Winning Space Race with Data Science

<Name>
<Date>

Outline

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

Executive Summary

- Exploratory data analysis, interactive visual analytics, dashboard and predictive analysis were used on this study
- A subset of variables can be used to predict if the Falcon 9´s first stage will land

Introduction

- Space X 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 Space X can reuse the first stage.
- The problem that we need to answer is if the first stage will land successfully

Section 1

Methodology

Methodology

Executive Summary

- Data collection methodology:
  - Data were collected thought a Restful API and web scraping
- Perform data wrangling
  - PayloadMass missing values were replaced by PayloadMass mean and A variable that represents the outcome of each launch was created
- Perform exploratory data analysis (EDA) using visualization and SQL
- Perform interactive visual analytics using Folium and Plotly Dash
- Perform predictive analysis using classification models
  - Different models were used and tested

Data Collection

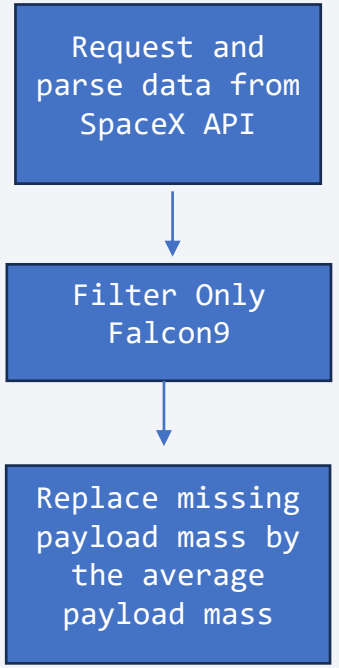
- The rocket launch data were collected from SpaceX API https://api.spacexdata.com/v4 by multiple calls to get all the needed information, include ID resolutions

- Data were also WebScraped from `List of Falcon 9 and Falcon Heavy launches` Wikipage

Data Collection – SpaceX API

- The rocket launch data were collected from SpaceX API

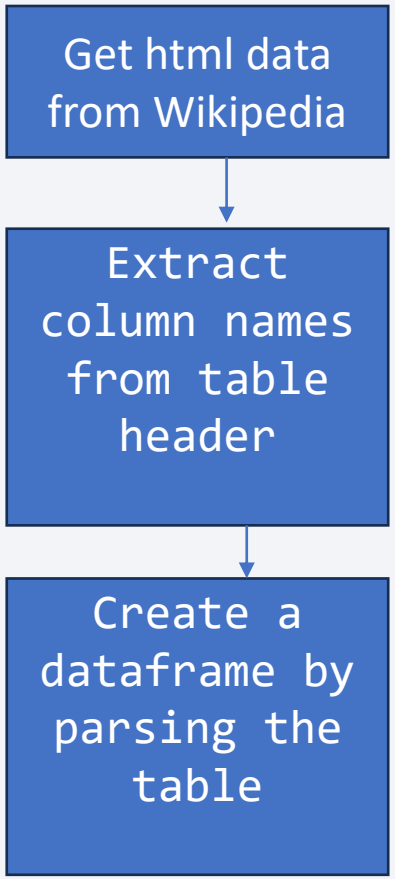
- https://github.com/rafaelwrrn3/datasciencecapstone/blob/main/jupyter-labs-spacex-data-collection-api.ipynb



Data Collection - Scraping

- Data were WebScraped from `List of Falcon 9 and Falcon Heavy launches` Wikipage

- https://github.com/rafaelwrrn3/datasciencecapstone/blob/main/jupyter-labs-webscraping.ipynb



Data Wrangling

- Calculated the number of launches on each site, the number and occurrence of each orbit and the number and occurence of mission outcome of the orbits

- PayloadMass missing values were replaced by PayloadMass mean

- A variable that represents the outcome of each launch was created

- https://github.com/rafaelwrrn3/datasciencecapstone/blob/main/labs-jupyter-spacex-Data%20wrangling.ipynb

EDA with Data Visualization

- Plotted a scatter point chart to visualize the relationship between Flight Number and Launch Site
- Plotted a scatter point chart to visualize the relationship between Payload and Launch Site
- Plotted a bar chart to visualize the relationship between success rate of each orbit type
- Plotted a scatter point chart to visualize the relationship between FlightNumber and Orbit type
- Plotted a bar chart to visualize the relationship between Payload and Orbit type
- Plotted a line chart to visualize the launch success yearly trend
- https://github.com/rafaelwrrn3/datasciencecapstone/blob/main/jupyter-labs-eda-dataviz.ipynb

EDA with SQL

- Query performed to display average payload mass carried by booster version F9 v1.1
- Query performed to List the date when the first succesful landing outcome in ground pad was achieved
- Query performed to List the names of the boosters which have success in drone ship and have payload mass greater than 4000 but less than 6000
- Query performed to List the total number of successful and failure mission outcomes
- Query performed to List the names of the booster_versions which have carried the maximum payload mass.
- Query performed to 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
- Query performed to 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
- https://github.com/rafaelwrrn3/datasciencecapstone/blob/main/jupyter-labs-eda-sql-coursera_sqllite.ipynb

Build an Interactive Map with Folium

- Created a Circle and a Marker for each launch site, created markers and markers clusters for each launch result, added a `MousePosition` on the map to get coordinate for a mouse over a point on the map, created a marker in a coastline point and draw a polyline between launch site and selected coastline point
- Objects were added to check if launch success rate depend on the location and proximities of a launch site, i.e., the initial position of rocket trajectories.
- https://github.com/rafaelwrrn3/datasciencecapstone/blob/main/lab_jupyter_launch_site_location.ipynb

