IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

Fabian Herrera
08/24/2025

Outline

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

Executive Summary

- Data was obtained by 2 methods, SpaceX API and web scraping.
- Data wrangling performed in order to get usable data, filling missing values, normalizing data, etc.
- Exploratory Data visualization performed to find relations between variables and the interaction/wight that some variables have over the final outcome.
- Developed 4 prediction models, measured the accuracy.
- All 4 modules works fine but some better than others, as final result the best performing model is the Support Vector Machine Model: Train accuracy = 86%; Test Accuracy = 83%. Showing best numbers than all the others.

Introduction

- The present report is intended to summarize the inputs, transform and outcomes for a data analysis transformation and modeling process for SpaceX launch data. The complete process od data collection, data wrangling and model construction are described in respective sections. The main propose of the project is to obtain as output some classification and predicting models capable of predict if for a given set of values (input conditions) the model can predict the success or failure for the fist stage landing.

The main idea is to implement this model to predict if the mission will be a success or a fail, and to predict if the stage can be reusable or not after a mission.

- The question to be answerer using the model created, is: Are we able to predict with high precision if the first stage will be having a success landing or not, given a set of conditions?

Section 1

Methodology

Methodology

Executive Summary

- Data collection methodology:
  - Data was collected by using SpaceX REST API & Web Scraping. By using Python tools allowing to access and obtain the data used to the analysis and model generation.
- Perform data wrangling
  - Data obtained is processed and formatted to obtain usable and compatible values that can be introduced to the model construction. Such as handling of missing values, transformation of categorical values and data normalization and binning.
- Perform exploratory data analysis (EDA) using visualization and SQL
- Perform interactive visual analytics using Folium and Plotly Dash
- Perform predictive analysis using classification models
  - By using the help of some Python libraries, the data is used is input to construct, train and test several classification and prediction models. Such models are tested and evaluated to obtain metrics that can help to understand and compare the models.

Data Collection

- Data for the project was obtained by using 2 different ways:

a. SpaceX API and some Python tools.

b. Web scrapping, obtaining the data from Wikipedia and the help of some Python codding tools.

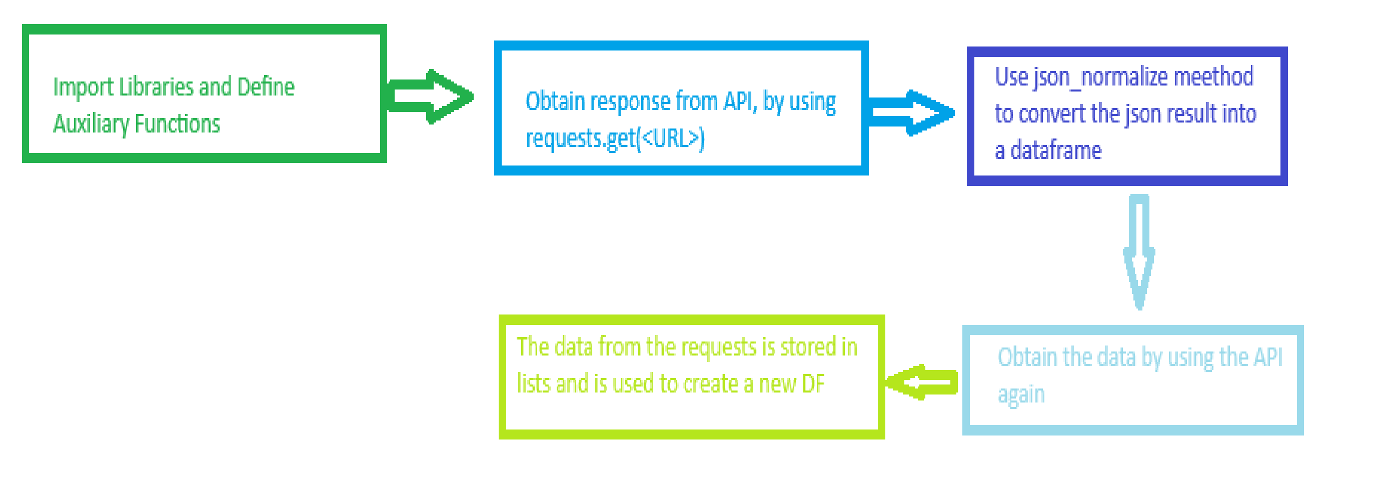
- For both cases, in next slides is presented a description of the general process used to collect the data. In both cases, the methodologies used are implementing a series of Python codding and the sue of some specific Libraries used to collect and process the data.

Data Collection – SpaceX API

- Here is presented a flowchart summarizing the process for data collection with SpaceX REST API calls.

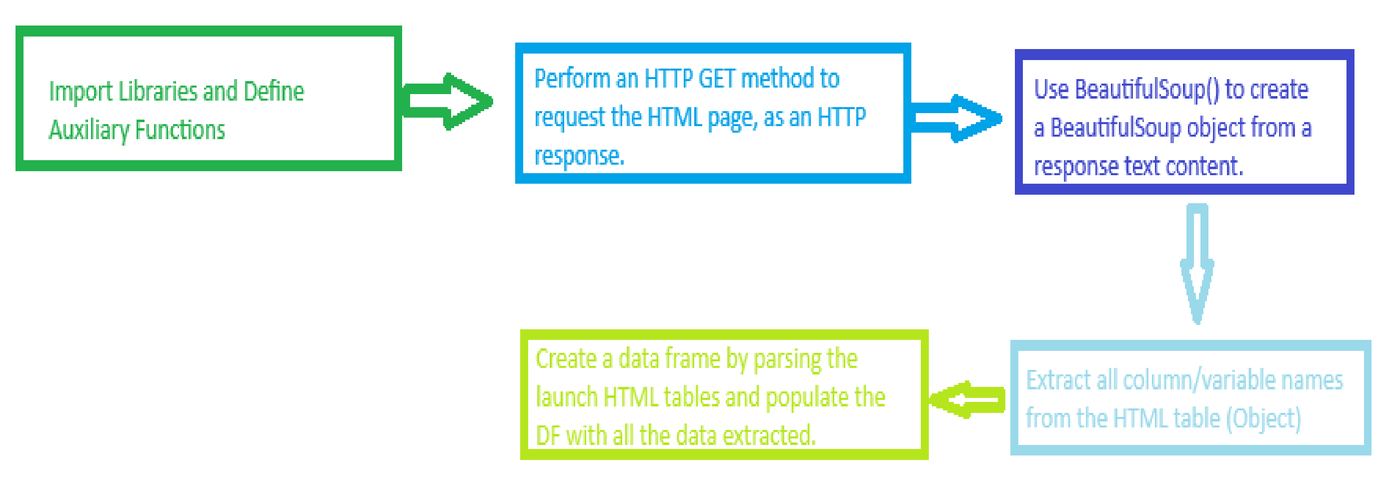
- As additional information, here is presented the GitHub URL of the completed SpaceX API calls notebook for technical review or additional questions.

https://github.com/fabianhq04/Final-Report-SpaceX-Data-modeling/blob/03143aaf5830339129f63334cd651d9b084e3668/jupyter-labs-spacex-data-collection-api.ipynb



Data Collection - Scraping

- Here is presented a flowchart summarizing the process for data collection with web Scrapping.
- As additional information, here is presented the GitHub URL of the completed SpaceX web scrapping data collection method notebook for technical review or additional questions

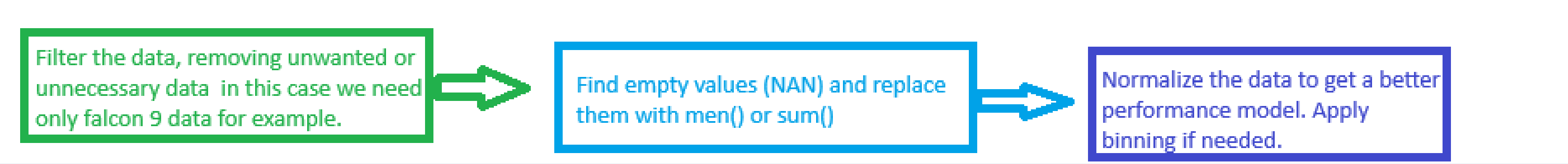


https://github.com/fabianhq04/Final-Report-SpaceX-Data-modeling/blob/03143aaf5830339129f63334cd651d9b084e3668/jupyter-labs-webscraping.ipynb

Data Wrangling

- Data was handled to be able to use it for the analysis and model construction process, this stage is named as data Wrangling. On this process we look for missing values and work with them, either filling missing records or deleting rows. Also, categorical columns get converted into logical and/or numerical values, and finally data is normalized. Here is presented a chart for the process.
- A GitHub URL of the completed Data Wrangling notebook is shared for technical review or additional questions.

https://github.com/fabianhq04/Final-Report-SpaceX-Data-modeling/blob/3fae7fb2db36cd844fdf320fc55e913bac99bb0f/labs-jupyter-spacex-Data%20wrangling.ipynb



EDA with Data Visualization

- After data was preparated for analysis, several charts are plotted in order to gain understanding of the data and the correlation between variable. Traying to identify which variables have the most relation with the final outcome and witch have no relation at all. Also attaching GitHub URL of the completed EDA with visualization notebook.

