IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

Vincent
August 2024

Outline

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

Executive Summary

Main steps for this analysis:

Data collection with API and web scrapping

Data wrangling and cleaning

Exploratory data analysis with SQL and visualization

Map creation with folium

Interactive dashboarding with dash

Training of machine learning models dans hyperparameter optimization.

Summary of all results

With the data collected, features can be extracted to train machine learning models and predict a launch outcome based on the history of launches.

A tree model reaches almost 95% accuracy

Introduction

- **Project background and context**

  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. 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 space X for a rocket launch.

- **Problems you want to find answers**

  - Perform exploratory Data Analysis and determine useful features
  - Evaluate the capacity to train an accurately predict a launch outcome based on selected features and recent history of lanches
  - Use this knowledge to determine the best launch site for a mission

Section 1

Methodology

Methodology

Executive Summary

Data collection methodology:

- Data was collected from the API provided by SpaceX
- Additional data was obtained through web scrapping on Wikipedia

Perform data wrangling

- Data was cleaned, missing values imputed and categorical features one-hot encoded

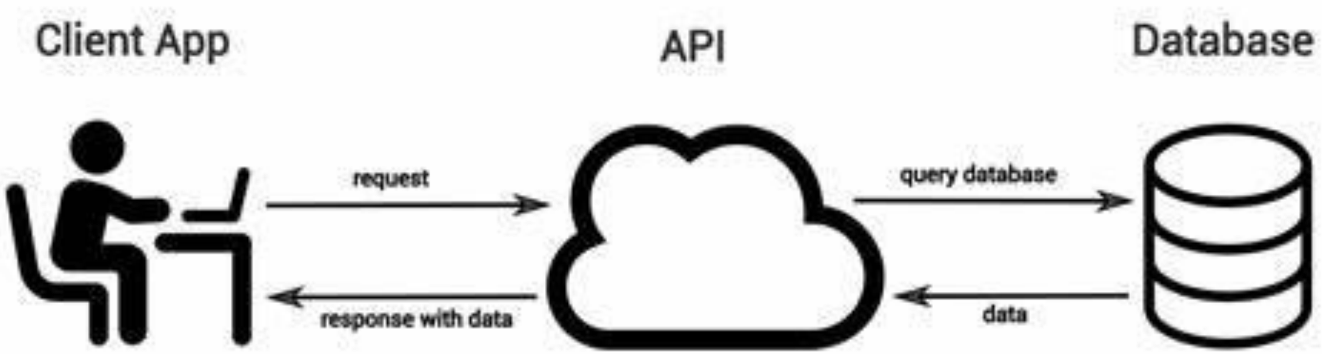
Perform exploratory data analysis (EDA) using visualization and SQL

Perform interactive visual analytics using Folium and Plotly Dash

Perform predictive analysis using classification models

- Various models were trained, hyperparameters selected with Grid search and final accuracies compared.

Data Collection – SpaceX API



- Data was queried from the API provided by Spacex with the requests library.
- Data was then cleaned, filtered and missing values were handled
- Additional was retrieved by scrapping tables from Wikipedia

Data Collection – SpaceX API

We used several endpoints from the SpaceX API to extract useful data:

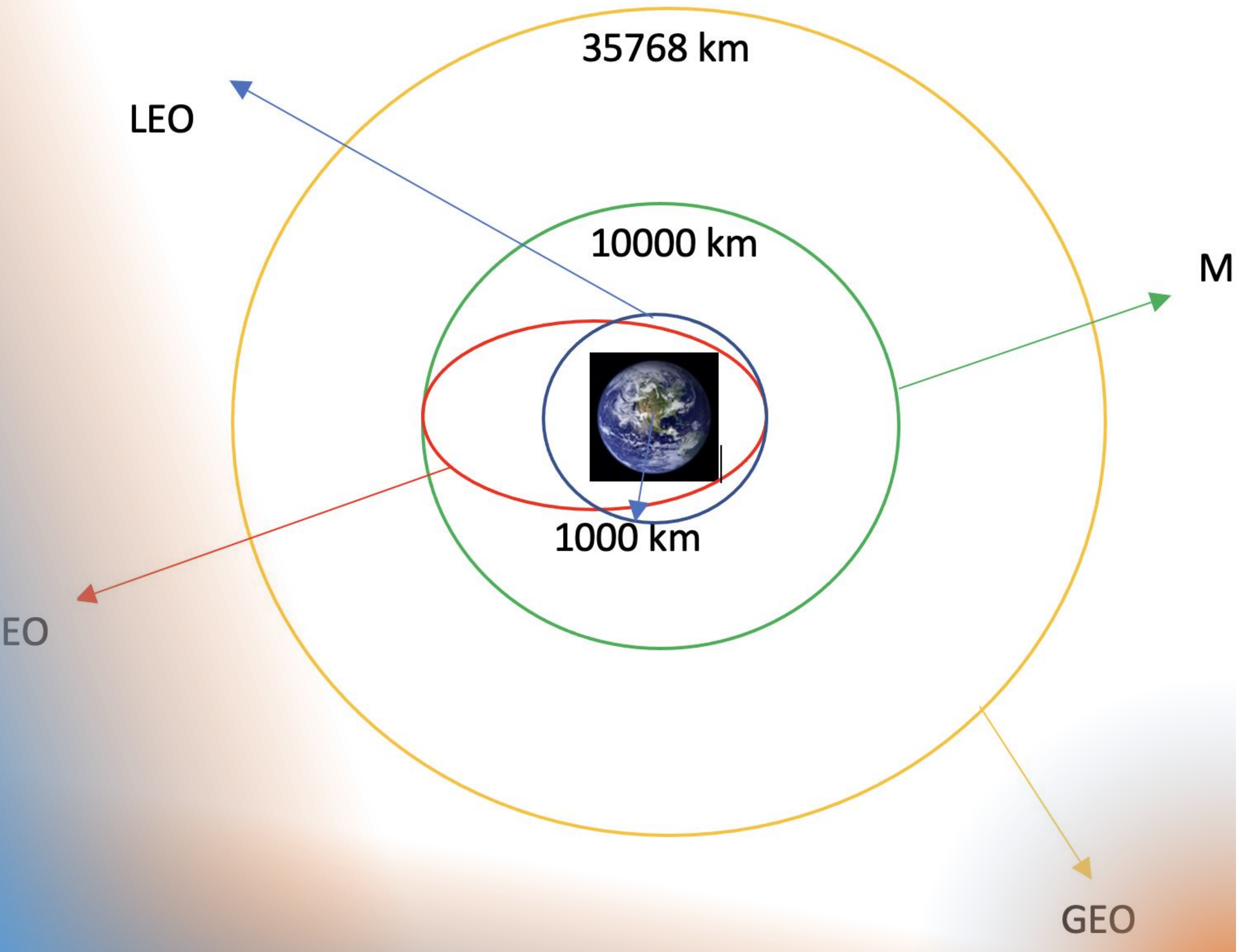
The ScaceX API Notebook is available here: IBMDataScience/jupyter-labs-spacex-data-collection-api.ipynb at main · VincentC75/IBMDataScience (github.com)

- https://api.spacexdata.com/v4/rockets/
- https://api.spacexdata.com/v4/launchpads/
- https://api.spacexdata.com/v4/payloads/
- https://api.spacexdata.com/v4/cores/

Data Collection - Scraping

We used the beautiful soup library to scrape data from tables in Wikipedia pages.

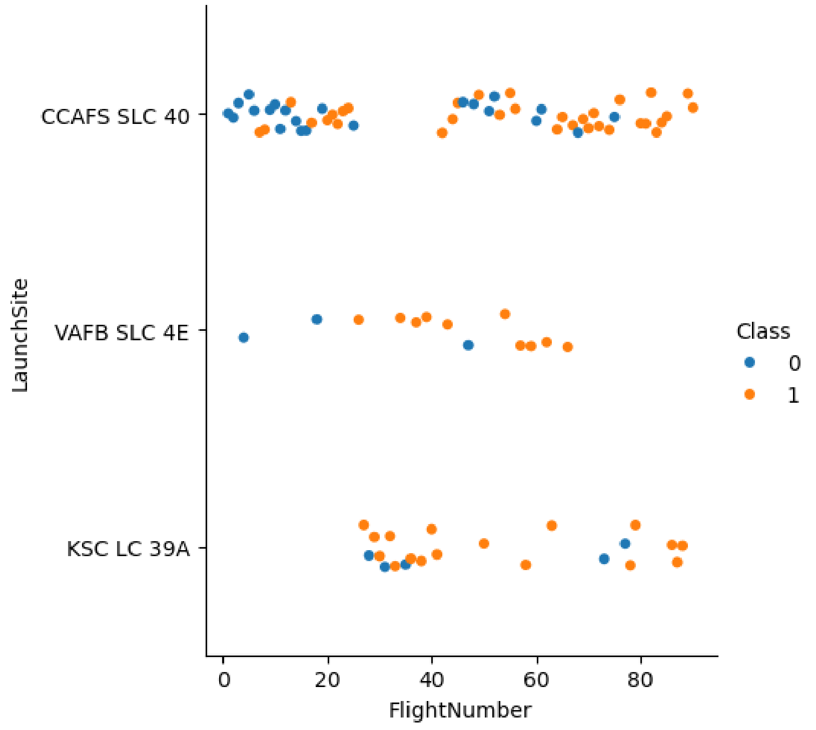
The web scraping notebook is available here: https://github.com/VincentC75/IBMDataScience/blob/main/jupyter-labs-webscraping.ipynb