Build a Dashboard with Plotly Dash

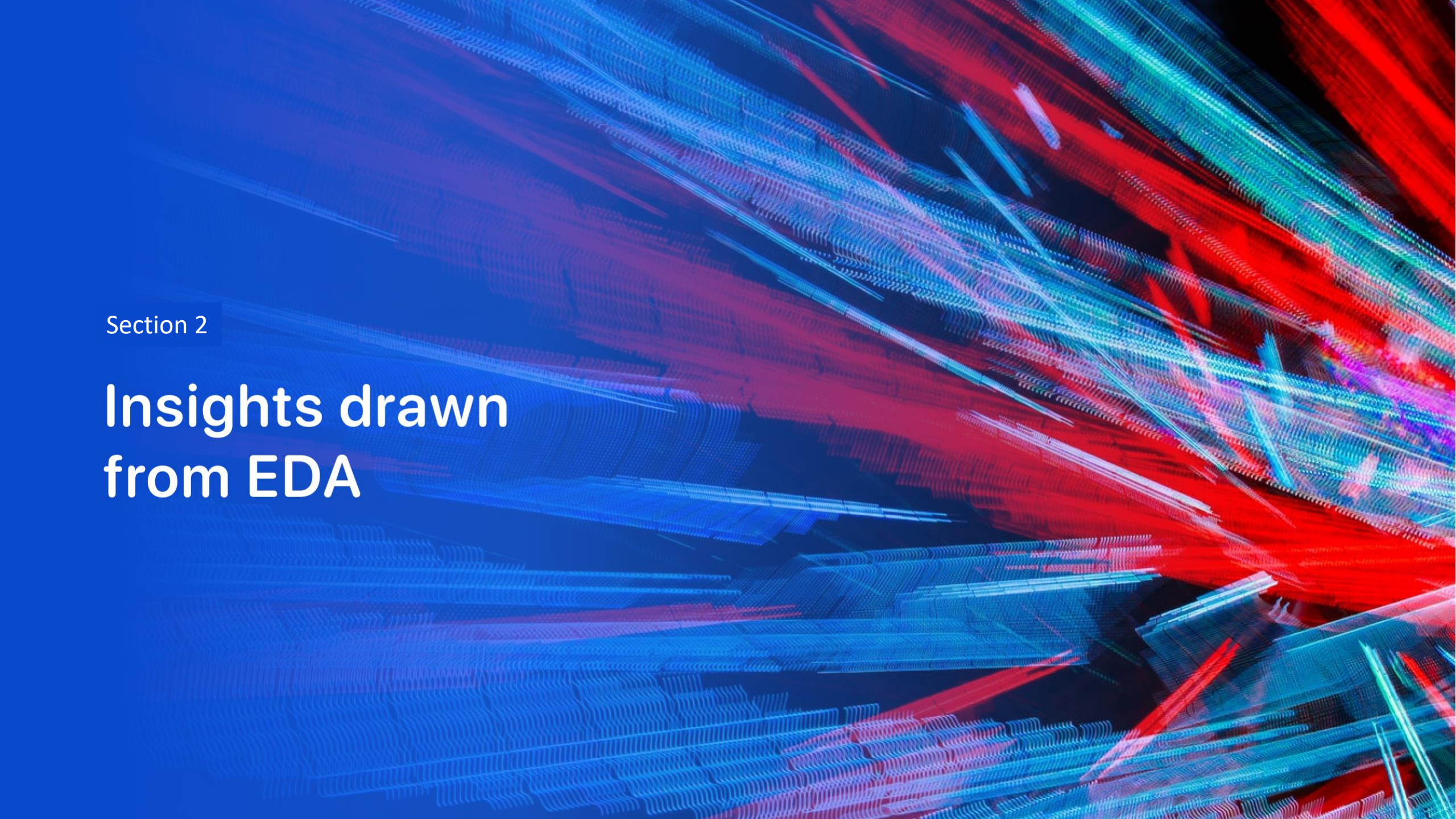
- Plots and interactions added: dropdown list to enable Launch Site selection, pie chart to show the total successful launches count for all sites or to show the Success vs. Failed counts for the site, if a specific launch site was selected, slider to select payload range and scatter chart to show the correlation between payload and launch success
- The plots and interactions were added for users to perform interactive visual analytics on SpaceX launch data in real-time. So that they can obtain some insights, like which F9 Booster version has the highest launch success rate
- https://github.com/rafaelwrrn3/datasciencecapstone/blob/main/spacex_dash_app.py

Predictive Analysis (Classification)

- Data were standardized, splitted into training and test data, GridSearchCV was used to find the best parameters, accuracy on train data and on test data, for 4 diferents classification models: logistic regression, support vector machine, decision tree classifier and k nearest neighbors.
- Confusion matrix was plotted for the 4 models.
- The accuracy with train and test data and the confusion matrix were used to find the best performing model
- https://github.com/rafaelwrrn3/datasciencecapstone/blob/main/module_4_SpaceX_Machine_Learning_Prediction_Part_5.jupyterlite.ipynb

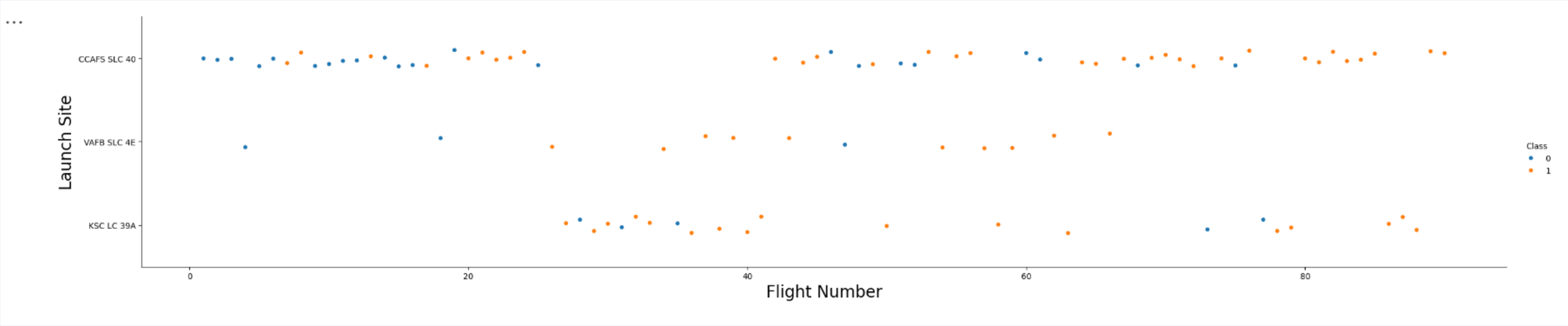
Results

- In the LEO orbit the Success appears related to the number of flights; on the other hand, there seems to be no relationship between flight number when in GTO orbit
- In the VAFB-SLC launchsite there are no rockets launched for heavypayload mass
- With heavy payloads the successful landing or positive landing rate are more for Polar,LEO and ISS
- Success rate increases over time
- Launch sites usually are close to the Equator line and to coastline
- The best classification model is the decision tree


