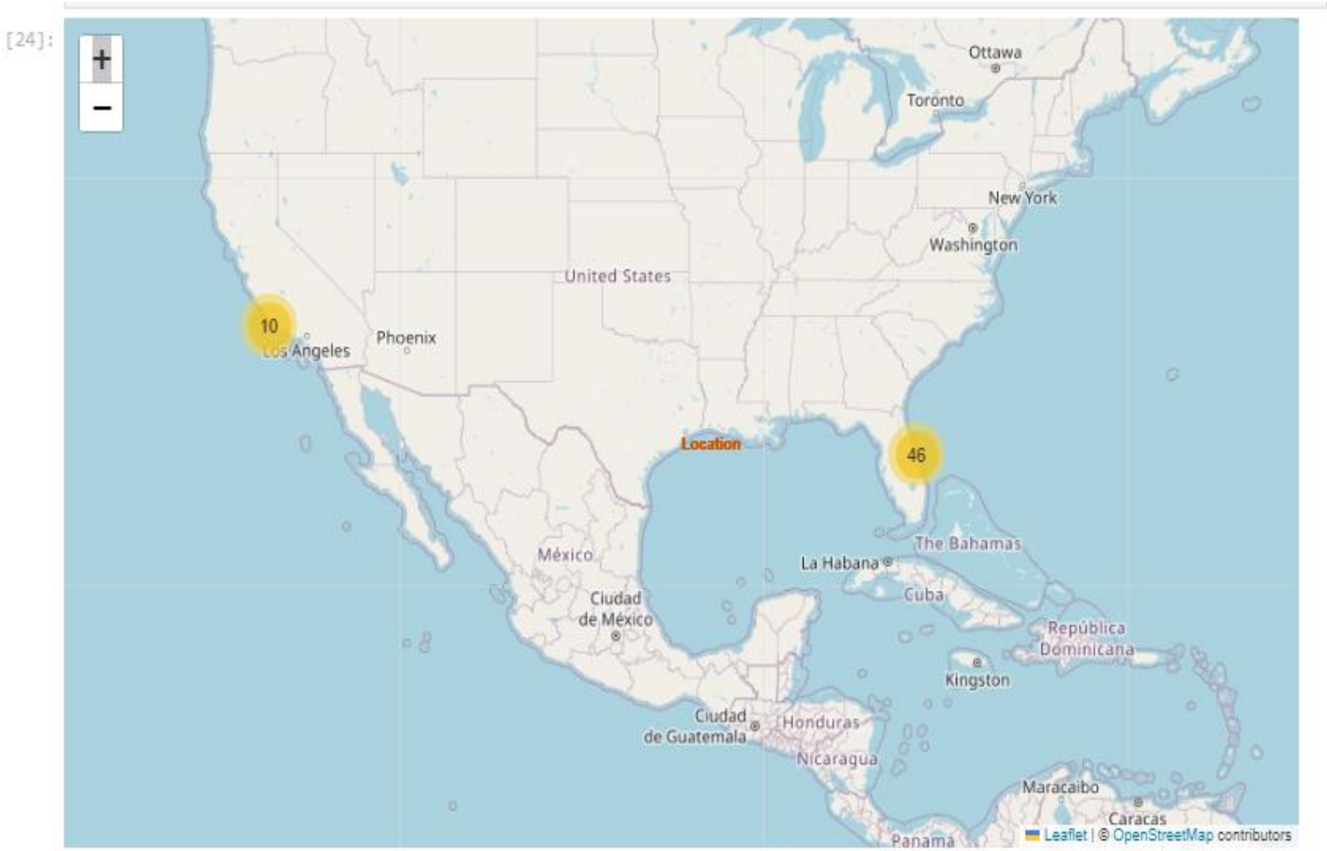
The **Launch Success Yearly Trend** was **positive over the years** and **increased from 2013 till 2020.**



RESULTS

SpaceX Launch Sites Locations

- **all sites locations** are inside the **United States**
- **all sites** are located **near to coastline** – west and east coast
- **all sites** locations are in **south located as possible as in equator line**