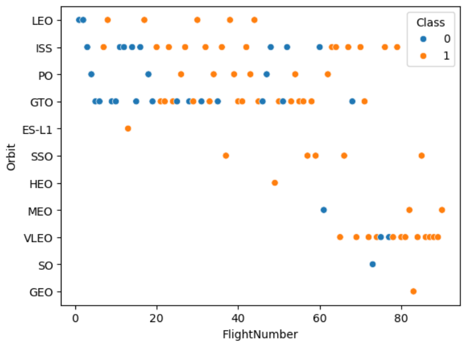


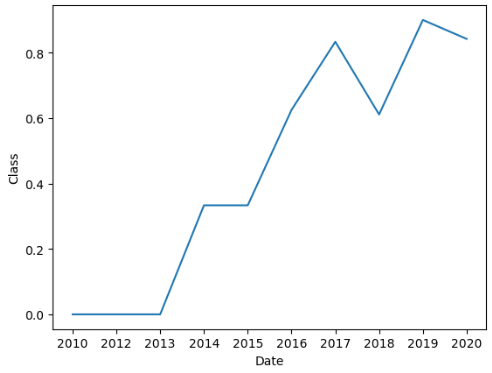
Data Wrangling

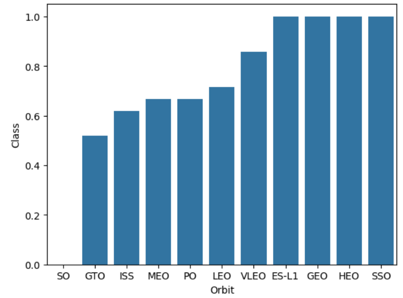
- During data wrangling, we:
  - Calculatied the number of launches on each site
  - Calculated the number and occurrence of each orbit
  - **Calculated the number and occurence of mission outcome of the orbits**
  - **Created a landing outcome label to be used as the dependent variable for machine learning**

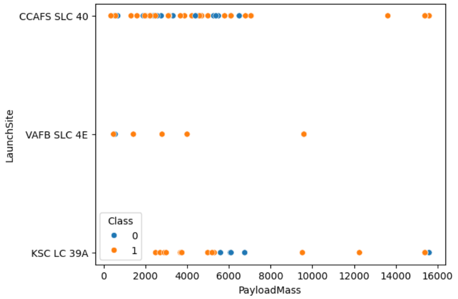
- The data wrangling notebook is available here: [IBMDataScience/labs-jupyter-spacex-Data wrangling.ipynb at main · VincentC75/IBMDataScience (github.com)]()











EDA with Data Visualization

- We used scatter plots, bar plots and lines charts to visualize the SpaceX data

- The EDA with data visualization notebook is available here: [IBMDataScience/edadataviz.ipynb at main · VincentC75/IBMDataScience (github.com)]()

EDA with SQL

We used SQL queries to:

Show DISTINCT values

Filter on partial strings with LIKE

Filter data on value with WHERE and on range with BETWEEN

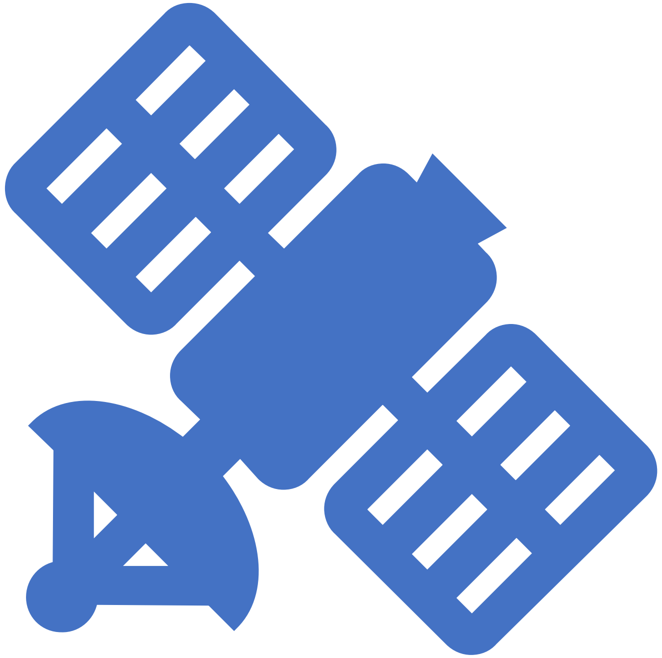
Group and summarize data with GROUP BY, SUM and AVG

Filter on the result of a subquery

Use multiple filters to answer complex questions

The completed EDA with SQL notebook is available here:
[IBMDataScience/jupyter-labs-eda-sql-coursera_sqllite.ipynb at main · VincentC75/IBMDataScience (github.com)]

Build an Interactive Map with Folium

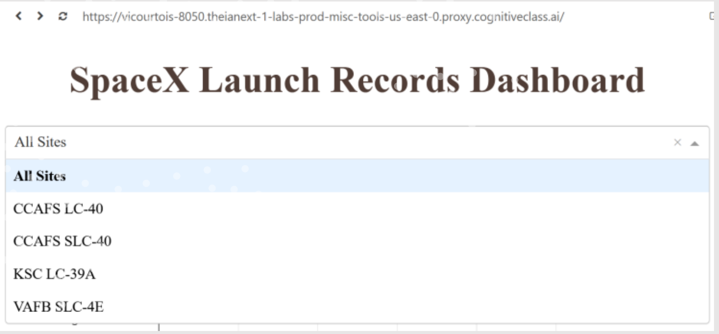


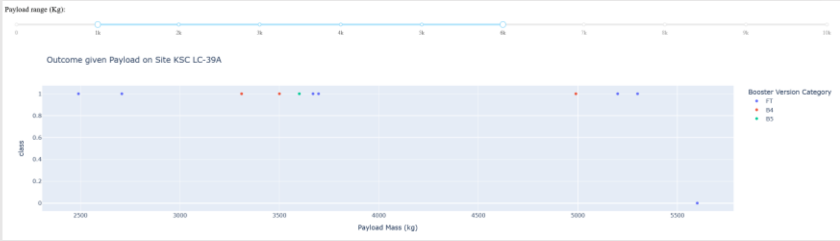
- On top of the US map, we added;
    - Circles to show the launch sites
    - Cluster of markers to indicate the launch outcomes
    - Lines to indicate the distance to coastline or the closest city
    - Text with calculated distance
- The interactive map with Folium notebook is available here: IBMDataScience/lab_jupyter_launch_site_location.ipynb at main · VincentC75/IBMDataScience (github.com)

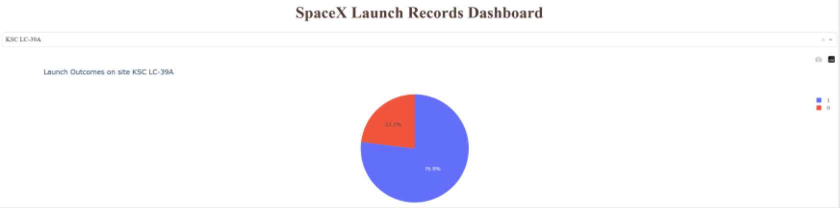
Build a Dashboard with Plotly Dash

- In the Dash interactive dashboard we used:
    - A dropdown box to select the site
    - A slider to select the payload
    - A Pie Chart
    - A scatterplot

- The Python code for the Dash lab is available here: [IBMDataScience/spacex_dash_app.py at main · VincentC75/IBMDataScience (github.com)]