Some of the charts plotted are:

- FlightNumver vs PayLoasMass
- FlightNumber vs Launchsite
- PayloadMass vs LaunchSite
- Orbit vs Success Rate
- FlightNumber vs Orbit
- PayLoadMass vs Orbit
- Year vs Success Rate.

https://github.com/fabianhq04/Final-Report-SpaceX-Data-modeling/blob/de07b1f4cb4af2766855cd6776ab6c9bff832495/edadataviz.ipynb

EDA with SQL

Some exploratory data was performed using AQL queries in order to better understand the data and to gain knowledge about relation between variables. Next some of the SQL queries performed and also attaching GitHub URL of the completed EDA with SQL notebook.

- Display the names of the unique launch sites in the space mission as:

```
%sql select distinct Launch_Site from SPACEXTABLE
```

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

```
%sql select * from SPACEXTABLE where Launch_Site like 'CCA%' limit 5
```

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

```
%sql select SUM(PAYLOAD_MASS__KG_) from SPACEXTABLE where Customer = 'NASA (CRS)'
```

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

```
%sql select AVG(PAYLOAD_MASS__KG_) from SPACEXTABLE where Booster_Version = 'F9 v1.1'
```

EDA with SQL Continuation

- List the date when the first successful landing outcome in ground pad was achieved.

```
%sql select MIN(Date) from SPACEXTABLE where Landing_Outcome = 'Success (ground pad)'
```

- 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 SPACEXTABLE where Landing_Outcome = 'Success (drone ship)' and PAYLOAD_MASS__KG_ between 4000 and 6000
```

- List the total number of successful and failure mission outcomes

```
%sql select Mission_Outcome, count(*) from SPACEXTABLE group by Mission_Outcome
```

- List all the booster_versions that have carried the maximum payload mass, using a subquery with a suitable aggregate function

```
%sql select distinct(Booster_Version) from SPACEXTABLE where PAYLOAD_MASS__KG_ in (select MAX(PAYLOAD_MASS__KG_) from SPACEXTABLE)
```

- 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

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

EDA with SQL Continuation

- 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(*) from SPACEXTABLE where Landing_Outcome in ('Failure (drone ship)', 'Success (ground pad)')
    and Date between '2010-06-04' and '2017-03-20' group by Landing_Outcome
```

https://github.com/fabianhq04/Final-Report-SpaceX-Data-modeling/blob/209ded9bf8ba7e1647f9db571eaef2059d4ec179/jupyter-labs-eda-sql-coursera_sqllite.ipynb

Build an Interactive Map with Folium

By using an important Python tool from Python we are able to generate interactive maps, to present visualization of physic location data to facilitate the special understanding of what we are working with. Attaching also GitHub URL of the completed Folium working notebook.

In this case we have created 3 Maps with Folium:

a. A facilities launch site: in this case the idea is to present a Map wiht all launch sites on the US map, by using makers and orange circles to visually locate what we want to present in the map.

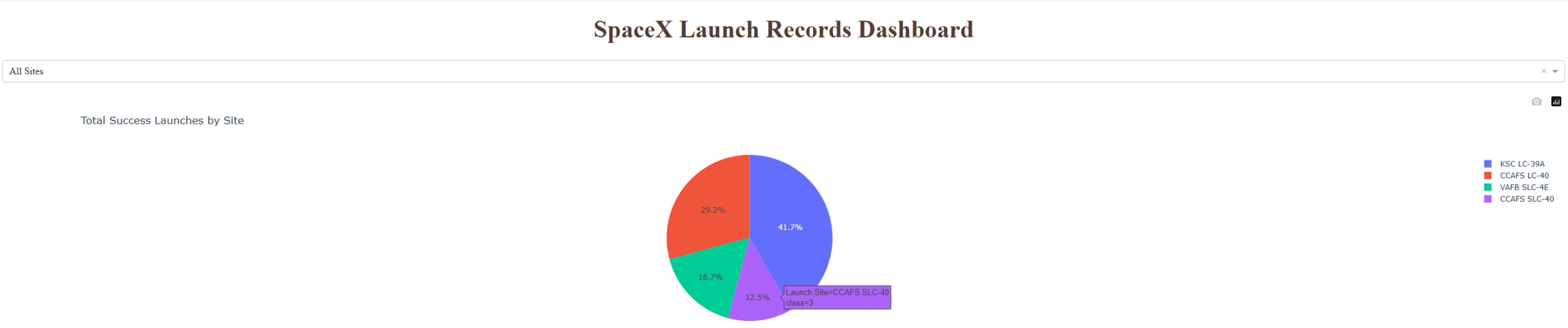
b. Success/Fail launches for each site Map: For this case the idea is to show in the Map all launches class, using makers on red color if the launch was fail and green marker if launch was successful.

c. Distances between launch sites to its proximities map: in this case the idea is to present in the map the ubication for each launch site and the facilities. Once presented using markers as reusing marker types as described in cases a and b, now the idea is to use lines to calculate the distances between the launch sites and the facilities.

https://github.com/fabianhq04/Final-Report-SpaceX-Data-modeling/blob/e397be995eb7f1a287c9cea3f16060d1d1afcc6b/lab_jupyter_launch_site_location.ipynb

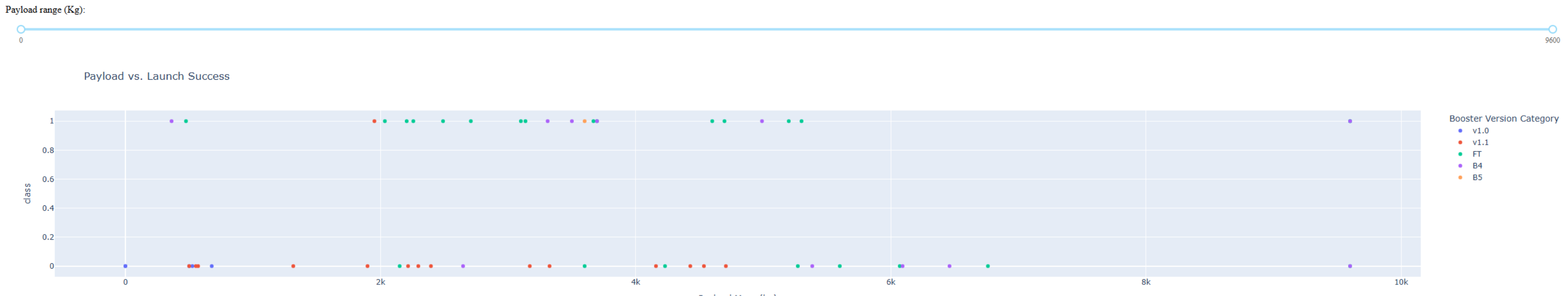
Build a Dashboard with Plotly Dash

- Using Python tools like Plotly and other libraries, we have created an interactive data dashboard, withing data from SpaceX presented in visual-interactive and user friendly way.
- The dashboard contains a Pie chart graph presenting the information about Success Rate lunches per site. Having an interactive dropdown menu, letting the user to select information for any specific site or even all of them. The idea for this presentation is to let the reader know information about the outcome data (Success or Fail launches per site) and to extract valuable information from it.



