



IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

Miguel Valentín-Gamazo Pontijas
1st-Jul-2023



Outline

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

Executive Summary

- **Summary of methodologies**
 - Collecting the data (API and Webscraping)
 - Data Wrangling
 - EDA with SQL
 - EDA with Pandas+Matplotlib
 - Interactive map with Folium
 - Interactive dashboard with Plotly Dash
 - Predictive analysis with machine learning
- **Summary of all results**
 - EDA results
 - Interactive analysis
 - Predictive analysis

Introduction

- **Project background and context**

SpaceX is a successful player in commercial space travel. One reason behind this success is the relatively inexpensive cost of rocket launch thanks to the reusability of the first stage -the most expensive part of a rocket launcher-.

- **Problems you want to find answers**

In this project, our job will be to determine the price of each launch which depends on whether the first stage would land successfully to be reused.

Section 1

Methodology

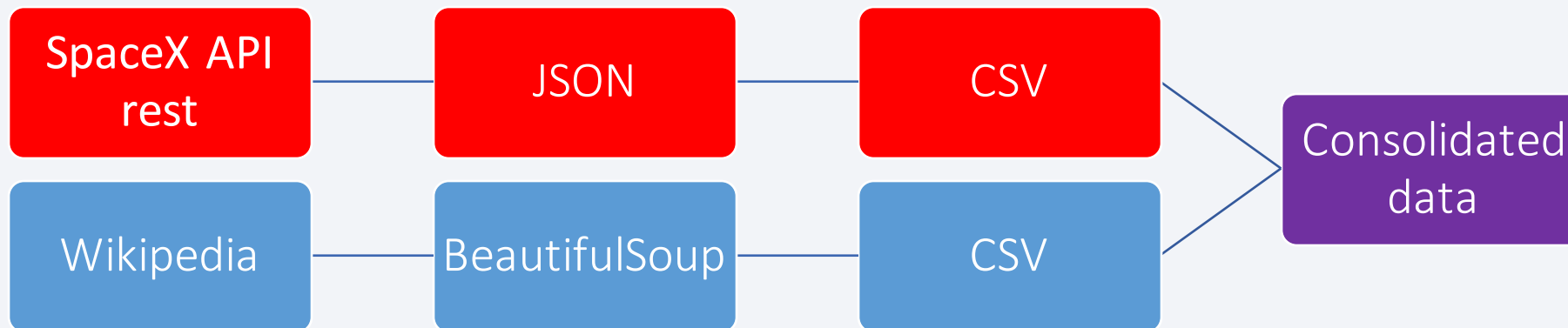
Methodology

Executive Summary

- Data collection methodology:
 - Space X rest API
 - Webscraping from Wikipedia using BeautifulSoup and Requests
- Perform data wrangling
 - Data were analyzed and new landing classes created using Pandas and Numpy
- Perform exploratory data analysis (EDA) using visualization and SQL
- Perform interactive visual analytics using Folium and Plotly Dash
- Perform predictive analysis using classification models
 - LR, KNN, SVM and DT models were created splitting the dataset into test and training

Data Collection

- First some data were extracted from the API rest of SpaceX using get Requests. Then received data were transformed from JSON into a csv
- A second set of data was downloaded from Wikipedia using Requests and BeautifulSoup. The result was a beautiful soup with html code, from which a table was extracted and converted into csv



Data Collection – SpaceX API

1. Get Request
2. Decode and normalize JSON code
3. Apply customized functions
4. Convert into dictionary and data frame
5. Data cleaning (missing values)
6. Export to csv

- <https://github.com/Zdarec56/Applied-Data-Science-Capstone-Coursera-IBM/blob/main/jupyter-labs-spacex-data-collection-api.ipynb>

1 Get request



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

2 JSON decode



```
decoded=response.json()  
data=pd.json_normalize(decoded)
```

3 Apply customized functions



```
getLaunchSite(data)
```

4 Convert into data frame



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

5 Cleaning and missing values



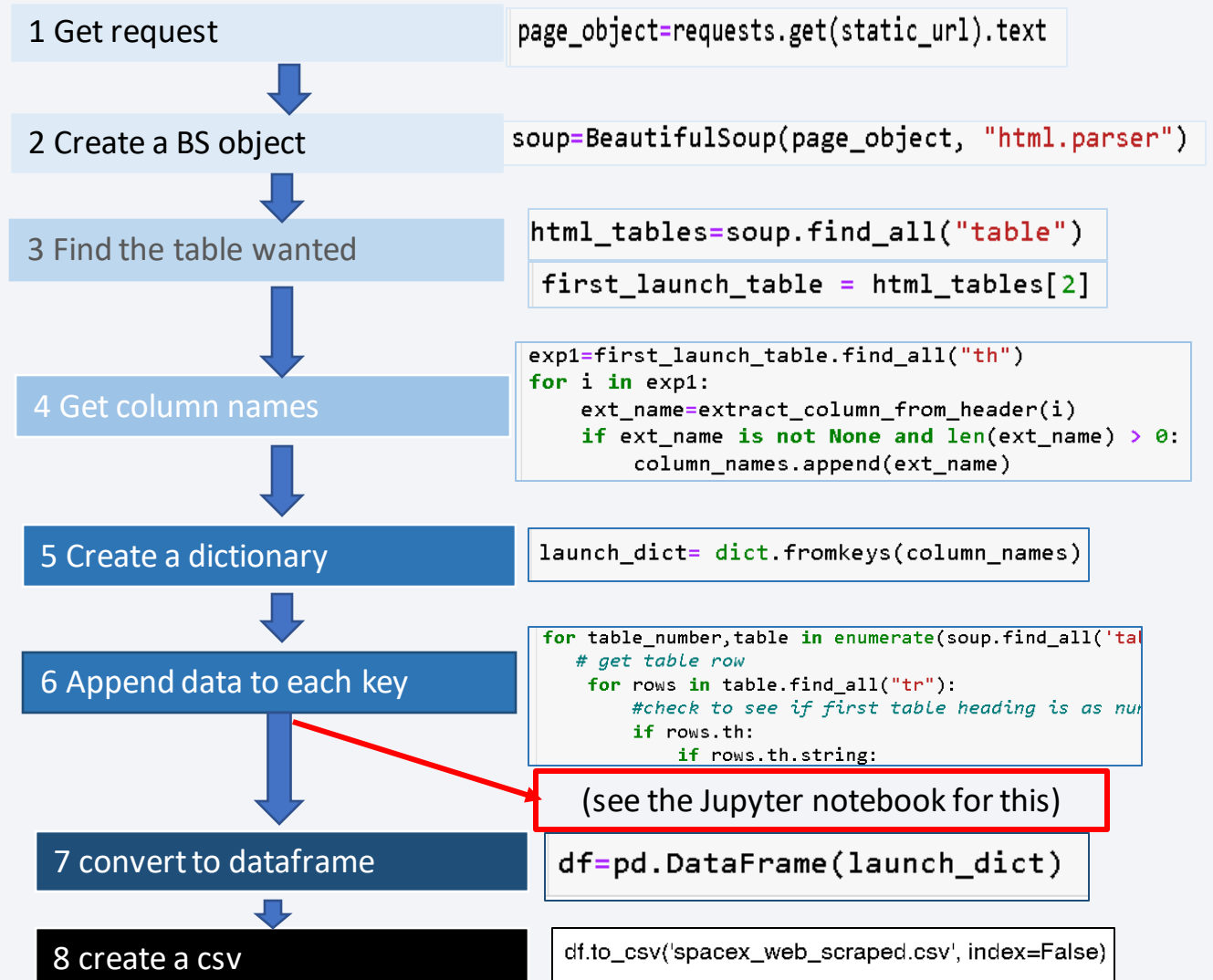
```
data_falcon9.isnull().sum()
```

6 Export to CSV

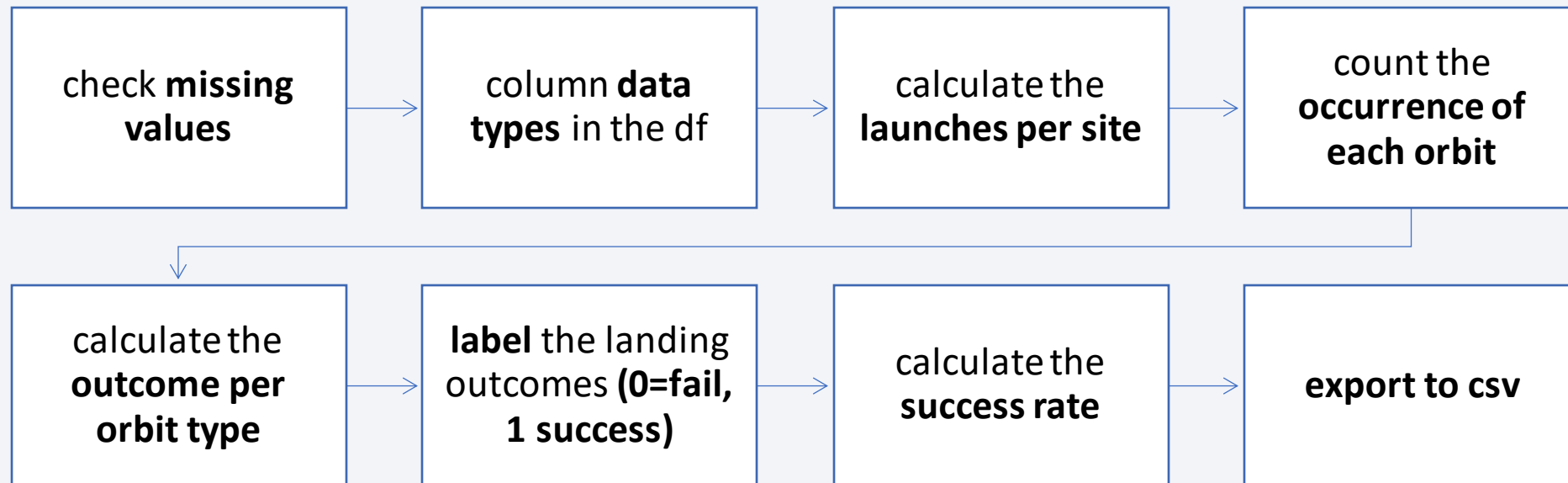
```
data_falcon9.to_csv(Path=r"C:\Users\Miguel\PycharmProjects\final pro
```