RESULTS

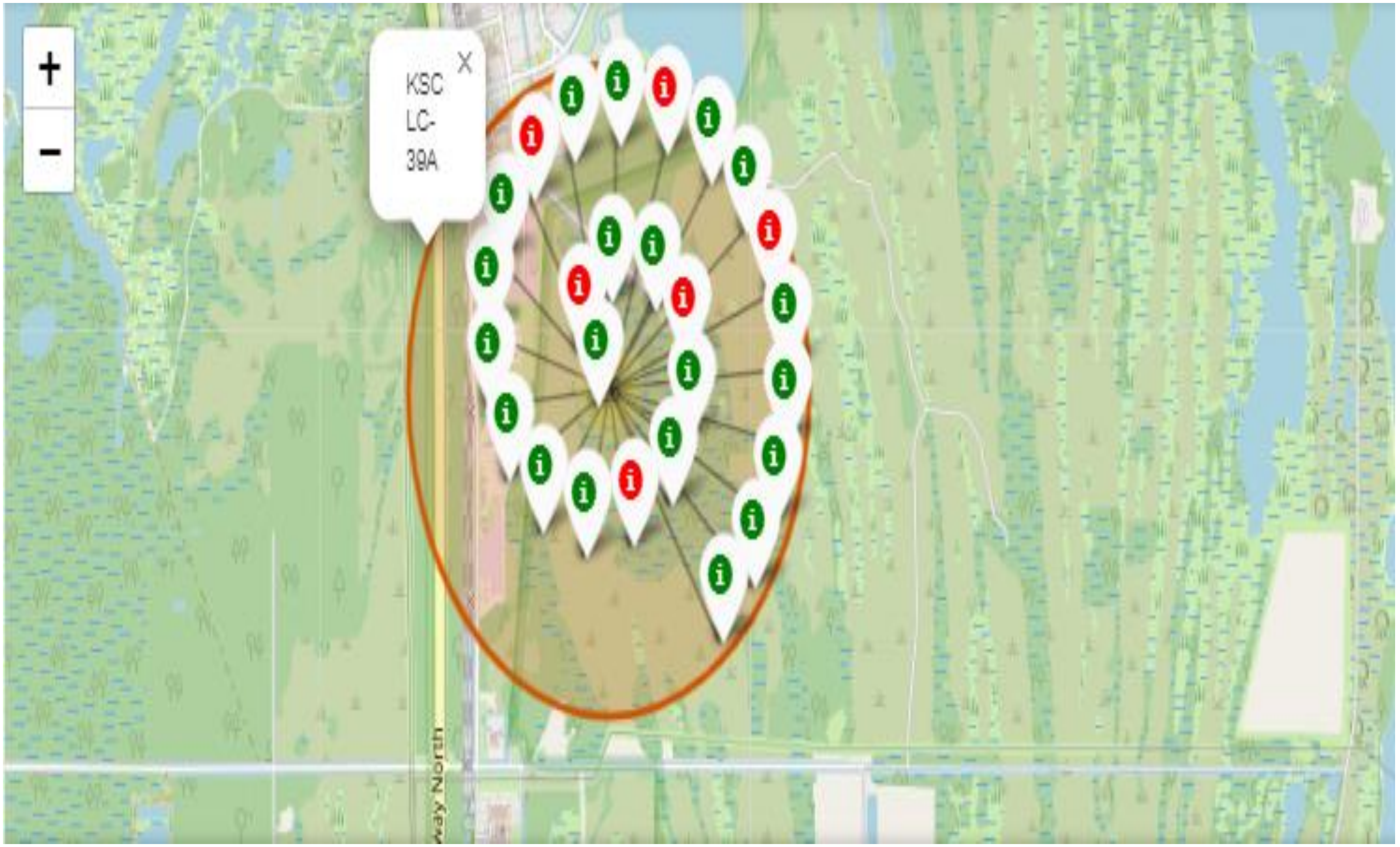
Colour-labeled launch records on map

- From the colour-labeled markers we should be able to easily identify which launch sites have relatively high success rates.