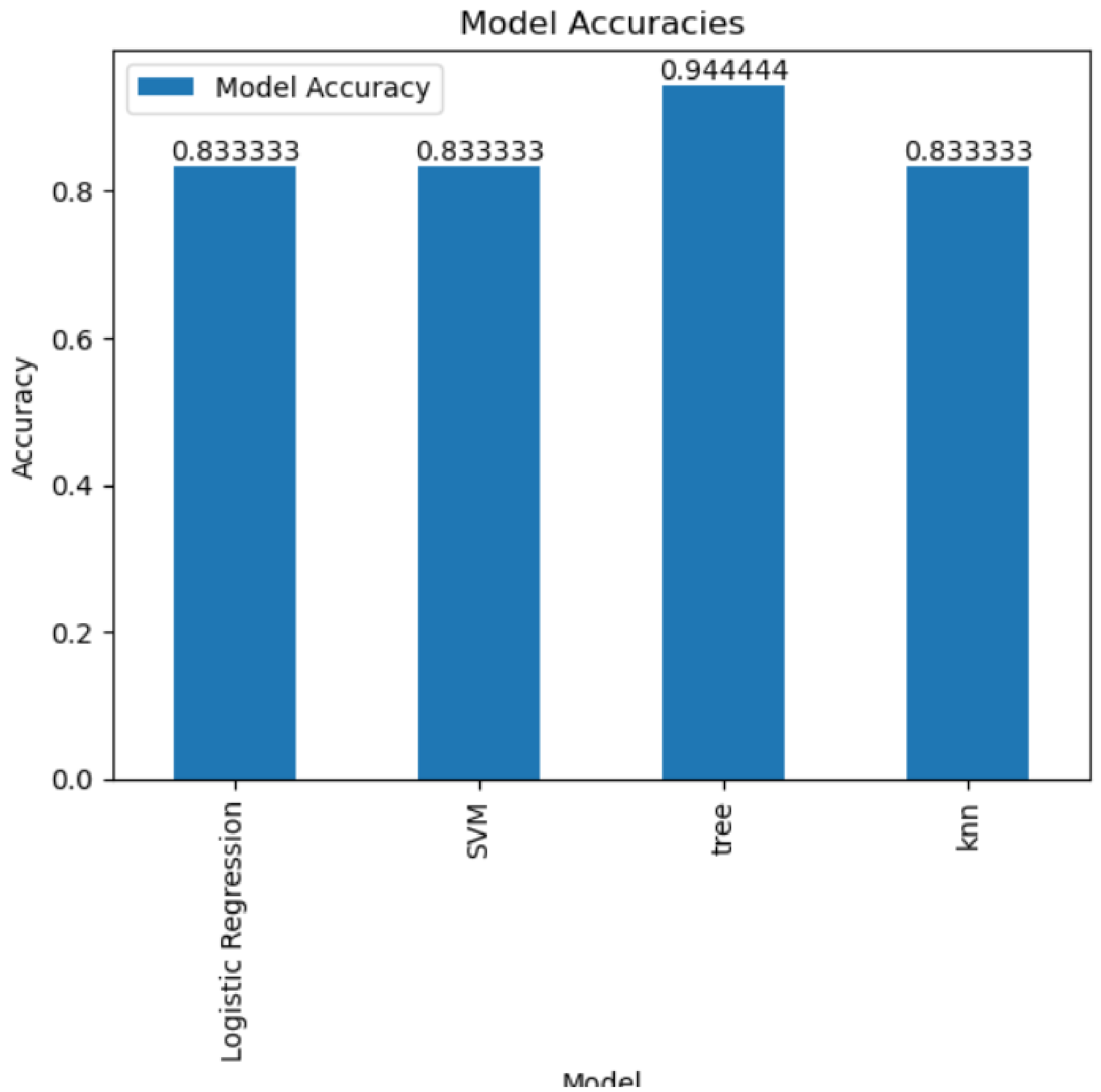






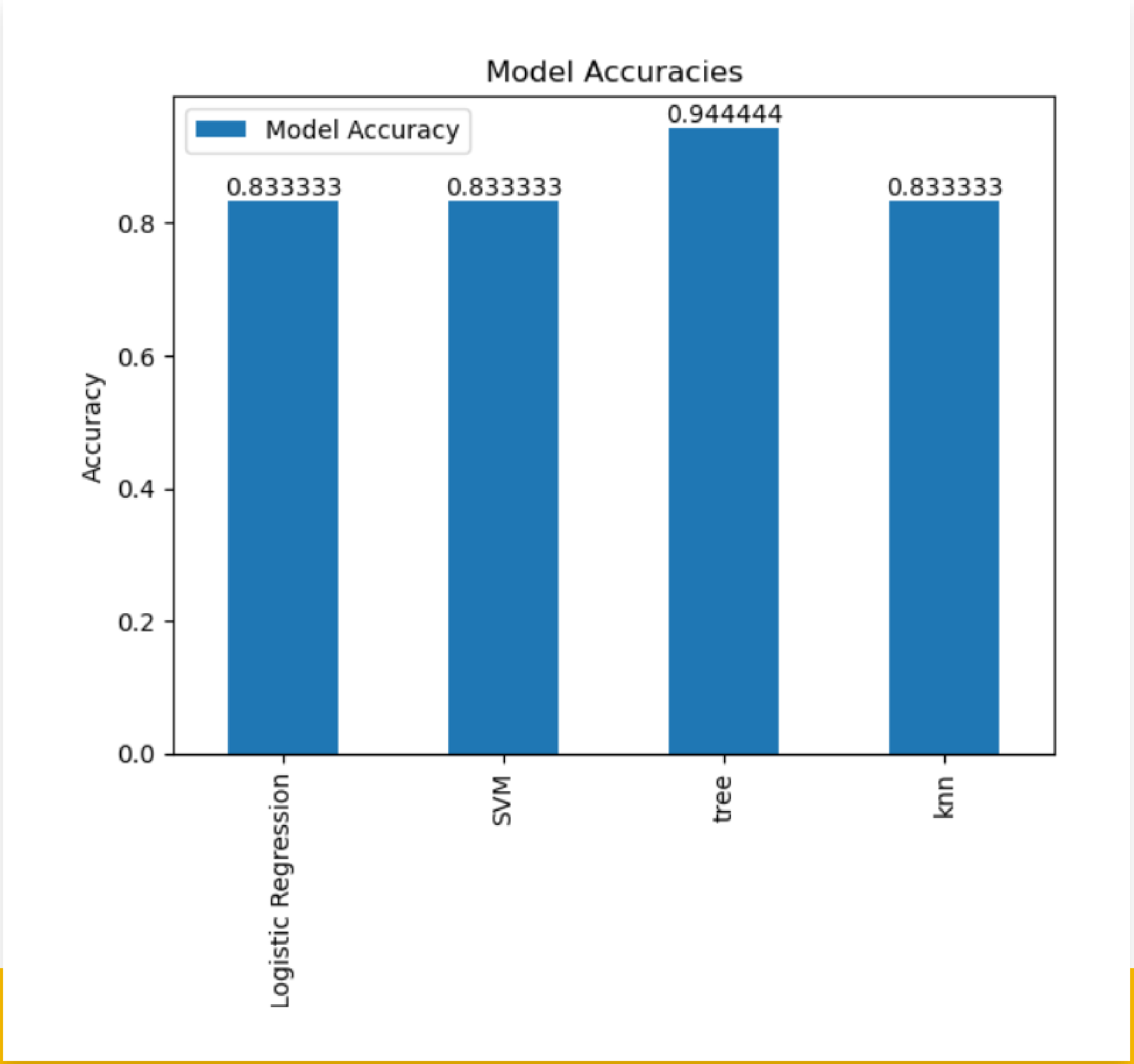
Predictive Analysis (Classification)

- The previous steps lefts use with a dataset with preselected features and the outcome dependent variables.
- We trained several types of classification models and used grid search to optimize the hyperparameters
- We computed and compared the accuracies on test data to select the best model.
- The machine learning notebook is available here: [IBMDataScience/SpaceX_Machine Learning Prediction_Part_5.ipynb at main · VincentC75/IBMDataScience (github.com)]



Results

- Exploratory data analysis allowed use to better understand our data and select the useful features for model building
- Interactive analytics dashboard was useful to further explore our data in an interactive and visual way
- Predictive analysis showed that a tree model allowed to predict with almost 95% accuracy the outcome of a launch.