Data Collection - Scraping

- Bscraping from wikipedia
"List of Falcon 9 and Falcon Heavy launches"
- <https://github.com/Zdarec56/Applied-Data-Science-Capstone-Coursera-IBM/blob/main/jupyter-labs-webscraping2.ipynb>



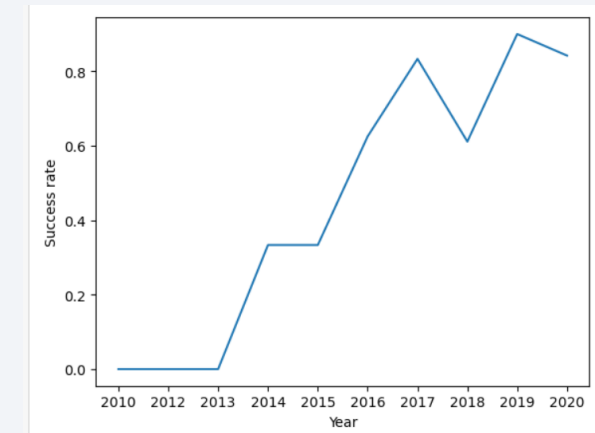
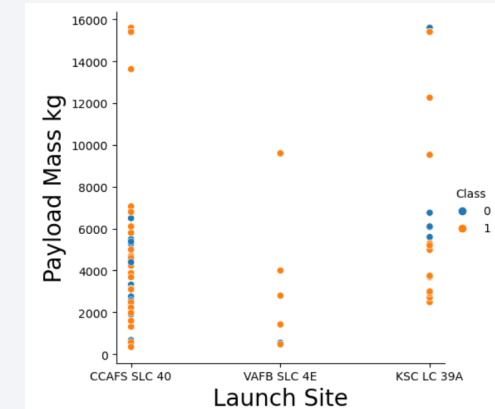
Data Wrangling



- <https://github.com/Zdarec56/Applied-Data-Science-Capstone-Coursera-IBM/blob/main/labs-jupyter-spacex-Data%20wrangling.ipynb>

EDA with Data Visualization

1. Relationship btw. **flight number and launch site** (by class): how often was each site used and whether at the end or at the beginning
2. **Launchsite vs. Payload mass**. From where were the heavy payloads launched (not from the second site)
3. **Orbit and class**. To show which orbits had higher success rate
4. **Flight number and orbit**. Which orbits were used at the beginning and which were used more often and success patterns
5. **Payload and orbit**. Which orbits are used for heavy or light payloads
6. **Year and success rate**. The success rate improves with time (tendence)



- https://github.com/Zdarec56/Applied-Data-Science-Capstone-Coursera-IBM/blob/main/IBM-DS0321EN-SkillsNetwork_labs_module_2_jupyter-labs-eda-dataviz.ipynb.jupyterlite.ipynb

EDA with SQL

- Display names of unique launch sites
 - Display records that start with CCA
 - Total payload mass by boosters launched by NASA
 - Average payload mass carried by boosters F9 v. 1.1
 - Date of first successful landing outcome
 - Name of boosters with success (drone ship) and mass btw. 4000-6000
 - Total number of successful and failure mission outcomes
 - Name of the booster_versions that have carried the maximum payload mass
 - List of records which month number + "failure (drone ship)" + booster version + launch_site for year 2015
 - Count of successful landings between 04-06-2010 and 20-03-2017
-
- https://github.com/Zdarec56/Applied-Data-Science-Capstone-Coursera-IBM/blob/main/jupyter-labs-eda-sql-coursera_sqlite_FINAL.ipynb

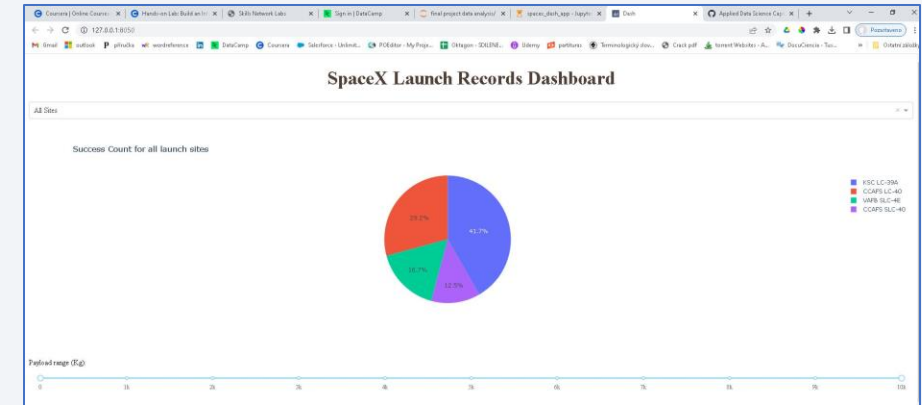
Build an Interactive Map with Folium

Different objects were created to the maps:

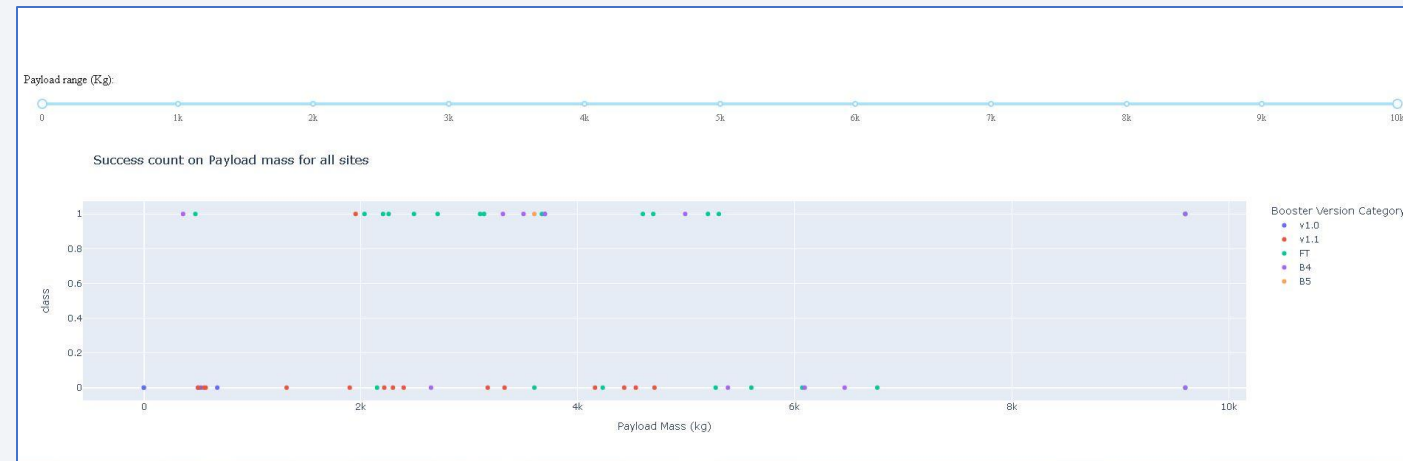
- **Circles and label names:** to show launch sites
 - **Marker clusters** (**green** and **red**): summarize all **successful** and **failed** launches
 - **Mouse position:** to show the latitude and longitude of pointer
 - **Polylines:** to measure distances between launch site and roads, coast...
-
- https://github.com/Zdarec56/Applied-Data-Science-Capstone-Coursera-IBM/blob/main/IBM-DS0321EN-SkillsNetwork_labs_module_3_lab_jupyter_launch_site_location.jupyterlite.ipynb

Build a Dashboard with Plotly Dash

- **Dropdown menu** to select the launch site
- **Pie chart** to show successful launches per site
- **Pie chart** to show success vs. failure of a specific site
- **Slider** to select payload range
- **Scatter plot**: correlation btw. Payload and launch success



https://github.com/Zdarec56/Applied-Data-Science-Capstone-Coursera-IBM/blob/main/spacex_dash_app.py



