IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

<Name>
<Date>

Outline

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

Executive Summary

This case involves analyzing data from Falcon 9 launches and predicting the outcome from different launch sites for its versions and checking the characteristics that may influence the outcome. Starting with data collection with the appropriate tools and libraries, continuing with the discussion, processing, exploration and visualization of the data, seeking information and relationships between variables, looking for preparing the data for later modeling and evaluation.

finally we are going to find results and correlations between some features and interesting visualizations that let us see outcomes grouped by site and success rates, some interesting data and the most important, obtain the predictive model for this case, a classification model.

Introduction

The Falcon 9 has five categories, v1.0, v1.1, FT, B4 and B5. Launches are recorded as of June 4, 2010. More information is obtained from the SpaceX API and Wikipedia, the data obtained from these sources is processed in order to analyze some relationships between variables, obtain insights and visualize interesting data.

After that we need to understand the problem and design a strategy to find and answer it, but what is the question to solve?

How can I know what the result of the next release is?

There are two options. Success or failure. So this question and the possible solutions will guide us to think about what type of model we should use and then to find the right model for a good prediction.

Section 1

Methodology

Methodology

Executive Summary

- Data collection methodology:

  Using the appropriate libraries of Python, the data was extracted from API SpaceX and Wikipedia.

- Perform data wrangling

  The data was processed to obtain some measures for columns, and a check any relationship between variables. After that, the table was modified to analyzing and constructing the predictive model.

- Perform exploratory data analysis (EDA) using visualization and SQL

  SQL language was necessary for Exploratory Data Analysis and obtain interesting measures for some variables

Methodology

Executive Summary

- Perform interactive visual analytics using Folium and Plotly Dash

  Visualization through maps and dashboard make easy to view and understand some relationships by aggrupation and location, for example, checking the geographic location of the events.

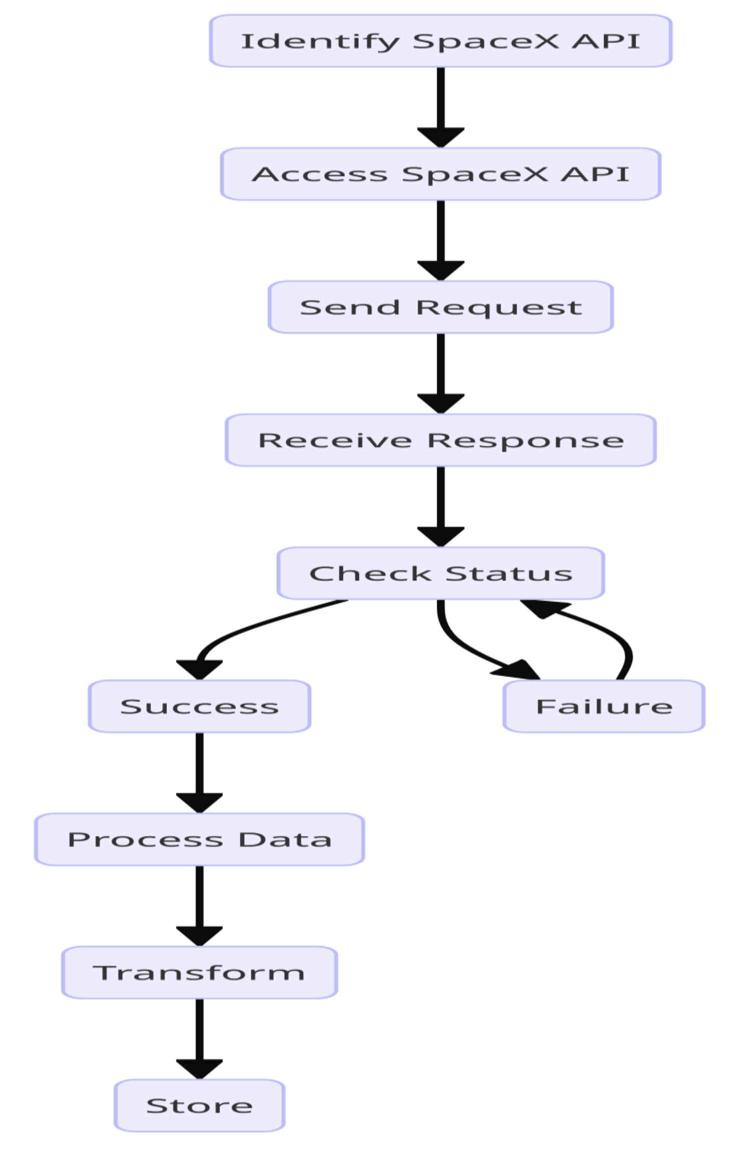
- Perform predictive analysis using classification models
  - Logistic Regression, Support Vector Machine, Decision Tree Classifier and KNeighbors Classifier were considered and evaluated to choose the best perform model.

Data Collection

- From the SpaceX API (‘https://api.spacexdata.com/v4/launches/past'), the data were obtained, processed and cleaned leaving just the ‘LandingPad’ column with None values to represent when landing pads were not used. The set was exported as ‘data_falcon9.to_csv('dataset_part_1.csv', index=False)’.
- By webscraping the The Falcon 9 launch records HTML was obtained from Wikipedia and exported as ‘df.to_csv('spacex_web_scraped.csv', index=False)’

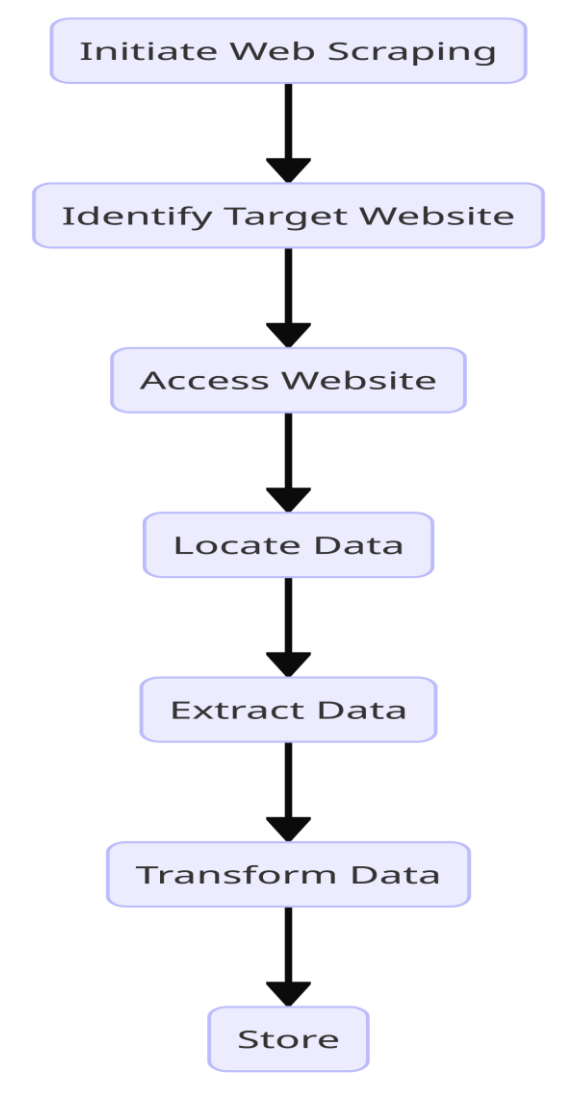
Data Collection – SpaceX API

- The next flowchart summarizes the process to get the data from the API.
- Follow the next URL to read with more details, you will find the notebook with the code step by step: https://github.com/Cupaban1/testrepo/blob/Capstone/jupyter-labs-spacex-data-collection-api.ipynb



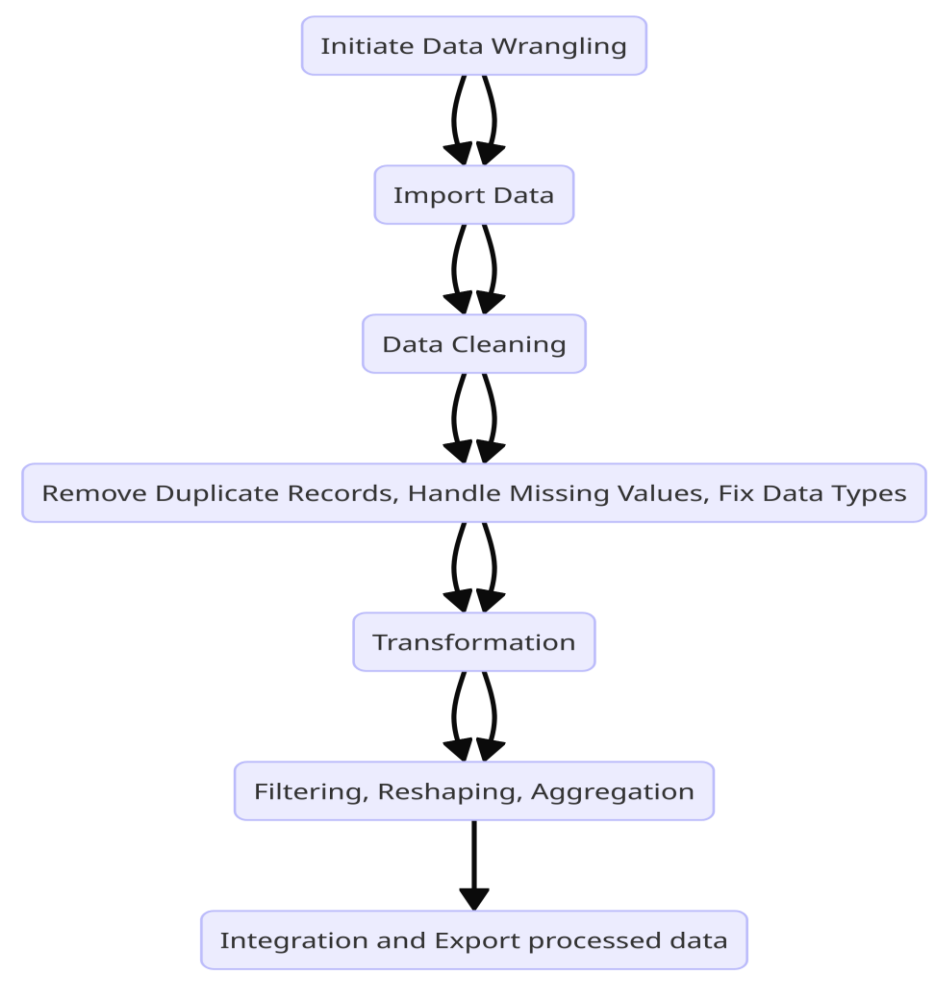
Data Collection - Scraping

- The flowchart shows the logic process to get the data from Wikipedia using Python and its appropriate libraries.
- Follow the next URL to read with more details, you will find the notebook with the code step by step: https://github.com/Cupaban1/testrepo/blob/Capstone/jupyter-labs-webscraping.ipynb



