IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

Eilind Karlsson
11.11.2024

Outline

- Executive Summary
- Introduction
- Methodology
- Results
- Conclusion

Executive Summary - required

- Summary of methodologies
  - Data collection through API and Web Scraping
  - Data Wrangling using e.g. OneHotEncoder
  - Exploratory Data Analysis using SQL and Seaborn (for visualisations)
  - Visual Analytics using Folium
  - Machine Learning Predictions using KNN, Decision tree, SVM and Logistic Regression
- Summary of all results
  - Visualisation revealed that some factors contribute more clearly to success than others
  - The most accurate machine learning algorithm to predict success is the Decision Tree

Introduction - required

- Project background and context

  SpaceX has a considerably lower cost of rocket launches than other competitors. Specifically, a Falcon 9 launch cost around 62 million dollars while a launch from other providers cost up to 165 million dollars. A big contribution to keeping the costs down for SpaceX comes from their ability to reuse the first stage of the rocket launch. However, this requires a successful landing of the first stage. Predicting if the first stage will land successfully is hence valuable information when predicting the cost of a rocket launch, useful for example when bidding on a project.

- Problems you want to find answers to
  - Which factors determine if a launch is successful?
  - How heavily do the above factors contribute towards success?
  - How do we obtain the most cost-effective launch protocol?

Section 1

Methodology

Methodology

Executive Summary

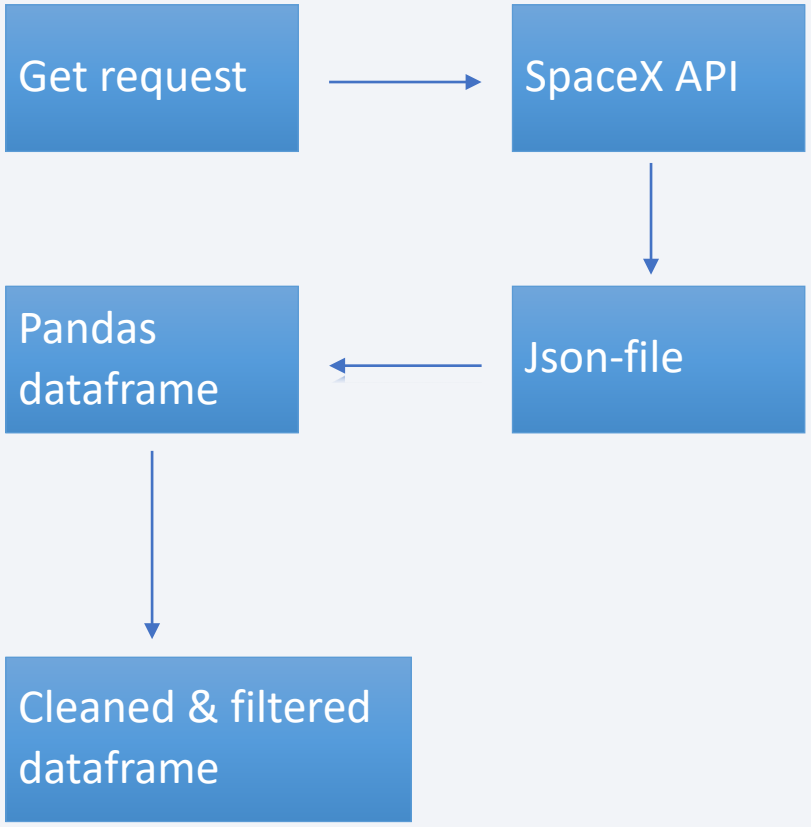
- Data collection methodology
  - Data was collected from the SpaceX API and web scraping from Wikipedia
- Perform data wrangling
  - Categorical features were made into binary columns using OneHotEncoder
- Perform exploratory data analysis (EDA) using visualization and SQL
- Perform interactive visual analytics using Folium and Plotly Dash
- Perform predictive analysis using classification models
  - Trained Machine Learning Models using Logistic Regression, SVM, Decision Tree and KNN to predict the outcome of launches

Data Collection

- The data was first imported using get requests from the SpaceX API (using a URL)
- The json results were normalised and turned into a Pandas dataframe
- The SpaceX API was again used to clean the data
  - Specifically, we used the IDs of each launch to get information about the rocket type, payloads, launchpad and cores by calling predefined functions
- The data frame was filtered to only contain Falcon 9 launches
- Missing values were replaced by e.g. the mean value of the column
- We also used Beautiful Soup to extract a Falcon 9 launch records HTML table from Wikipedia and then convert it into a Pandas dataframe

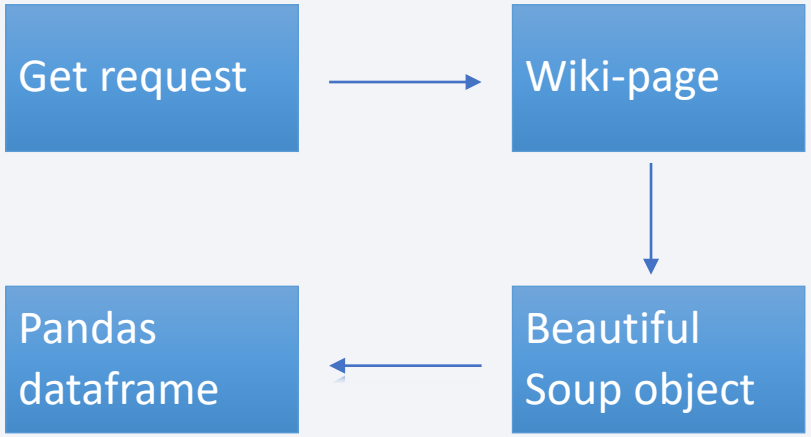
Data Collection – SpaceX API

- Get request to extract a Json-file from the SpaceX API
- Convert the Json-file to a pandas dataframe
- Used the Launch IDs to extract info about rocket, payload, launchpad and cores
- Filtered dataframe for Falcon 9 launches
- Github URL



Data Collection - Scraping

- HTTP GET request to get the Falcon 9 Launch HTML Page as a Beautiful Soup object
- Extract the tables and corresponding column names
- Parse through the Beautiful Soup object and extract it as a Pandas dataframe
- Github URL



Data Wrangling

- Load the SpaceX dataset from earlier data collection-process
- First we calculated the number of launches at each site and the number and occurrence of each orbit
- Extracting the unsuccessful landings we then created a landing outcome label (‘Class’) in a new column (where 1=successful landing, 0=unsuccessful)

- GitHub URL

EDA with Data Visualization

- Used scatter plots to examine possible relationships between e.g. Flight Number and Launch Sites, Payload Mass and Launch Site etc.
- Bar chart was created to examine the success rate of the different orbits and find the most successful orbits
- Line plot illustrating how the success rate has progressed over time (yearly from 2013-2020)
- Also used OneHotEncoder to turn categorical feature to one-hot encoded format for future analysis

- GitHub URL

EDA with SQL

- Having established a connection to the database we used standard SQL queries like SELECT DISTINCT, SELECT .. WHERE, GROUP BY, SUM, COUNT(*), MIN and AVG to solve the EDA-tasks

- GitHub URL

Build an Interactive Map with Folium

- We first initialised a folium-map with the different launch sites (using folium.Circle and folium.map.Marker)
- Then we created a new column with a colour-coding of the outcome of the launch (green/red) and added this to the map using add_child and folium.map.Marker
- Lastly, we calculated the distance from a chosen launch site to the ocean, closest highway, closest railroad and closest city and indicated this in the map by adding a line together with the distance being displayed
- All of this was done to highlight important considerations when choosing a launch site, e.g. we can see that proximity to the ocean is important from looking at the map
- GitHub URL

Build a Dashboard with Plotly Dash

- Used piecharts to visualize success rate at all sites respectively the individual launch sites
- Used scatter plots to visualize the success rate of the launches colored by booster version with a slider that lets the used determine the range of payload mass (in kg)
- The above allows the user to examine which launch site is the most successful, as well as which booster-version and payload mass is the best at all respectively a chosen launch site
- GitHub URL