Results

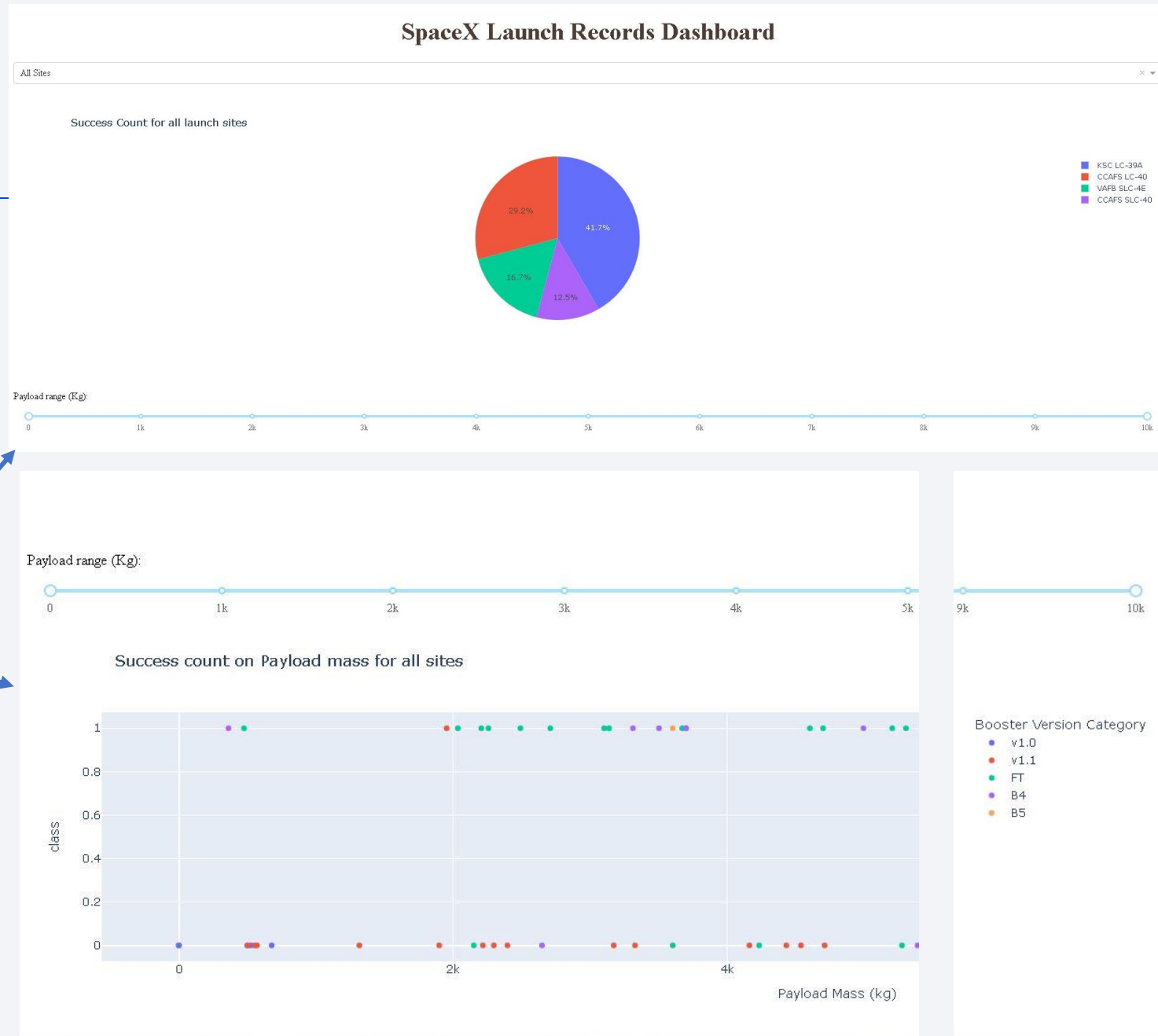
Exploratory data analysis results

- Not all launch sites were suitable for all payload masses.
- The success rate improved with time (highest in the last years)
- Average payload mass= 2928.4 kg

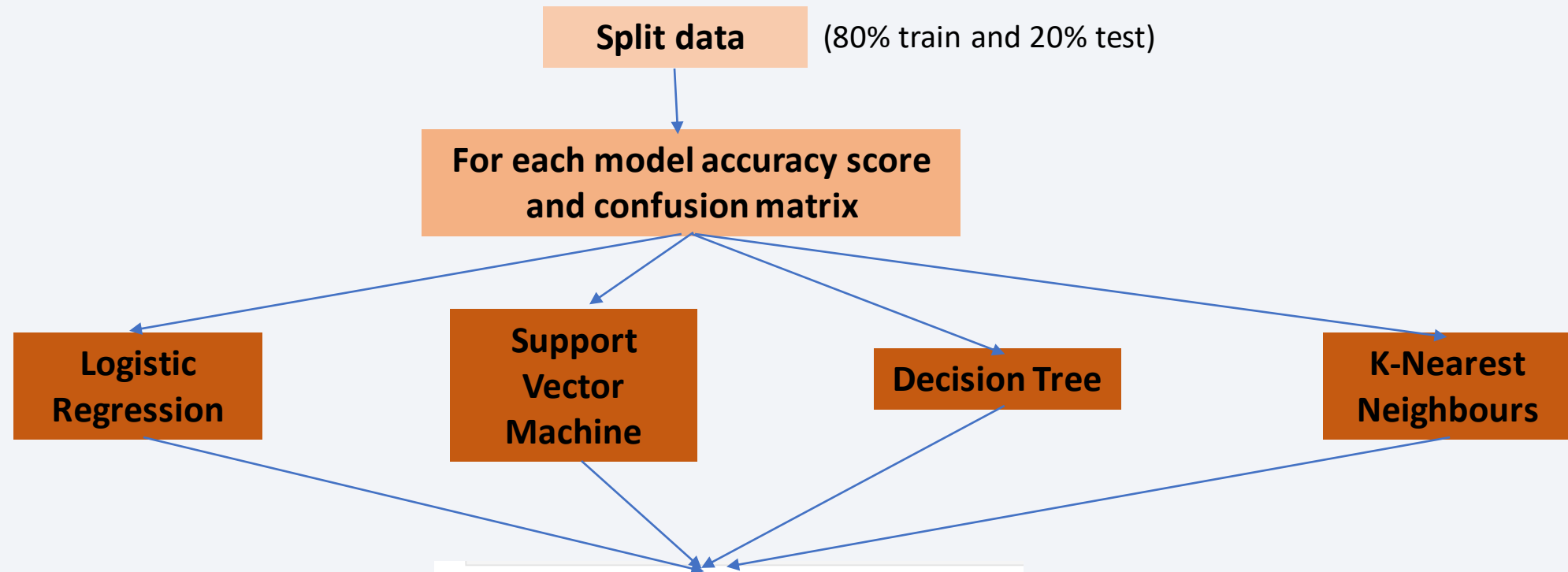
Interactive analytics demo in screenshots

Predictive analysis

- Decision tree had slightly better accuracy but poorer prediction score
- KNN was in general the best. Had the same prediction score than SVM and LogReg, but slightly better accuracy.



Predictive Analysis (Classification)



4] :	Model	Accuracy	Prediction score
0	LogisticRegression()	0.8464285714285713	0.8333333333333334
1	SVC()	0.8482142857142856	0.8333333333333334
2	DecisionTreeClassifier()	0.875	0.7777777777777778
3	KNeighborsClassifier()	0.8482142857142858	0.8333333333333334

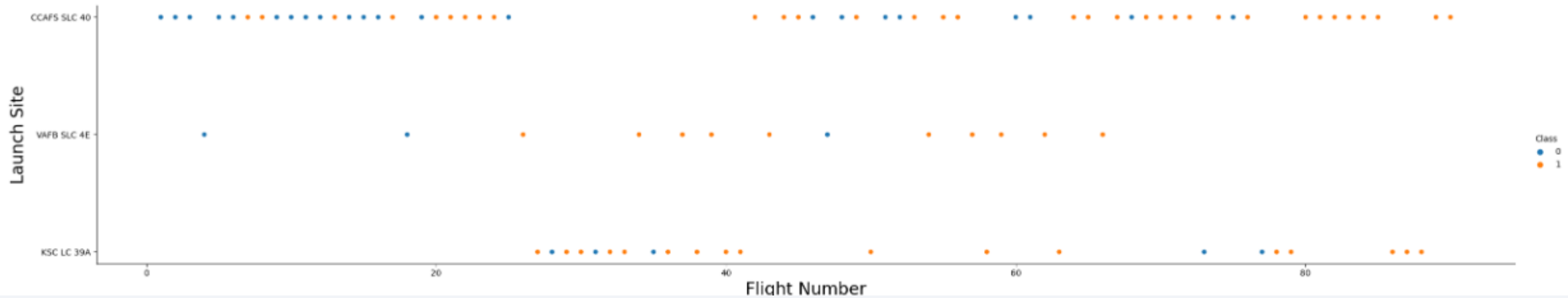
Decision tree: slightly better accuracy but worse prediction score
All the rest same score but KNN a bit better accuracy --> **KNN**

The background of the slide is an abstract composition. It features a dark blue base color. Overlaid on this are numerous diagonal streaks in shades of blue and red, creating a sense of motion or data flow. A faint, light blue grid pattern is also visible, particularly in the lower-left quadrant. The overall effect is high-tech and digital.