  Green Marker = Successful Launch

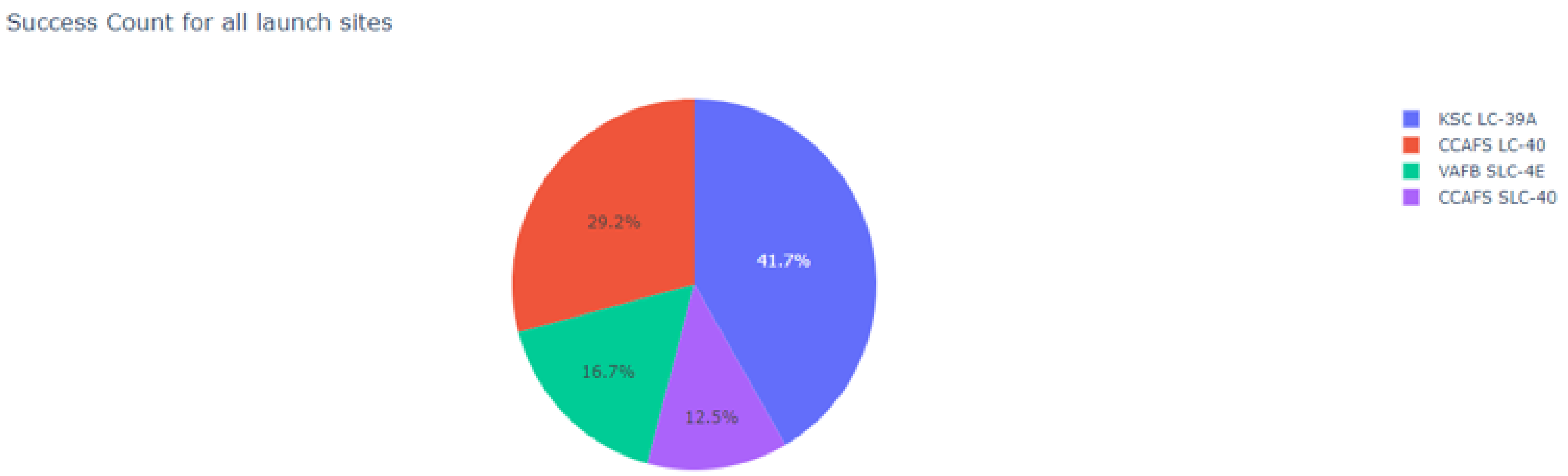
  Red Marker = Failed Launch

- Launch Site **KSC LC-39A** has a very **high Success Rate**.