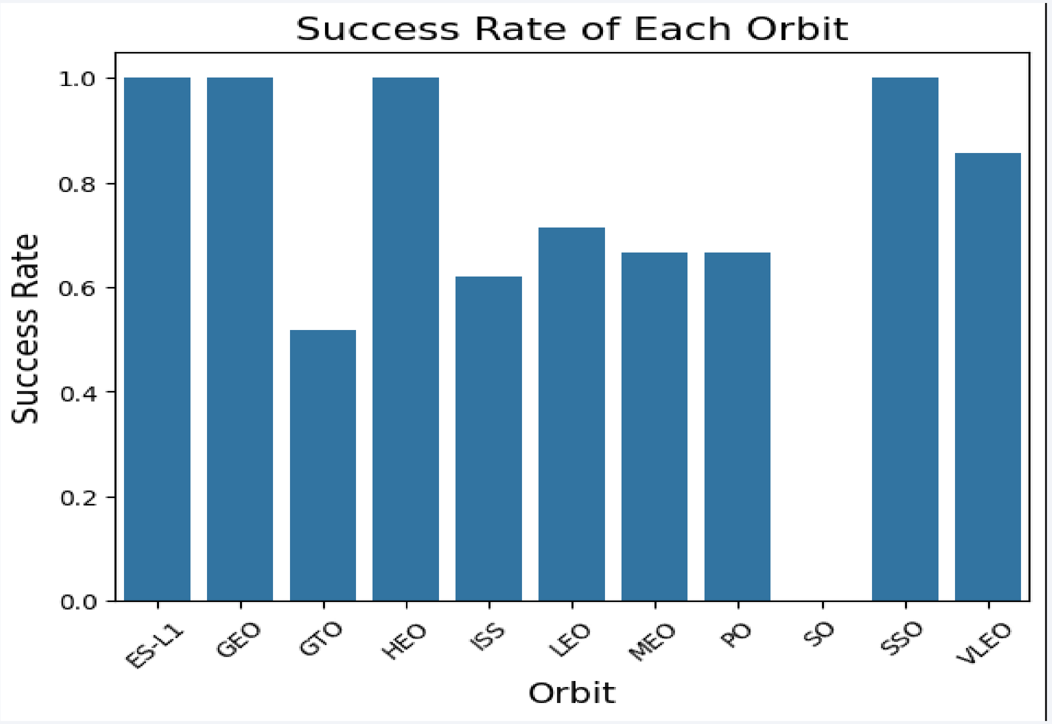
Data Wrangling

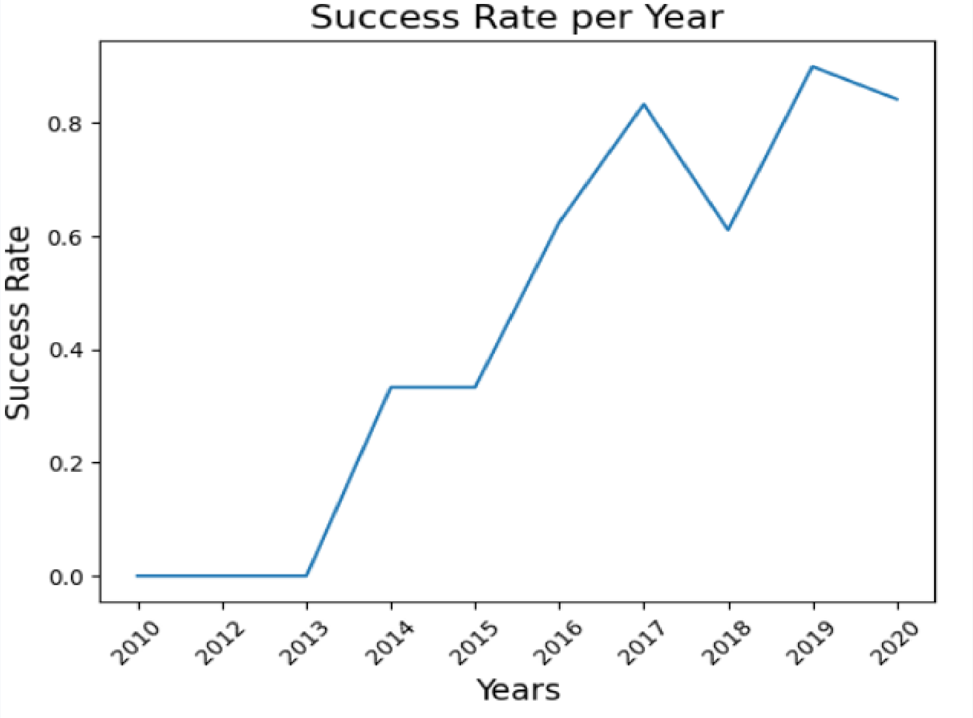
- The data were processed as showed in the flowchart. The data used is the collected from the SpaceX API.
- Follow the next URL to read with more details, you will find the notebook with the code step by step: https://github.com/Cupaban1/testrepo/blob/Capstone/labs-jupyter-spacex-Data%20wrangling.ipynb



EDA with Data Visualization

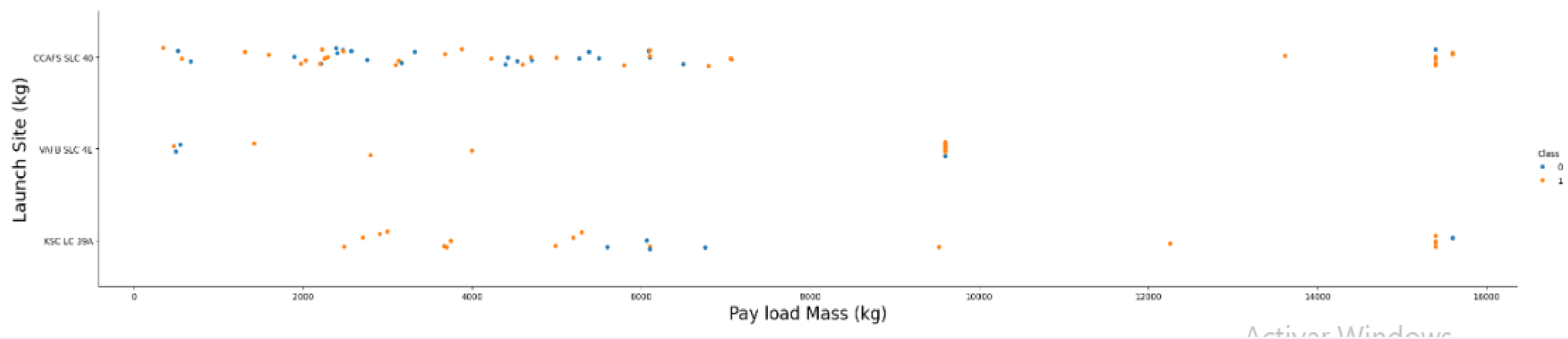
- As we can view, The success rate is higher year after year and there is no correlation between success rate and orbit, so the Orbit is not a variable of interest to study.





EDA with Data Visualization

- If Pay load mass increases, the count of successful launches increases.
- CCAFS SCL-40 has the lowest successful launches count.
- Can we see the KSC LC 39A is the Launch Site with highest success launch rate.



EDA with SQL

- Making some queries we can check details like total number of successful missions and failure missions or Booster versions with the maximum payload