Section 2

Insights drawn from EDA

Flight Number vs. Launch Site



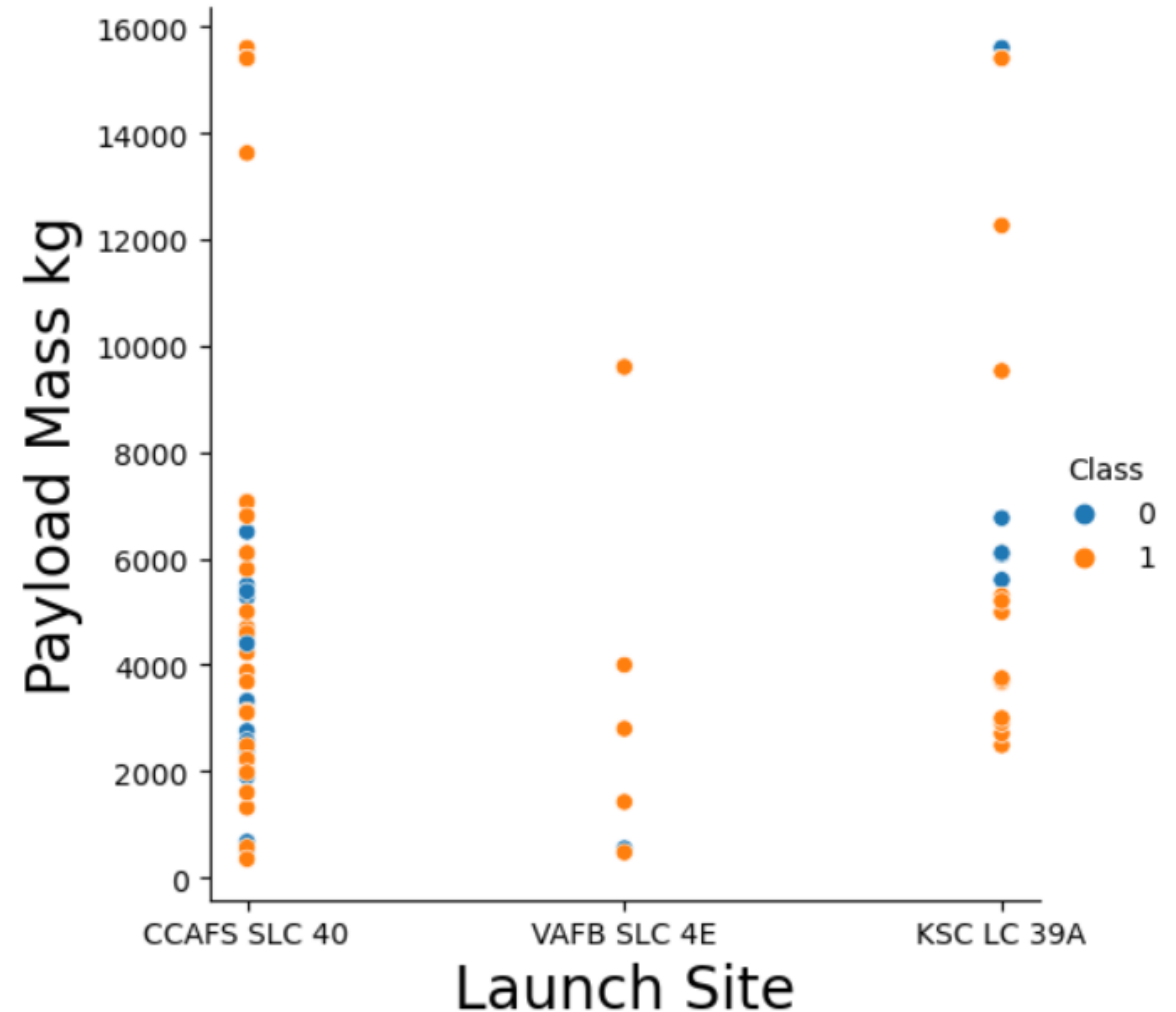
At the beginning, almost all launches were sent from CCAFS SLC 40

There is a period in the middle, during which LSC LC 39A substituted CCAFS SLC 40 (possible repair works?)

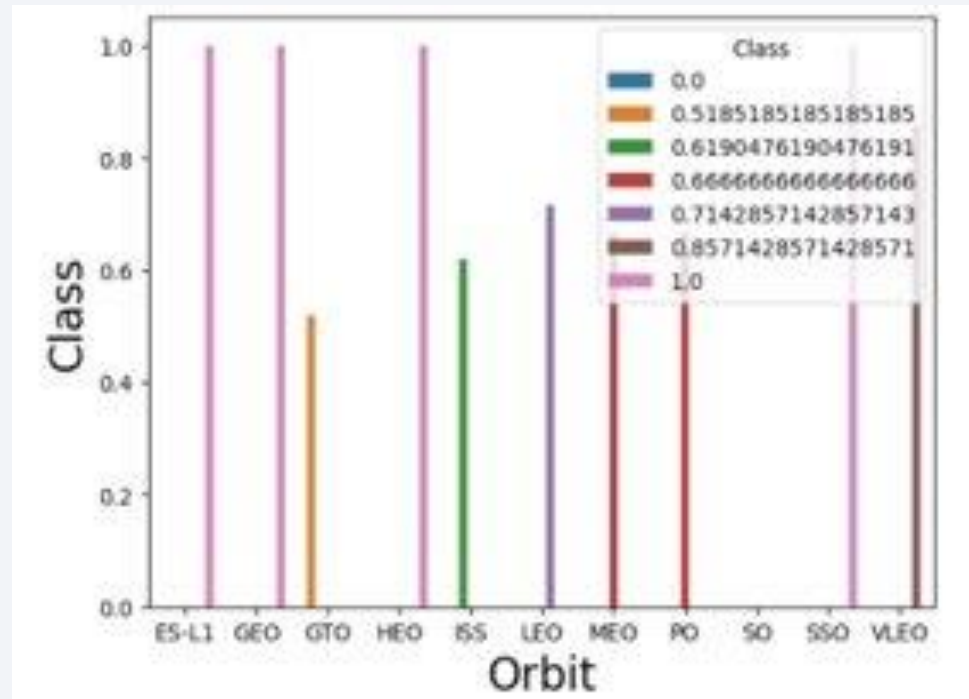
VAFB SLC 4E was less used than the others

Payload vs. Launch Site

- There are almost no launches from site VAFB SLC 4E
- From site VAFB SLC 4E there are not launches with payload higher than 10000 kg
- Class 1 was more widely used than class 0



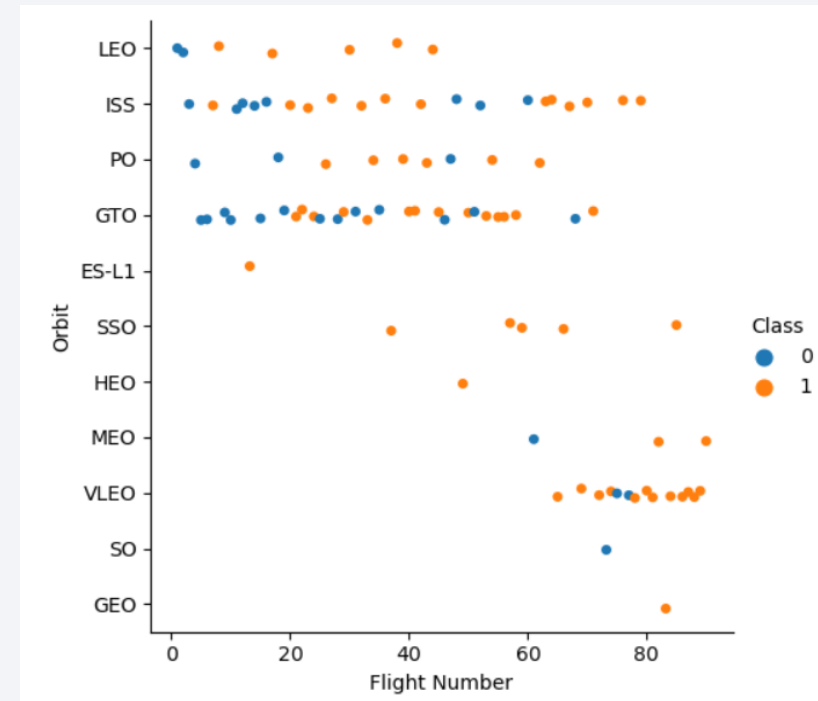
Success Rate vs. Orbit Type



SO the less successful (but only 1 launch see next slide)
GTO and ISS were the next less successful orbits
ES-L1, GEO, HEO and SSO the most successful

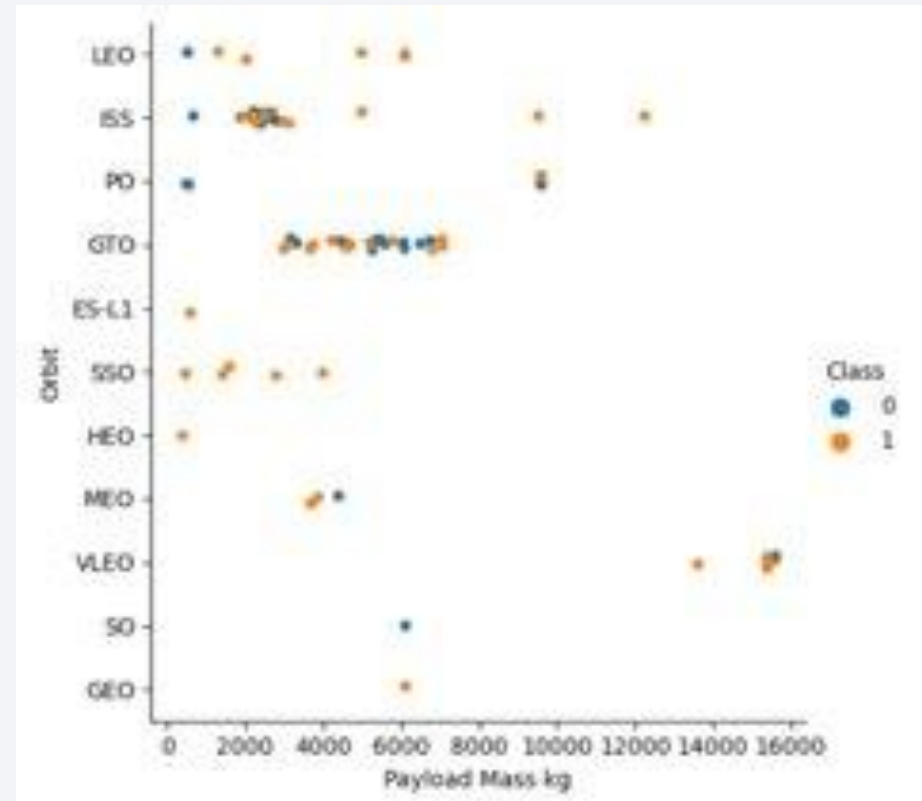
Flight Number vs. Orbit Type

- SSO had 5 out of 5 successful launches
- ES-L1, GEO, HEO, also 100% successful but only 1 launch
- LEO only fails at the beginning.
- No clear pattern for GTO