RESULTS

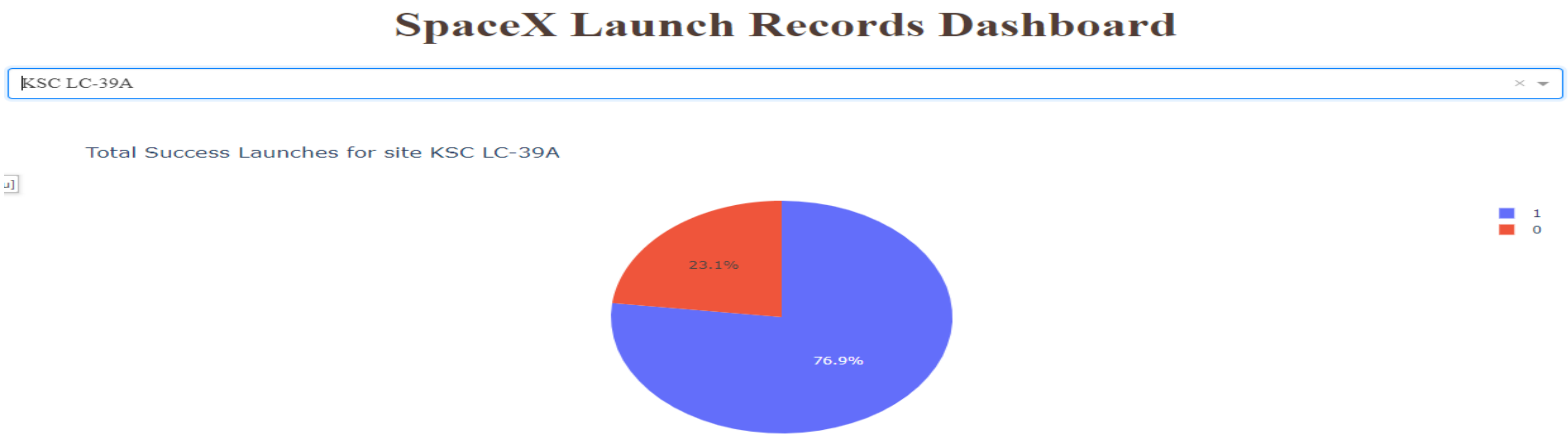
% of Success By Each Launch Site



KSC LC-39A has the **most successful launch** with **41.7%** from all successfull landings

RESULTS

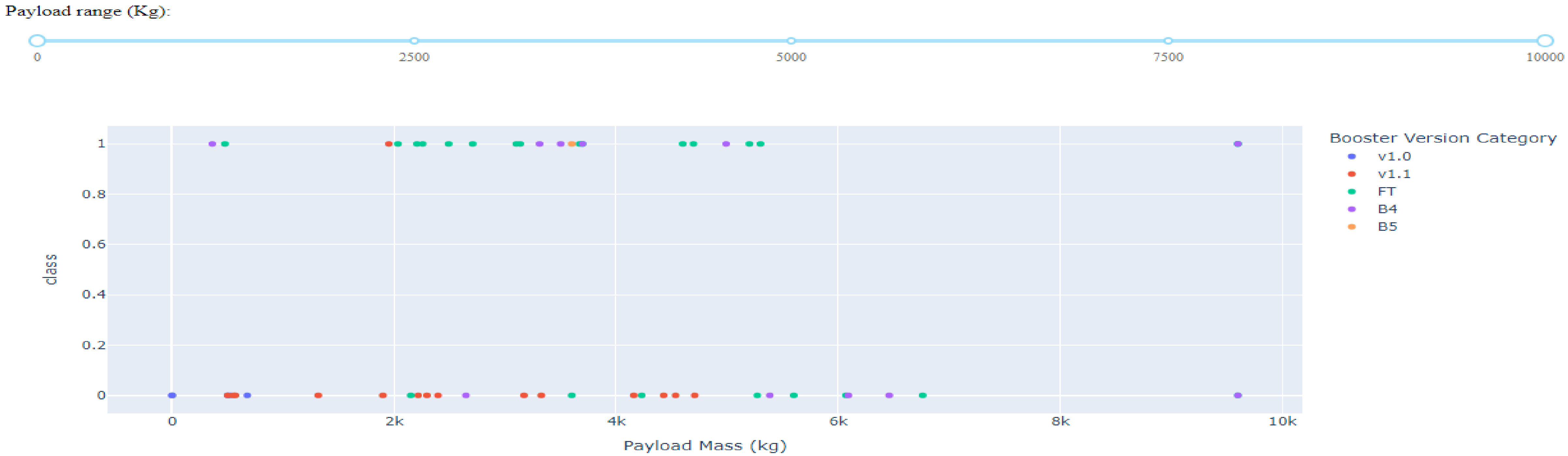
The highest launch-success ratio



KSC LC-39A has **76.9% success rate** while getting **23.1% failure rate**

RESULTS

Payload Mass vs Launch Outcome



The chart shows that **the higest success rate** is for **playload mass** between **2 000 kg and 5 500 kg** for **all site**

RESULTS

Classification Accurancy

Decision Tree method is **the best** to predict **successful landings** with **Accurancy 89%**