Section 2
Insights drawn from EDA

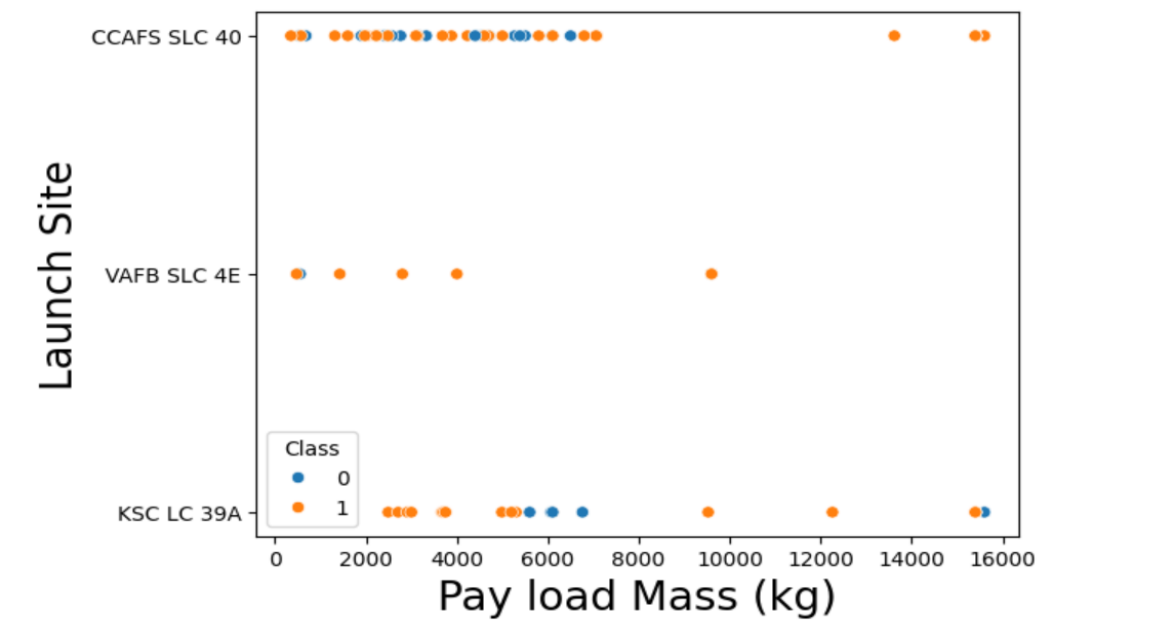
Flight Number vs. Launch Site

- CCAFS SLC 40 was the most used launch site. Its success rate improved over time
- KSC LC 35A has the best success rate



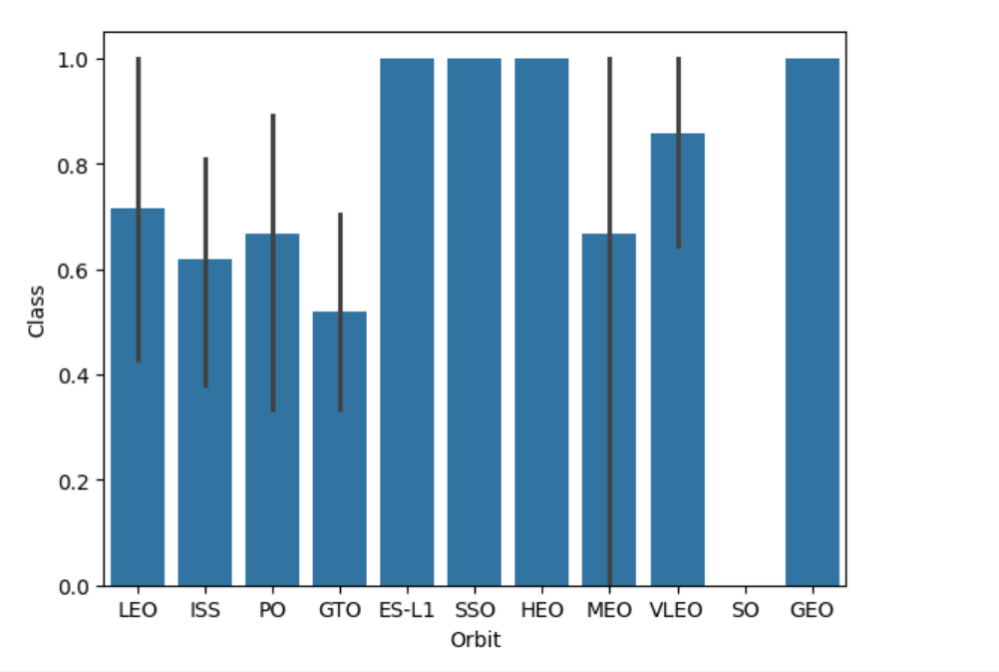
Payload vs. Launch Site

- VAFB-SLC launchsite there are no rockets launched for heavypayload mass(greater than 10000).
- CCAFS SLC 40 has high success rate for heavy payload mass