Build a Dashboard with Plotly Dash Continuation

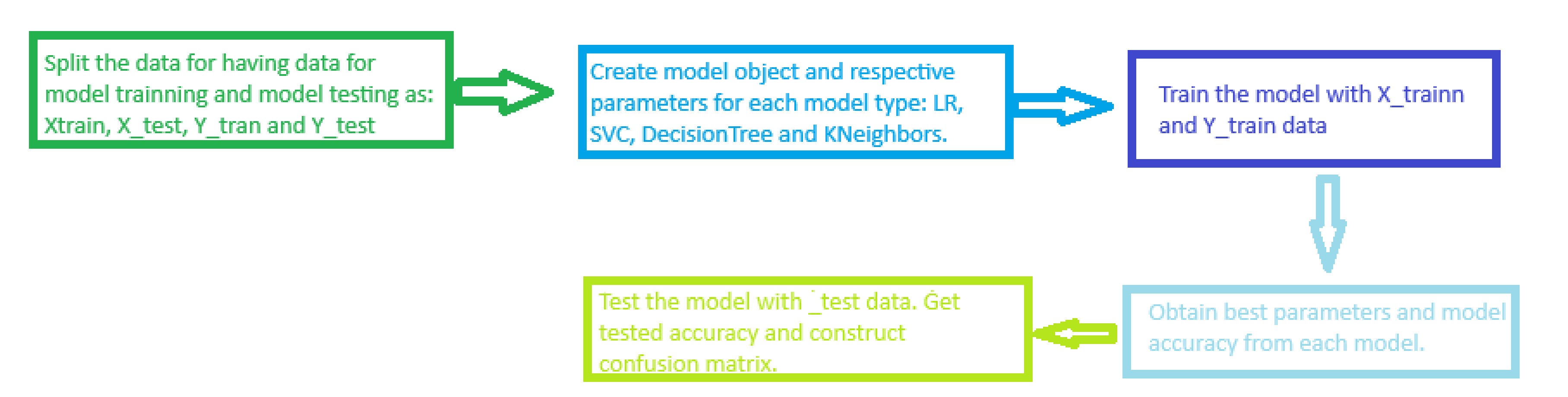
- Also, the dashboard includes a scatter plot, presenting information of Payload Mass (Kg) vs Class (Fail =0, success =1) and having extra interaction where data is differenced by colors, using as classifier the Booster Version Category. Offering to the reader, an extra interaction with the data, having a slide bar where the Payload range can be adjusted to present only selected Mass wanted. The idea of this section is to let the reader know information about the correlation between Payload Mass and how it is related to the Class, being able to classify the data per booster Version also.



https://github.com/fabianhq04/Final-Report-SpaceX-Data-modeling/blob/765b44bd5e52e009a651c877f44435e3c70a86dc/spacex-dash-app%20(1).py

Predictive Analysis (Classification)

- In order to give an answer to the question: Are we able to predict with high precision if the first stage will be having a success landing or not, given a set of conditions?. We have developed 4 predictive models using different methodologies, each model was evaluated and finally compared to each other. Finding the best fit and best performance model that is able to present best results with a measurable way, comparing the accuracy.
- After data is standardized and prepared for module creation, the process used for each model creation is presented in next flow chart.



Predictive Analysis (Classification) Continuation

- Each model constructed is automated to return the best fit object with the parameters defined, so it is guarantied that is the best fit possible for each model construction.
- The models are valuated by calculating the accuracy twice, first using the model parameters and construction (_train data) and finally the accuracy for _test data.

https://github.com/fabianhq04/Final-Report-SpaceX-Data-modeling/blob/1204511c967e38874e599fa6ee3d870ead83b72f/SpaceX_Machine%20Learning%20Prediction_Part_5.ipynb

Results

- On Exploratory data analysis we were able to find very important results, like being able to identify which variable have the most impact on the final outcome (Class, as success or fail landing). And how those variables are related to each other and how they can influence the model with different weights. For example, found that variables like Payload Mass or Booster Version are examples of variables with most impact on final outcome.
- The interactive dashboard created is an important key result and crucial tool for reader/user, as it let the user to visualize the data in interactive way. Being able to identify how the SpaceX Launch records data is spread. There we can extract the successful rate per site, or also the relation between Payload Mass vs Class. Being able to identify which sites have the best or worst success rate or identify which Payload Mass range is the one with better rate, or even to identify which Booster version is the one having best results.
- Site with best success rate is KSC LC-39A, site with worst rate is CCAFC SLC-40.
- The Payload Mass range with better rate is 3000 to 4000 Kg with 70% rate.
- The best Booster version (Having most success rate) is FT.

Results Continuation

- On Predictive analysis, 4 models were developed. Presented here with respective outcome:
- Logistic regression Model: Train accuracy = 85%; Test Accuracy = 83%.
- Support Vector Machine Model: Train accuracy = 86%; Test Accuracy = 83%.
- Decision Tree classifier Model: Train accuracy = 89%; Test Accuracy = 77%.
- K Nearest Neighbors Model: Train accuracy = 86%; Test Accuracy = 77%.

- As final outcome, the best model fit is Support Vector Machine Model since the accuracy trained is high enough (not indicating overfeeding) and the test accuracy still higher that the other ones.

Section 2

Insights drawn from EDA

Flight Number vs. Launch Site

- Failures (blue) are more common in earlier flight numbers, across all sites.
- Successes (orange) dominate in later flights, showing improvement over time.
- KSC LC 39A seems to have the highest success rate, followed by VAFB SLC 4E, while CCAFS SLC 40 had the roughest start with many failures.

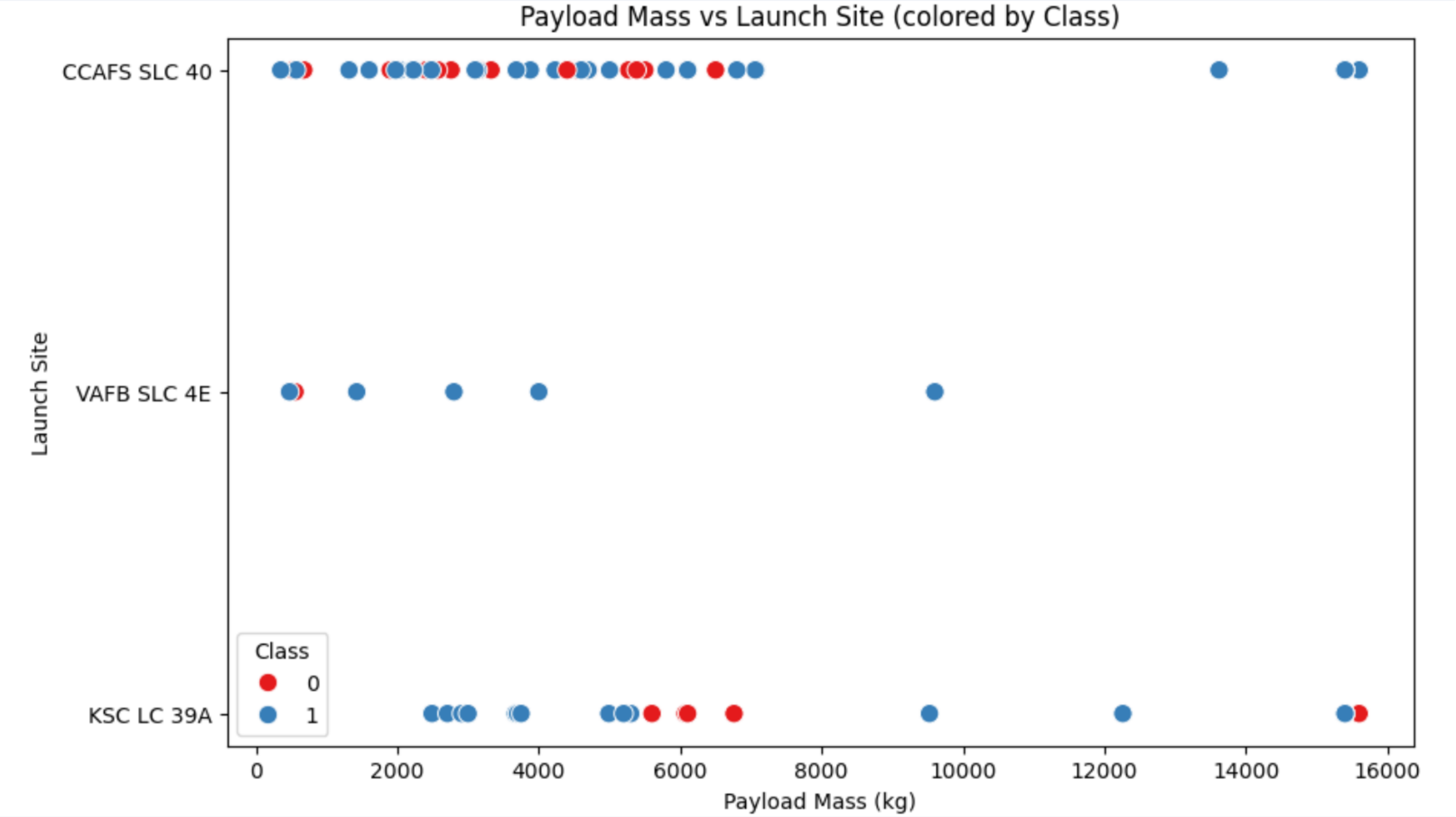
Payload vs. Launch Site

•CCAFS SLC 40 has the widest variation in outcomes (less predictable).