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

total_number_of_successful_missions	total_number_of_failure_missions
100	1

```
 * 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

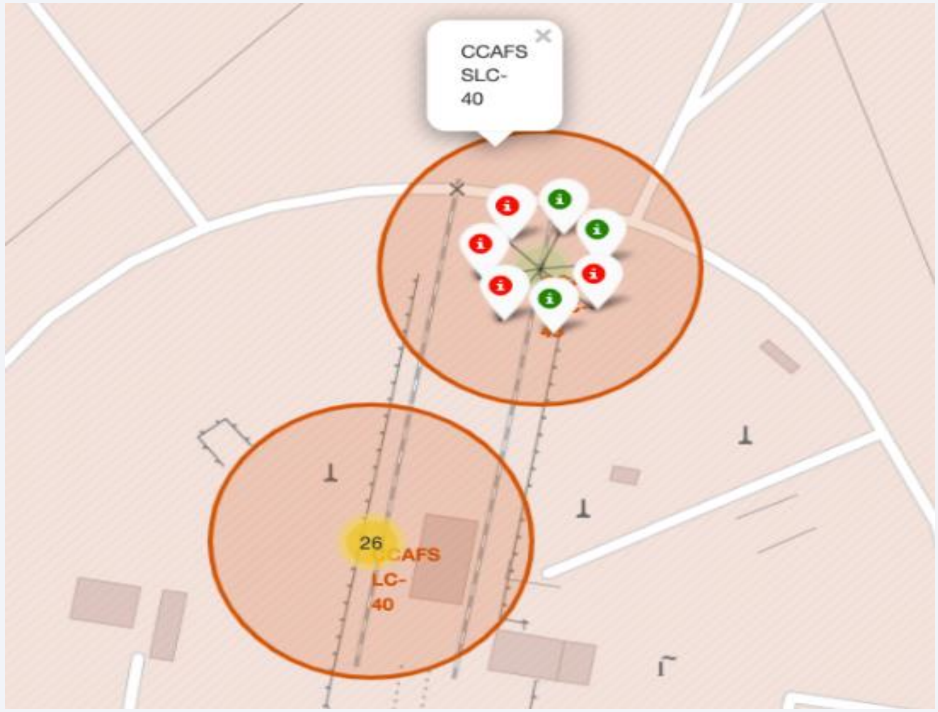
Build an Interactive Map with Folium

- The next graphs show relevant geographical information of the launch sites. These were selected because they can provide information with details about the location in the map.

	Launch Site	Lat	Long
0	CCAFS LC-40	28.562302	-80.577356
1	CCAFS SLC-40	28.563197	-80.576820
2	KSC LC-39A	28.573255	-80.646895
3	VAFB SLC-4E	34.632834	-120.610745

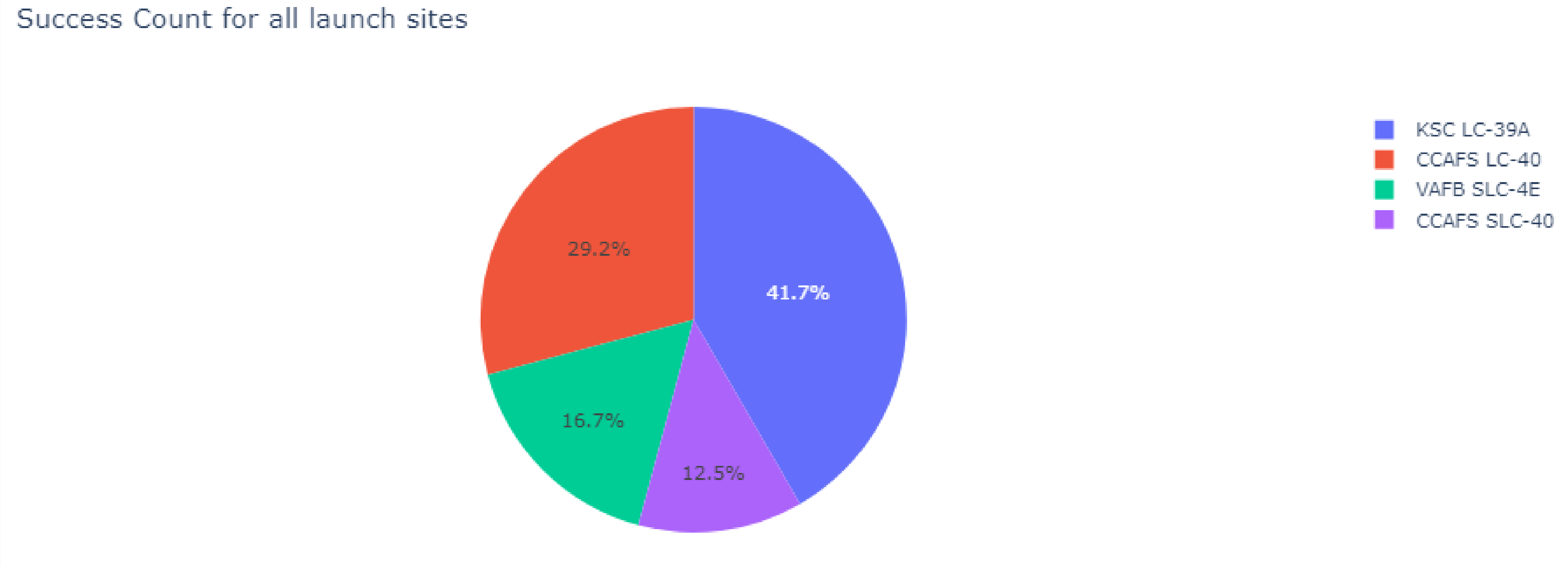
Build an Interactive Map with Folium

For example, we can view the cases per location, in this case CCAF SLC-40

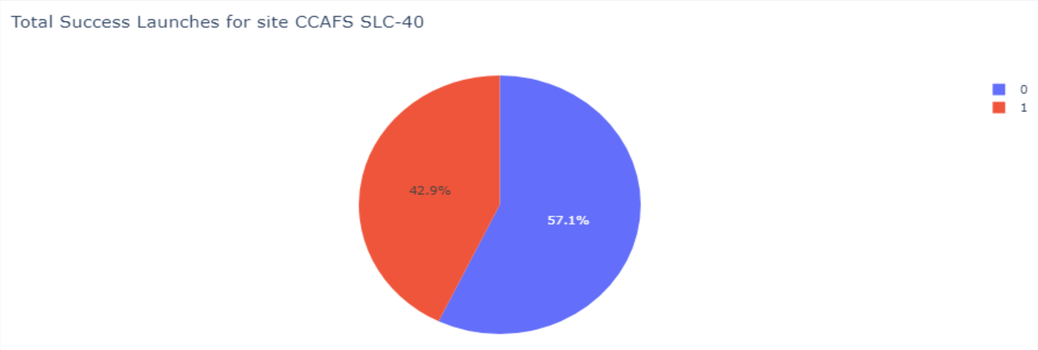


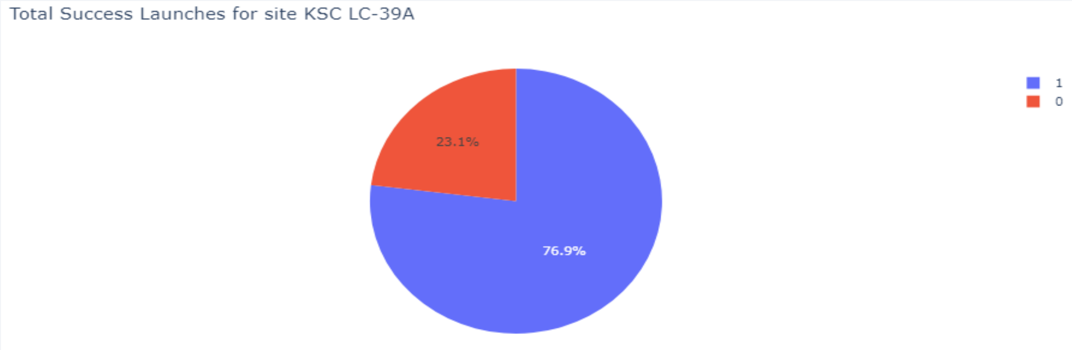
Build a Dashboard with Plotly Dash

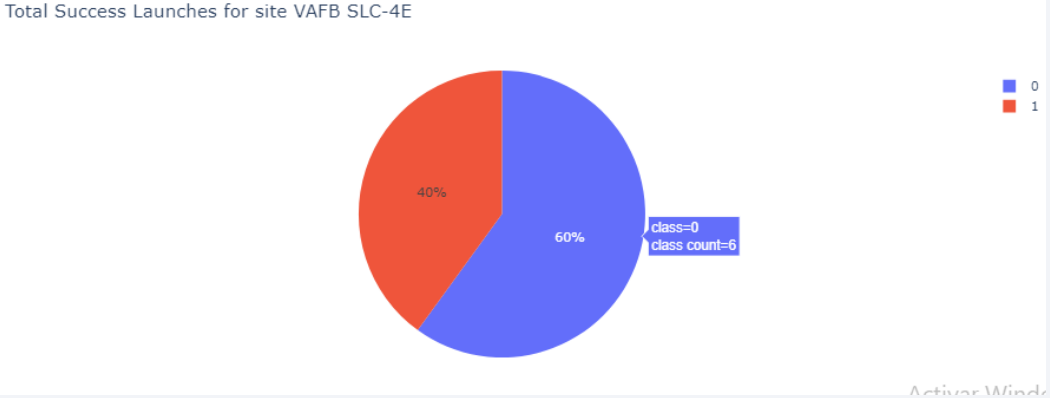
- The next graphs are taken from a dashboard constructed to interact with variables and see relations and results according the launch site.
- This graph shows percentage of success for each site or all sites.

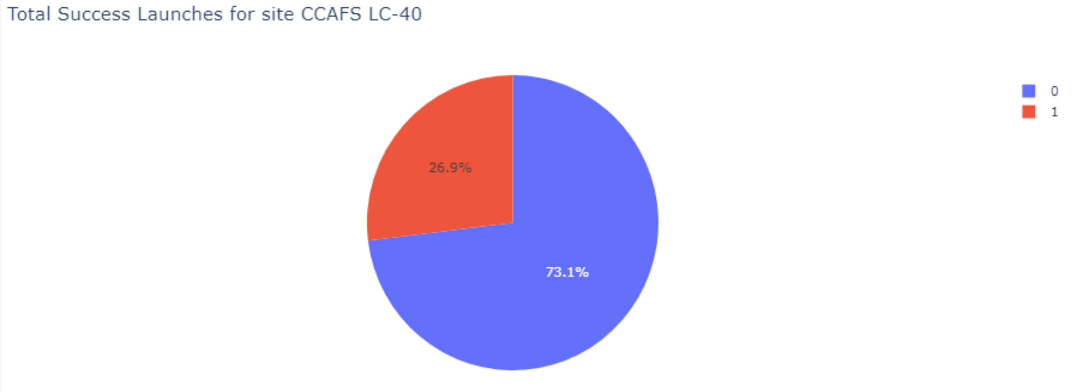


Build a Dashboard with Plotly Dash



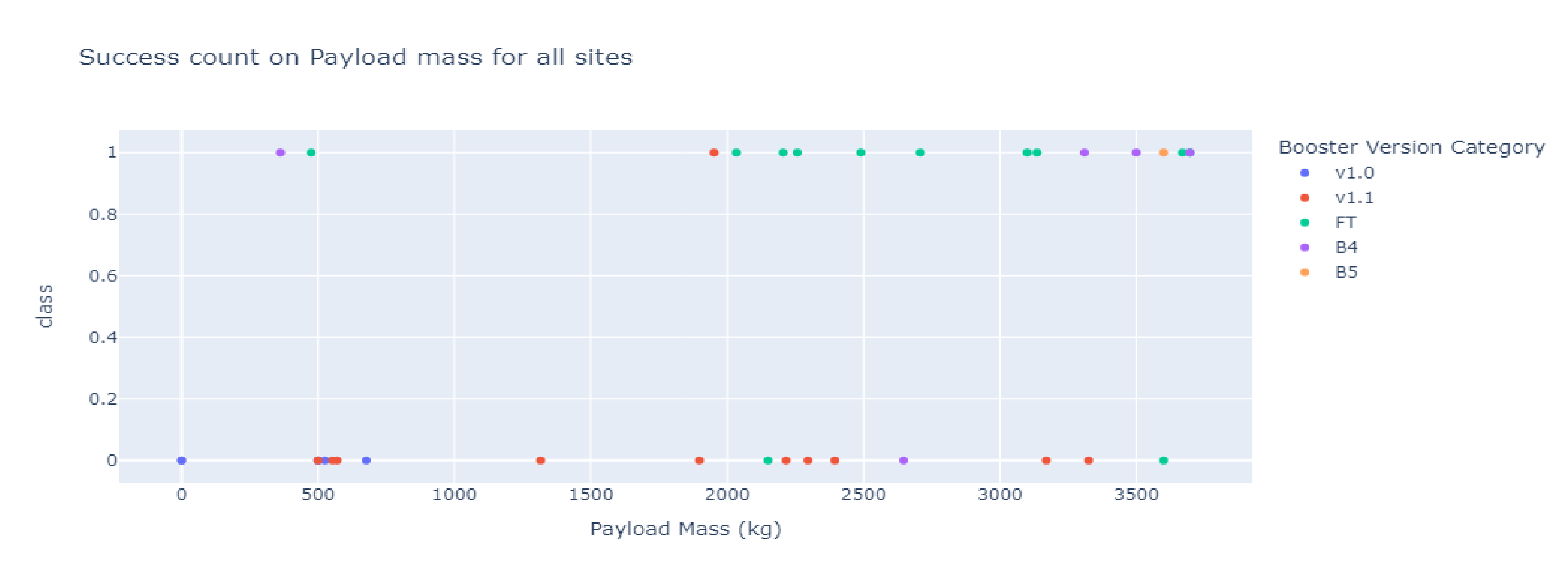






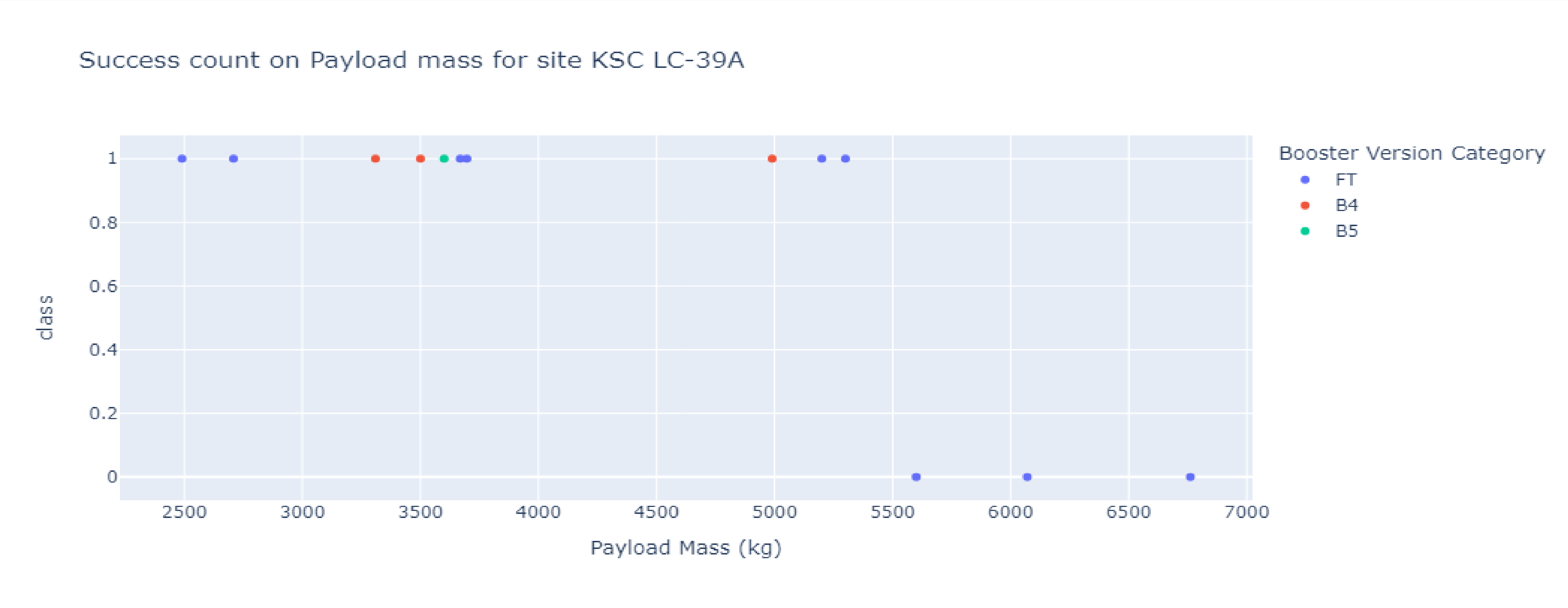
Build a Dashboard with Plotly Dash