Predictive Analysis (Classification)

- We first standardise the data in the data frame and split it into training and test data
- For the 4 different models we fit the model on the training-data to find the best parameters using GridSearchCV
- Using the best parameters we compute the accuracy score of each model on the test-data
- For each model we also produced a confusion matrix to get a better feel for what kind of classification-errors each model had (false-negative vs false-positive)

- GitHub URL

Results

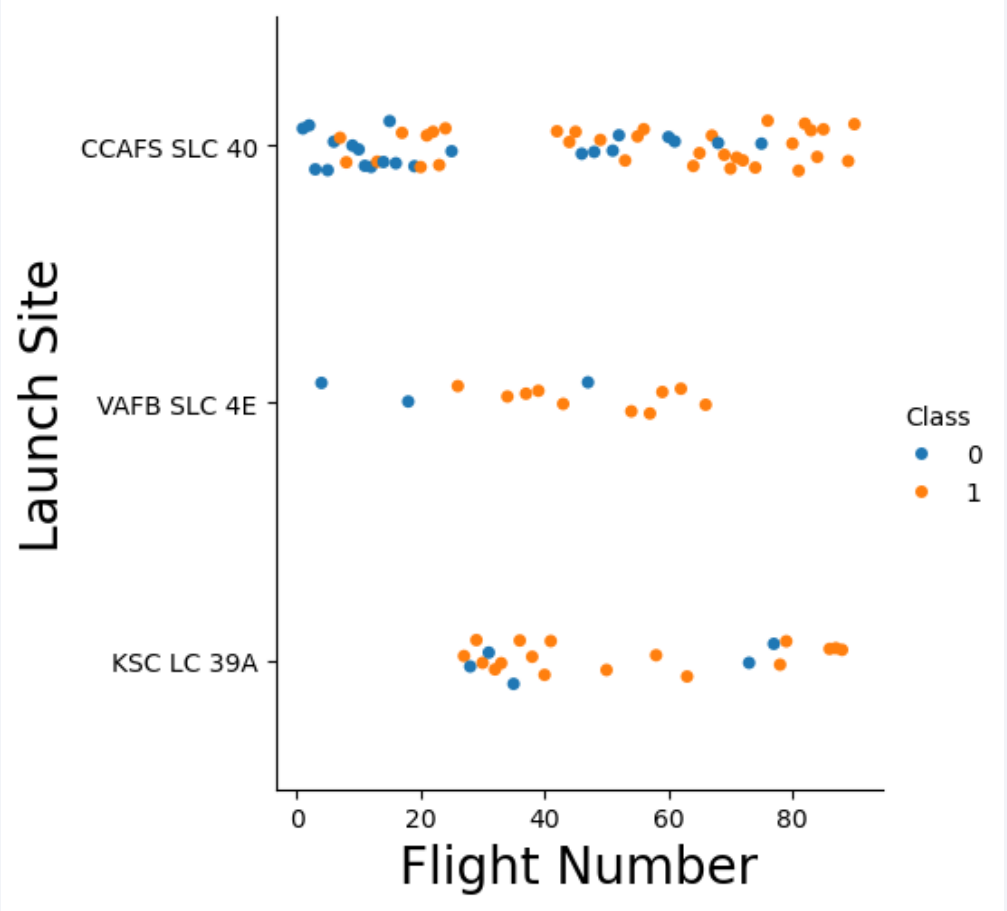
- EDA showed that flight number, orbit, payload mass, booster version and launch site were particularly relevant columns for predicting success
- The dashboard built allows the user to examine the relationship between different factors and success themselves
- The Decision Tree model could predict the outcome of a launch with 88.9% accuracy on the test-data

Section 2

Insights drawn from EDA

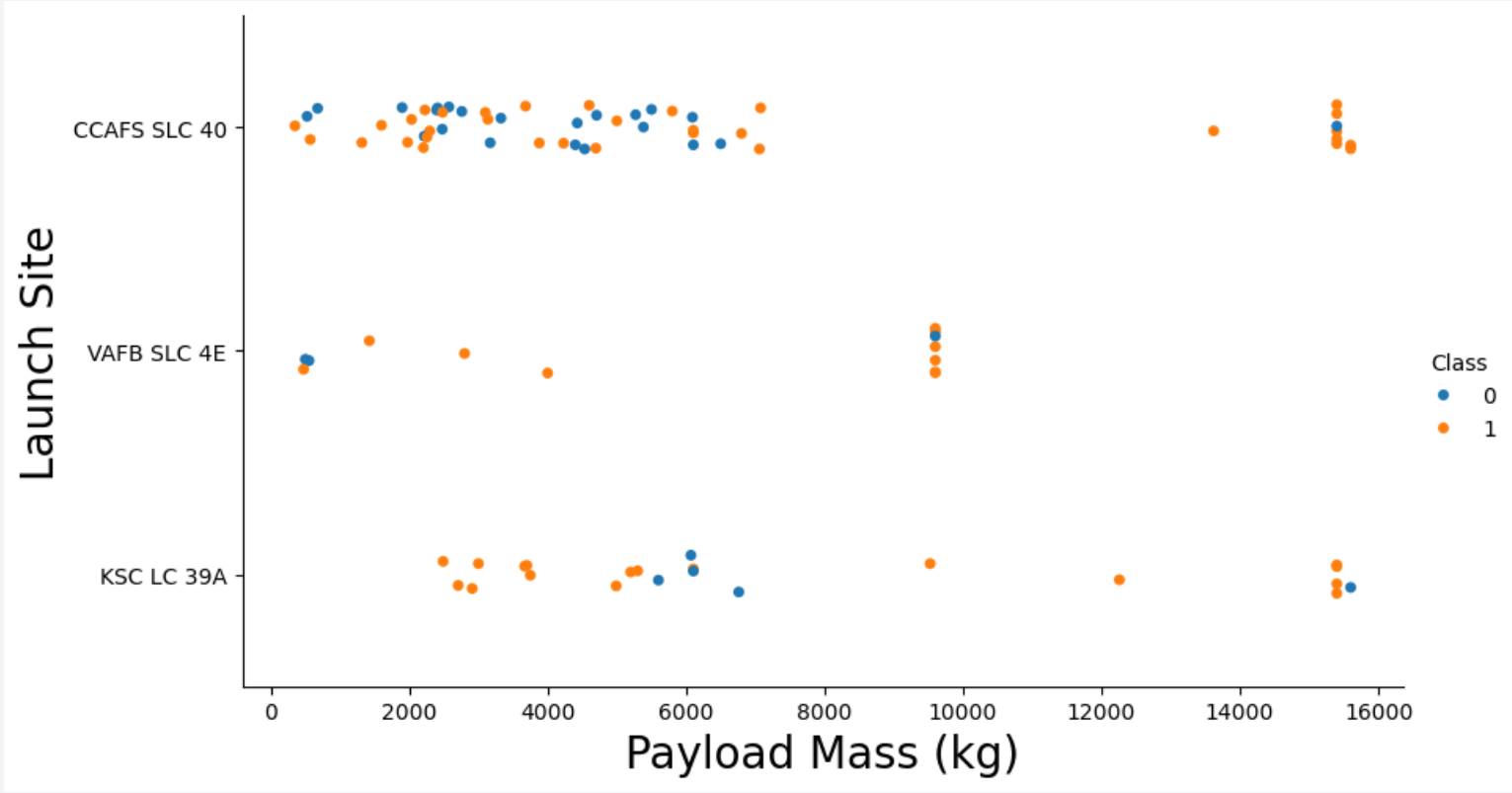
Flight Number vs. Launch Site

- Scatter plot of the Flight Number vs the Launch Site
- Shows that the higher the flight number the more likely the launch is successful at all of the 3 launch sites