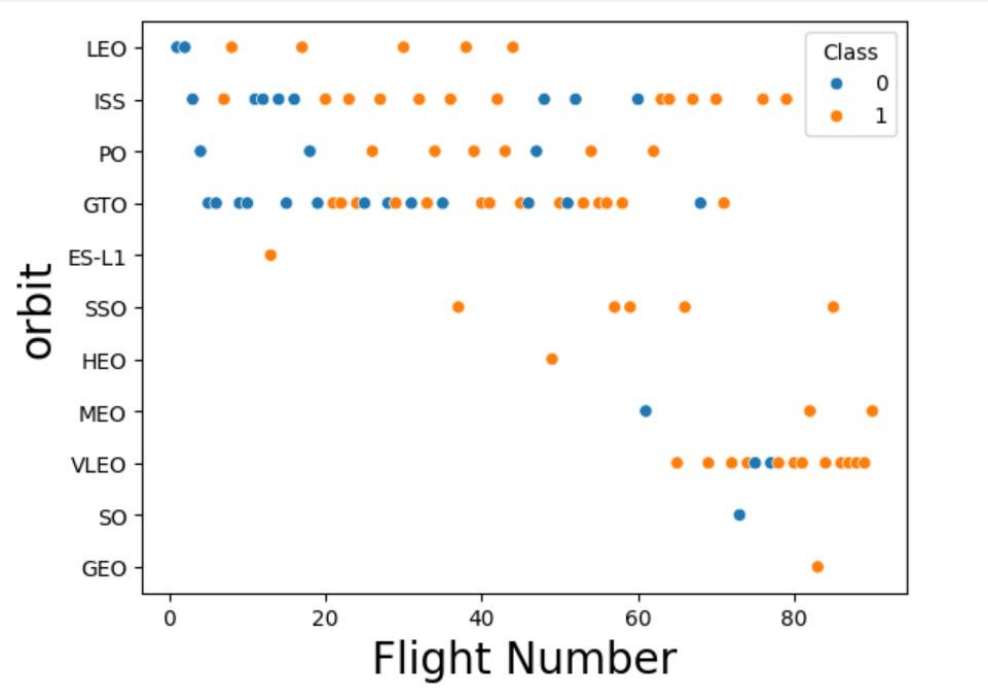
Success Rate vs. Orbit Type

- ES-L1, SSO, HEO and GEO have the highest success rate
- SO has the lowest success rate



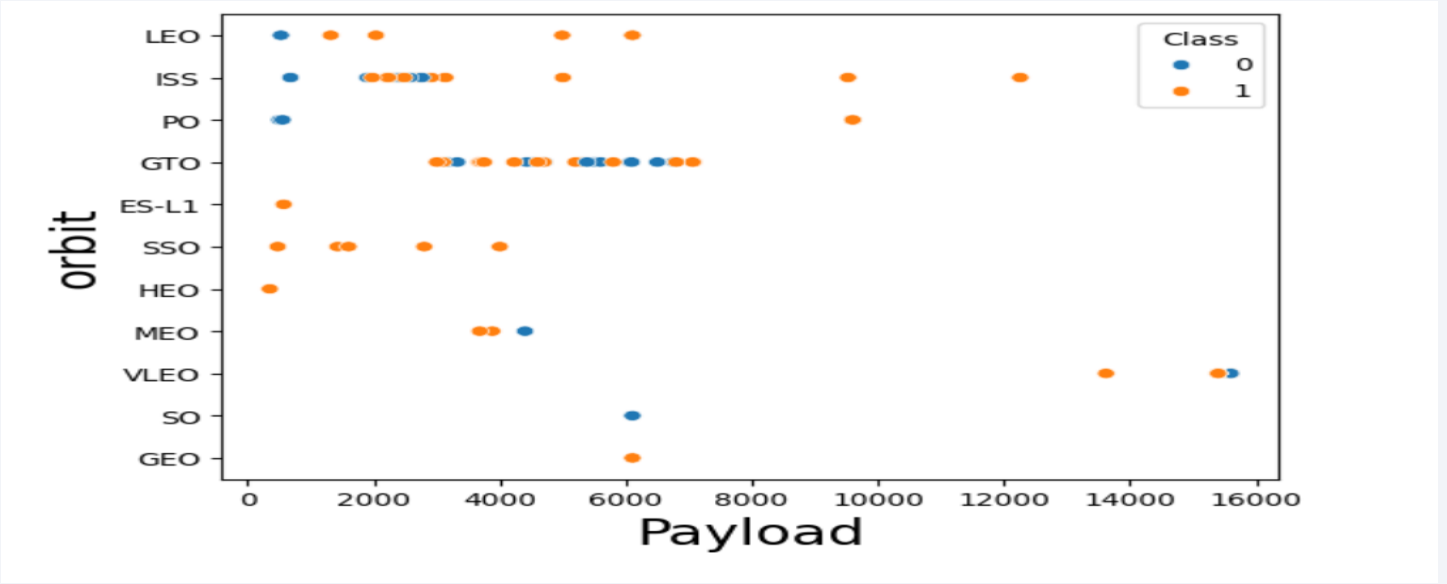
Flight Number vs. Orbit Type

- LEO orbit the Success appears related to the number of flights
- On the other hand, there seems to be no relationship between flight number when in GTO orbit