- The dashboard let us check the count of success and failures according to Pay load mass for each Booster version category for all launch sites o each one:



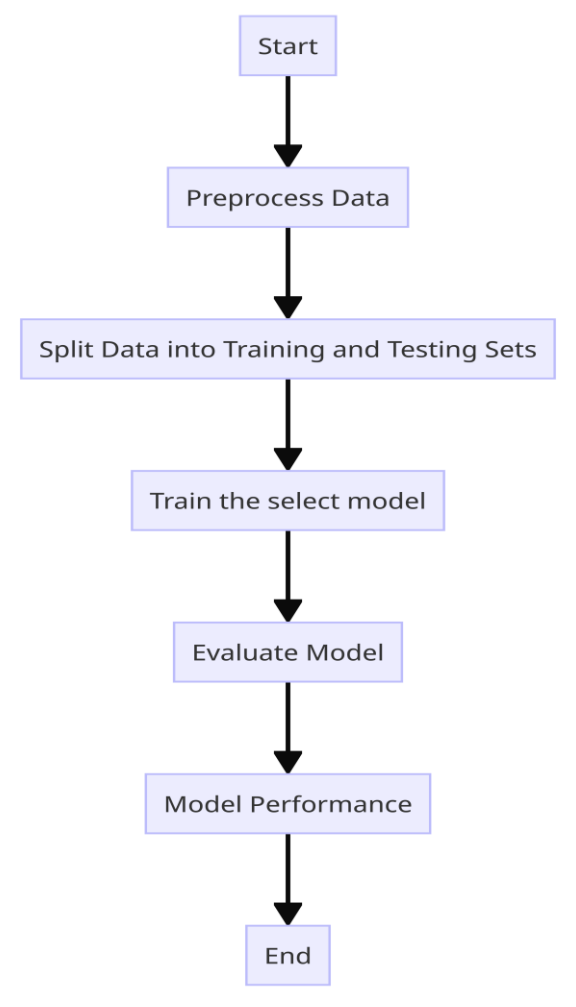
Build a Dashboard with Plotly Dash

- The version BT was the only failed in the launch site KSC LC-39A.



Predictive Analysis (Classification)

- For predictive analysis three models were considered: Logistic Regression, SVM and Decision Tree Classifier. All of those were evaluated according their score. The next diagram show the process for all of three models.
- The three models have the same score so any of those might be selected as good model. For more details, follow th next URL
- Add the GitHub URL of your completed predictive analysis lab, as an external reference and peer-review purpose



Results

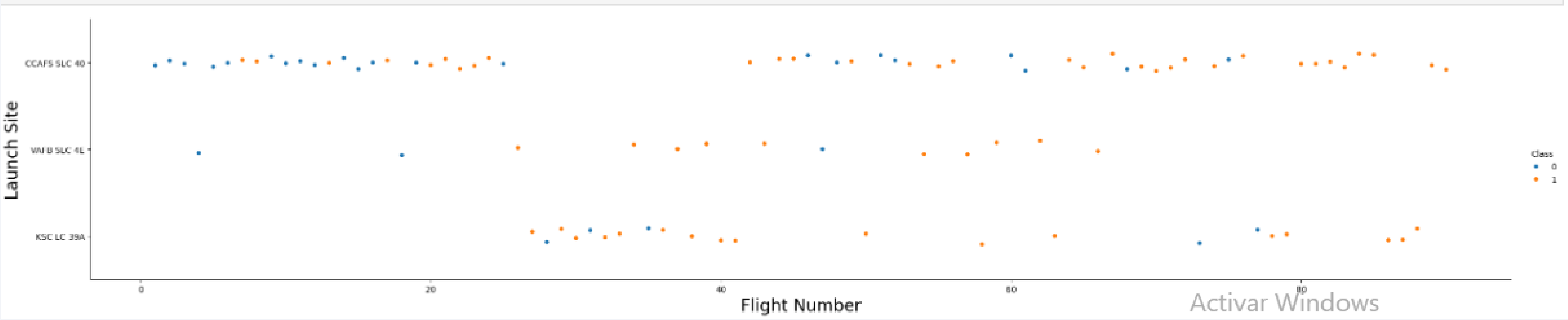
- Exploratory data analysis results
- Interactive analytics demo in screenshots
- Predictive analysis results

Section 2

Insights drawn from EDA

Flight Number vs. Launch Site

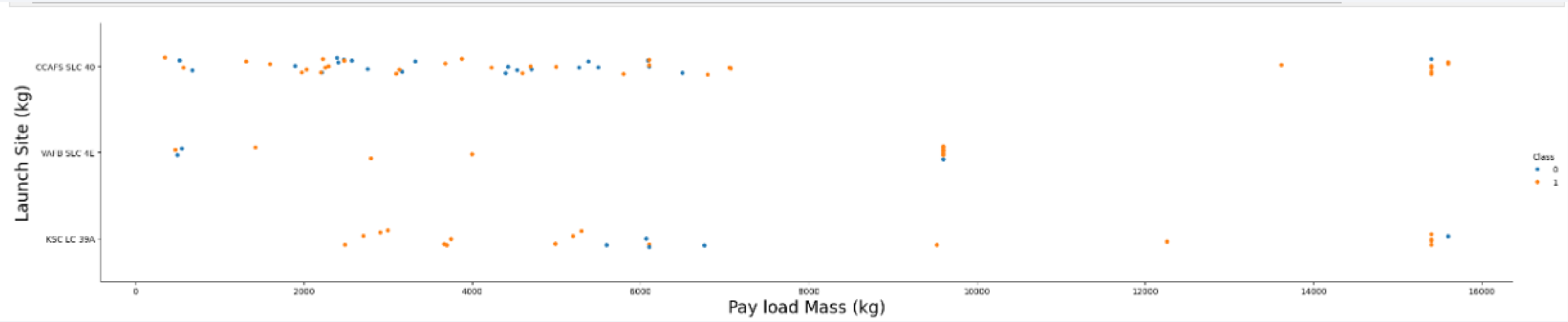
Scatter plot of Flight Number vs. Launch Site



- We can see that as the number of flights increases, the launch result improved. Even in a fast counting, the KSC LC 39A shows its launches with the best performance (blue: unsuccessful, orange: successful)

Payload vs. Launch Site

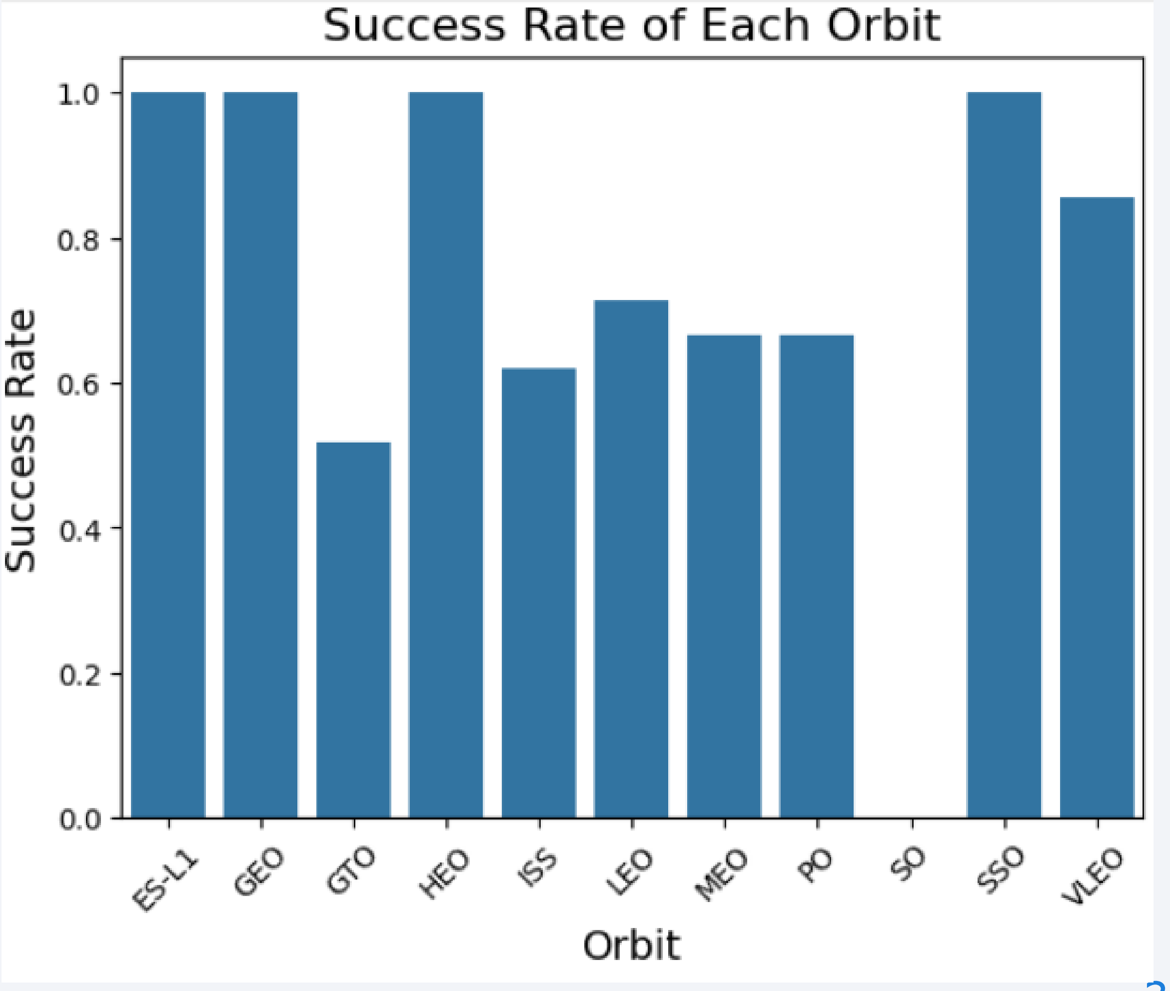
Scatter plot of Payload vs. Launch Site