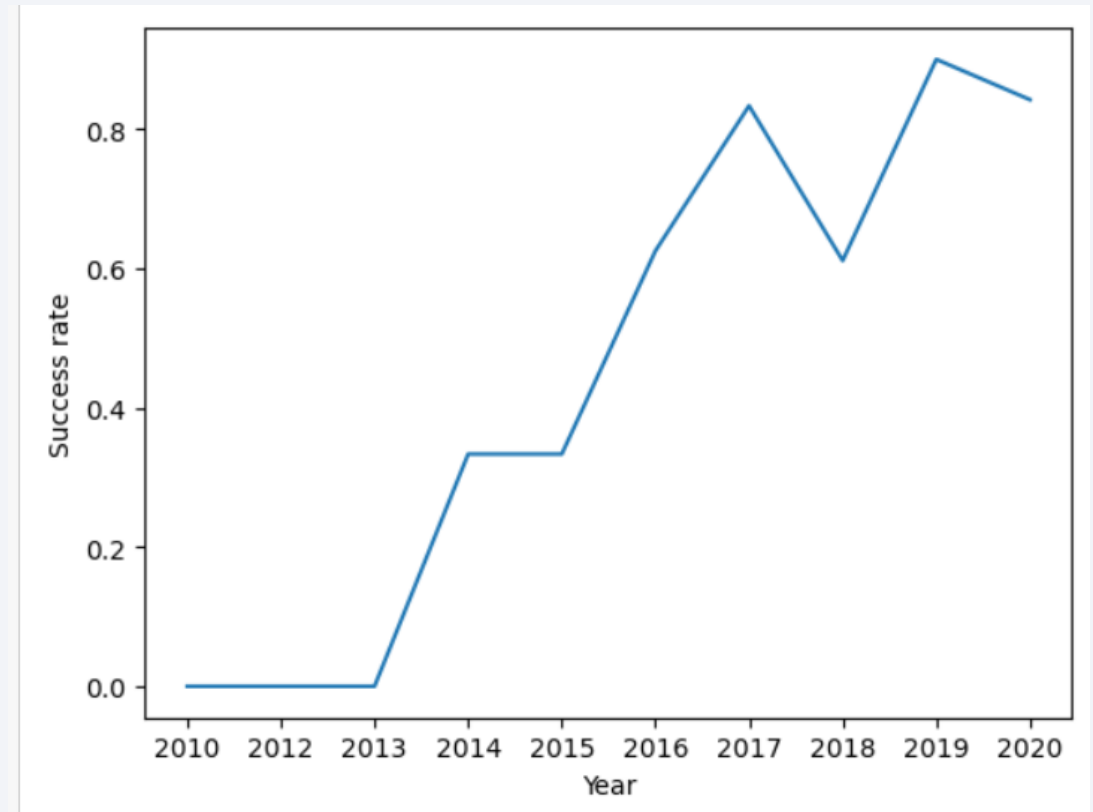
Payload vs. Orbit Type

- No clear pattern for GTO
- In LEO and ISS an increase of payload seems to increase the success
- SSO only low payloads but all successful



Launch Success Yearly Trend

The success rate seems to increase by time. That is due to the cumulated knowledge/experience).



All Launch Site Names

- Find the names of the unique launch sites

%sql : to indicate that the line is not Python but SQL

SELECT: the start of most SQL requests

DISTINCT Launch_Site: from the column Launch Site we want to select only unique names (no repeat)

FROM SPACEXTBL: the table from which data are retrieved is called SPACEXTBL

```
] : %sql SELECT DISTINCT Launch_Site FROM SPACEXTBL
* sqlite:///my_data1.db
Done.
] : Launch_Site
   CCAFS LC-40
   VAFB SLC-4E
   KSC LC-39A
   CCAFS SLC-40
```


Launch Site Names Begin with 'CCA'

- Find 5 records where launch sites begin with 'CCA'

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

WHERE Launch_Site LIKE 'CCA%': restricts the search to elements that start by CCA.

LIMIT 5: restricts the search to the first 5 elements



Launch_Site
CCAFS LC-40
CCAFS LC-40
CCAFS LC-40
CCAFS LC-40
CCAFS LC-40

Total Payload Mass

- Calculate the total payload carried by boosters from NASA

```
%sql SELECT SUM(PAYLOAD_MASS__KG_) FROM SPACEXTBL WHERE Customer='NASA (CRS)'
```

`SUM(PAYLOAD_MASS__KG_)`: sums all values from that column

`WHERE Customer='NASA (CRS)'`: restricts the selection only to lines where the customer is NASA



```
* sqlite:///my_data1.db
Done.
[9]:  SUM(PAYLOAD_MASS_KG_)
      45596
```

Average Payload Mass by F9 v1.1

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

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

AVG(PAYLOAD_MASS__KG_)
2928.4

- The clause WHERE is to restrict the selection. The name of the booster shall be between simple quotation marks '___'

First Successful Ground Landing Date

- The first successful landing date

```
1 %sql SELECT min(Date) FROM SPACEXTBL WHERE [Landing _outcome]='Success (ground pad)'  
2
```

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

min(Date)

01-05-2017

Successful Drone Ship Landing with Payload between 4000 and 6000

- Names of the boosters:

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

```
%sql SELECT Booster_Version FROM SPACEXTBL WHERE [Landing_Outcome]='Success  
(drone ship)' AND PAYLOAD_MASS__KG_ BETWEEN 4000 and 6000
```

- WHERE --> to insert conditions. In this case, there are two conditions (successful landing+ payload btw 4k and 6k); therefore, we use AND to ensure both conditions are fulfilled.

Total Number of Successful and Failure Mission Outcomes

```
%sql SELECT MISSION_OUTCOME AS Name_of_outcome, COUNT(MISSION_OUTCOME) AS  
numer_outcomes FROM SPACEXTBL GROUP BY MISSION_OUTCOME;
```

Name_of_outcome	numer_outcomes
Failure (in flight)	1
Success	98
Success	1
Success (payload status unclear)	1

There are 100 successful mission outcomes vs. only 1 failure

Boosters Carried Maximum Payload

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

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

There is a nested subquery in the clause where, to select those launches with maximum payload mass

2015 Launch Records

```
%sql SELECT substr(Date, 4, 2) AS month, [LANDING_OUTCOME], BOOSTER_VERSION, LAUNCH_SITE FROM  
SPACEXTBL WHERE [LANDING_OUTCOME]='Failure (drone ship)' AND substr(Date, 7, 4)='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

- In this case, the failed ones are required.
- SQLite does not support month names. So instead we use *substr(Date, 4, 2)* as month and *substr(Date, 7, 4)='2015'* for year.

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