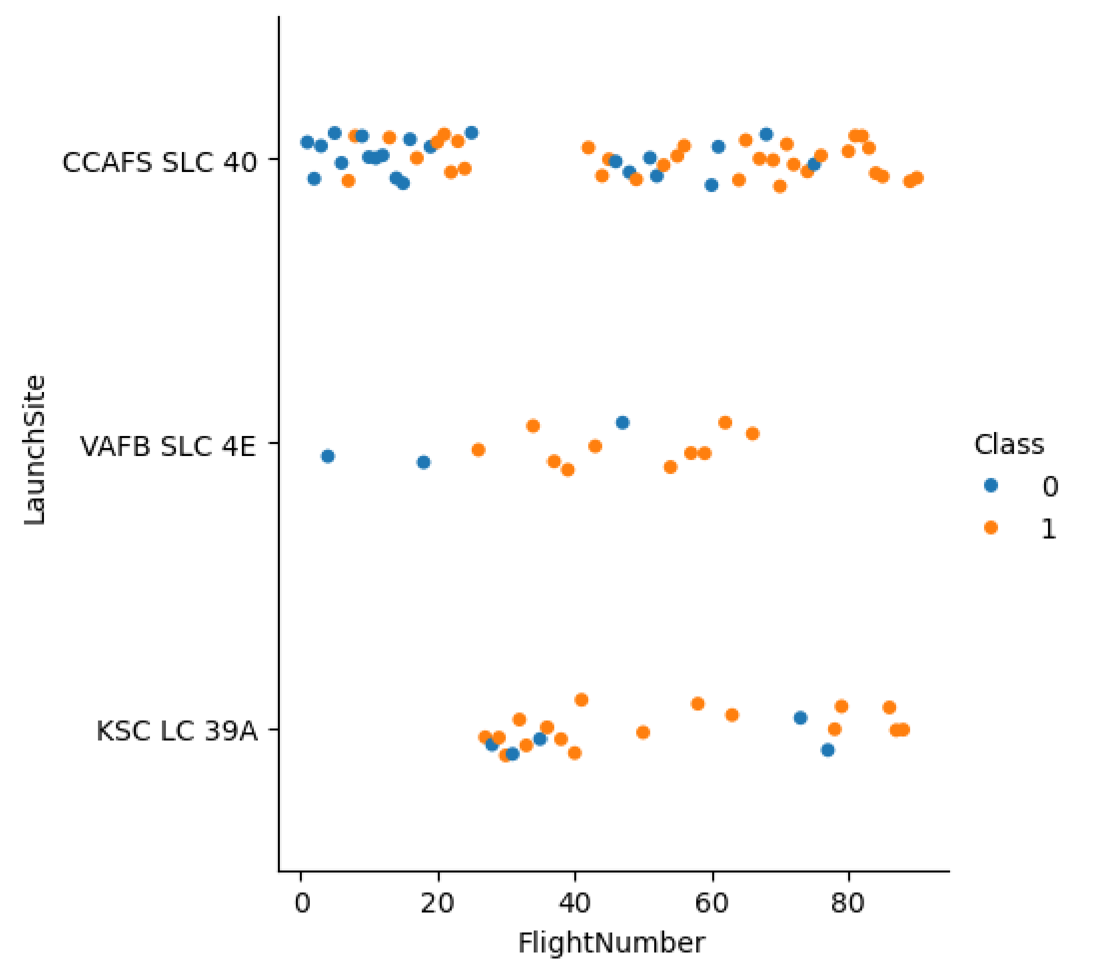


Section 2

Insights drawn from EDA

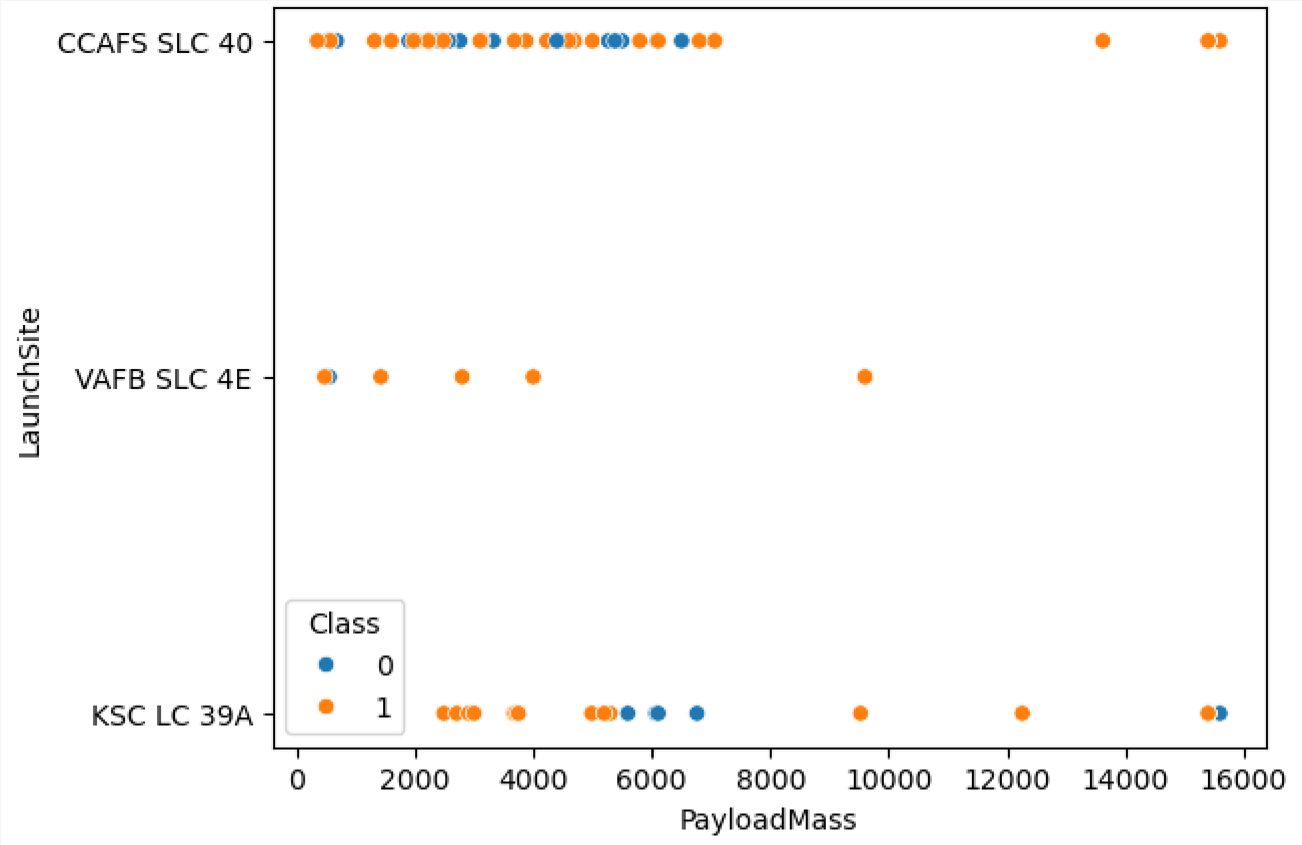
Flight Number vs. Launch Site

- This scatter plot show the Flight Number vs. Launch Site
- The outcome (failure in blue, success in orange) is color coded. It show that as flight number increases there are more successes.



Payload vs. Launch Site

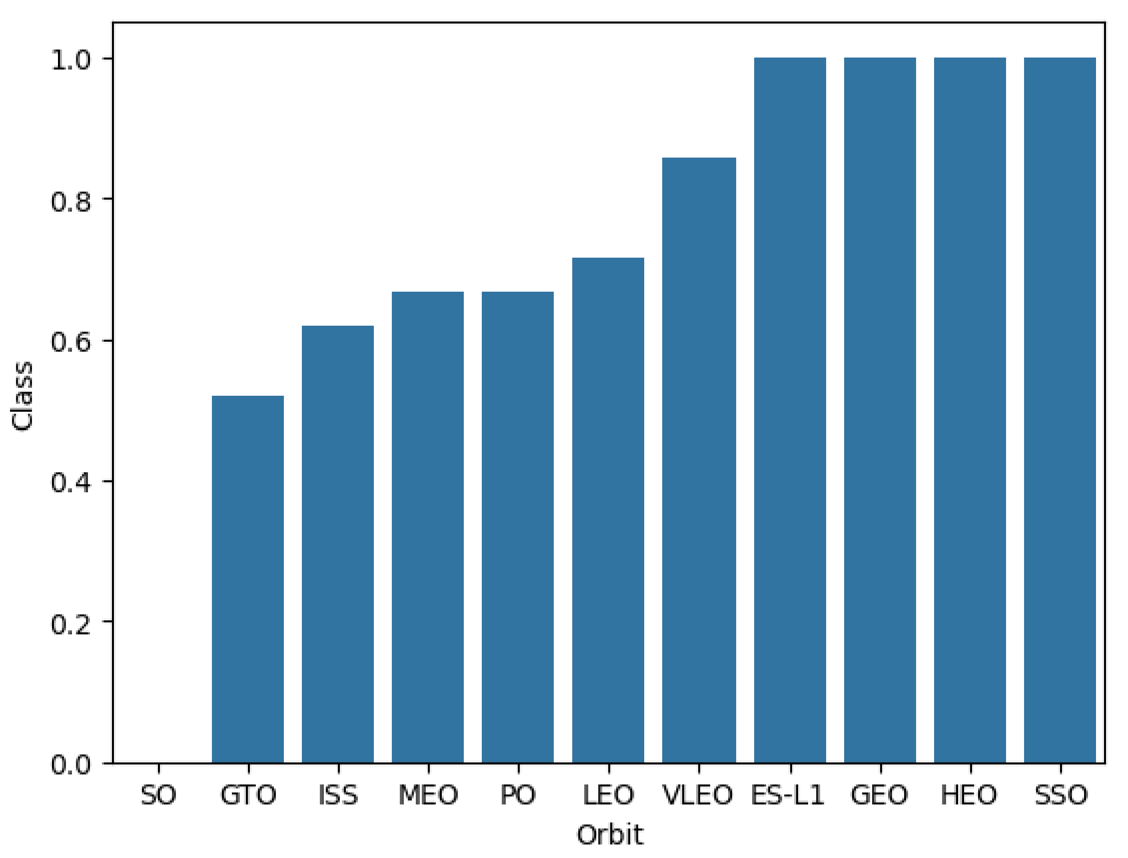
- This scatterplot shows that the VAFB SLC 4E site is dedicated to smaller payloads and has much less launches.