- We can see that as Pay load mass increases for each launch site increases the successful launches increases (blue: unsuccessful, orange: successful)

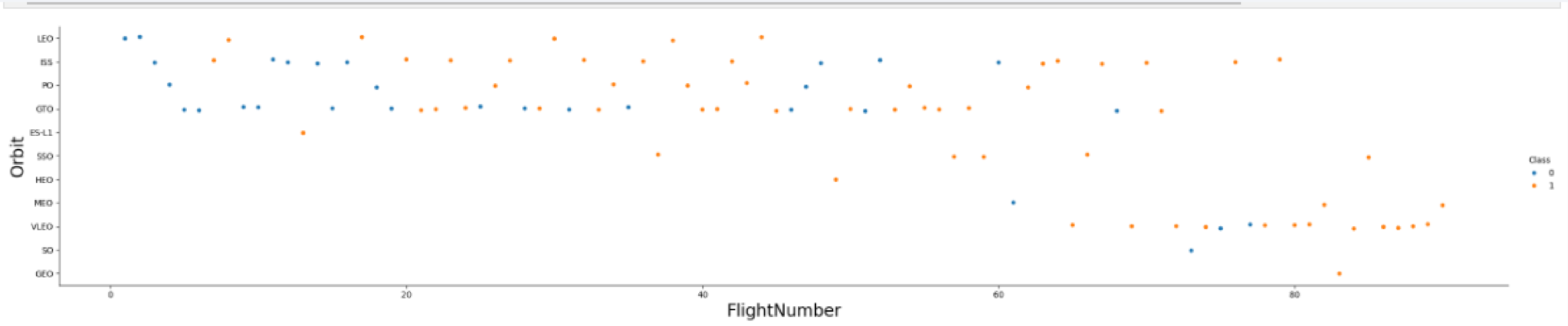
Success Rate vs. Orbit Type

- We can see 100% success rate for some orbits but those had just 1 launch and just one with 0% because of the same.
- Checking with more detail the number of launches we might say VLEO and Polar have the best rate.



Flight Number vs. Orbit Type

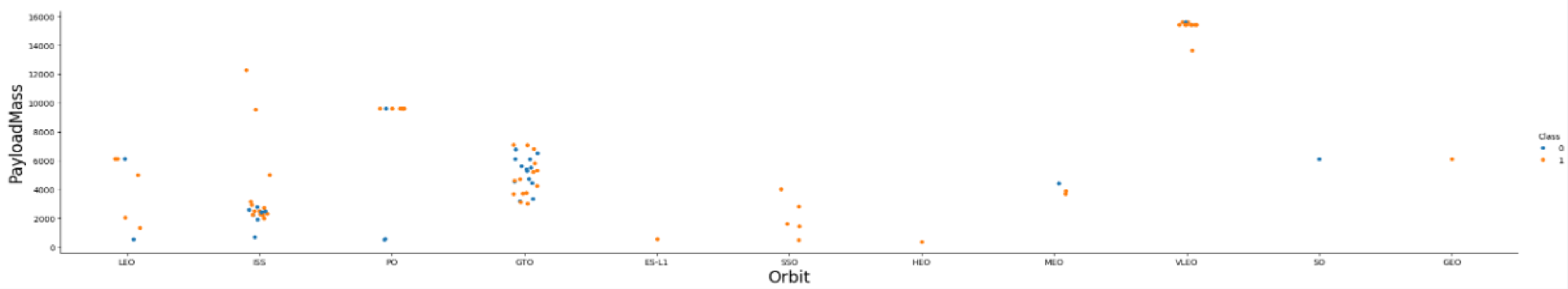
- Scatter plot of Flight number vs. Orbit type



- With heavy payloads the successful landing or positive landing rate are more for Polar (PO), LEO and ISS.
- With light payloads the successful landing or positive landing rate is for SSO

Payload vs. Orbit Type

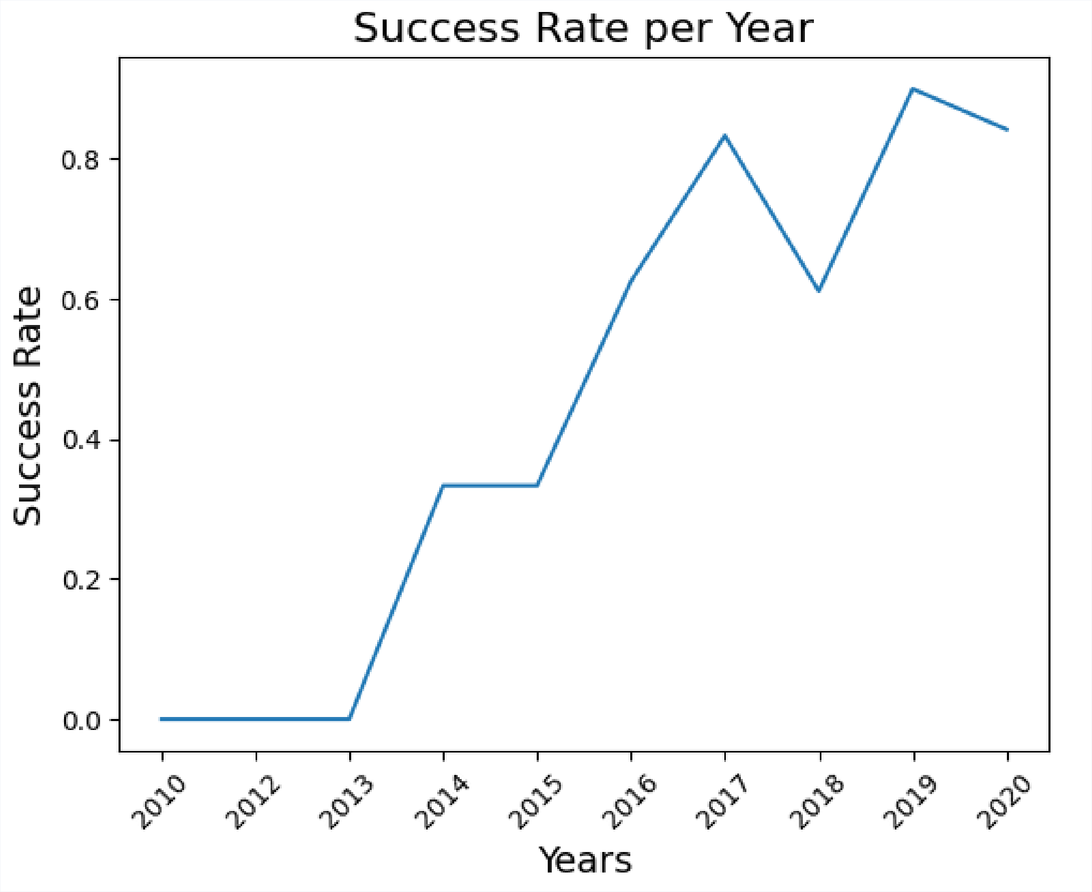
- Scatter plot of Payload vs. Orbit type



- With heavy payloads the successful landing or positive landing rate are more for Polar (PO), LEO and ISS.
- With light payloads the successful landing or positive landing rate is for SSO

Launch Success Yearly Trend

- the trend is success rate increases year after year, just had a small fall in 2018 and something minimal in 2020



All Launch Site Names

- Making a query, these are the Launch Sites names.
- The best explanation is the code of the query:

%sql SELECT DISTINCT(Launch_Site) FROM SPACEXTABLE;

Those sites are showed in other graphs and maps and is important to know about the launch sites for analysis.