•VAFB SLC 4E looks more reliable (mostly successes).

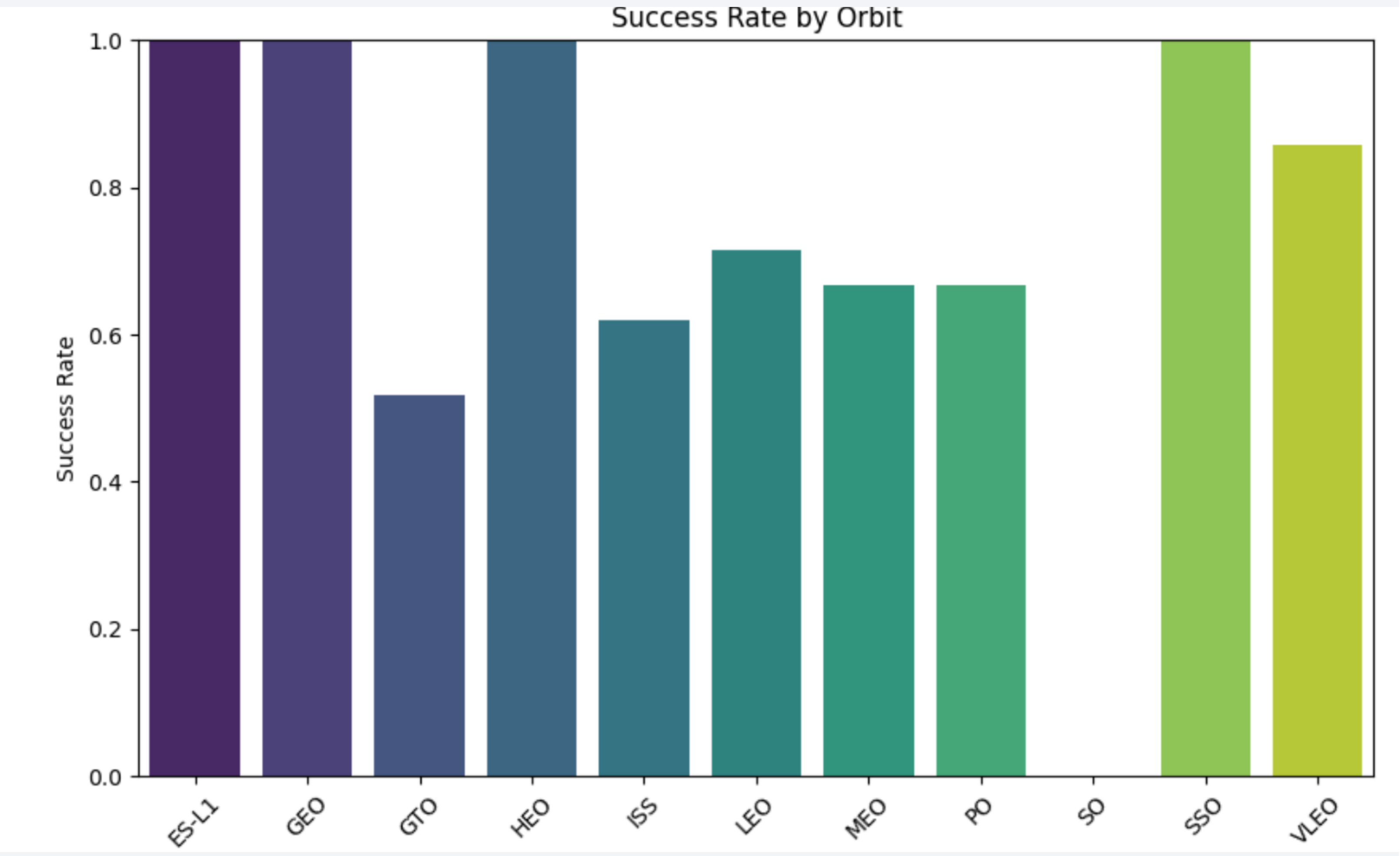
•KSC LC 39A shows success at higher payloads.

•Failure Patterns: Failures (red) cluster more in the mid-to-heavy payload range at CCAFS and KSC, but rarely at VAFB.



Success Rate vs. Orbit Type

- Orbits: ES-L1, GEO, HEO and SEO have rate 100%
- SO orbit is 0% and GTO is 50% rate.



Flight Number vs. Orbit Type

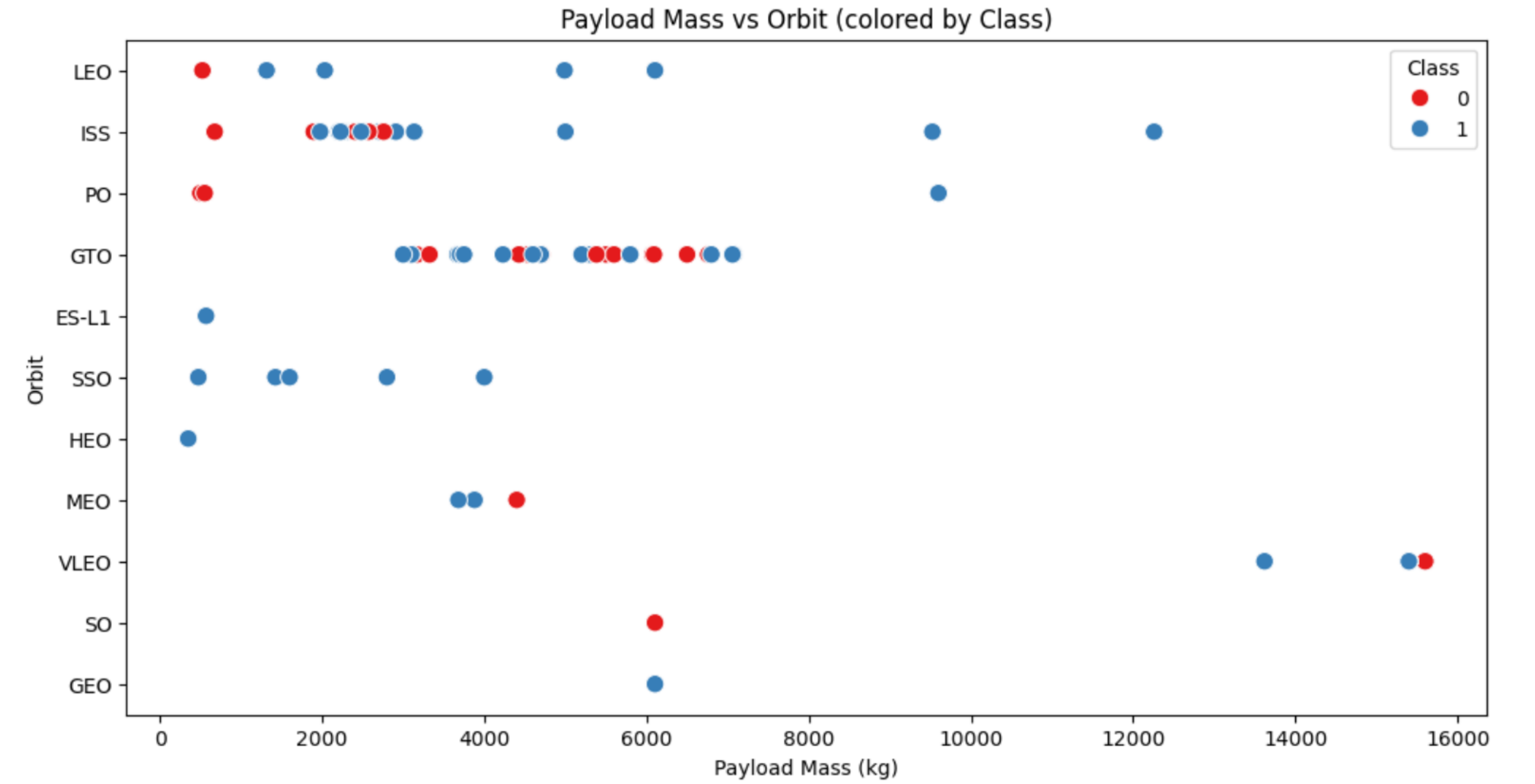
- In the LEO orbit, success seems to be related to the number of flights

- In the GTO orbit, there appears to be no relationship between flight number and success