```
%sql SELECT LANDING_OUTCOME, COUNT([LANDING_OUTCOME]) AS count_number FROM SPACEXTBL WHERE  
Date BETWEEN '04-06-2010' AND '20-03-2017' GROUP BY Landing_Outcome ORDER BY count_number DESC
```

Landing_Outcome	count_number
Success	20
No attempt	10
Success (drone ship)	8
Success (ground pad)	7
Failure (drone ship)	3
Failure	3
Failure (parachute)	2
Controlled (ocean)	2
No attempt	1

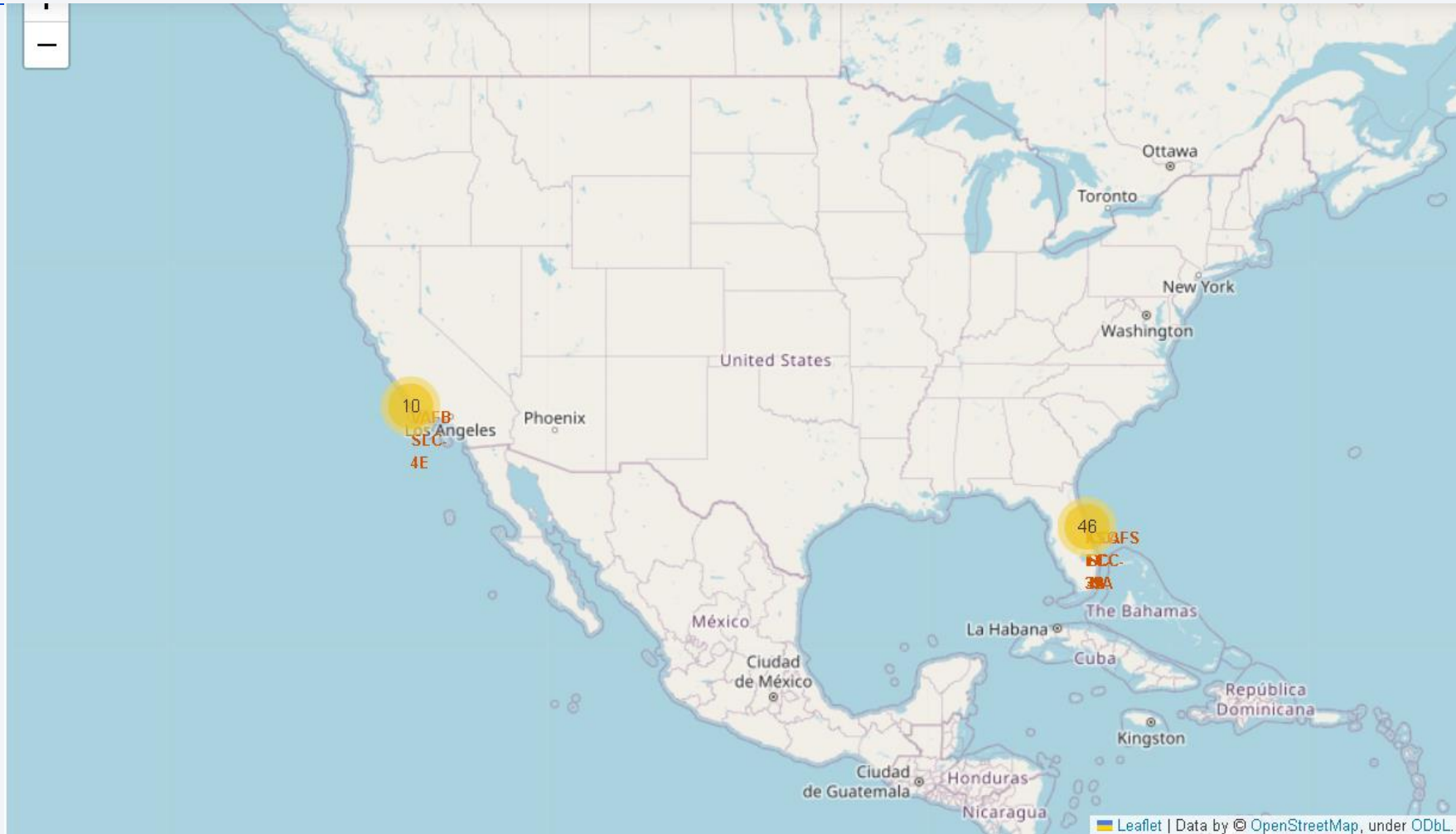
Success and *No attempt* are the most common landing outcomes.

A satellite view of Earth from space, showing the curvature of the planet and city lights at night. The image is a composite of a dark blue sky with stars and a view of the Earth's surface from space. The Earth's surface is mostly dark blue, with a thin layer of white clouds. A curved horizon line separates the dark sky from the Earth's surface. On the right side, there are bright, glowing yellow and orange lights, likely representing city lights or urban areas. The overall image has a high-contrast, cinematic quality.

Section 3

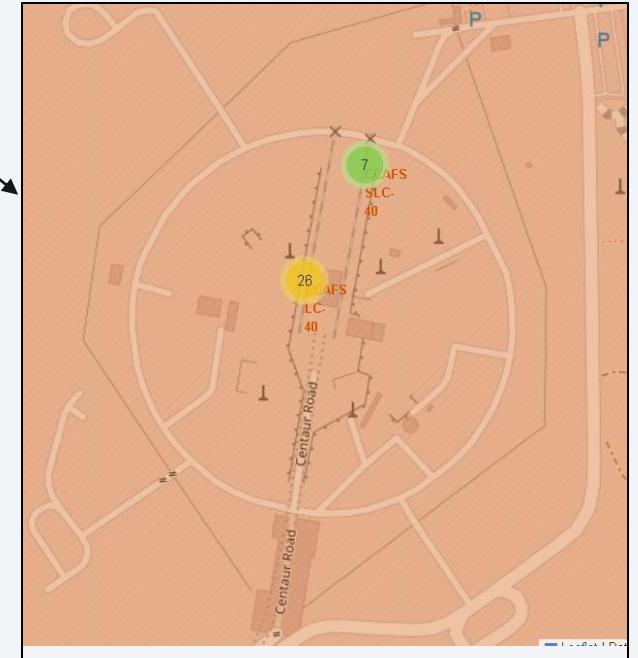
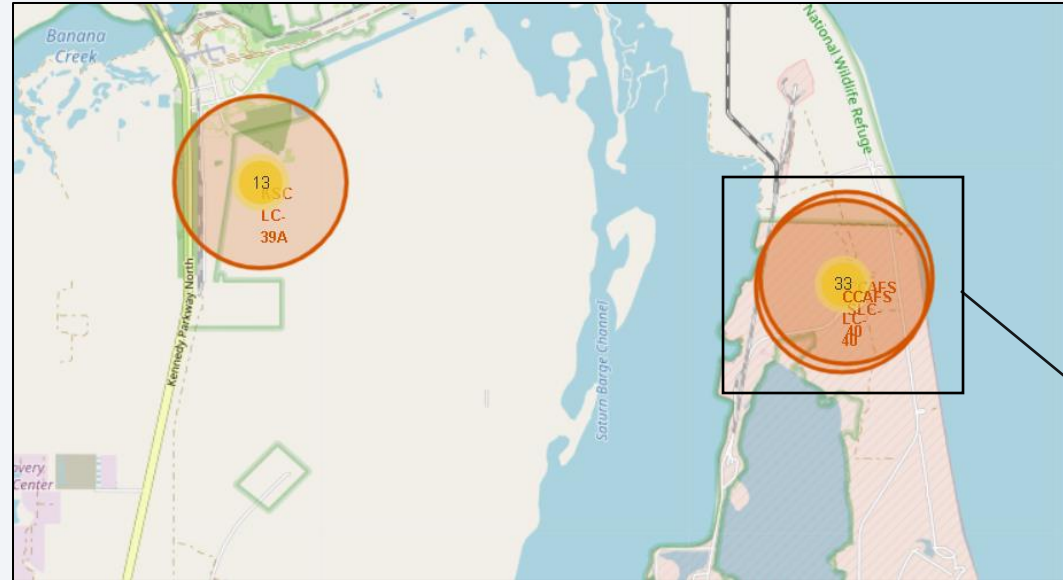
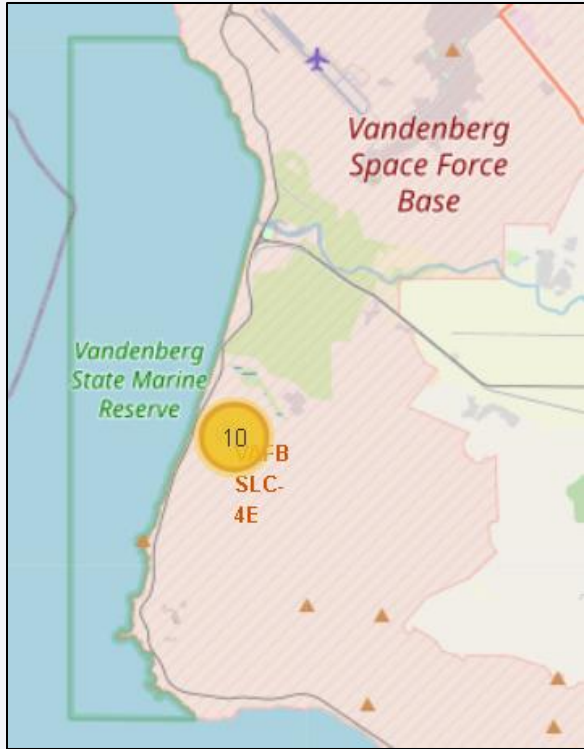
Launch Sites Proximities Analysis