```
 * 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'

These are the first 5 elements of the query of launch sites beginning with ‘CCA’ . With a small modification in the code (..SELECT count(*)…) we can know how many element are in total with this specification.

```
%sql SELECT * FROM SPACEXTABLE 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 showed in the next image with the code of the query
- This amount is the total of payload mass for all the launches done by NASA

```
%sql SELECT SUM(PAYLOAD_MASS__KG_) as total_payload_mass FROM SPACEXTABLE WHERE Customer like '%NASA (CRS)%'
```

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

total_payload_mass
48213

Average Payload Mass by F9 v1.1

The next image shows the average payload mass carried by booster version F9 v1.1,this includes F9 v1.1 B1010, F9 v1.1 B1011 and more…

```
%sql SELECT AVG(PAYLOAD_MASS__KG_) as average_payload_mass FROM SPACEXTABLE WHERE Booster_Version like '%F9_v1.1%'
```

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

average_payload_mass
2534.6666666666665

First Successful Ground Landing Date

- This is date of the first successful landing outcome on ground pad and the code of the query

```
%sql SELECT MIN(Date) FROM SPACEXTABLE WHERE Landing_Outcome like 'Success (ground pad)'
```

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

MIN(Date)
2015-12-22

Successful Drone Ship Landing with Payload between 4000 and 6000

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

```
%sql SELECT Booster_Version FROM SPACEXTABLE WHERE Landing_Outcome like '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

Here we will see the total number of failure and successful missions and its code for the query done.

```
%sql SELECT COUNT(CASE WHEN Mission_Outcome LIKE '%Success%' THEN 1 END) as total_number_of_successful_missions, COUNT(CASE WHEN Mission_Outcome LIKE '%F
```

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

total_number_of_successful_missions	total_number_of_failure_missions
100	1

Boosters Carried Maximum Payload

We would like to see which Boosters have carried the maximum payload mass in any mission:

```
%sql SELECT Booster_Version FROM SPACEXTABLE WHERE PAYLOAD_MASS__KG_= (SELECT MAX(PAYLOAD_MASS__KG_) FROM SPACEXTABLE)
```

```
 * 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

This is the list of the failed landing_outcomes in drone ship, their booster versions, and launch site names for in year 2015

```
%sql SELECT substr(Date, 6, 2) AS Month, Mission_Outcome, Landing_Outcome, Booster_Version, Launch_Site FROM SPACEXTABLE WHERE substr(Date, 0, 5) = '2015
```

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

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

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

The next image shows the rank of 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(*) as count_of_landing_outcomes FROM SPACEXTABLE WHERE Landing_outcome like 'Failure (drone Ship)' or Landing_outcome
```

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

Landing_Outcome	count_of_landing_outcomes
Failure (drone ship)	5
Success (ground pad)	3

Section 3

Launch Sites Proximities Analysis

Geographical location of launch sites on world map

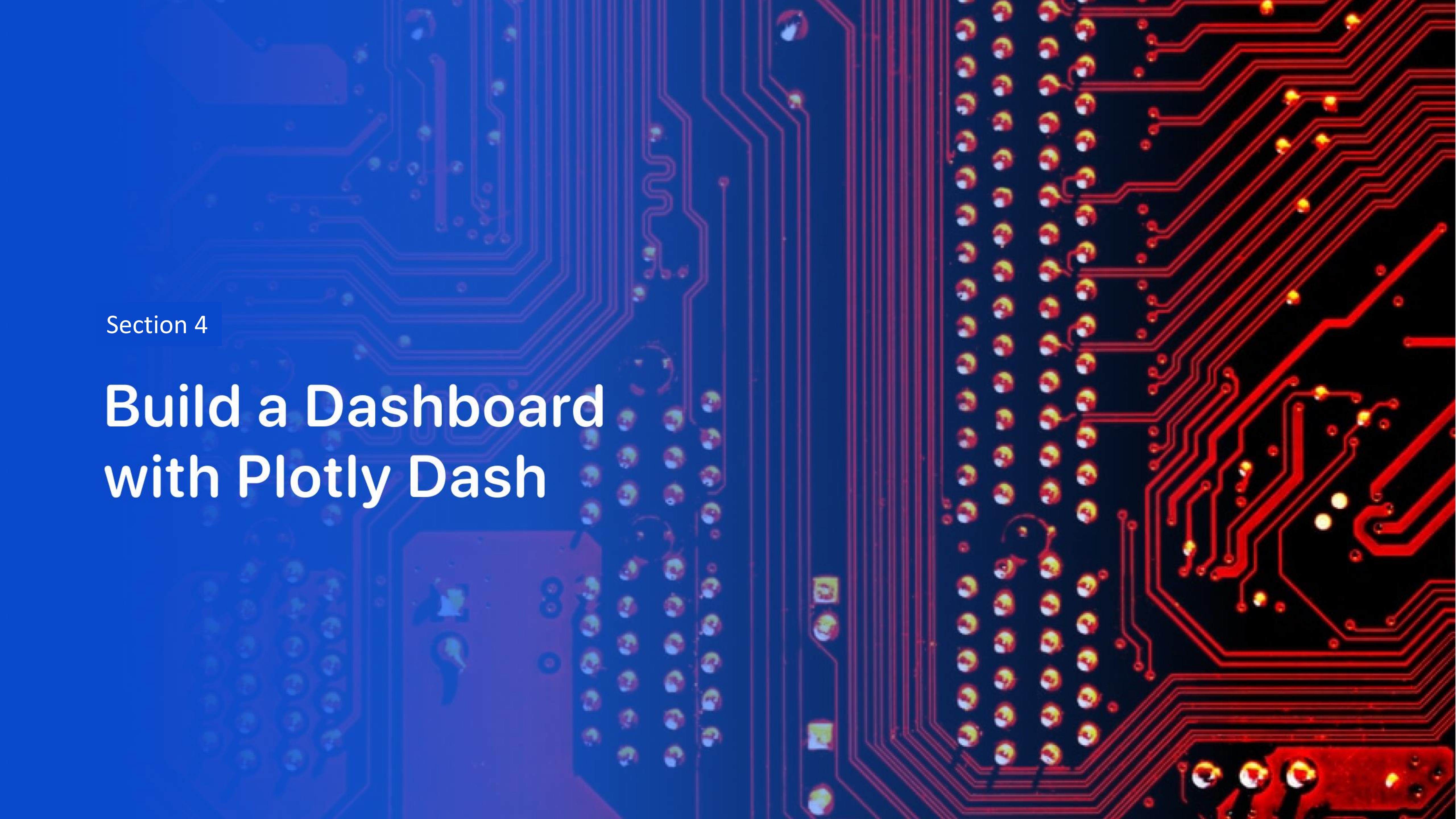
From the folium map we extract the location launch sites on a global map. As we can see those sites are located near to the sea.