Payload vs. Orbit Type

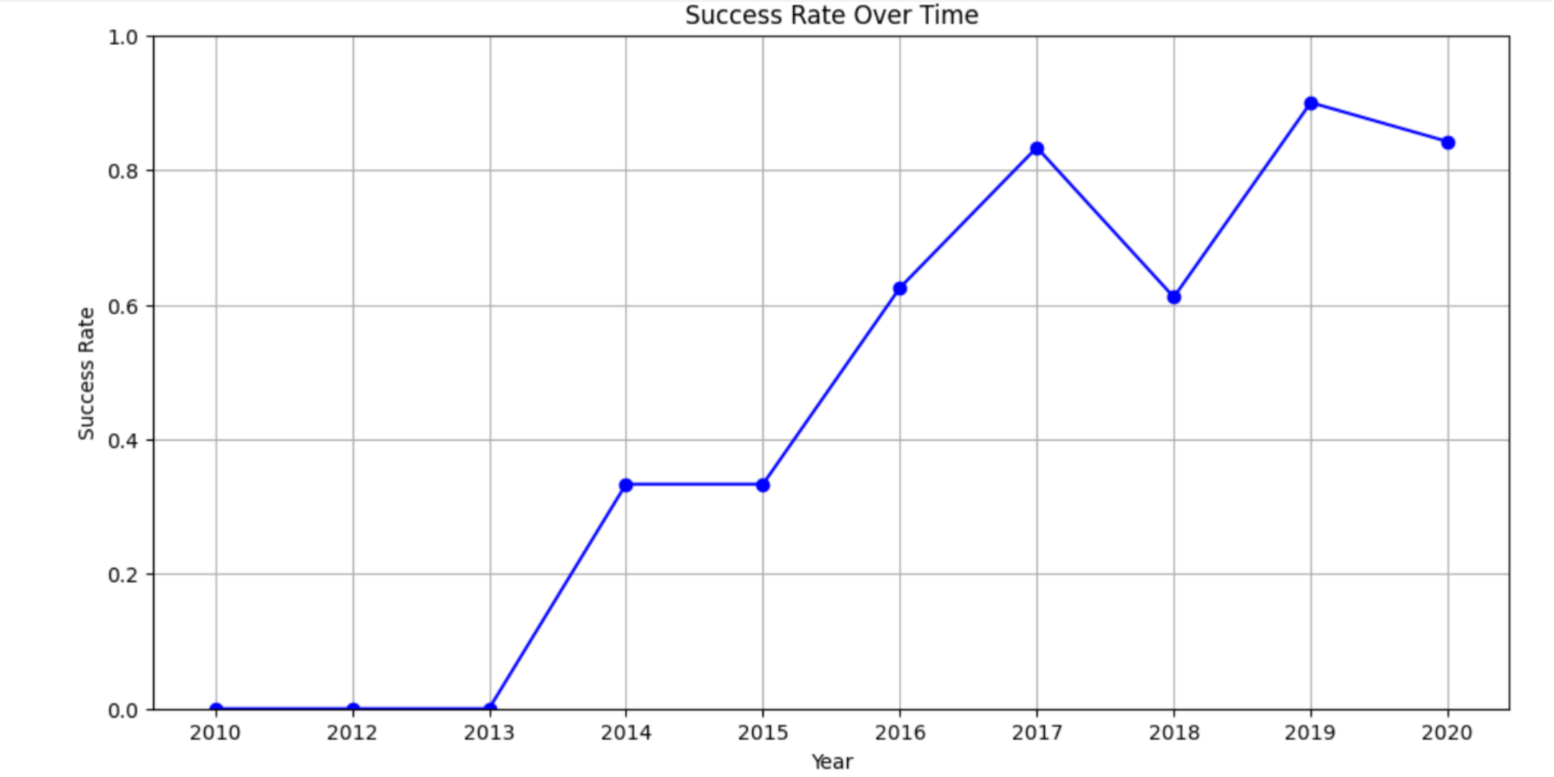
- With heavy payloads the successful landing or positive landing rate are more for Polar, LEO and ISS.

- For GTO, it's difficult to distinguish between successful and unsuccessful landings as both outcomes are present.



Launch Success Yearly Trend

- Success rate since 2013 kept increasing till 2020.
- 2018 had small fall, butgeneral trending still increasing after that.



All Launch Site Names

- Selecting all distinct site names:

```
In [10]: %sql select distinct Launch_Site from SPACEXTABLE

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

Out[10]:

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`
- Just using launch site like "CCA%" and limit the result to 5.

In [11]:
```
%sql select * from SPACEXTABLE where Launch_Site like 'CCA%' limit 5
```

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

Out[11]:

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

- Calculate the total payload carried by boosters from NASA
- Selecting Payload data only for cases where customer = “NASA (CRS)”, and applying SUM to the data

```
In [12]: %sql select SUM(PAYLOAD_MASS__KG_) from SPACEXTABLE where Customer = 'NASA (CRS)'

 * sqlite:///my_data1.db
Done.

Out[12]:
```

SUM(PAYLOAD_MASS__KG_)
45596

Average Payload Mass by F9 v1.1

- Calculate the average payload mass carried by booster version F9 v1.1
- Select the Mass Payload for Booster version = “F9 v1.1” and apply AVG

```
In [13]: %sql select AVG(PAYLOAD_MASS__KG_) from SPACEXTABLE where Booster_Version = 'F9 v1.1'

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

Out[13]:

AVG(PAYLOAD_MASS__KG_)
2928.4

First Successful Ground Landing Date

- Find the dates of the first successful landing outcome on ground pad
- Select the MIN(Date), where the landing outcome is “Success (ground pad)”

```
In [14]: %sql select MIN(Date) from SPACEXTABLE where Landing_Outcome = 'Success (ground pad)'

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

Out[14]:

MIN(Date)
2015-12-22

Successful Drone Ship Landing with Payload between 4000 and 6000

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

- Selecting the Booster version, by applying conditions Landing_Outcome = 'Success (drone ship)' and PAYLOAD_MASS__KG_ between 4000 and 6000

```
[20]: %sql select distinct Booster_Version from SPACEXTABLE where Landing_Outcome = 'Success (drone ship)' and PAYLOAD_MASS__KG_ between 4000 and 6000

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

[20]:

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

Total Number of Successful and Failure Mission Outcomes

- Calculate the total number of successful and failure mission outcomes
- Select by applying function count(), while grouping by mission Outcome

```
[22]: %sql select Mission_Outcome, count(*) from SPACEXTABLE group by Mission_Outcome

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

[22]:

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

Boosters Carried Maximum Payload

- List the names of the booster which have carried the maximum payload mass
- Perform a nested select extracting the max payload, using this result to look for the distinct booster versions that carried such payload

```
[26]: %sql select distinct(Booster_Version) from SPACEXTABLE where PAYLOAD_MASS__KG_ in (select MAX(PAYLOAD_MASS__KG_) from SPACEXTABLE)

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

[26]:

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

- List the failed landing_outcomes in drone ship, their booster versions, and launch site names for in year 2015
- Selecting the required information, but using function substr(Date,a,b) to extract the month and the year from the date. As substr(Date,0,5)='2015' for the year and substr(Date,6,2) for the Month.

```
[30]: %sql select substr(Date, 6,2) as Month, Launch_Site, Booster_Version, Landing_Outcome
         from SPACEXTABLE where substr(Date,0,5)='2015' and Landing_Outcome = 'Failure (drone ship)'

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

[30]:

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

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

- 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
- Grouping by Landing Outcome, and order by the count as desc.

```
•[33]: %sql select Landing_Outcome, count(*) from SPACEXTABLE where Landing_Outcome in ('Failure (drone ship)', 'Success (ground pad)')
           and Date between '2010-06-04' and '2017-03-20' group by Landing_Outcome order by count(*) desc

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

[33]:

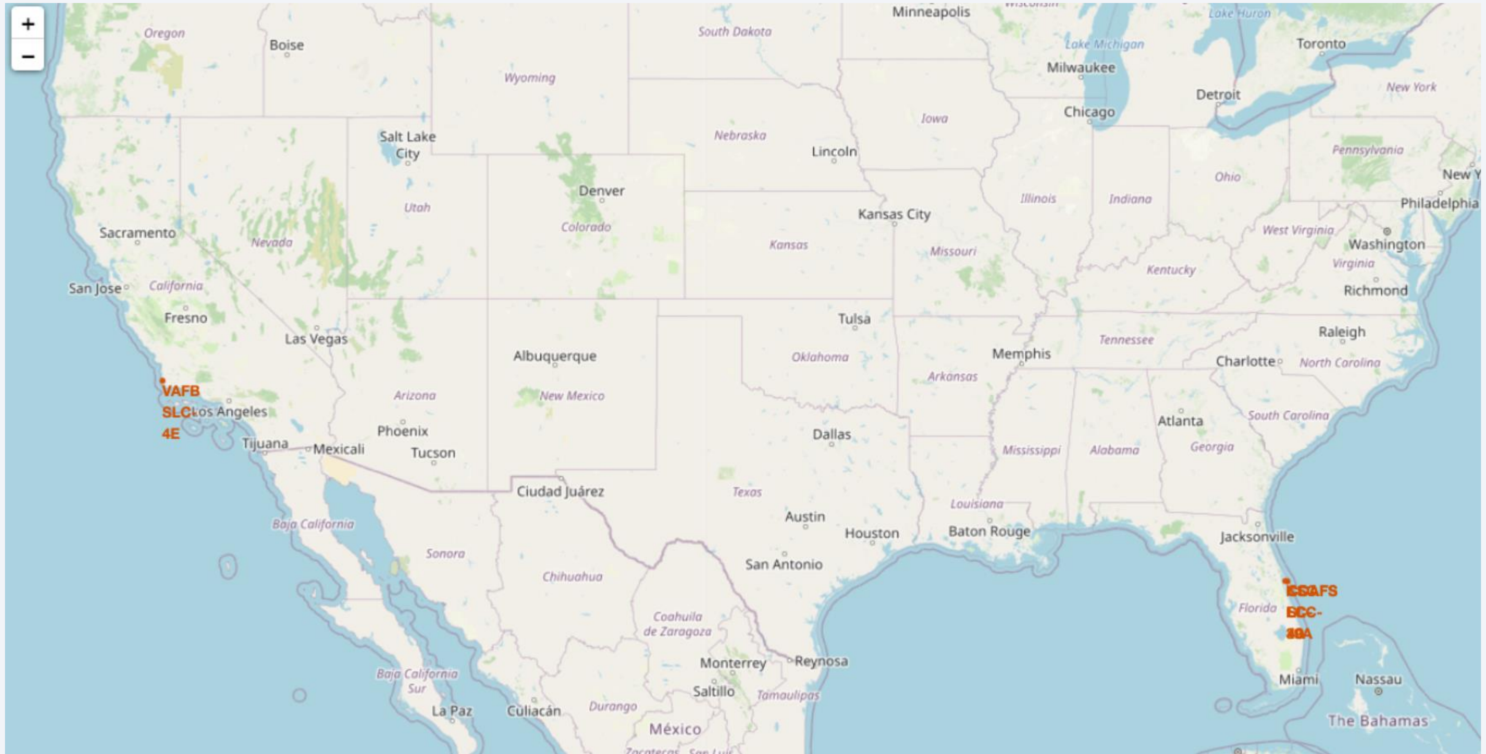
Landing_Outcome	count(*)
Failure (drone ship)	5
Success (ground pad)	3

Section 3

Launch Sites Proximities Analysis

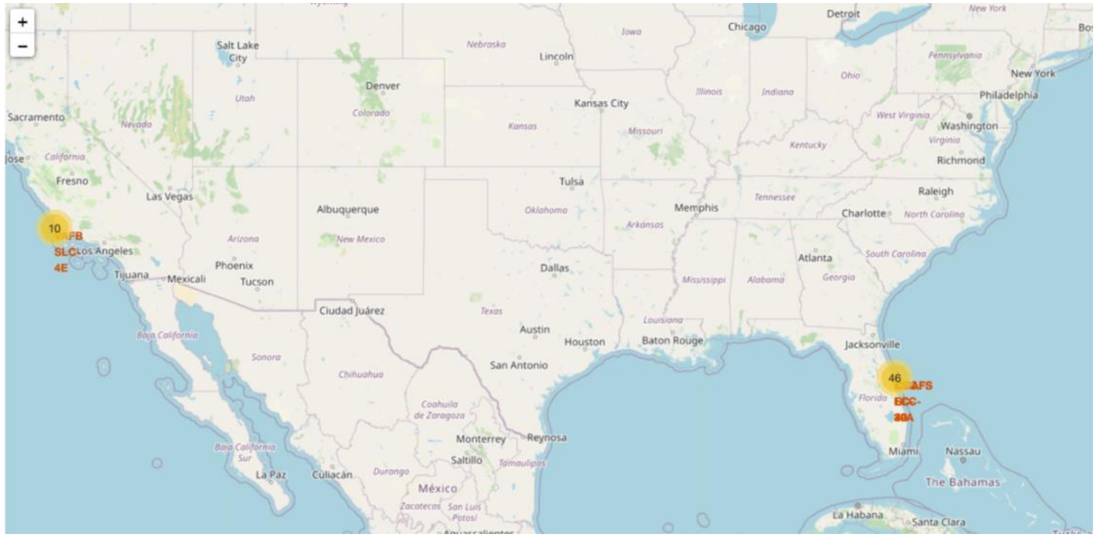
Site locations on the Map

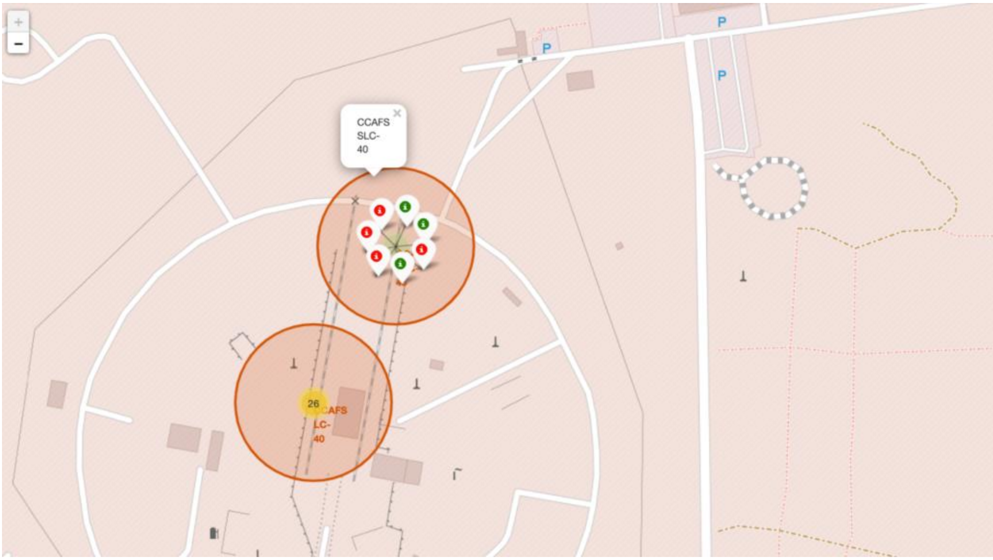
- The main topic on this map is to present all site locations as presented in the data, in visualization mode on the map.
- The sites are marked using markers as circles on orange color, by every single site location (coordinates).



Success/Failed Launches for each Site on the Map

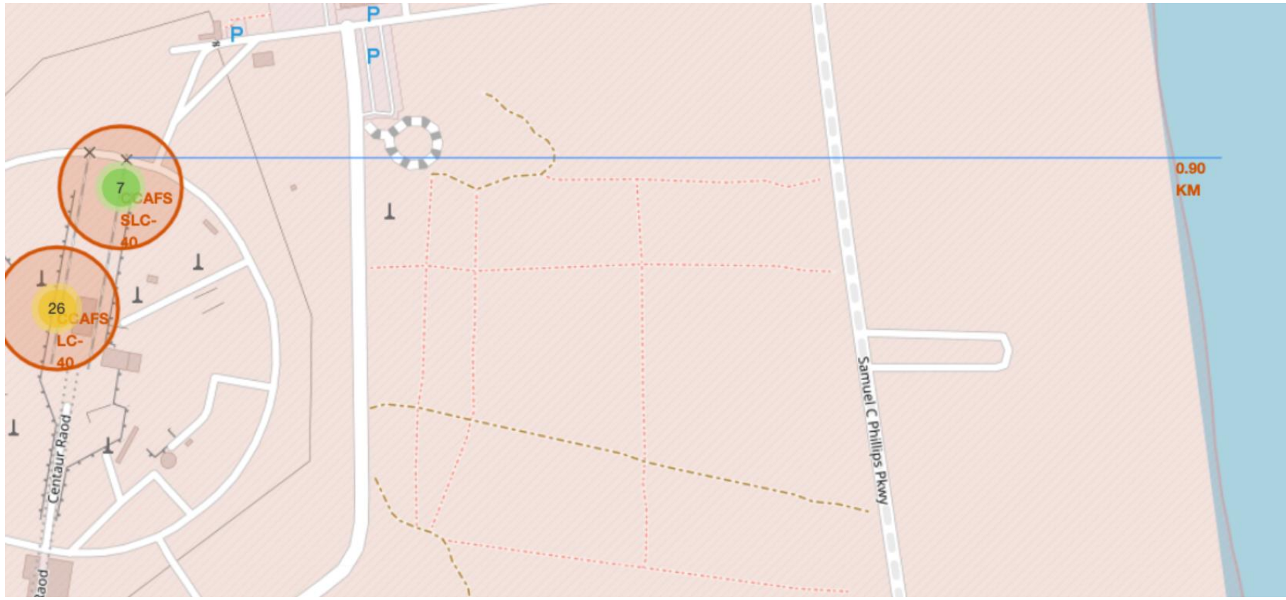
- The main topic on this map is to present all site locations as presented in the data, in visualization mode on the map, s in previous case.
- After that, remark every single launch location, in green if it is class =1 and red if it is class =0. In this case, we are using makers and labels according to the color/class to visually identify every case.