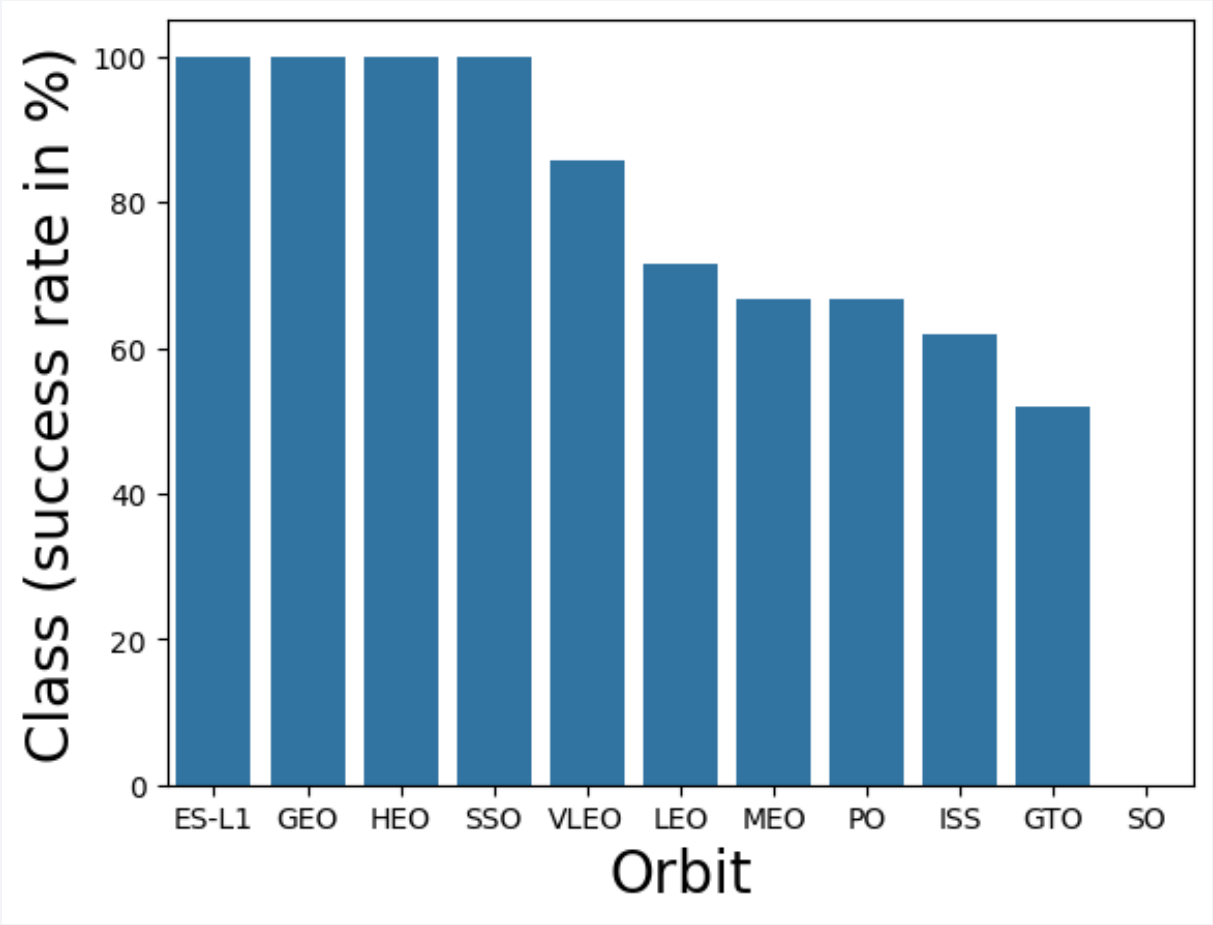
Payload vs. Launch Site

- Scatter plot of Payload Mass (kg) vs Launch Site
- Looks like higher Payload Mass increases the chance of success
- We also observe that at the Launch Site VAFB SLC 4E there are no launches with very heavy rockets



Success Rate vs. Orbit Type

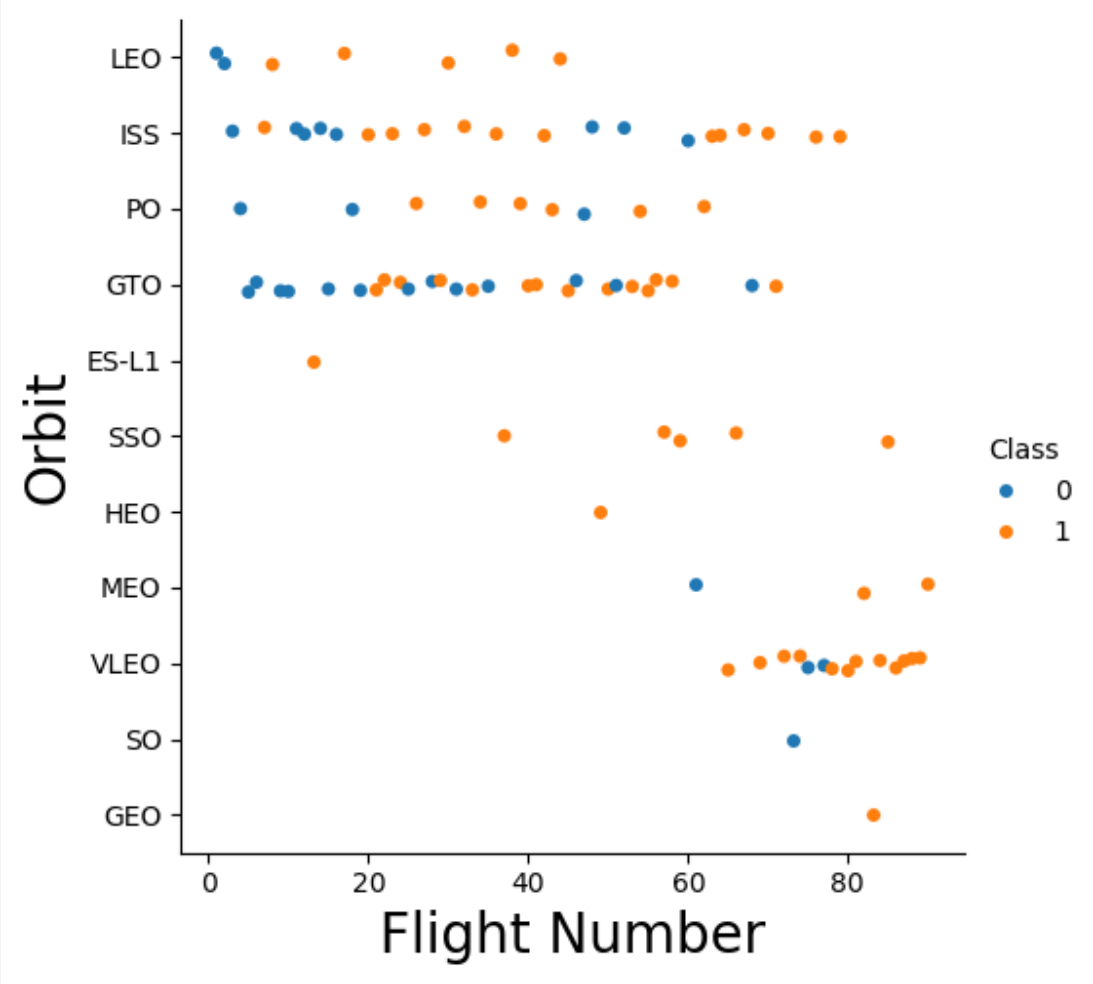
- Bar chart of the orbit vs the Success Rate
- The success rate of certain orbits is significantly higher than others
- Concretely, the orbits ES-L1, GEO, HEO and SSO have the highest chance of success