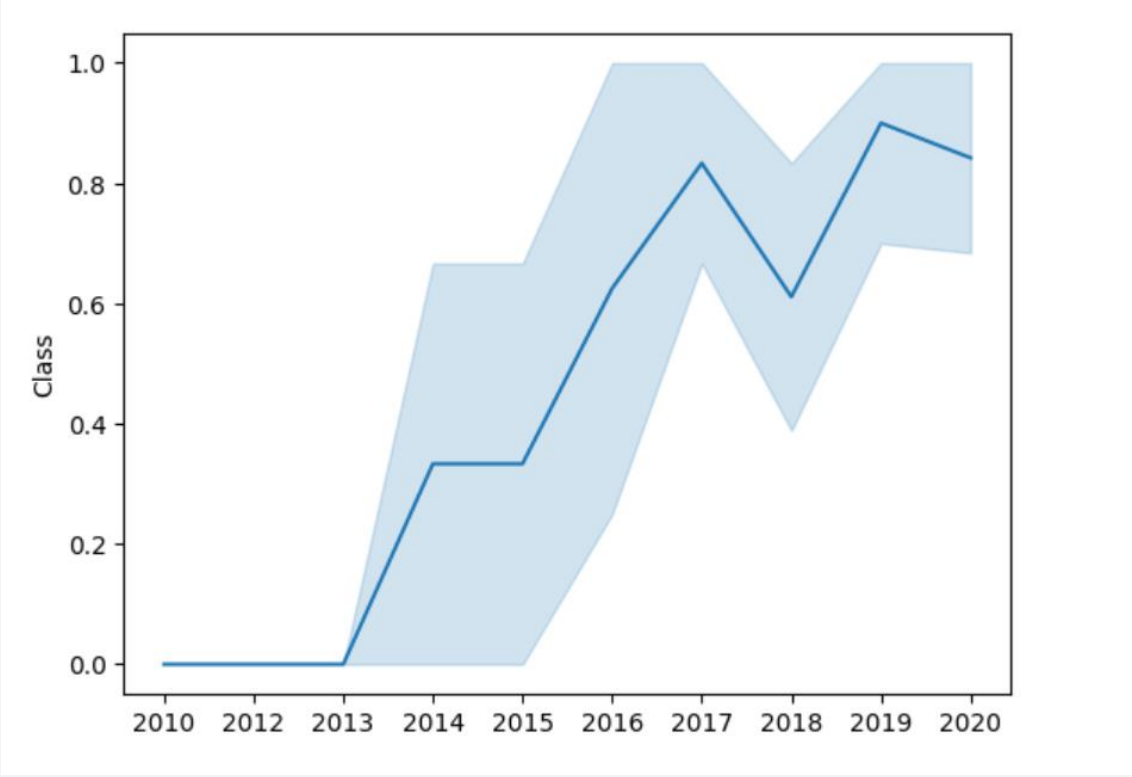
Payload vs. Orbit Type

- With heavy payloads the successful landing or positive landing rate are more for Polar,LEO and ISS
- For GTO we cannot distinguish this well as both positive landing rate and negative landing(unsuccessful mission) are both there here



Launch Success Yearly Trend

- Success rate increased over time



All Launch Site Names

```
%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

- The query was implemented with the use of select distinct to get the unique launch sites names

Launch Site Names Begin with 'CCA'

```
%sql select * from SPACEXTBL where "Launch_Site" like "CCA%" limit 5
```

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

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

- The query was done with the use of the “limit” keyword to get just 5 records and regular expression comparation with the “like” keyword to get the launch sites starting with “CCA”

Total Payload Mass

```
%sql select sum("PAYLOAD_MASS__KG_") from SPACEXTBL where "Customer" like "NASA%"
```

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

sum("PAYLOAD_MASS__KG_")
99980

- The query was implemented with the use of regular expression comparation with the keyword “like” to get only Nasa customer and the use of the sum to get the total payload mass

Average Payload Mass by F9 v1.1

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

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

avg("PAYLOAD_MASS__KG_")
2534.6666666666665

- The query was build with the use of regular expression comparation to get only F9 V1.1 booster and avg to calculate the average mass

First Successful Ground Landing Date

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

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

min("Date")
2015-12-22

- The query were build filtering only the Success (ground pad) landing outcome with the where clause and using the min(“Date”) to get the first occurrence

Successful Drone Ship Landing with Payload between 4000 and 6000

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

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

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

- List the names of boosters which have successfully landed on drone ship and had payload mass greater than 4000 but less than 6000

- The query was built with the use of the where clause to filter only successful drone ship landing outcome and payload mass between 4000 and 6000

Total Number of Successful and Failure Mission Outcomes

```
%sql select count(*)  from SPACEXTBL where "Landing_Outcome" like "%Success%"
```

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

count(*)
61

```
%sql select count(*)  from SPACEXTBL where "Landing_Outcome" like "%Failure%"
```

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

count(*)
10

- Both queries were built with the use of regular expression with the like clause to get the success or failure landing outcomes and the use of the count(*) to count the ocurrences

Boosters Carried Maximum Payload

```
%sql select "Booster_Version" from SPACEXTBL where "PAYLOAD_MASS__KG_" == (select max("PAYLOAD_MASS__KG_") from SPACEXTBL)
```

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

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 was build with the use of a subquery to get the max payload mass and then filtering the booster versions that carried this maximum payload

2015 Launch Records

```
%sql select substr(Date, 6,2) as "Month", "Landing_Outcome","Booster_Version", "Launch_Site"  from SPACEXTBL
where "Landing_Outcome"=="Failure (drone ship)" and substr(Date,0,5) == "2015"
```

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

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

- The query was built with the use the substr to get the month and the year of the Date and filtering the failure drone ship landing outcomes and the 2015 year with the where clause

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