Map of all launch sites



There are 56 Launch sites, 46 in the East coast (Cape Canaveral) and 10 by the West coast.

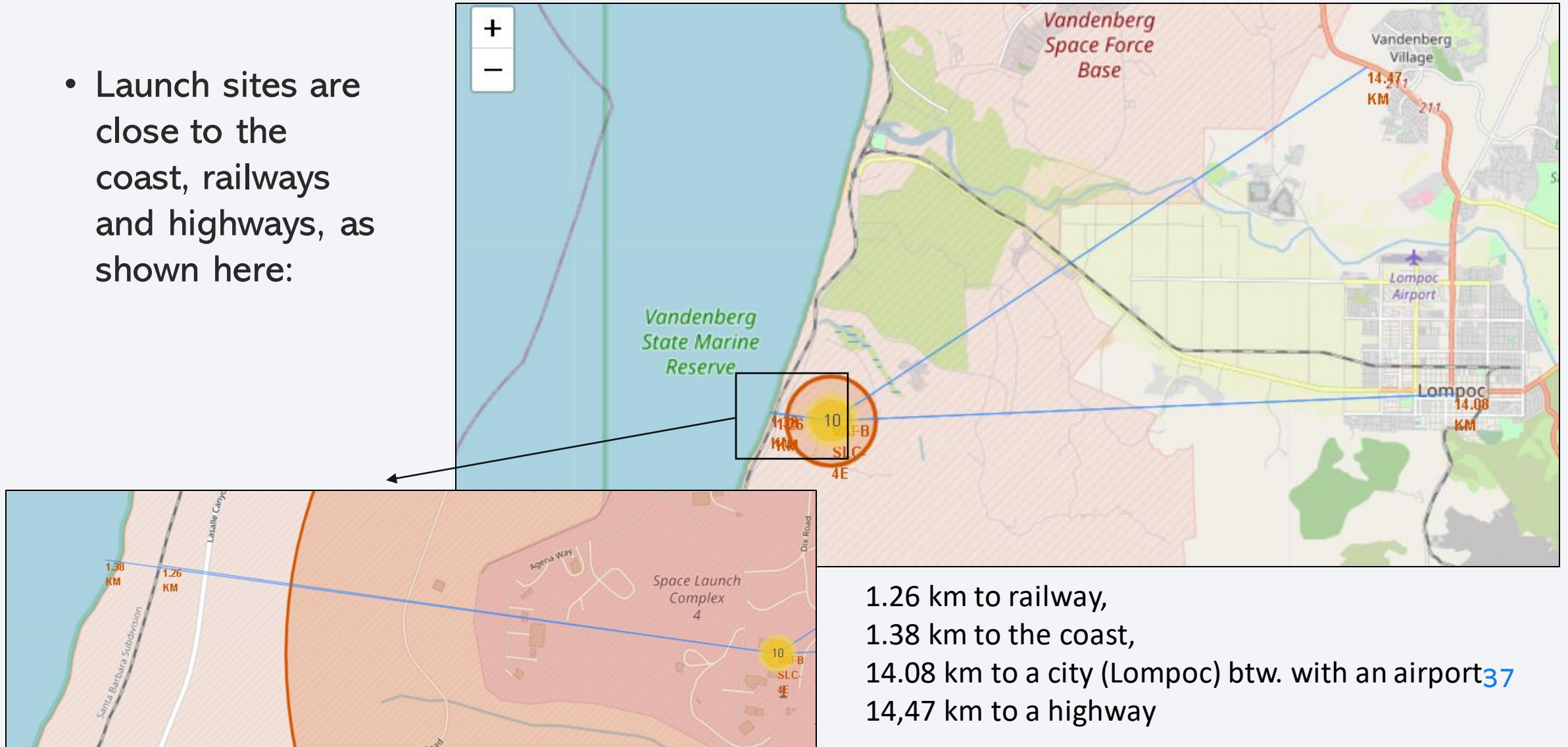
Success/failed launches for each site



Only one site
shows failed AND
successful launches
(CCAFS-LC-40)

Launch site: distance to infrastructures

- Launch sites are close to the coast, railways and highways, as shown here:

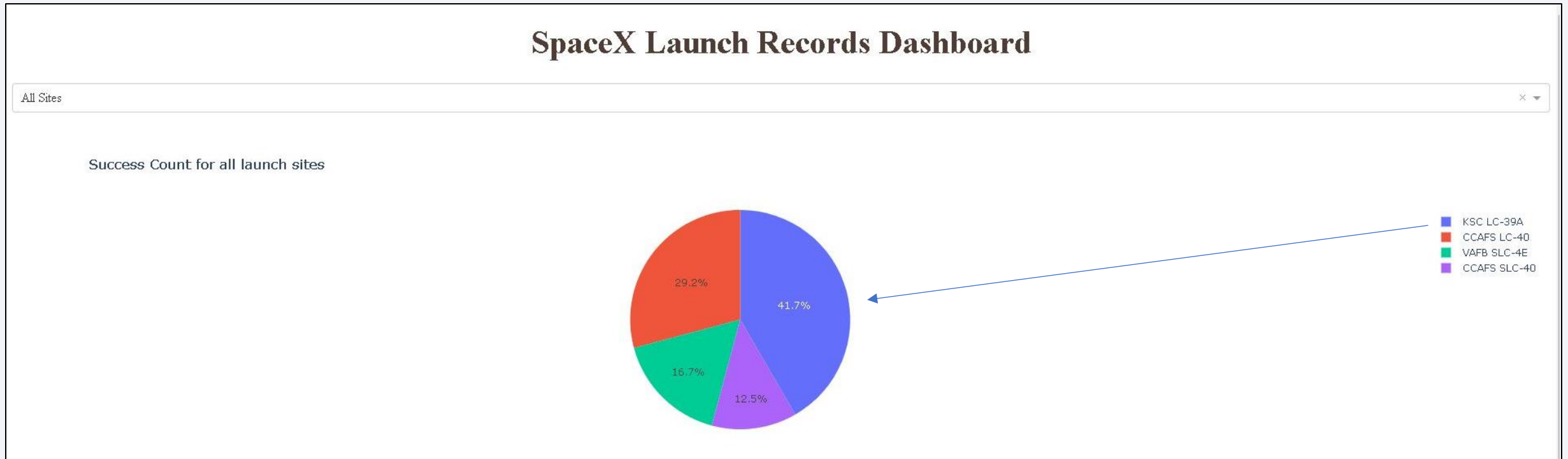




Section 4

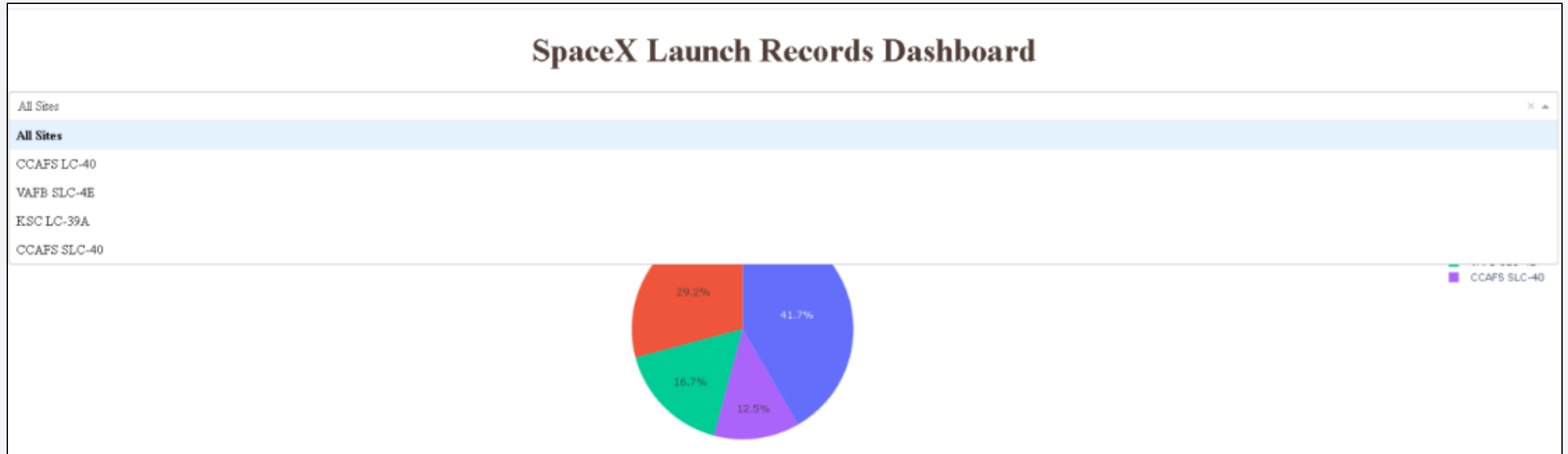
Build a Dashboard with Plotly Dash

Total successful launches (by all sites)



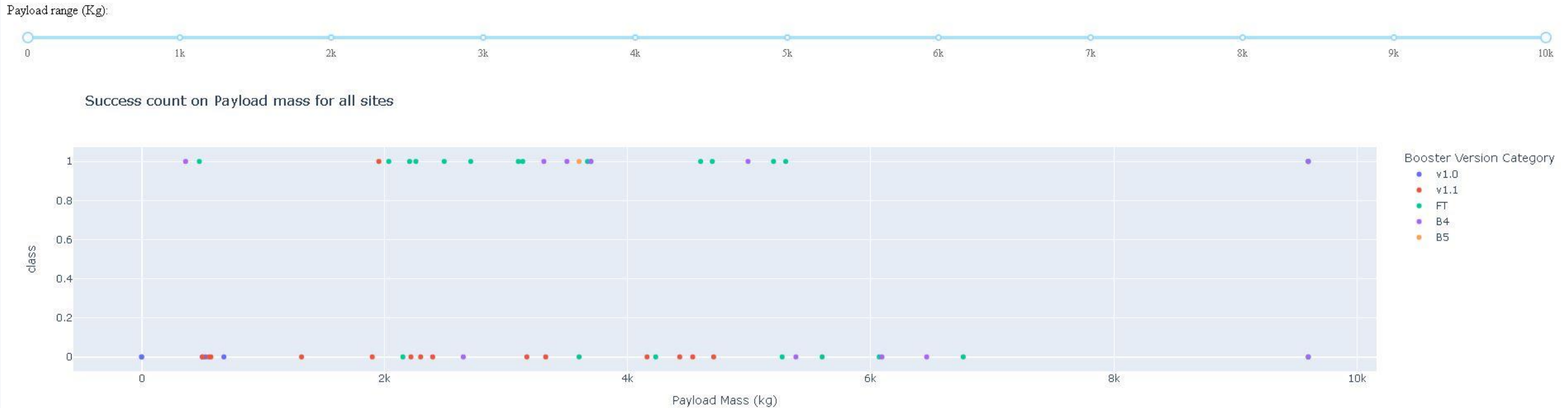
Most successful launches are from the site KSC LC-39A (41,7%)

Piechart for each site



Using the selectable menu it is possible to generate a pie chart for any specific launch site

Payload vs Launch outcome

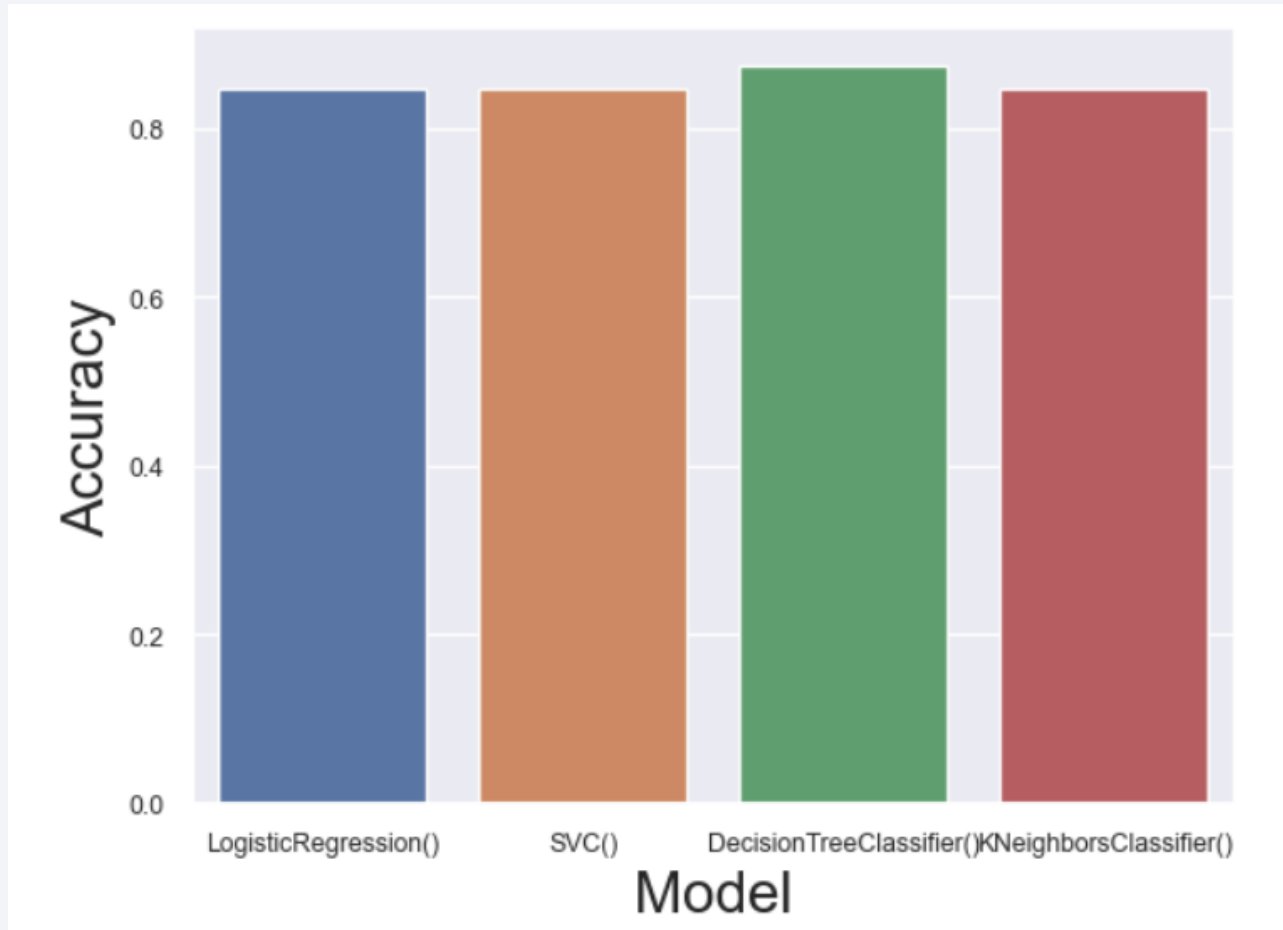


Success rate is higher for payloads between 2k and 4k

Section 5

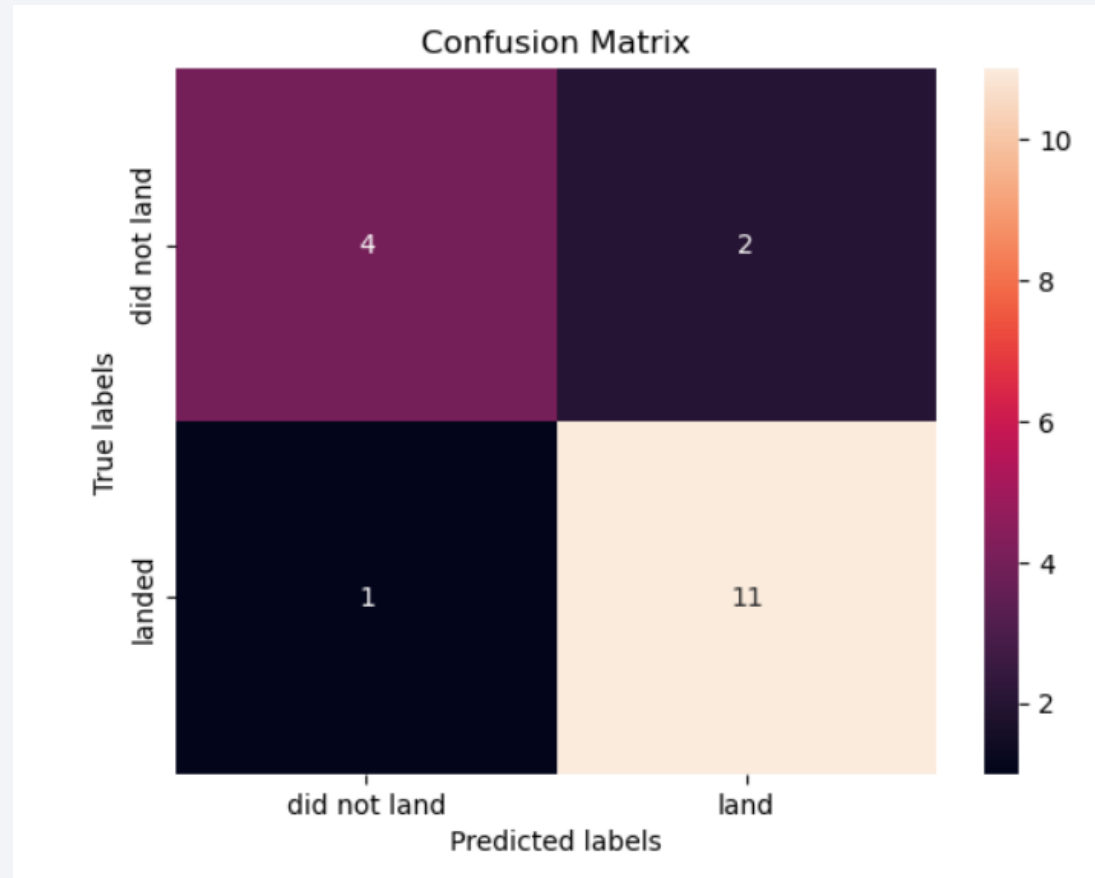
Predictive Analysis (Classification)

Classification Accuracy



Decision Tree provides the highest classification accuracy

Confusion Matrix