The success/failed launches for each site on the map

Exploring the folium map, we can see with more detail the success and failed launches

Some proximities to CCAFS SLC - 40

Here we can see the distance from the launch site CCAFS SLC-40 to some proximities like the coastline (0,86 km), closest highway (0,58 km) and the railway (1,28 km)

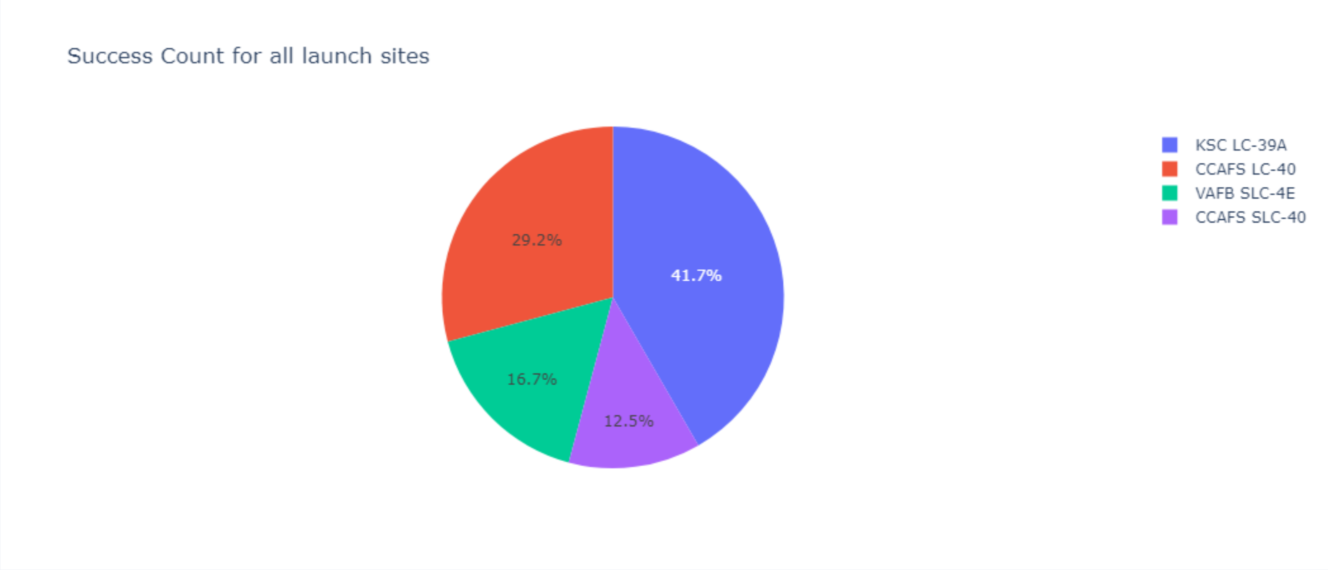


Section 4

Build a Dashboard with Plotly Dash

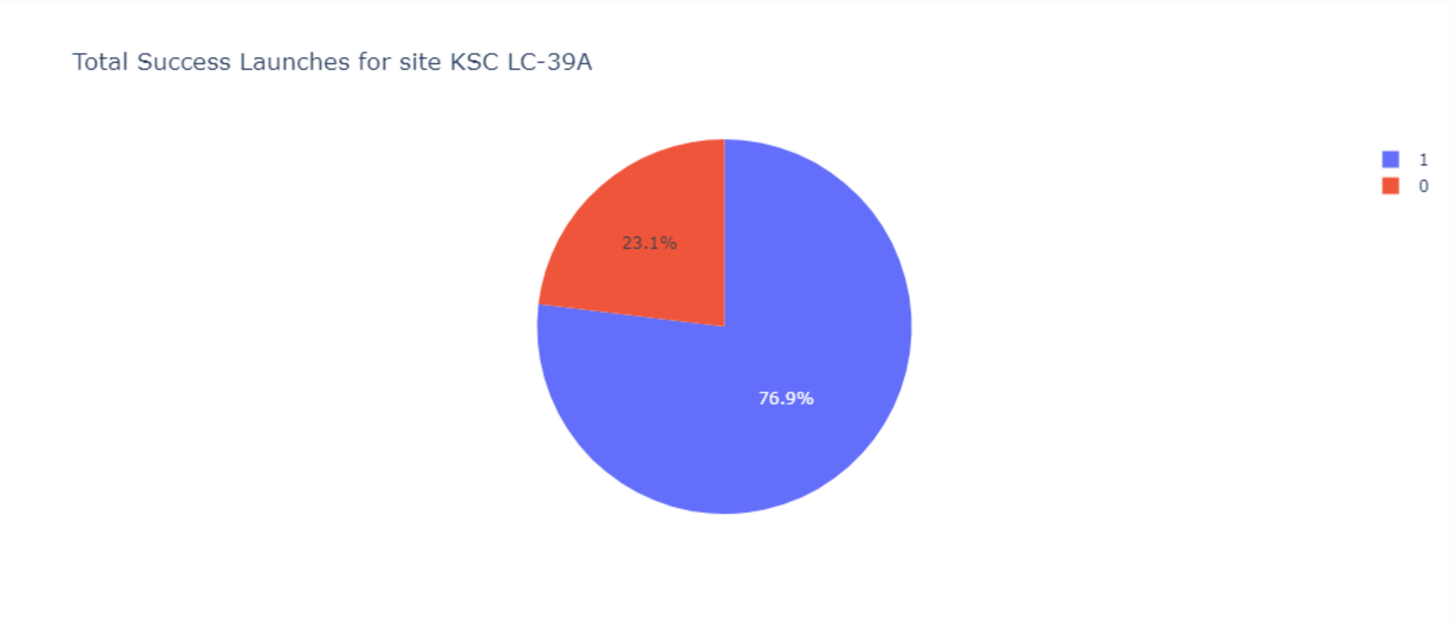
Rate success percentage per launch site

This image shows how KSC LC-39A is the launch site with the highest success rate and oppositely CCAFS SLC-40 the lowest.



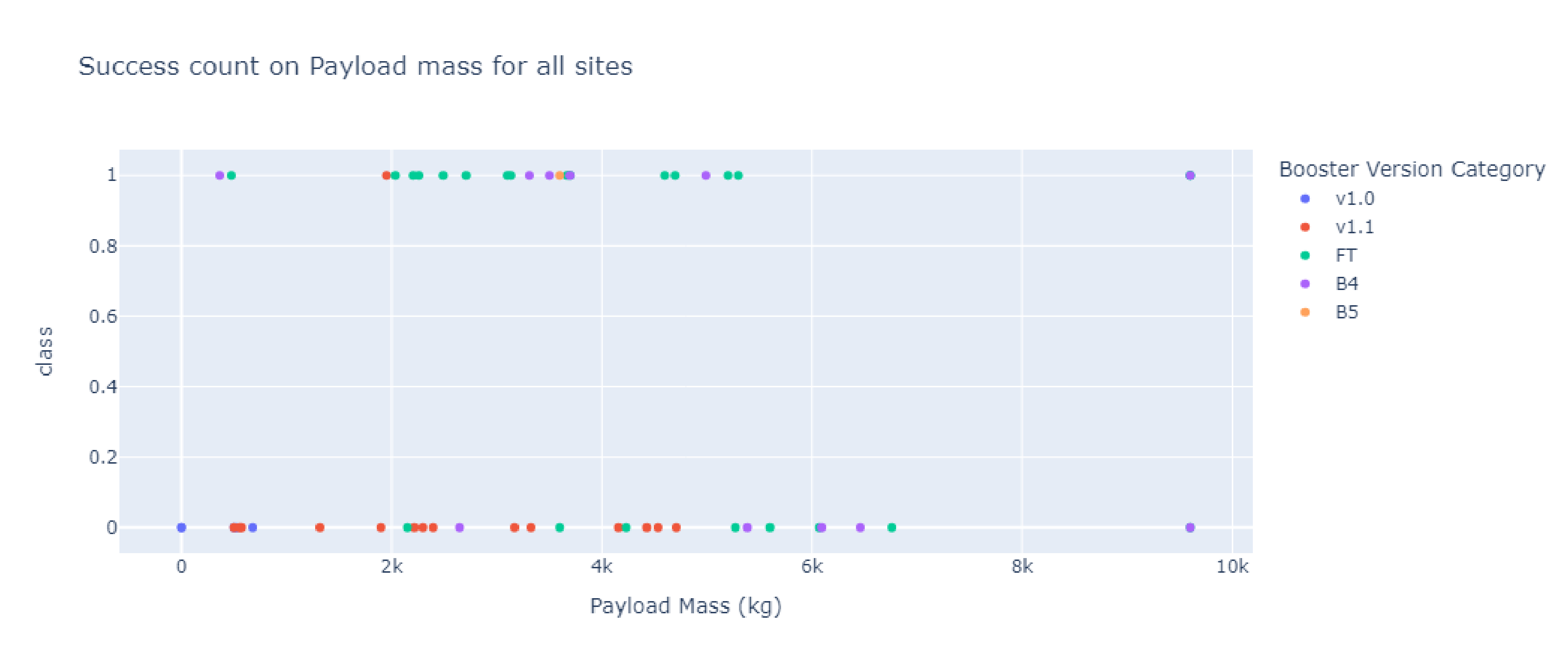
Launch site with Highest Success rate

The KSC LC-39C site has the highest success rate



Payload mass for all sites

- The image (with a range slider for payload mass) shows the payload for all the versions all launches, their class for all launch sites. The range slider helps to see the range for selected values, in this case the maximum is 10000.
- Only the B4 Booster version has the maximum payload with mass one success and one failure.