Flight Number vs. Orbit Type

- Scatter plot of Flight Number vs Orbit Type

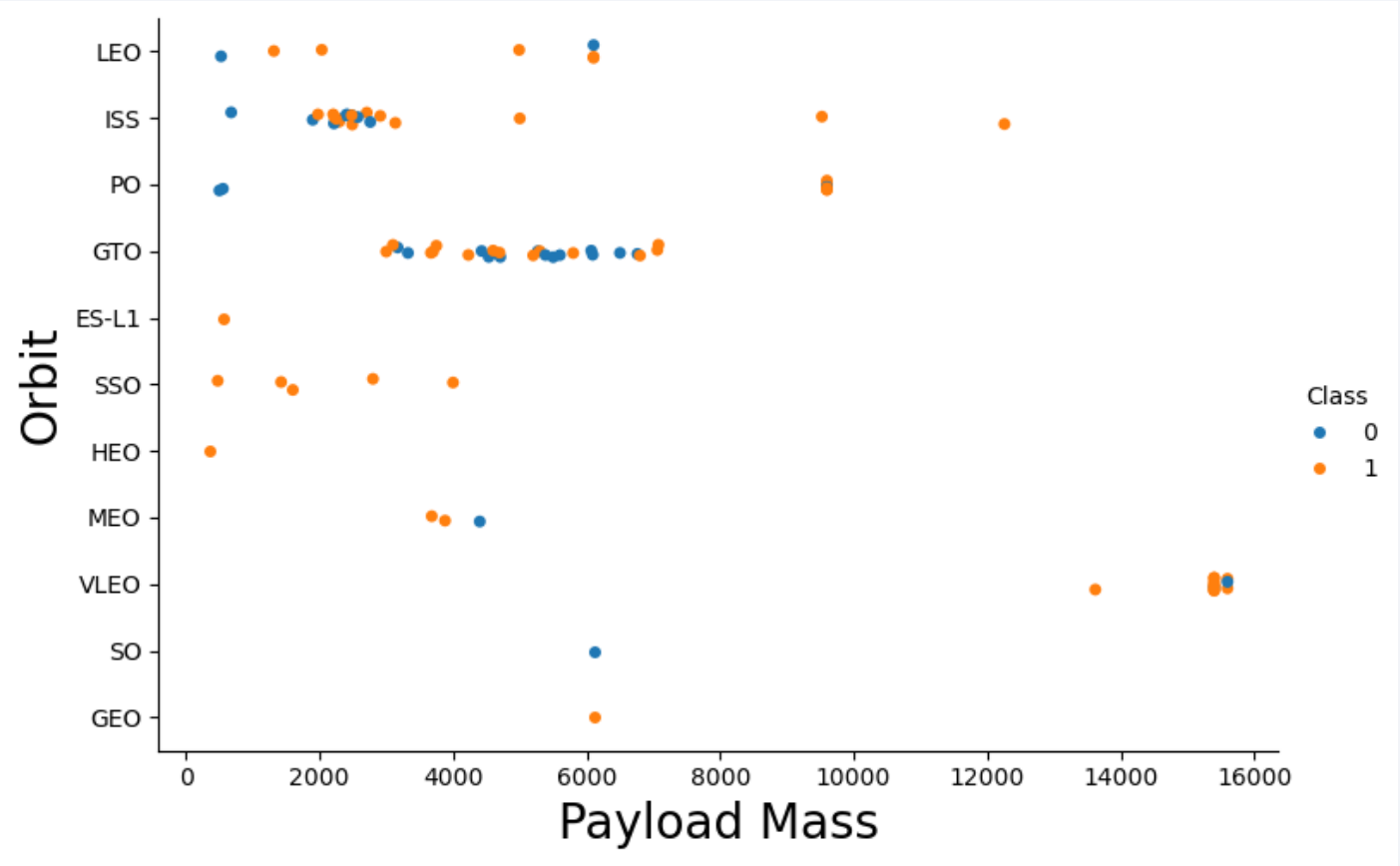
- Generally the higher the flight number the more likely the launch will succeed regardless of orbits
- Note that e.g. the GTO orbit does not quite fit the above pattern
- Some orbits also can be seen to be more successful; e.g. SSO does not have a single failed launch



Payload vs. Orbit Type

- Scatter plot of Payload Mass vs Orbit Type

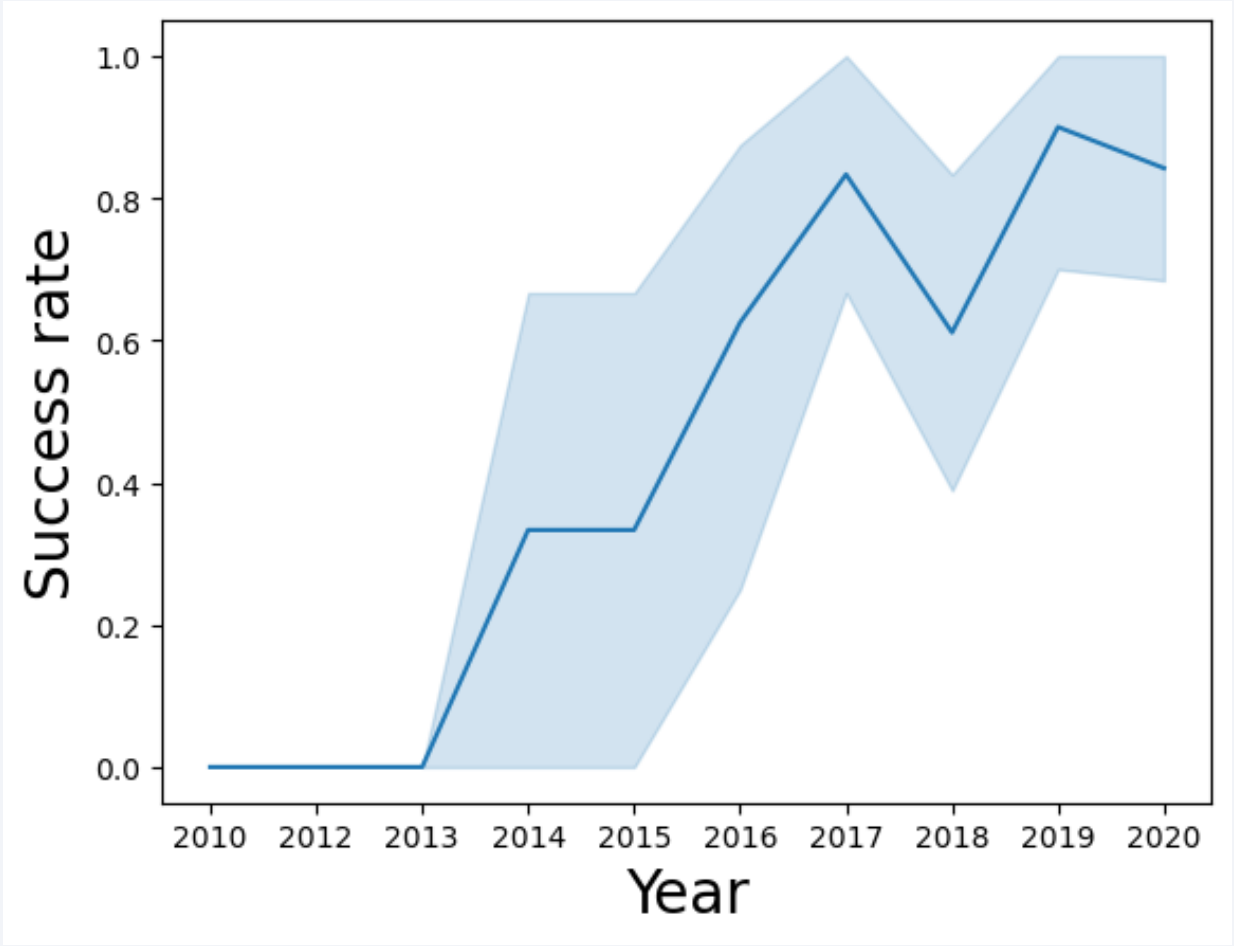
- For the orbits Polar, LEO and ISS we see that higher payload mass means higher success rate
- For e.g. the orbit GTO it is impossible to conclude a similar relationship