```
%sql select "Landing_Outcome", count(*) as contagem  from SPACEXTBL where "Date" between "2010-06-04" and "2017-03-20"
group by "Landing_Outcome" order by "contagem" desc
```

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

Landing_Outcome	contagem
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

- The query was done with the use of the group by clause to group the records by the count and the dates where filtered with the where clause


Section 3
Launch Sites
Proximities Analysis

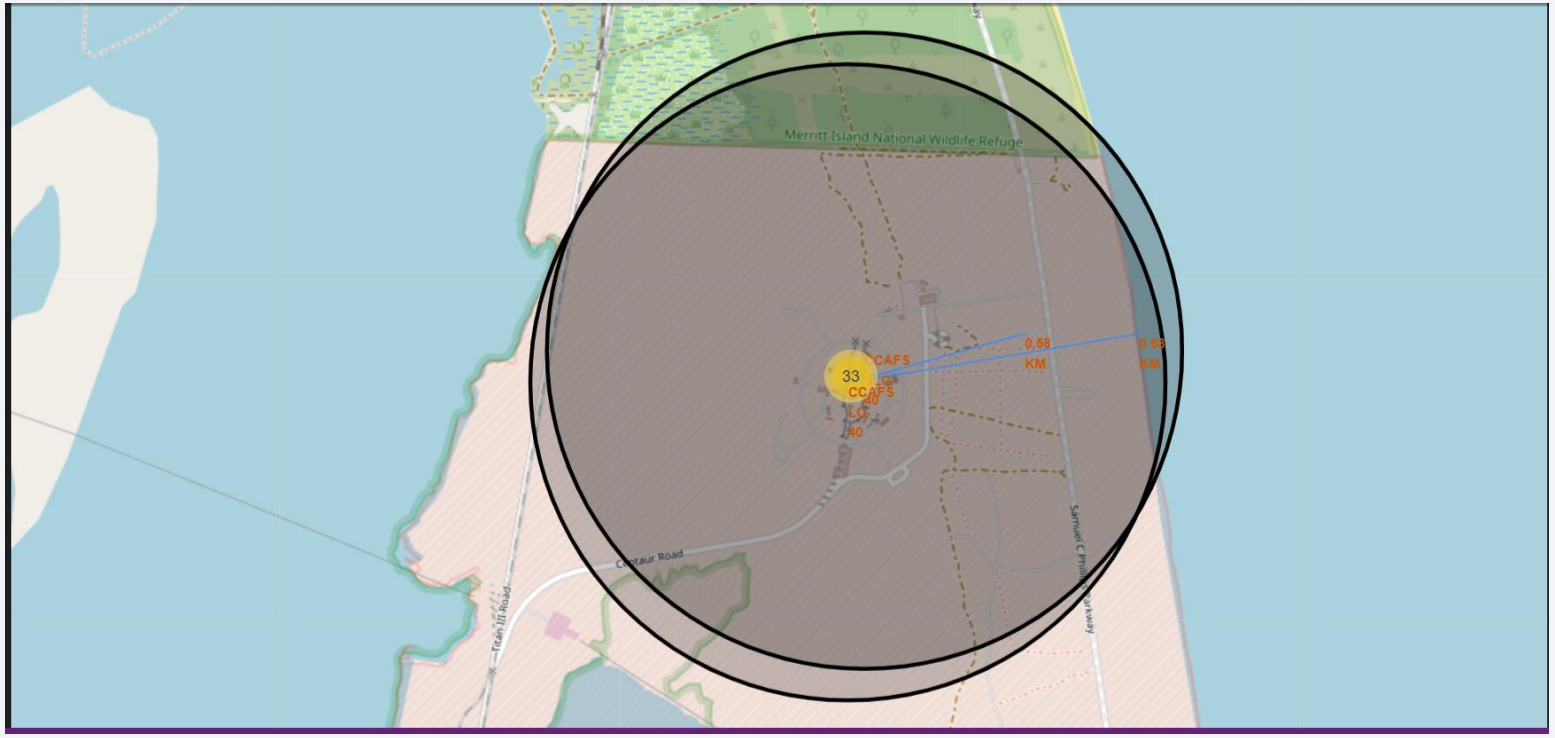
Map with all launch sites

- All launch sites are close to the Equator and to the coast

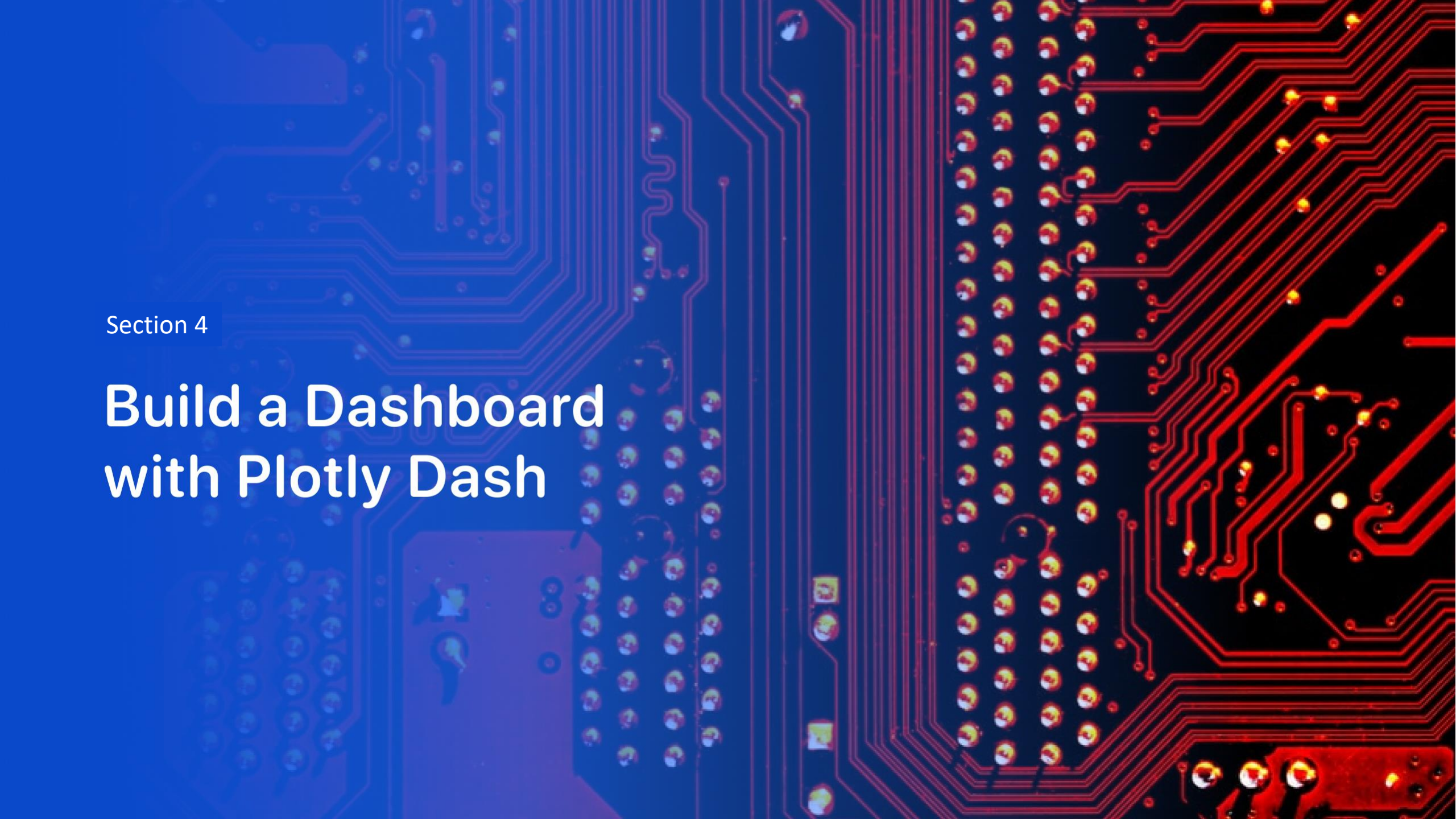
Map with markers for success/failed launches

- With the color-labeled markers it is easy to identify which launch sites have relatively high success rates.