Success Rate vs. Orbit Type

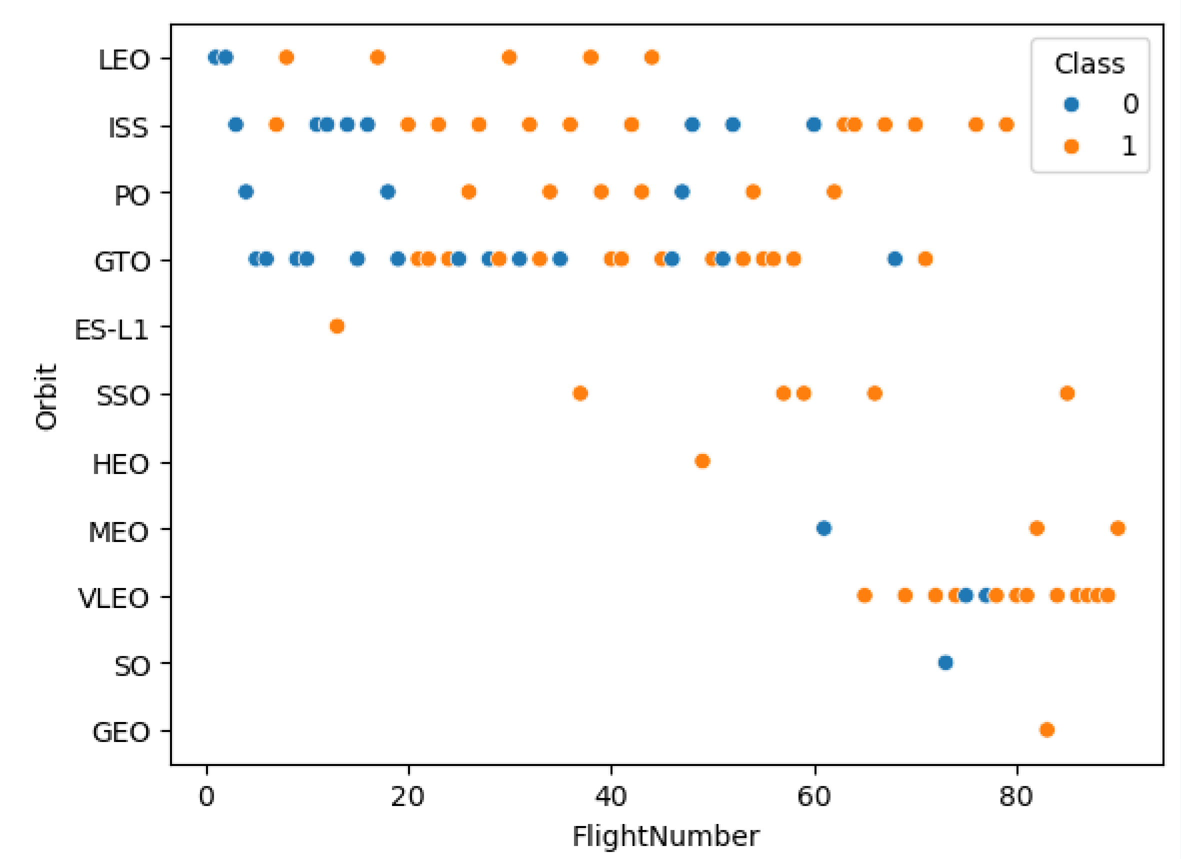
- Show a bar chart for the success rate of each orbit type

- Show the screenshot of the scatter plot with explanations



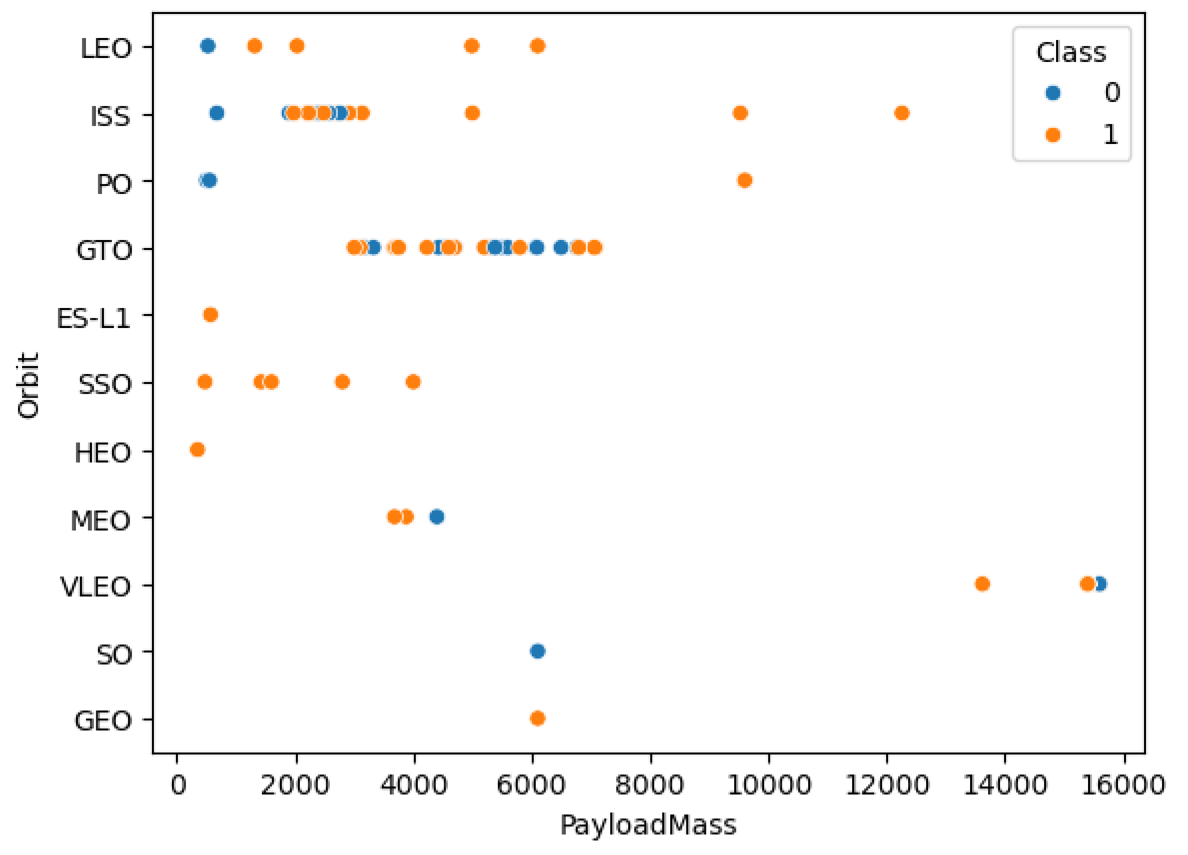
Flight Number vs. Orbit Type

- Show a scatter point of Flight number vs. Orbit type
- Show the screenshot of the scatter plot with explanations