Launch Success Yearly Trend

- Line chart of the success rate over time

- We see that the success rate in general improved over time, with a slight dip in 2018



All Launch Site Names

- We found the names of the launch sites using the SELECT DISTINCT SQL query
- The table had 4 unique launch sites as seen in the screenshot

Task 1

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

```
[60]: %sql SELECT DISTINCT LAUNCH_SITE AS "Launch_Sites" FROM SPACEXTABLE;
```

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

[60]:

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

Launch Site Names Begin with 'CCA'

- We found 5 records where Launch site begins with CCA by using the code in the screenshot.
- Note in particular that we used LIKE ‘CCA%’ to allow for arbitrary text after the string

Task 2

Display 5 records where launch sites begin with the string 'CCA'

[61]:
```
%sql SELECT * FROM SPACEXTABLE WHERE LAUNCH_SITE LIKE 'CCA%' LIMIT 5;
```

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

[61]:

Date	Time (UTC)	Booster_Version	Launch_Site	Payload
2010-06-04	18:45:00	F9 v1.0 B0003	CCAFS LC-40	Dragon Spacecraft Qualification Unit
2010-12-08	15:43:00	F9 v1.0 B0004	CCAFS LC-40	Dragon demo flight C1, two CubeSats, barrel of Brouere cheese
2012-05-22	7:44:00	F9 v1.0 B0005	CCAFS LC-40	Dragon demo flight C2
2012-10-08	0:35:00	F9 v1.0 B0006	CCAFS LC-40	SpaceX CRS-1
2013-03-01	15:10:00	F9 v1.0 B0007	CCAFS LC-40	SpaceX CRS-2

Total Payload Mass from NASA

- For this task we used SUM() to add together the different Payload masses from launches where the customer was NASA
- The total payload mass from NASA was 45596kg

Task 3

Display the total payload mass carried by boosters launched by NASA (CRS)

```
[62]: %sql SELECT SUM(PAYLOAD_MASS__KG_) FROM SPACEXTABLE WHERE CUSTOMER = 'NASA (CRS)';
 * sqlite:///my_data1.db
Done.
```

[62]:

SUM(PAYLOAD_MASS__KG_)
45596

Average Payload Mass by F9 v1.1

- We calculated the average payload mass carried by booster version F9 v1.1 by using the AVG function on the entries of the table WHERE the booster version was F9 v1.1.
- The result is 2928.4kg

Task 4

Display average payload mass carried by booster version F9 v1.1

```
[63]: %sql SELECT AVG(PAYLOAD_MASS__KG_) FROM SPACEXTABLE WHERE BOOSTER_VERSION = 'F9 v1.1';
 * sqlite:///my_data1.db
Done.
```

[63]:

AVG(PAYLOAD_MASS__KG_)
2928.4

First Successful Ground Landing Date

- Using the MIN function we found the date (2015-12-22) of the first successful landing in ground pad - what a fantastic Christmas present for the company!

Task 5

List the date when the first succesful landing outcome in ground pad was acheived.

Hint:Use min function

```
[64]: %sql SELECT MIN(DATE) FROM SPACEXTABLE WHERE LANDING_OUTCOME = 'Success (ground pad)';
```

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

[64]:

MIN(DATE)
2015-12-22

Successful Drone Ship Landing with Payload between 4000 and 6000

- We used WHERE … AND to sort out only the successful drone ship landings with a payload between 4000 and 6000.
- There were 4 successful such landings by JCSAT-14, JCSAT-16, SES-10 and SES-10/EchoStar 105

Task 6

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

```
[65]: %sql SELECT PAYLOAD FROM SPACEXTABLE WHERE LANDING_OUTCOME = 'Success (drone ship)' AND PAYLOAD_MASS__KG_ BETWEEN 4000 AND 6000;
 * sqlite:///my_data1.db
Done.
```

[65]:

Payload
JCSAT-14
JCSAT-16
SES-10
SES-11 / EchoStar 105

Total Number of Successful and Failure Mission Outcomes

- We used COUNT(*) (only for the mission_outcome) grouped by mission_outcome to get the total number of the different outcomes.
- For example we see that we have 1 failure in flight

Task 7

List the total number of successful and failure mission outcomes

```
[66]: %sql SELECT MISSION_OUTCOME, COUNT(*) AS total_number FROM SPACEXTABLE GROUP BY MISSION_OUTCOME;
```

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

[66]:

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

Boosters Carried Maximum Payload

- We sorted through the spacextable by using WHERE and a subquery to find the max payload mass.
- There were 12 booster versions with maximal payload mass

Task 8

List the names of the booster_versions which have carried the maximum payload mass. Use a subquery

```
[67]: %sql SELECT BOOSTER_VERSION FROM SPACEXTABLE WHERE PAYLOAD_MASS__KG_ =(SELECT MAX(PAYLOAD_MASS__KG_) FROM SPACEXTABLE);
 * sqlite:///my_data1.db
Done.
```

[67]:

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

2015 Launch Records

- We used SUBSTR(“DATE”, 6,2) as month and picked the relevant columns (booster_version, launch_site, landing_outcome) WHERE the landing_outcome was failure (drone ship) and the year was 2015.
- We see that two launches fit this: one in January and one in April

Task 9

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.

Note: SQLLite does not support monthnames. So you need to use substr(Date, 6,2) as month to get the months and substr(Date,0,5)='2015' for year.

```
%sql SELECT SUBSTR("DATE",6,2) as month, "BOOSTER_VERSION", "LAUNCH_SITE", "LANDING_OUTCOME" FROM SPACEXTABLE WHERE "LANDING_OUTCOME" = 'Failure (drone ship)' AND substr("DATE", 0,5) = '2015';
```

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

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

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

- We ranked the landing outcomes between 2010-06-04 and 2017-03-20 in a descending manner (using DESC) as below.
- It can be seen that e.g. the most common landing_outcome was no attempt (10), followed by success (drone ship) and failure (drone ship) (both with 5)

Task 10

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, in descending order.

[78]:
```
%sql SELECT [Landing_Outcome], COUNT(*) as count_outcomes FROM SPACEXTABLE WHERE DATE BETWEEN '2010-06-04' AND '2017-03-20' GROUP BY [Landing_Outcome] ORDER BY count_outcomes DESC;
```

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

[78]:

Landing_Outcome	count_outcomes
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