Success/Failed Launches distance to facilities on the Map

- The main topic on this map is to present all site locations as presented in the data, in visualization mode on the map, s in previous case.
- After that, remark every single launch location, in green if it is class =1 and red if it is class =0. In this case, we are using makers and labels according to the color/class to visually identify every case as in previous case.
- Finally, adding lines to the map to measure and show the distance from each launch location to the facilities location.



Section 4

Build a Dashboard with Plotly Dash

SpaceX launch Records information for all sites

- **KSC LC-39A (Blue) – 41.7%**

This site has the largest share successful launches.
Nearly half of all launches happened here, showing it's a major hub for SpaceX missions.

- **CCAFS LC-40 (Red) – 29.2%**

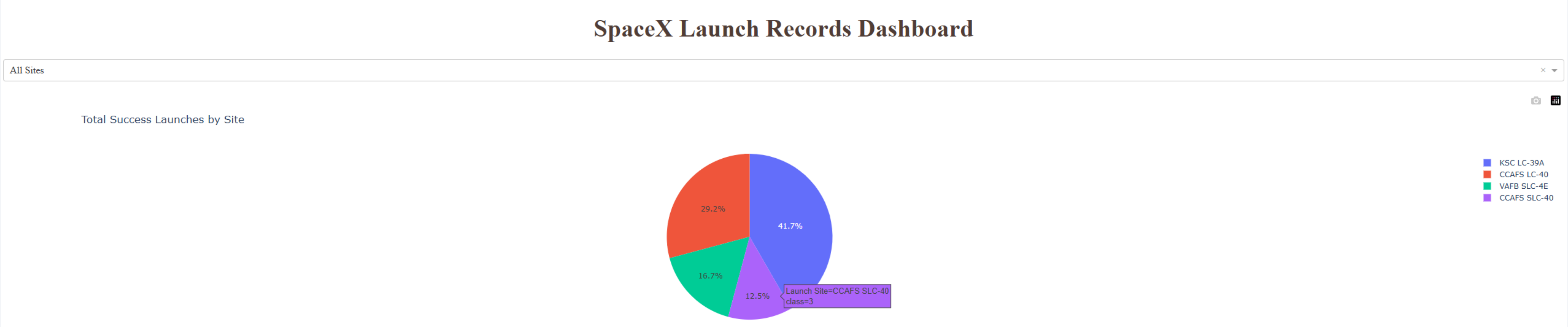
Second most successful site, with almost a third of the launches.
Important, but not as dominant as LC-39A.

- **VAFB SLC-4E (Green) – 16.7%**

Fewer launches overall but still contributes a meaningful share of launches.
Historically used for polar orbit launches.

- **CCAFS SLC-40 (Purple) – 12.5%**

This is the one with smallest success launch count.

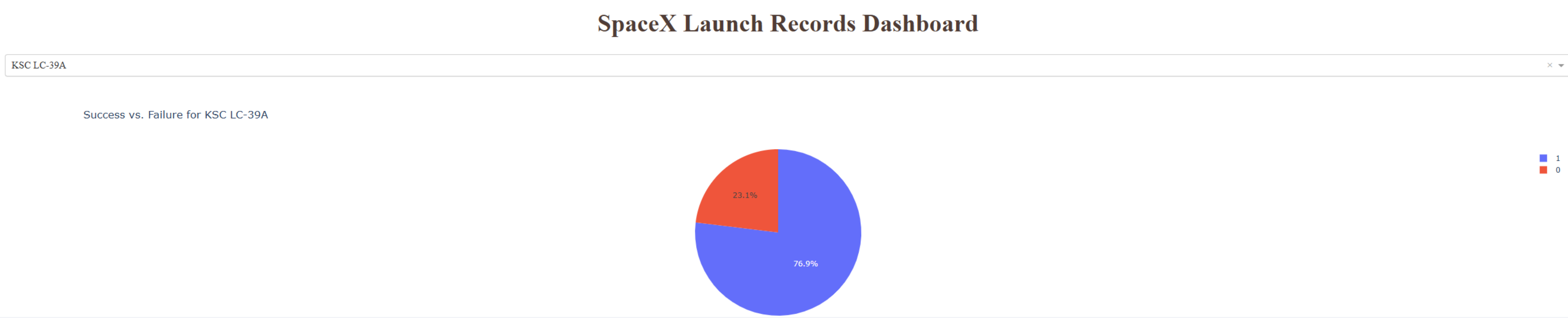


Site with higher successful ratio

- **KSC LC-39A (Blue) – 41.7% overall**

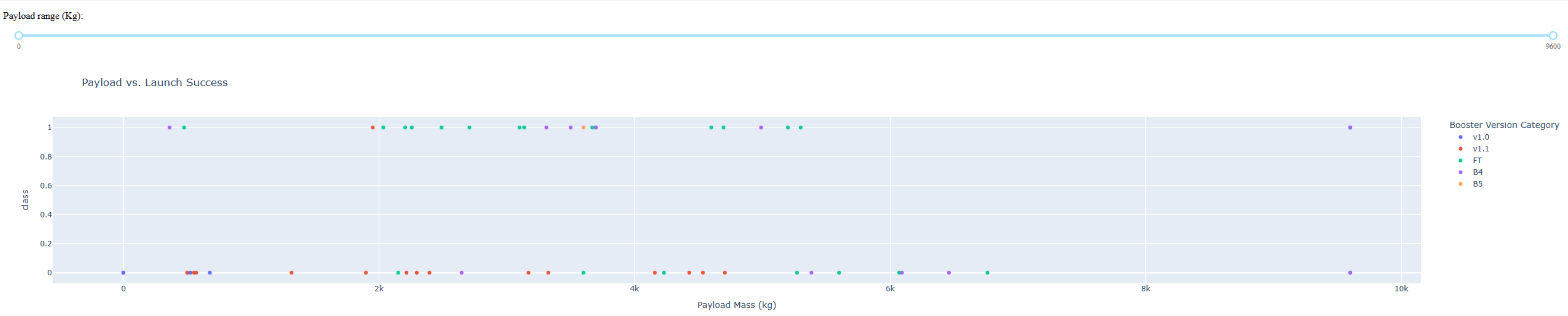
This site has the largest share successful launches.
Nearly half of all launches happened here, showing it's a major hub for SpaceX missions.

Presenting a ration of 76.9% of success vs fail launches, highest ration between sites.



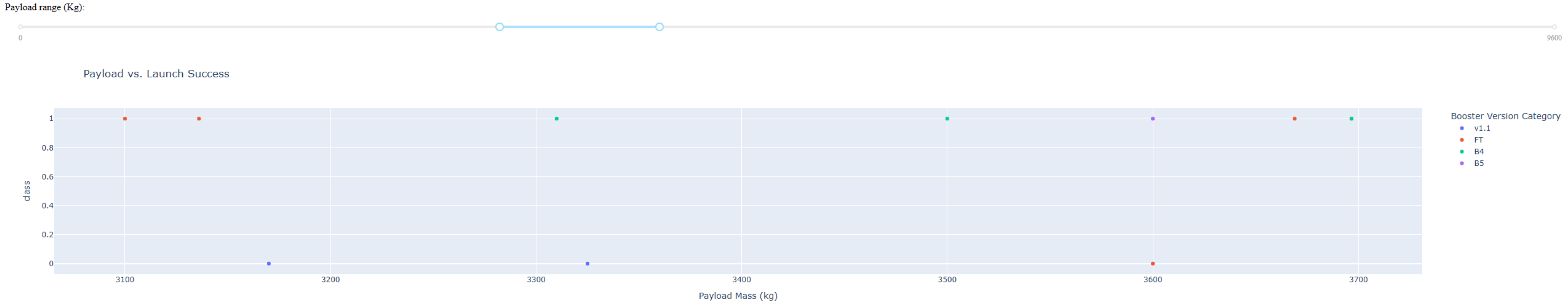
Payload vs Launch Success for all Booster Versions