Decision tree model has the best accuracy. Also we see that it is good at distinguishing between the different classes. We see that the error is distributed both in false positives (2) and false negatives (1).

Conclusions

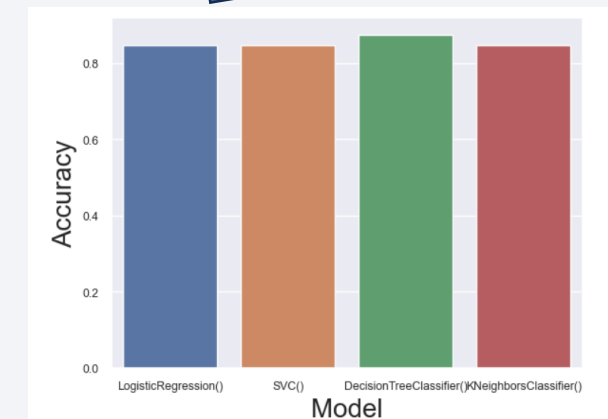
- ES-L1, GEO, HEO and SSO the most successful orbits
- In orbits LEO and ISS an increase of payload seems to increase the success
- The success rate of launches seems to increase by time. That is due to the cumulated knowledge/experience).
- The first successful landing date was the 01-05-2017
- There are 100 successful mission outcomes vs. only 1 failure
- Most successful launches are from the site KSC LC-39A (41,7%)
- All four ML models provide similar results. Decision Tree provides the highest classification accuracy

Appendix

- Python code to create the barchart on the slide 43

```
sns.set(font_scale=0.8) #to reduce the font scale so that the
labels do not overlap
results_df['Accuracy'] = pd.to_numeric(results_df['Accuracy'])
results_df['Prediction
score'] = pd.to_numeric(results_df['Prediction score'])
sns.barplot(x="Model",y="Accuracy",data=results_df)
plt.xlabel("Model",fontsize=20)
plt.ylabel("Accuracy",fontsize=20)

plt.show()
```



Thank you!