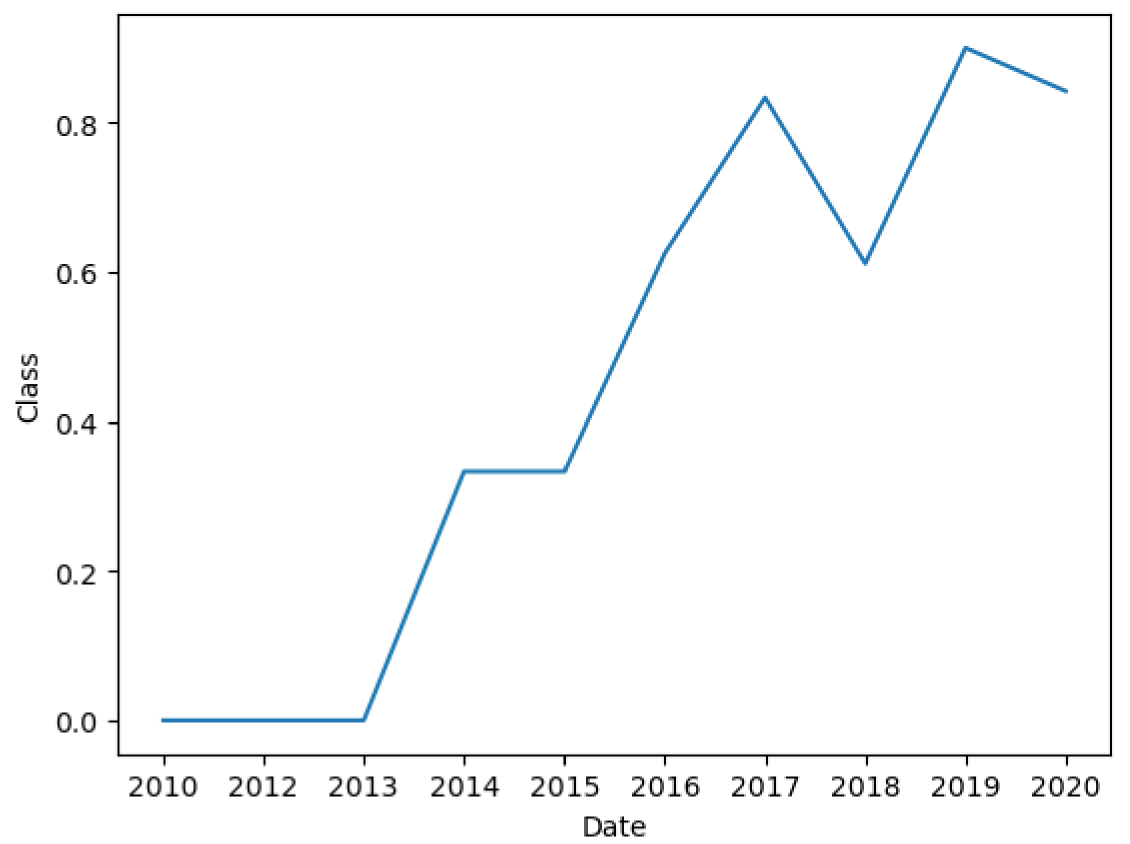
Payload vs. Orbit Type

- Show a scatter point of payload vs. orbit type
- Show the screenshot of the scatter plot with explanations



Launch Success Yearly Trend

- Show a line chart of yearly average success rate
- Show the screenshot of the scatter plot with explanations



All Launch Site Names

- To find the names of the unique launch sites we include the DISTINCT keyword to eliminate duplicates in the result.

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

```
%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'

- This query requires the use of the LIKE command to filter the launch site by name and the LIMIT clause to choose the number of results to display.

Display 5 records where launch sites begin with the string '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

Total Payload Mass

- The total payload carried by boosters from NASA is calculated with the SUM aggregation command. The customer is filtered in the WHERE clause.

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

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

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

SUM(PAYLOAD_MASS__KG_)
45596

Average Payload Mass by F9 v1.1

- The average payload mass carried by booster version F9 v1.1is calculated by using the AVG function and filtering the Booster version with a LIKE clause.

Display average payload mass carried by booster version 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

First Successful Ground Landing Date

- The date of the first successful landing outcome on ground pad is found by first filtering the relevant outcomes and applying the MIN function to select the earliest date.

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

Hint:Use min function

```
%sql SELECT MIN(Date) FROM SPACEXTBL WHERE Landing_Outcome ='Success (ground pad)';
```

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

MIN(Date)
2015-12-22

Successful Drone Ship Landing with Payload between 4000 and 6000

- We use the DISTINCT clause to eliminate duplicates and filter on the Landing Outcome and Payload Mass in the WHERE clause. The BETWEEN clause allows to select the payload between 4000 and 6000.

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

```
%sql SELECT DISTINCT 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

Total Number of Successful and Failure Mission Outcomes

- We use the GROUP BY clause to group the count by Mission Outcome
- We could also modify the query to group together the three outcomes which are successes by have a different values in the Mission_Outcome column.

List the total number of successful and failure mission outcomes

```
%sql SELECT Mission_Outcome, COUNT(Mission_Outcome) FROM SPACEXTBL GROUP BY Mission_Outcome;
```

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

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

Boosters Carried Maximum Payload

Here we use a subquery to find the maximum payload. The result is then reused in the WHERE clause.

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

```
%sql SELECT DISTINCT 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

2015 Launch Records

- The SUBSTR function is used to extract the month and the year from the Date filed.

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.