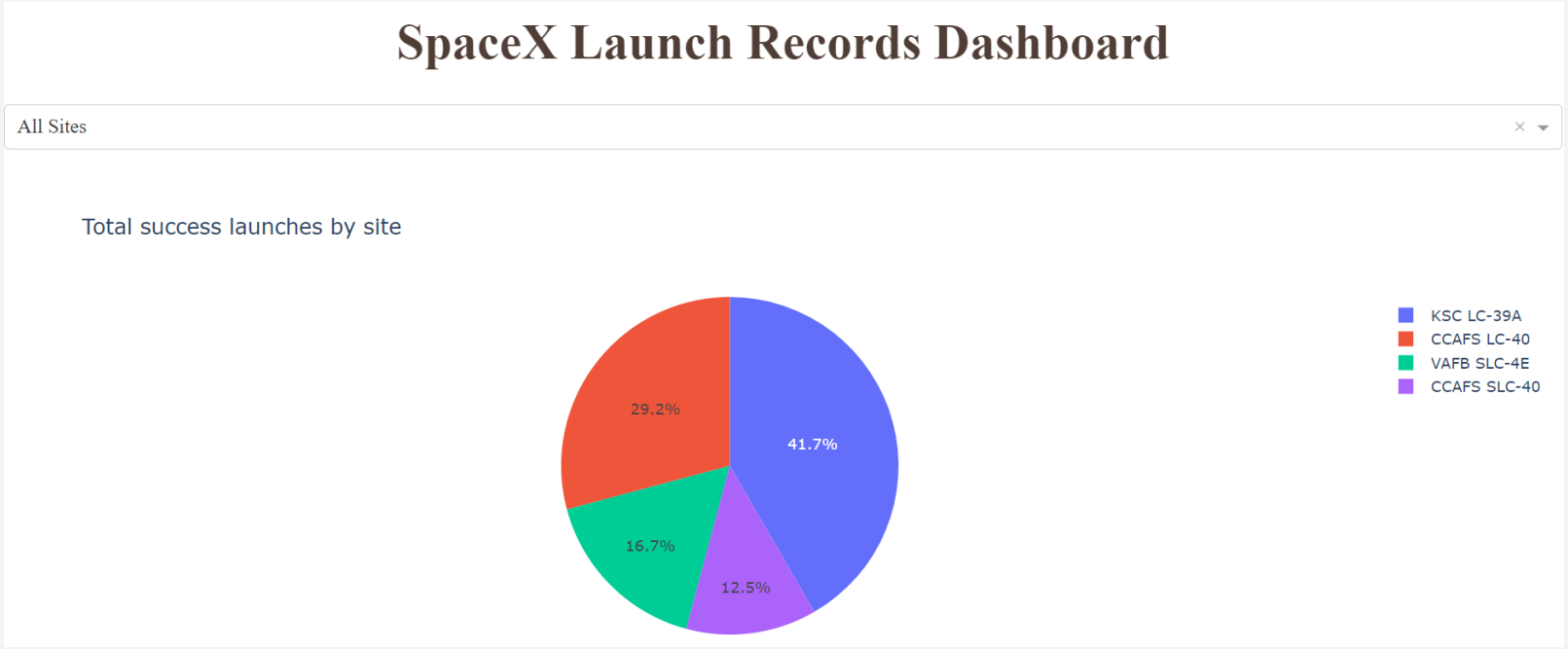
Map with launch site and proximities



- CCAFS LC-40 is close to the coast, to railways and to highways

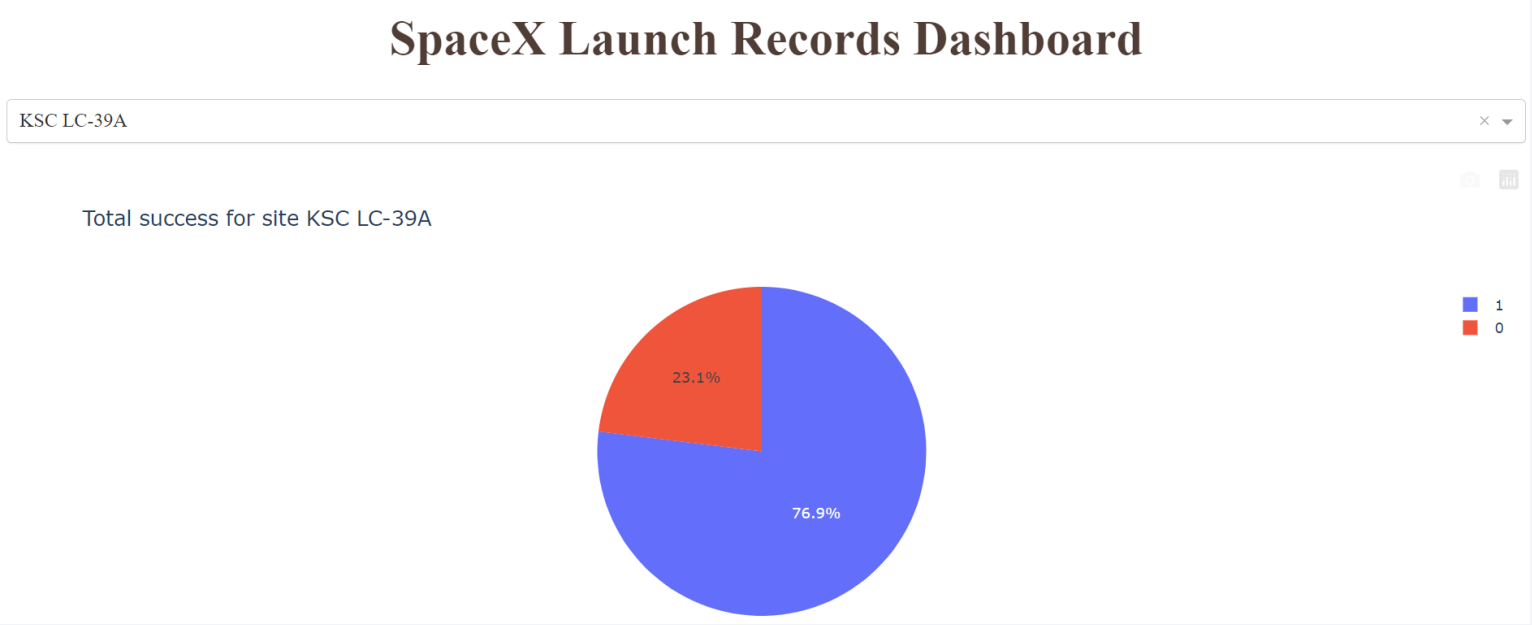

Section 4
Build a Dashboard with Plotly Dash

Dashboard - Piechart of launch success count for all sites



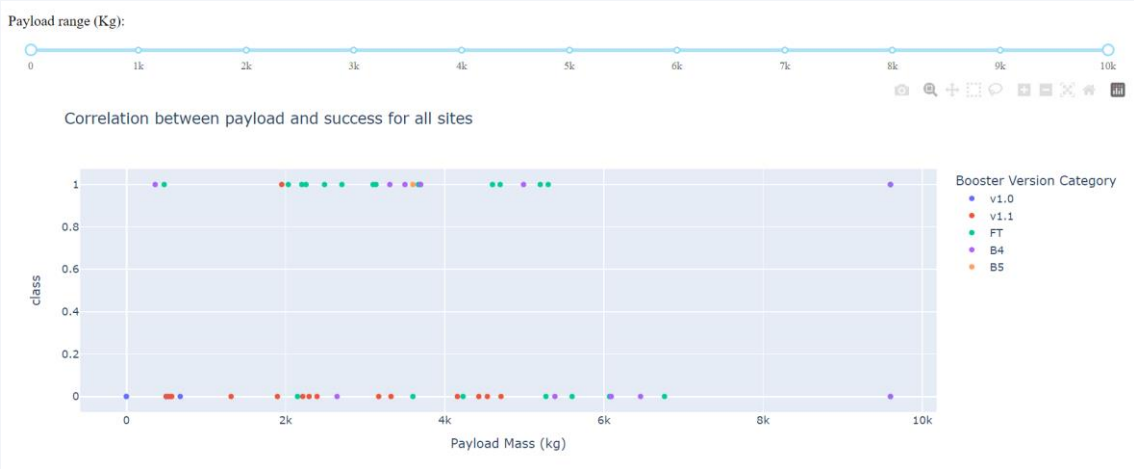
- KSC LC-39A has the biggest number of success lauches and CCAFS LC-40 has the second biggest

Dashboard – Success rate for KSC LC-39A



- KSC LC-39A has the biggest success rate, 76,9%

<Dashboard Screenshot 3>



- Lower payload mass has bigger success rate

Section 5

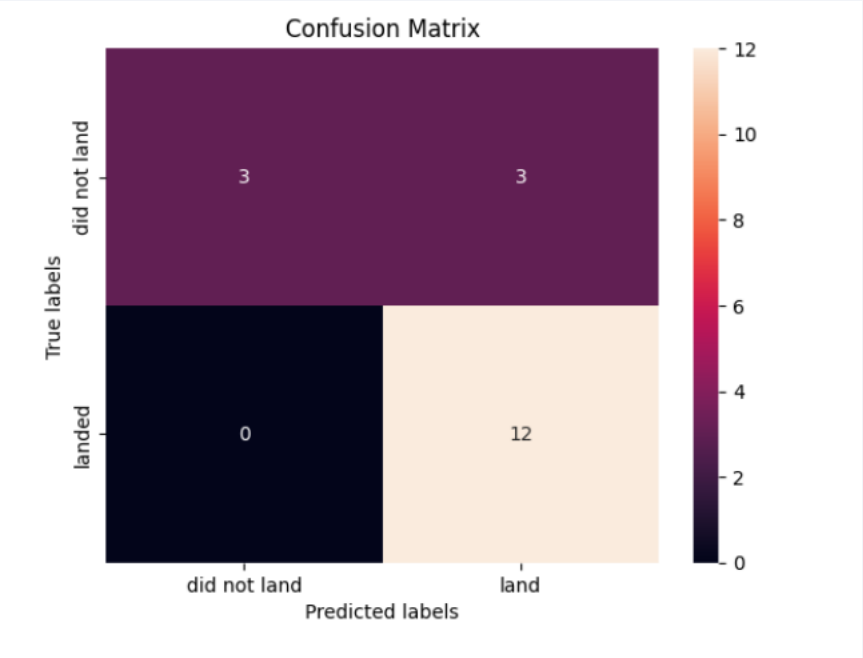
Predictive Analysis (Classification)

Classification Accuracy

	Train	Test
logreg	84.64	83.33
svm	84.82	83.33
tree	88.92	83.33
knn	84.82	83.33

- As we can see from the table, the decision tree classifier has the best classification accuracy

Confusion Matrix



- Decision tree classifier can distinguish between the different classes. The major problem is false positives.

Conclusions

- A set of variables can be used to predict if the first stage will land successfully, as the payload mass, orbit, launch site
- Success rate increases over time
- KSC LC-39A has the biggest success rate, 76,9%
- Launch sites usually are close to the Equator line and to coastline
- The best classification model is the decision tree

Appendix

- Include any relevant assets like Python code snippets, SQL queries, charts, Notebook outputs, or data sets that you may have created during this project

Thank you!