Find the method performs best:

```
[33]: algorithms = {'KNN':knn_cv.best_score_,'Tree':tree_cv.best_score_,'LogisticRegression':logreg_cv.best_score_}
bestalgorithm = max(algorithms, key=algorithms.get)
print('Best Algorithm is',bestalgorithm,'with a score of',algorithms[bestalgorithm])
if bestalgorithm == 'Tree':
    print('Best Params is :',tree_cv.best_params_)
if bestalgorithm == 'KNN':
    print('Best Params is :',knn_cv.best_params_)
if bestalgorithm == 'LogisticRegression':
    print('Best Params is :',logreg_cv.best_params_)
```

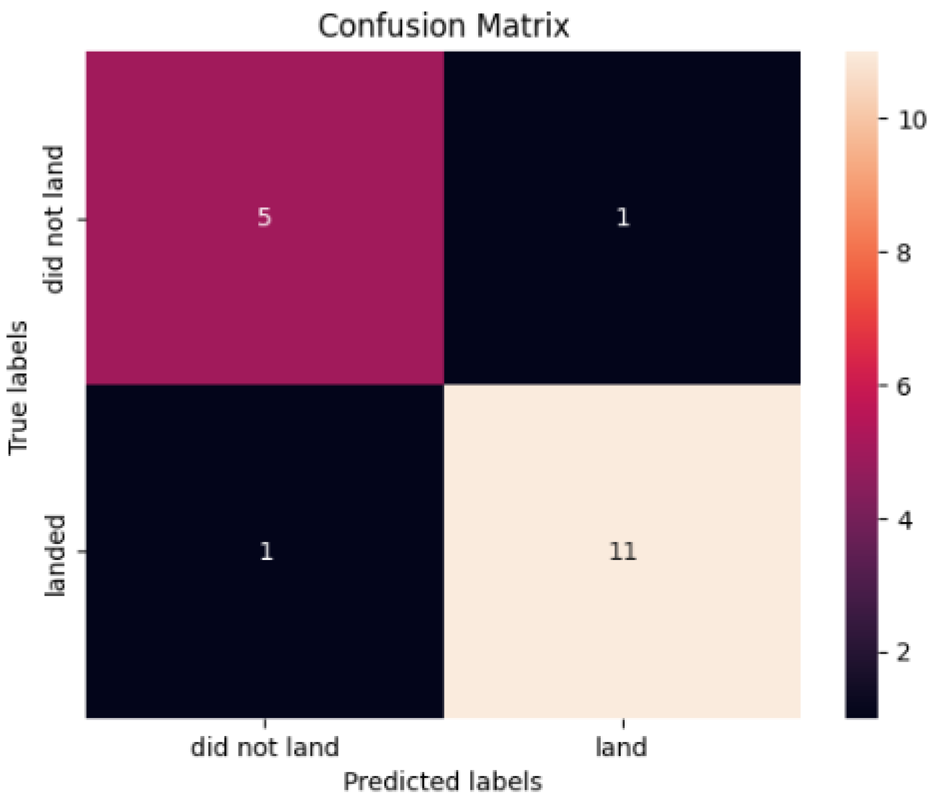
```
Best Algorithm is Tree with a score of 0.8875
Best Params is : {'criterion': 'entropy', 'max_depth': 10, 'max_features': 'sqrt', 'min_samples_leaf': 4, 'min_samples_spli
t': 5, 'splitter': 'random'}
```

RESULTS

Confusion Matrix

Examining the confusion matrix, we see that **decision tree** can **distinguish** between the **different classes.**

We see that the major problem is **false positives** i.e. unsuccessfull landing marked as successfull landing by clasifier.



CONCLUSION

- Decision Tree Model is the best algorithm for this dataset.
- Launches with a low payload mass show better results than launches with a larger payload mass.
- Most of launch sites are in proximity to the Equator line and all the sites are in very close proximity to the coast.
- The success rate of launches increases over the years.
- KSC LC-39A has the highest success rate of the launches from all the sites.
- Orbits ES-L1, GEO, HEO and SSO have 100% success rate.

APPENDIX

To get more details about Data Science

[IBM Data Science]

Jupyter notebooks and the presentation is available from

[GitHub]

THANK YOU