```
ELECT  substr(Date, 6,2) as month, Landing_Outcome, Booster_Version FROM SPACEXTBL WHERE Landing_Outcome='Failure (drone ship)' AND substr(Date,0,5)='2015
```

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

month	Landing_Outcome	Booster_Version
01	Failure (drone ship)	F9 v1.1 B1012
04	Failure (drone ship)	F9 v1.1 B1015

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

- This query has a filter on a date range and is grouped by Landing Outcome.
- The DESC clause reverses the order to get the greatest count on top.

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.

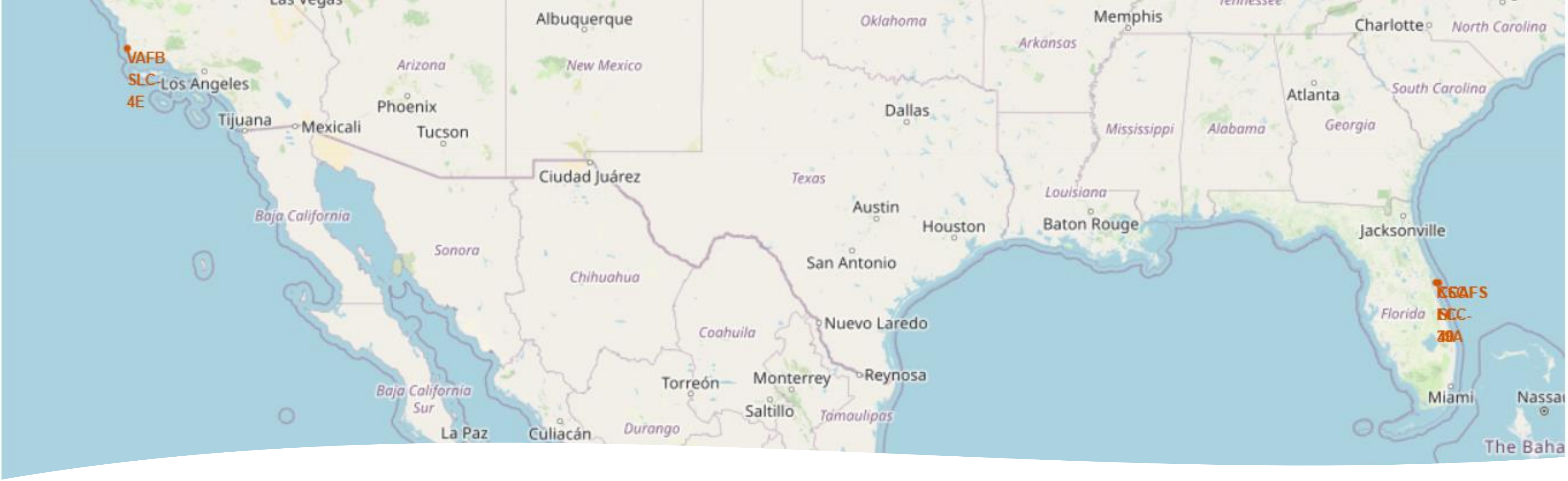
```
%sql SELECT Landing_Outcome, COUNT(Landing_Outcome) AS Count FROM SPACEXTBL WHERE Date BETWEEN '2010-06-04' and '2017-03-20' \
GROUP BY Landing_Outcome ORDER By Count DESC;
```

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

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

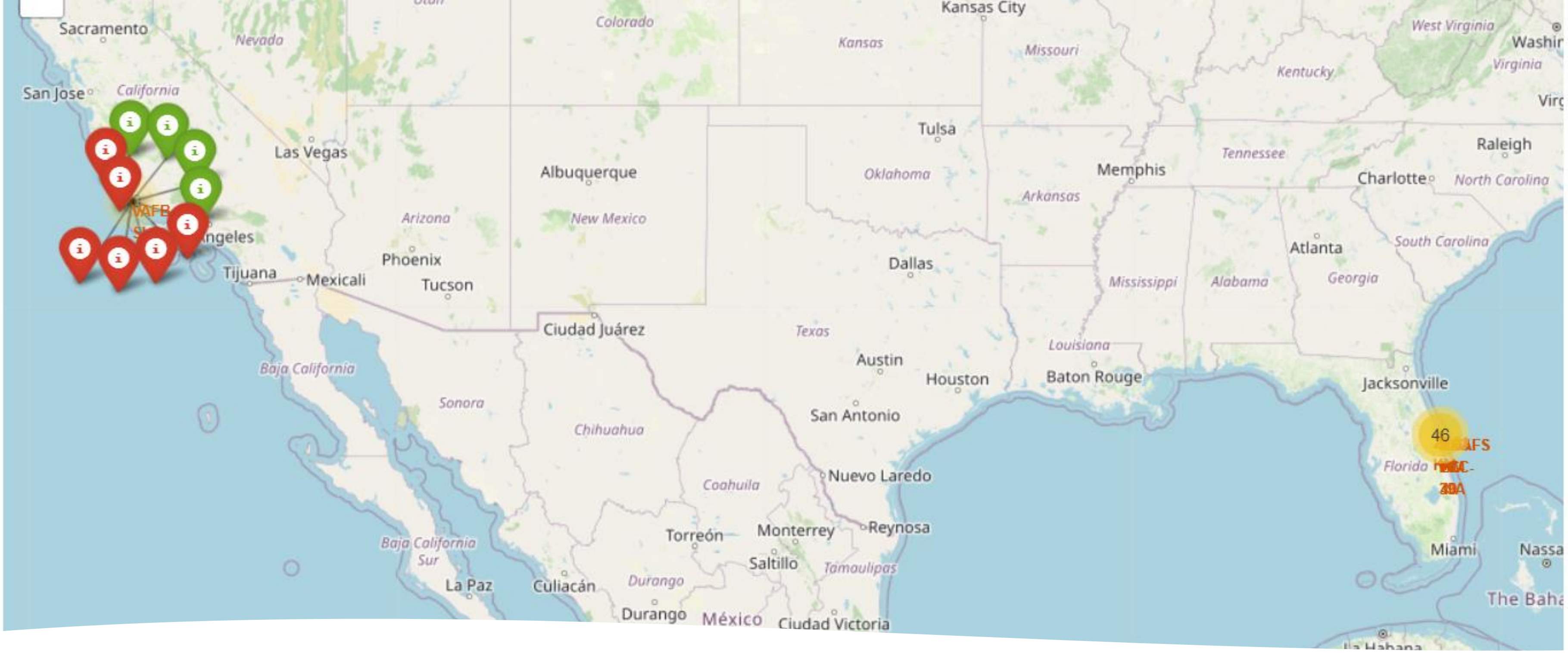
Section 3

Launch Sites Proximities Analysis



SpaceX Launch Sites Location

- SpaceX launch sites are located in the south of the United States to be close to the equator.
- There is a launch site on the west coast in California and two launch sites on the west coast in Florida.



Color-labeled Launch Outocomes

- Color-coded markers show the outcomes when zooming on a given site.

Proximity to coastline and city

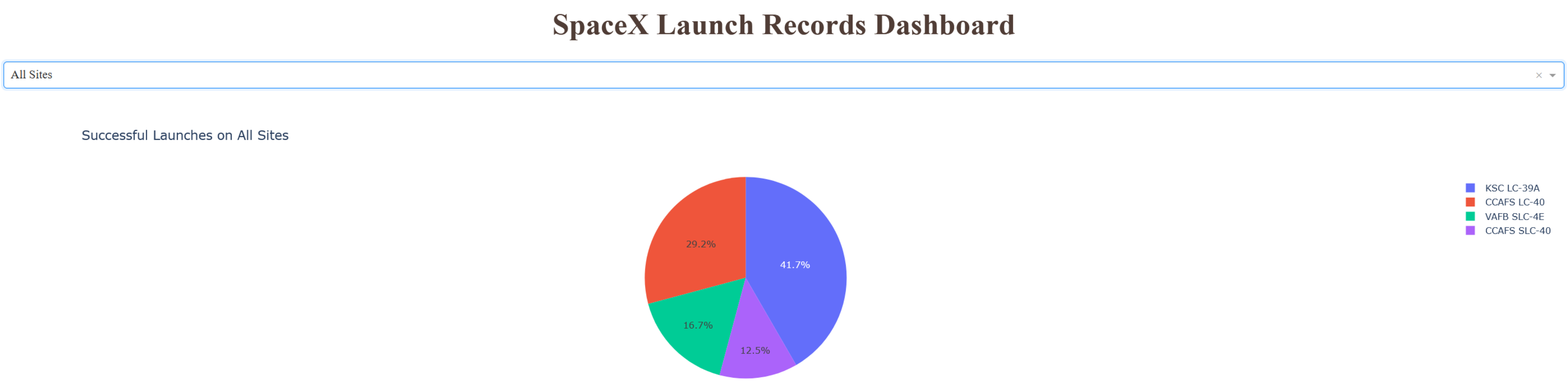
- An indicator can be shown on the map to underline the distance to the coast or to the closest city.