Section 3

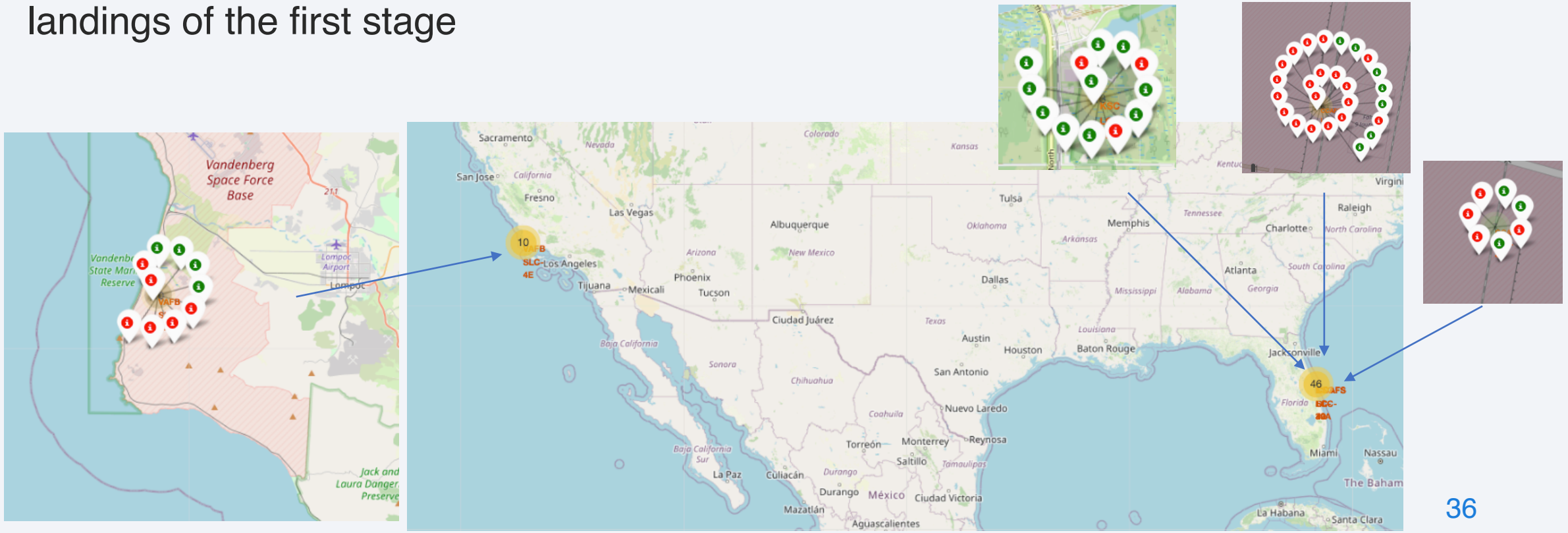
Launch Sites Proximities Analysis

Map of Launch Sites

- The generated Folium map highlights the different rocket launch sites
- We see that all of the launch sites are close to the equator
- It is also clear that all of the launch sites are close to the coast

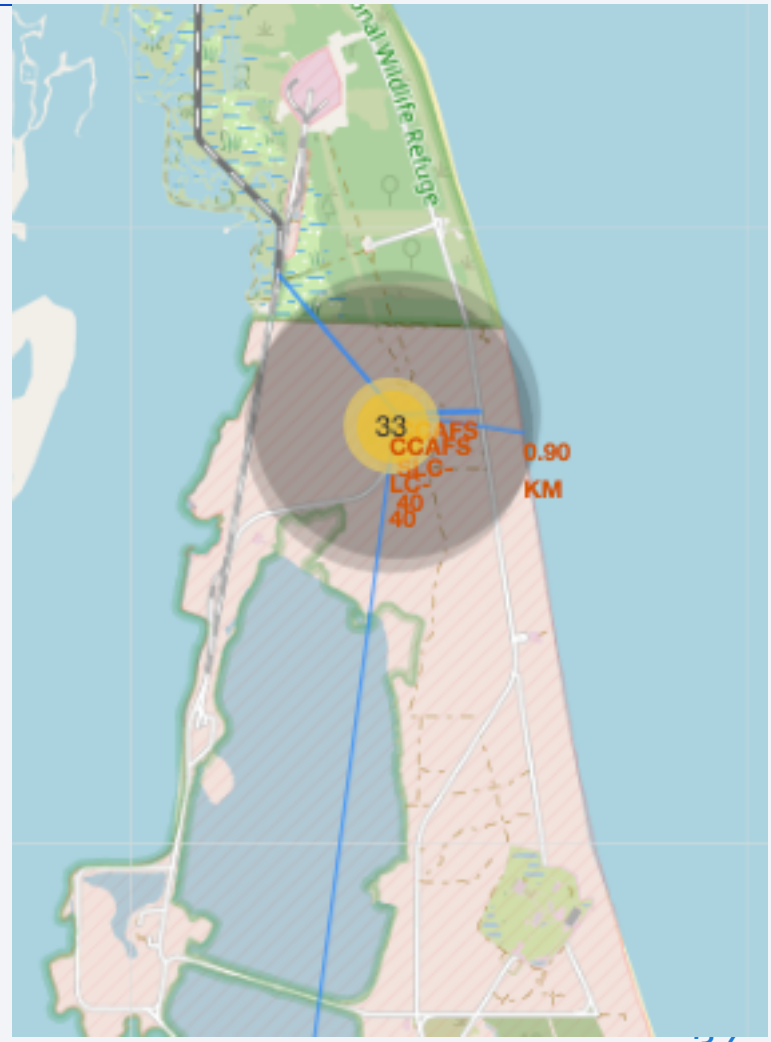
Map including Launch Outcomes

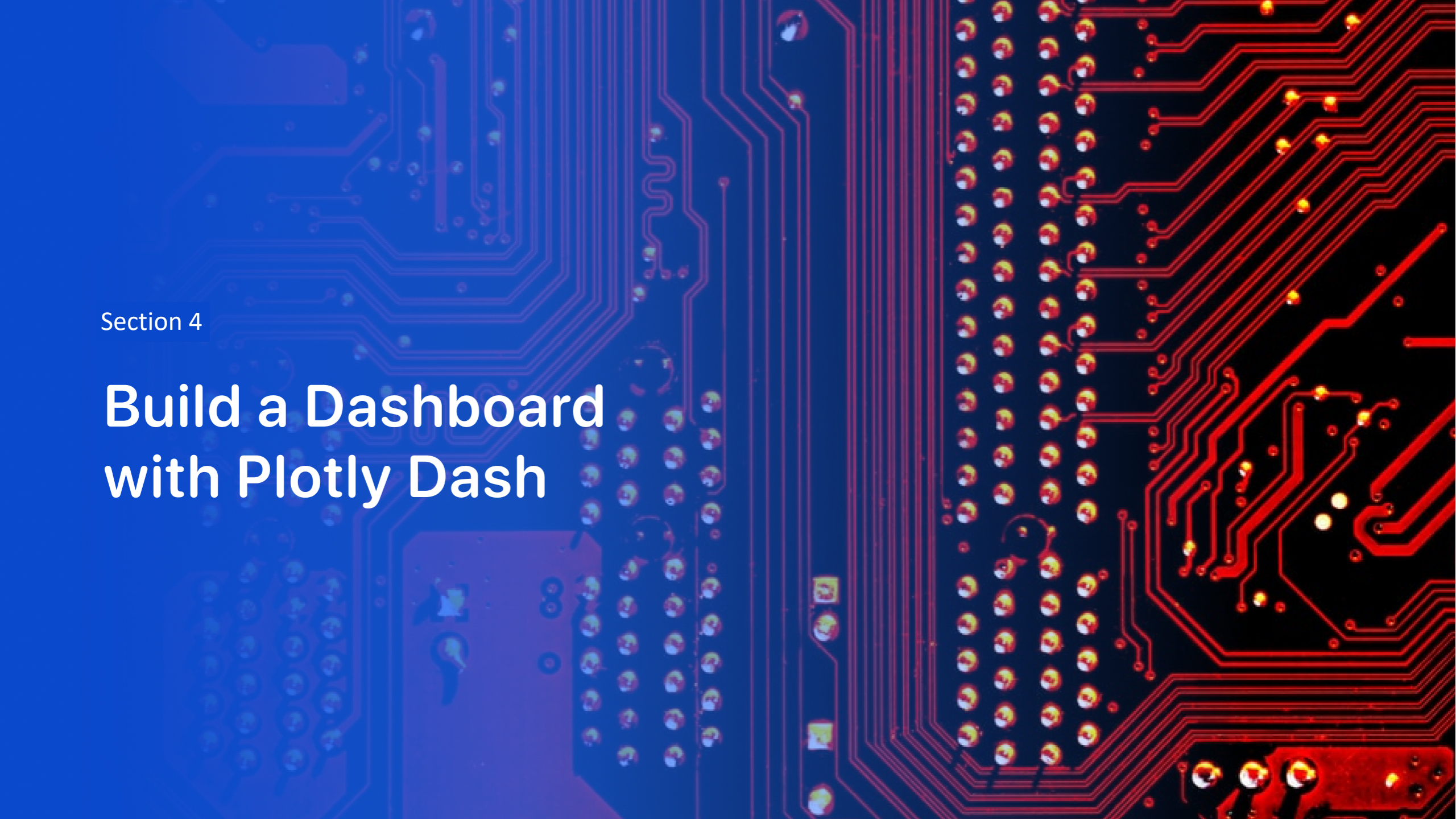
- We have added colour-coded markers for each launch at the different launch sites: green means successful launch while red is unsuccessful
- For example we see that the launch site KSC LC-39A has many successful landings of the first stage