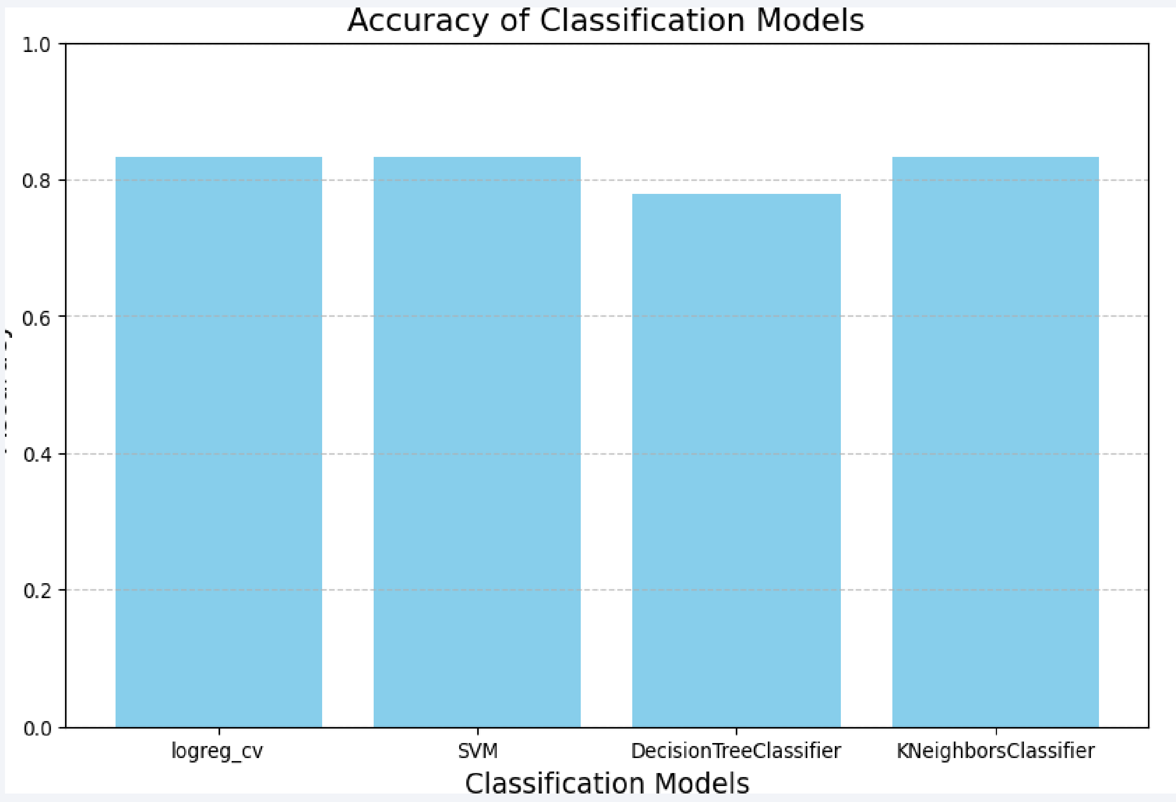


Section 5

Predictive Analysis (Classification)

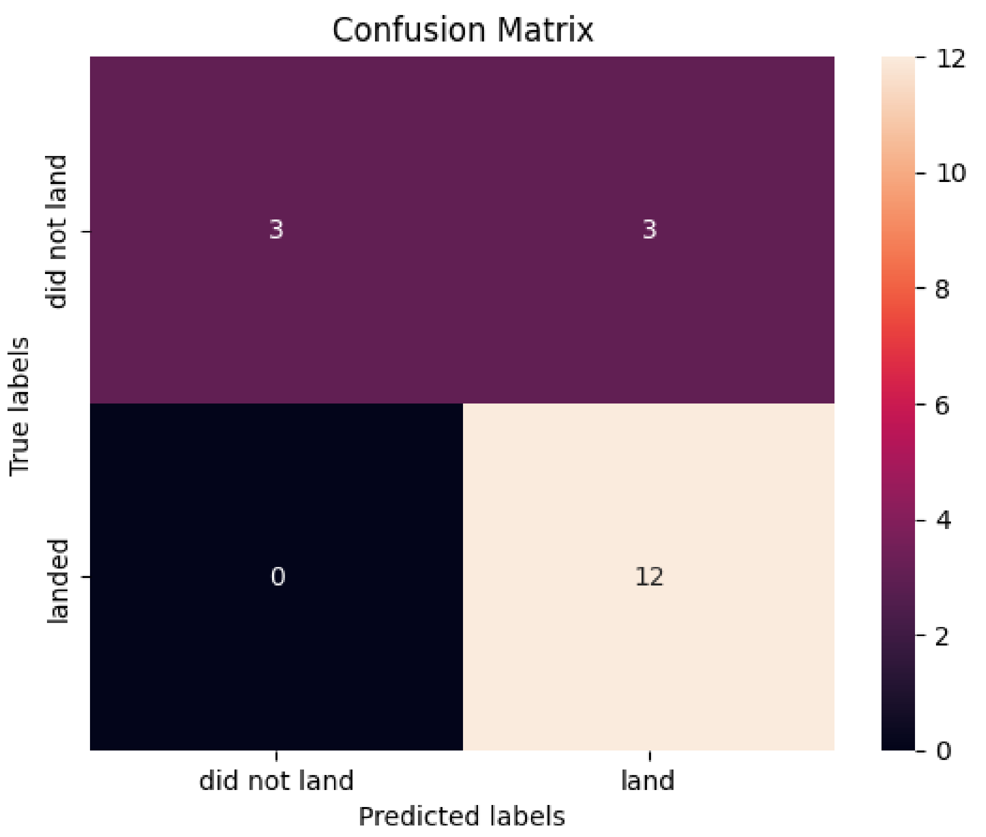
Classification Accuracy

- This graph shows three of four models have the same score 0,8333 and lowest 0,7777 (not bad) for Decision Tree Classifier.
- That means Logistic Regression, Support Vector Machine and Kneighbor Classifier are highest accuracy models



Confusion Matrix

- This matrix shows 15 correct estimated values (sum diagonal) from 18 values of the training set. With more detail, 3 values for ‘did not land’ when they did not land and 12 values for ‘land’ when they land.
- The mentioned above is showed in the score= 0,83333… or 15/18.



Conclusions

- The best success rate is for KSC LC-39 with 76,9% and the lowest is for CCAFS SLC-40 with 57,1%
- Only F9 B5 Booster Version and its categories were launched with the maximum payload.
- The launches with more payload have success rate higher than medium and low payload mass
- Any of SVM or Logistic Regression or KNeighbor Classifier are the best predictive models for this case of study

Appendix

Next you can find some lines of code of interest used in this project:

- Code to load the dataset for modelling:

```
df=pd.read_csv("https://cf-courses-data.s3.us.cloud-object-storage.appdomain.cloud/IBM-DS0321EN-SkillsNetwork/datasets/dataset_part_2.csv")
```

- Code to transform to dummies variable:

```
features_one_hot = pd.get_dummies(features, columns=['Orbit', 'LaunchSite', 'LandingPad', 'Serial’])

features_one_hot.astype(float)
```

- Code to create the logistic model:

```
parameters ={'C':[0.01,0.1,1], 'penalty':['l2'], 'solver':['lbfgs’]}

…
```

Appendix

…

```
lr=LogisticRegression()

logreg_cv=GridSearchCV(lr,parameters,cv=10)

logreg_cv.fit(X_train, Y_train)

GridSearchCV(cv=10, estimator=LogisticRegression(), param_grid={'C': [0.01, 0.1, 1], 'penalty':
['l2'], 'solver': ['lbfgs']})

print("tuned hpyerparameters :(best parameters) ",logreg_cv.best_params_)

print("accuracy :",logreg_cv.best_score_)

accuracy = logreg_cv.score(X_test, Y_test)

print("Accuracy on Test Data:", accuracy)
```

The code for the other models are similar, just set the parameters according to the model. You can read and learn more here: https://scikit-learn.org/1.4/index.html

Appendix

```
…

lr=LogisticRegression()

logreg_cv=GridSearchCV(lr,parameters,cv=10)

logreg_cv.fit(X_train, Y_train)

GridSearchCV(cv=10, estimator=LogisticRegression(), param_grid={'C': [0.01, 0.1, 1], 'penalty':
['l2’], 'solver': ['lbfgs’]})

print("tuned hpyerparameters :(best parameters) ",logreg_cv.best_params_)

print("accuracy :",logreg_cv.best_score_)

accuracy = logreg_cv.score(X_test, Y_test)

print("Accuracy on Test Data:", accuracy)
```

The code for the other models are similar, just set the parameters according to the model. You can read and learn more here: https://scikit-learn.org/1.4/index.html

Appendix

The next links show all the tasks that support the results of this study:

Data Collecting from de SpacesX API:
https://github.com/Cupaban1/testrepo/blob/Capstone/jupyter-labs-spacex-data-collection-api.ipynb

Web Scraping from Wikipedia:

https://github.com/Cupaban1/testrepo/blob/Capstone/jupyter-labs-webscraping.ipynb

Data Wrangling:

https://github.com/Cupaban1/testrepo/blob/Capstone/labs-jupyter-spacex-Data%20wrangling.ipynb

Appendix

The next links show all the tasks that support the results of this study:

EDA with SQL:
https://github.com/Cupaban1/testrepo/blob/Capstone/jupyter-labs-eda-sql-coursera_sqllite.ipynb

Visualization:

https://github.com/Cupaban1/testrepo/blob/Capstone/jupyter-labs-eda-dataviz.ipynb

Interactive visualization in maps:

https://github.com/Cupaban1/testrepo/blob/Capstone/lab_jupyter_launch_site_location.ipynb

Appendix

The next links show all the tasks that support the results of this study:

EDA with SQL:
https://github.com/Cupaban1/testrepo/blob/Capstone/jupyter-labs-eda-sql-coursera_sqllite.ipynb

Visualization:

https://github.com/Cupaban1/testrepo/blob/Capstone/jupyter-labs-eda-dataviz.ipynb

Modelling:

https://github.com/Cupaban1/testrepo/blob/Capstone/SpaceX_Machine_Learning_Prediction_Part_5.jupyterlite.ipynb

Thank you!