Section 4

Build a Dashboard with Plotly Dash

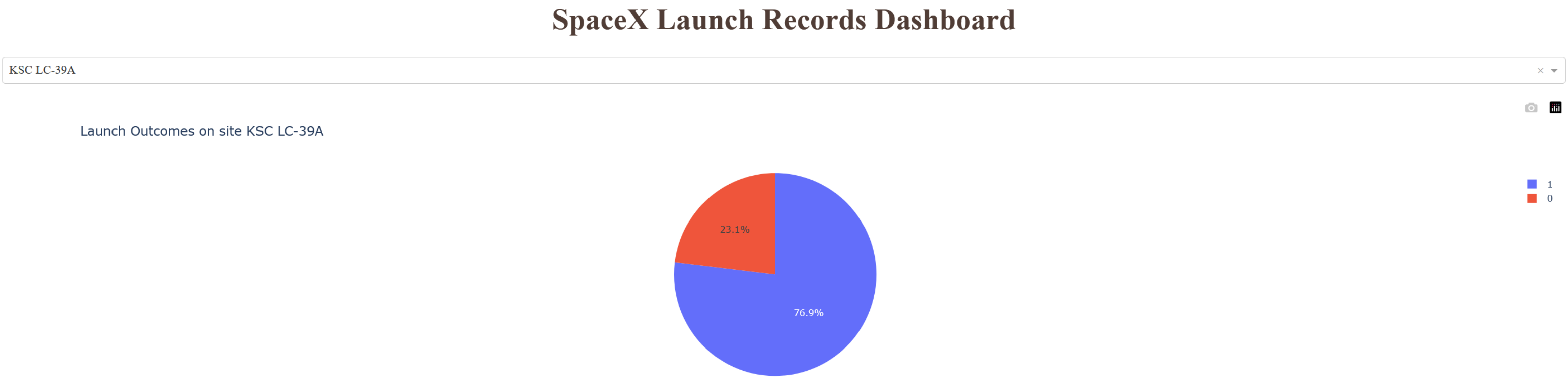
Pie Chart of launch success count for all sites

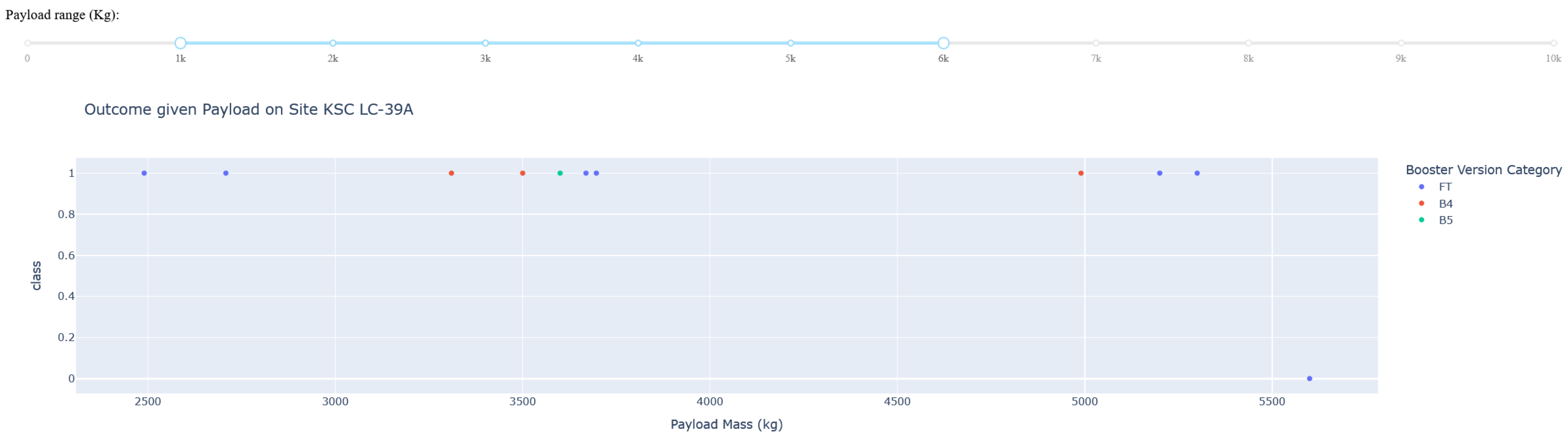
- A pie chart allows to show how each site contributed to successful launches.



Launch Outcomes on site KSC-LC-39A

- The dashboard allows to zoom on a given site and show the proportion of successes on this site.





Scatterplot of Outcome given Payload on site KSC-LC-39A

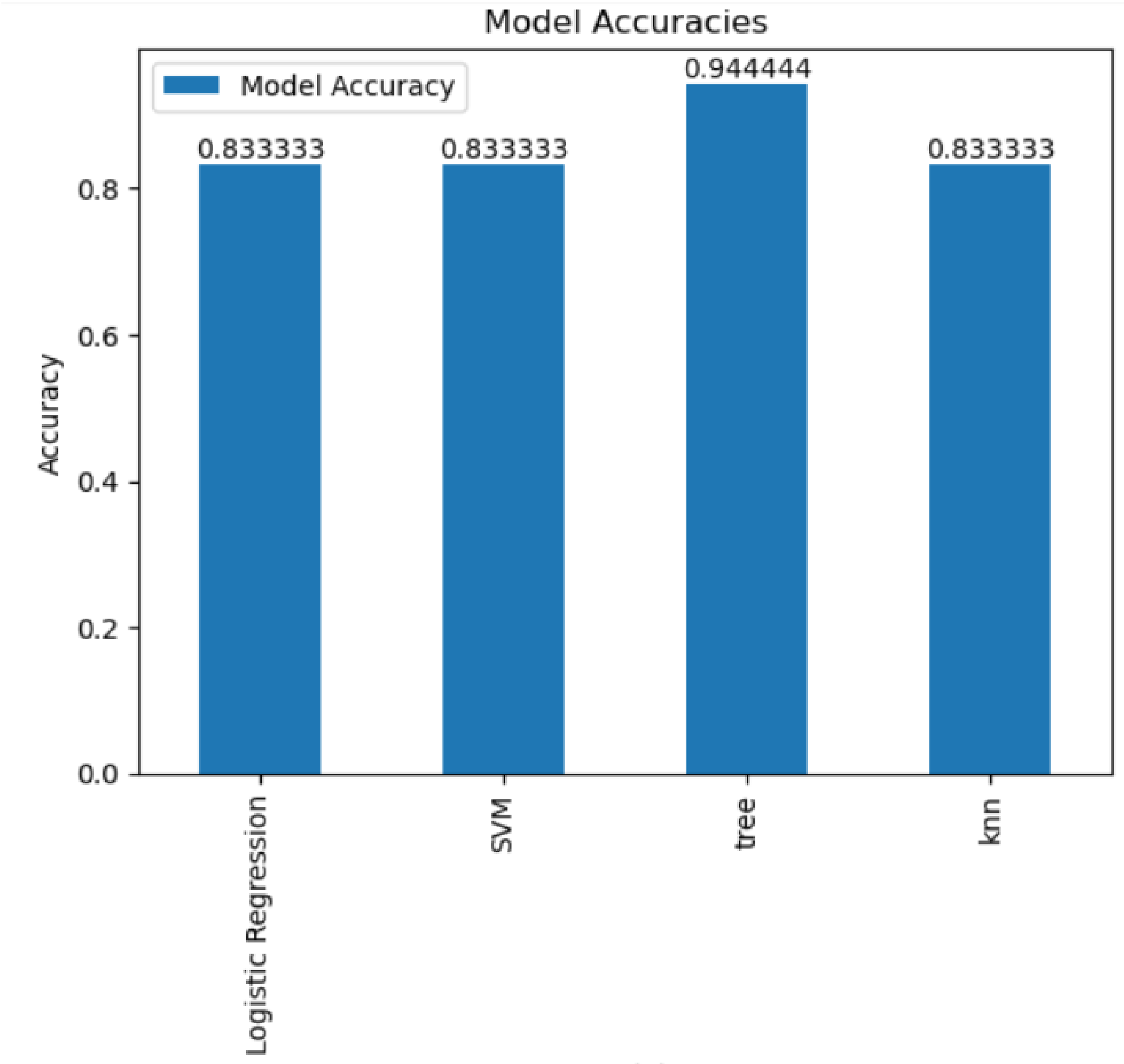
- The dashboard contains a slider to select the payload and a combo box for the site.

Section 5

Predictive Analysis (Classification)

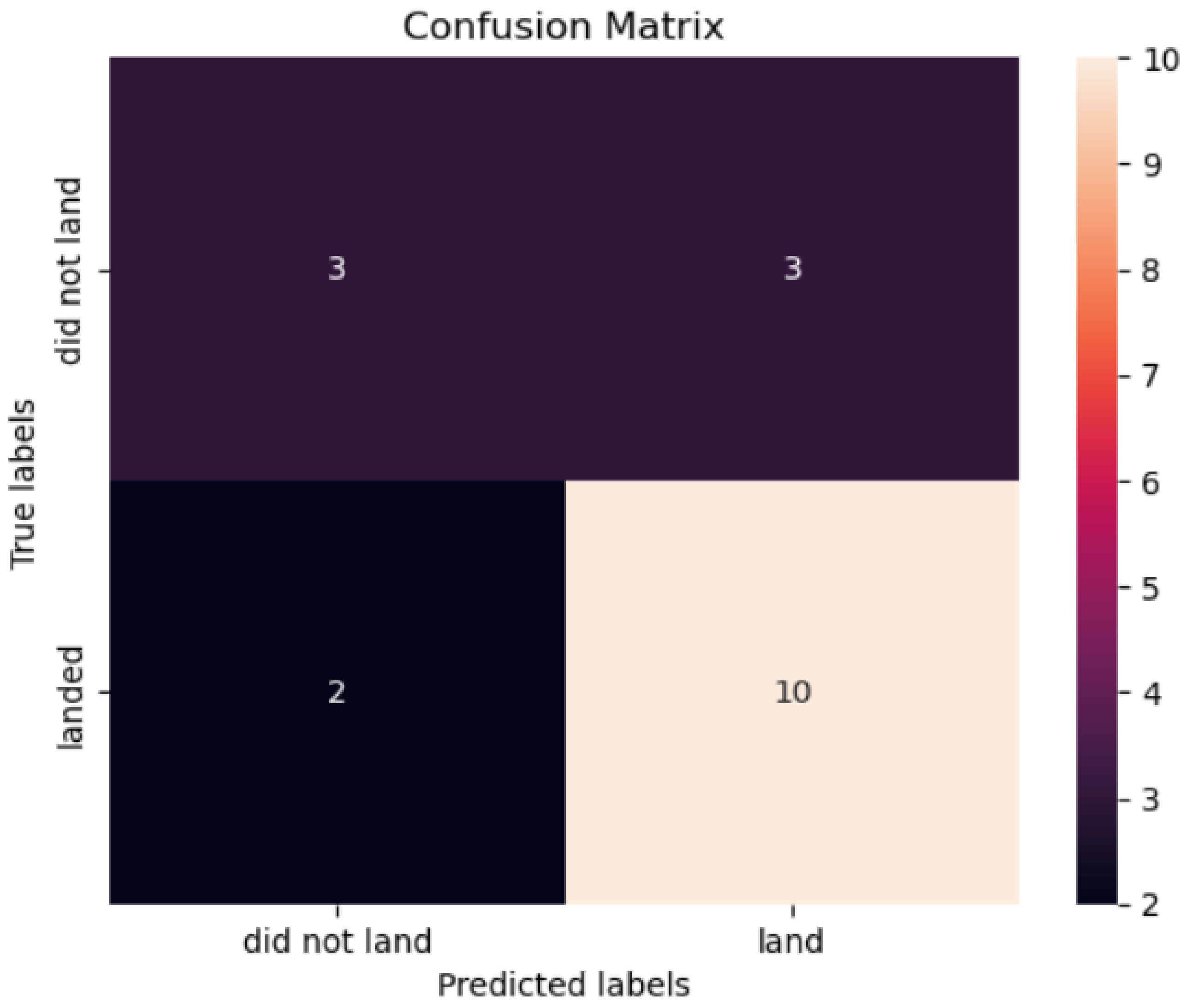
Classification Accuracy

The Tree classifier has the best accuracy of all models.



Confusion Matrix

The tree classifier makes only 5 errors, with 2 false positives and 3 false negatives.



Conclusions

This project shows that the success of SpaceX launches can be predicted with a reasonable accuracy when models are trained on chosen features.

With time the proportion of successes increases so we could put a bigger weight on recent data to improve our predictions

We could also combine our different models. They may not do the same mistakes and the combined model performs usually better.

Appendix

All notebooks, source code and helper files are available in my Github repository for this project:

[VincentC75/IBMDataScience: IBM Data Science on Coursera (github.com)]

Thank you!