Distance to the ocean and important infrastructure

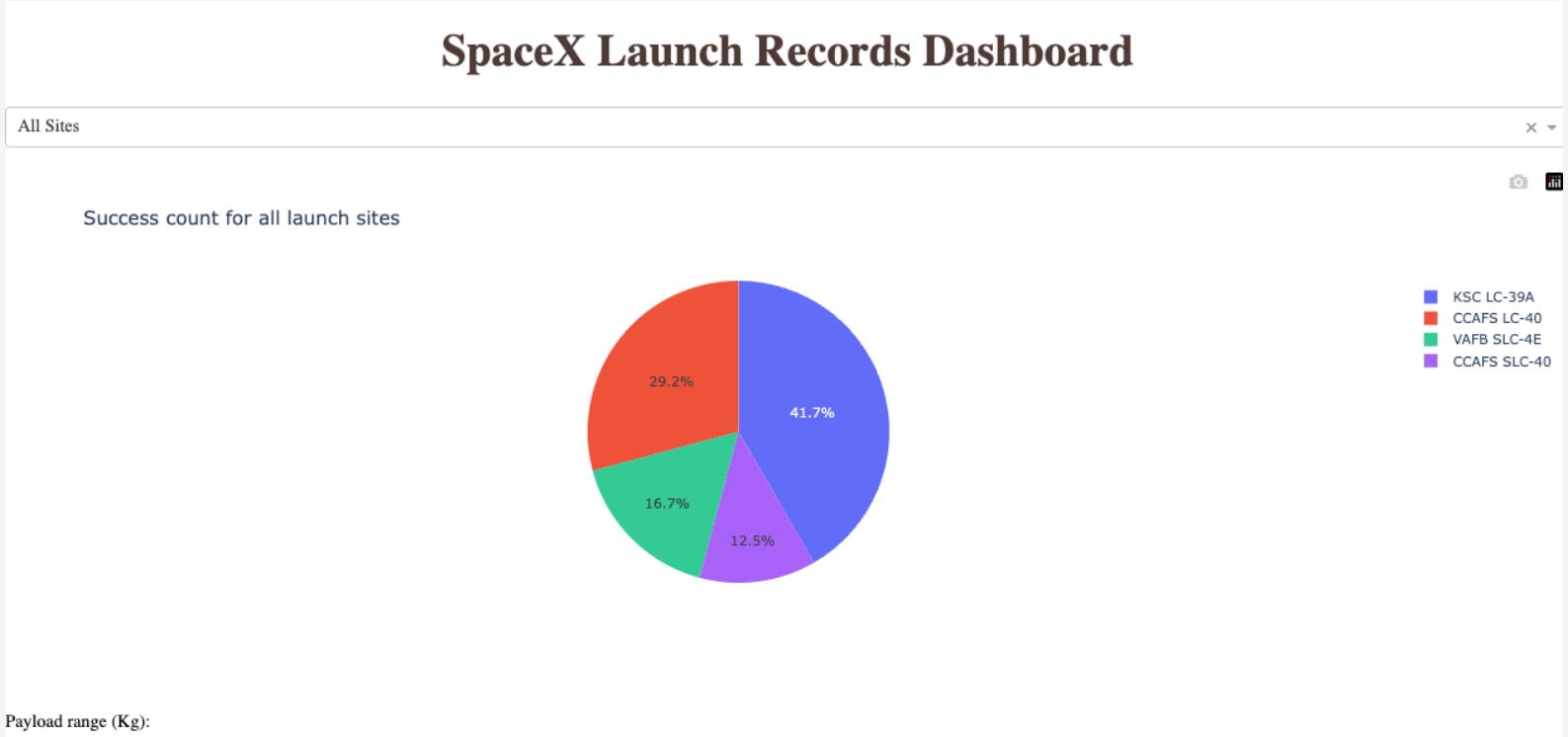
- We see that the chosen launch site is close to both the ocean, a highway and a railroad
- It is moderately distanced from the closest city




Section 4
Build a Dashboard with Plotly Dash

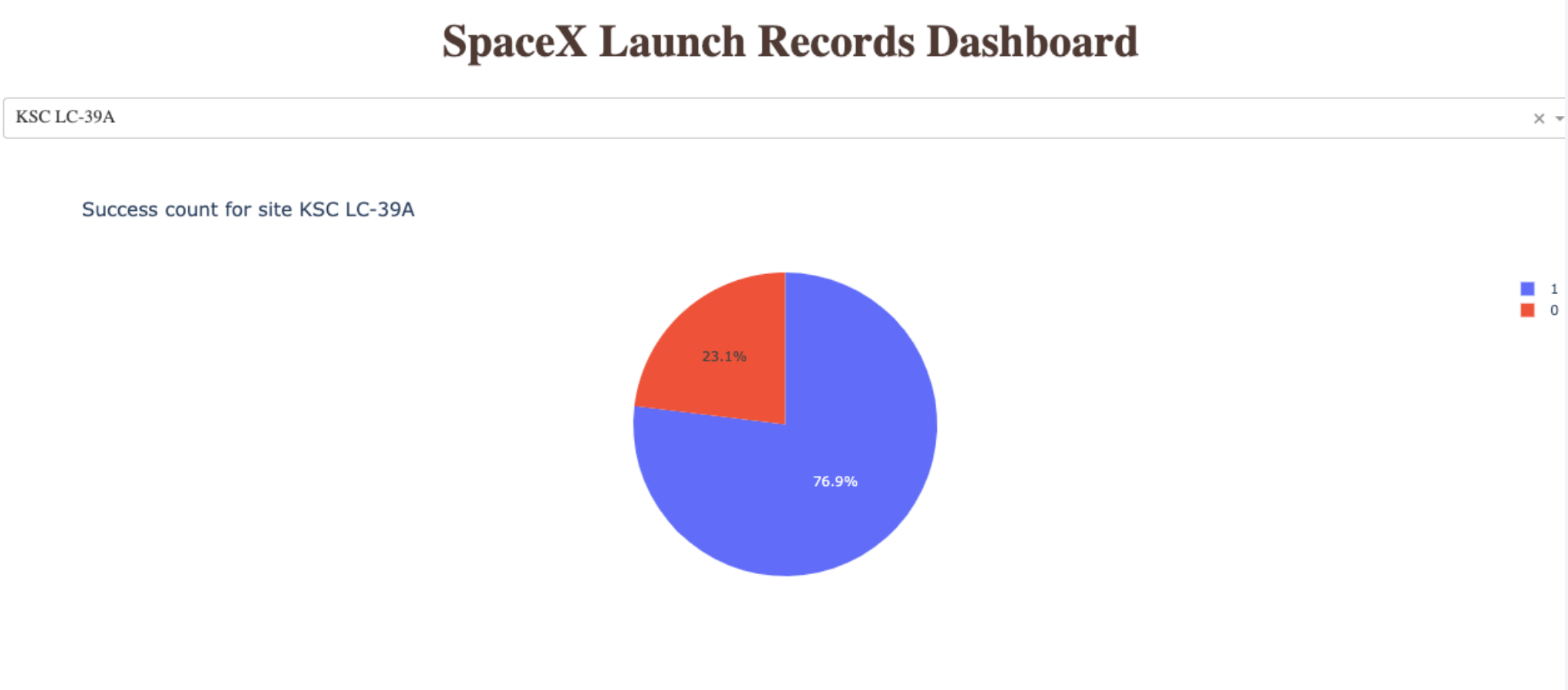
Launch Success for all Launch Sites

- The screenshot of the launch success count for all sites (as a piechart) shows that the launch site with the highest success count is KSC LC-39A