- The dashboard includes a scatter plot, presenting information of Payload Mass (Kg) vs Class (Fail =0, success =1) and having extra interaction where data is differenced by colors, using as classifier the Booster Version Category.
- Here we can see how the Booster version with higher success rate is FT



Payload vs Launch Success for all Booster Versions Continuation

- We can appreciate how the range with better success rate for Payload Mass is between 3000 and 4000 Kg



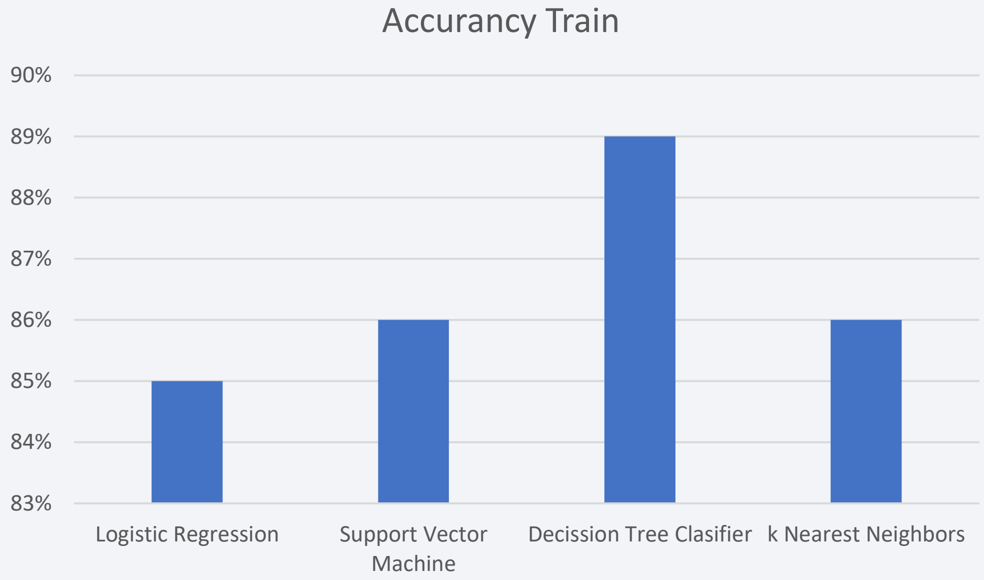
Section 5

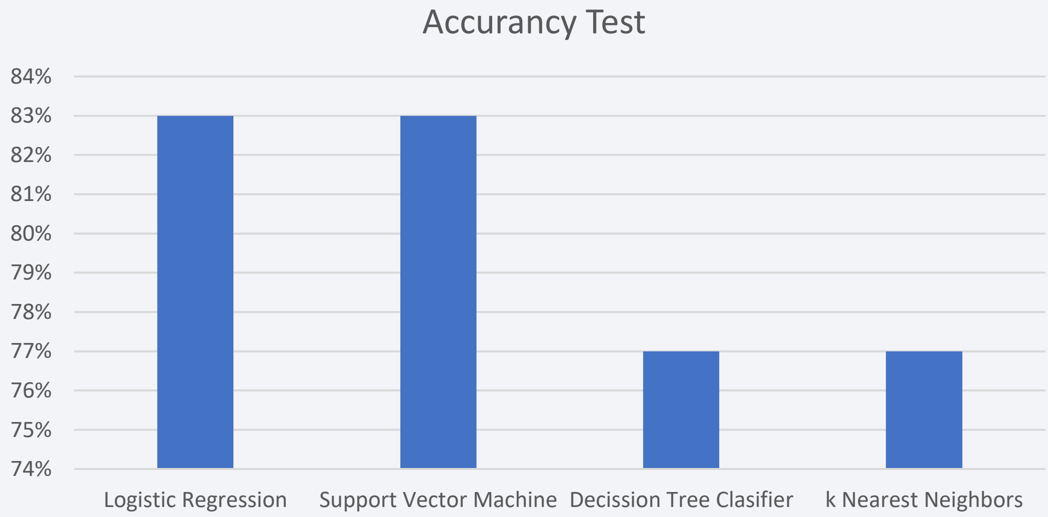
Predictive Analysis (Classification)

Classification Accuracy

- On Predictive analysis, 4 models were developed. Presented here with respective outcome:
- Logistic regression Model: Train accuracy = 85%; Test Accuracy = 83%.
- Support Vector Machine Model: Train accuracy = 86%; Test Accuracy = 83%.
- Decision Tree classifier Model: Train accuracy = 89%; Test Accuracy = 77%.
- K Nearest Neighbors Model: Train accuracy = 86%; Test Accuracy = 77%.

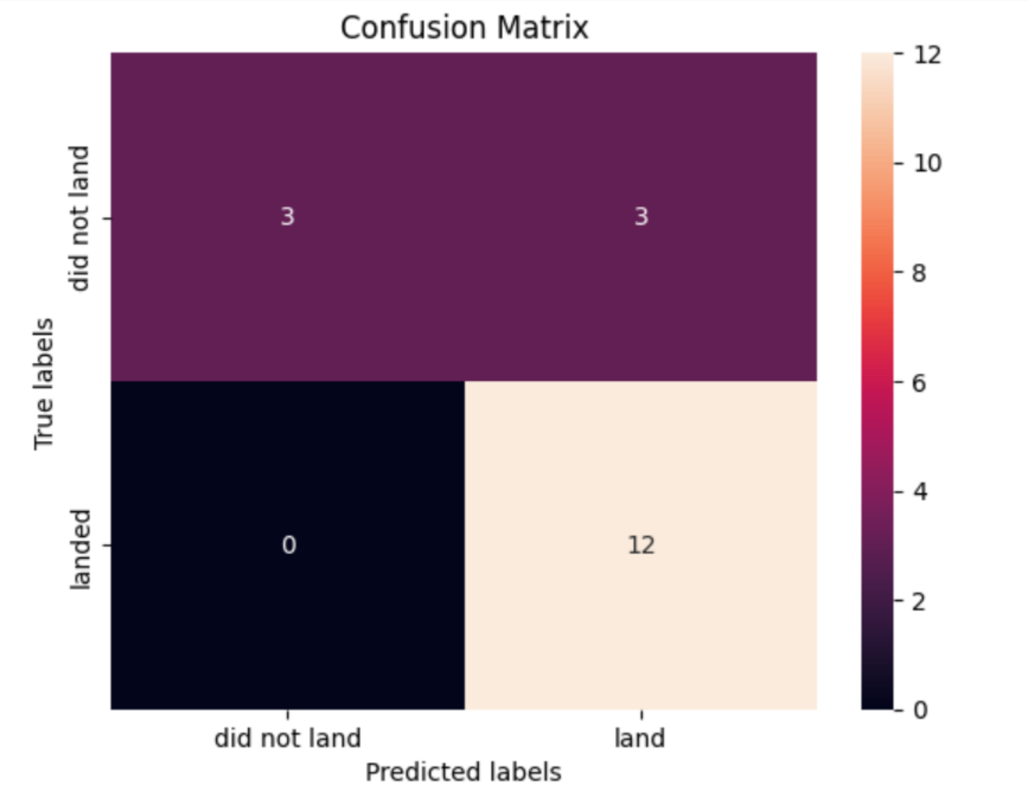
- As final outcome, the best model fit is Support Vector Machine Model since the accuracy trained is high enough (not indicating overfeeding) and the test accuracy still higher that the other ones.





Confusion Matrix for SVC Model

- Presenting the confusion Matrix for the model with better metrics and the one selected, in this case Support Vector Machine Model.



•**Accuracy** = =0.833 (83%)
•**Precision (Positive Predictive Value)** = 0.80 (80%)
•**Recall (Sensitivity, TPR)** =1.00 (100%)
•**Specificity (TNR)** = 0.50 (50%)
•**F1 Score** = 2·Precision·RecallPrecision+Recall=0.89

Conclusions

- SpaceX data needs some wrangling before being able to use it for prediction model construction.
- Exploratory data is crucial for data understanding, so we can evaluate what variables can be used and what variable are irrelevant.
- Data presentation in interactive way, as dashboards are key items for understanding true meaning of the data.
- Several Prediction models can be constructed for prediction.
- We can chose the best performing model based on accuracy values.
- Are we able to predict with high precision if the first stage will be having a success landing or not, given a set of conditions?

The answer is Yes, by selecting the Support Vector Machine Model, we are able to predict the success or fail for landing with 83% accuracy.

Thank you!