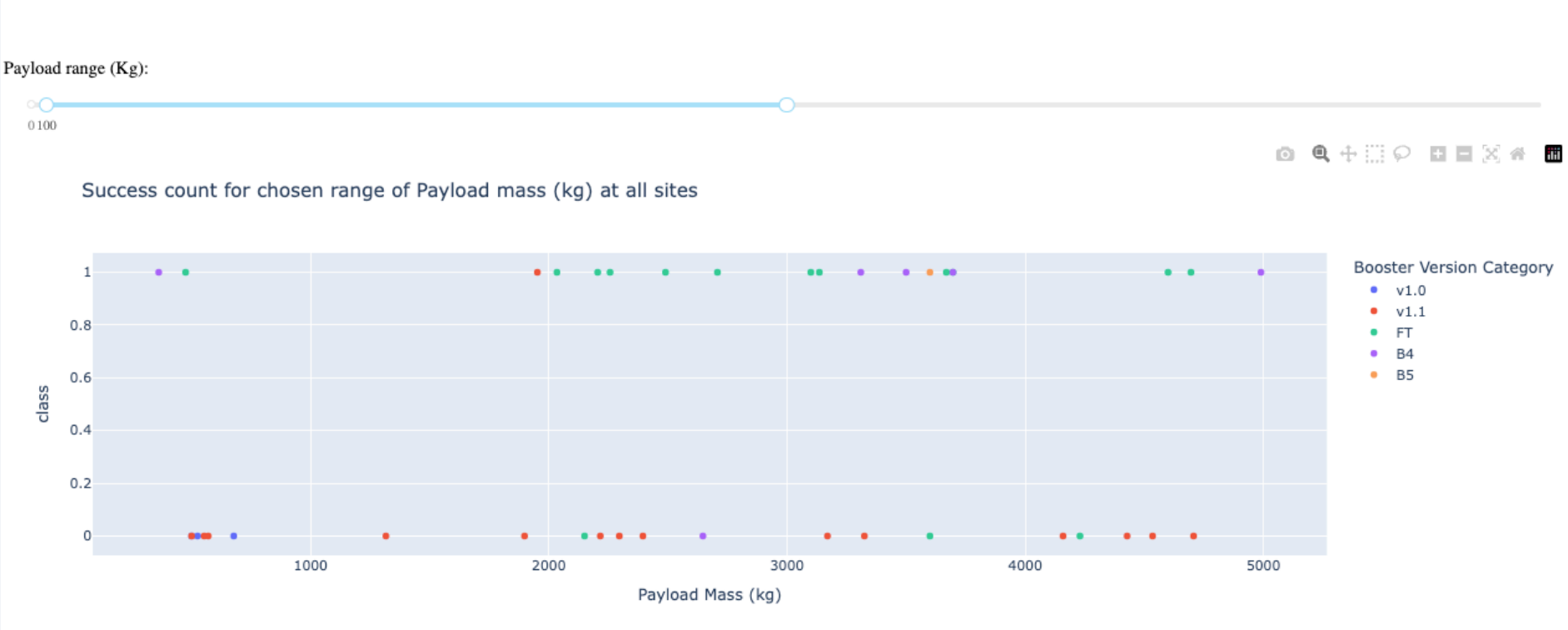
The most successful launch sites

- The piechart for the launch site with the highest success ratio is shown below
- This site is KSC LC-39A



Success count vs Payload

- For payload between 100kg and 5000kg we see the Launch outcome scatter plot for all sites from the Dashboard below
- We see that for example the booster version FT (green) has a lot of successful launches, while many of the launches with booster version v1.1 are unsuccessful



Section 5

Predictive Analysis (Classification)

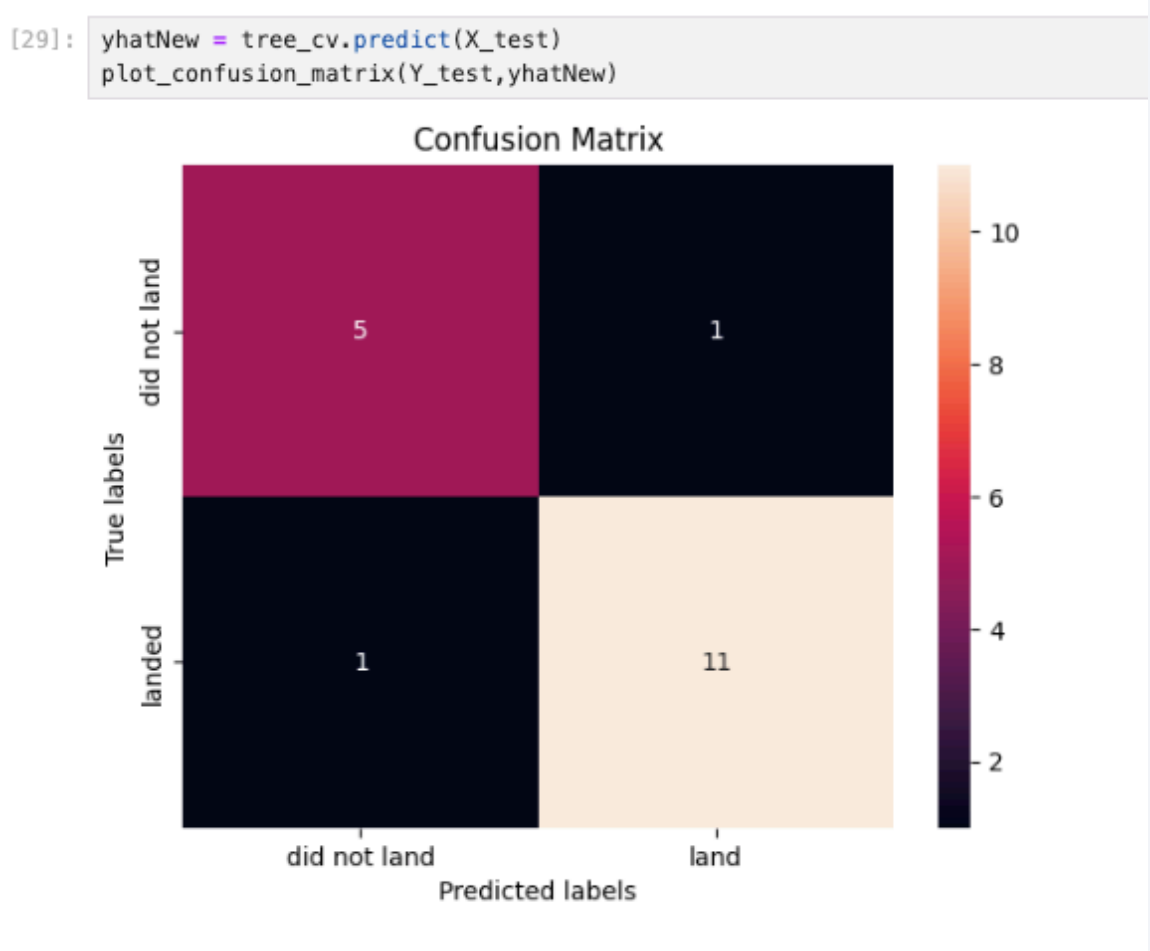
Classification Accuracy

- The screenshot shows the test data accuracy for each of the 4 models we considered in this lab
- We see that the **Decision Tree** performed the best

Method	Test Data Accuracy
Logistic Regression	0.833333
SVM	0.833333
Decision Tree	0.888889
KNN	0.833333

Confusion Matrix of Decision Tree

- The confusion matrix of the decision tree model shows that the model correctly predicted 11 landings and 5 non-landings
- Moreover, it only wrongly identified one landing as unsuccessful and one non-landing as successful



Conclusions

- The launch sites with higher flight number seems to be more successful
- The launch success rate increased with time from 2013 to 2020, with a smaller dip at the year 2018
- The orbits ES-L1, GEO, HEO and SSO have the highest chance of success
- The launch site KSC LC-39A is the most successful launch site
- The best machine learning algorithm for this task was the Decision Tree

Thank you!