IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

Ekin Bayçın
09/03/2023

Outline

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

Executive Summary

- Our aim is to find the best variables to increase the success rate of landings and find a suitable model to predict the outcome. First, data needs to be gathered, we have done that by getting the public data from SpaceX API and some tables containing launch specifications from wikipedia. After they are obtained, they have been processed and irrelevant variables have been dropped and meaningless data turned into meaningful ones. To gain insight, different graphs have been made comparing one variable against the other to see their contribution to success. An interactive dashboard has been made to compare them quickly in real-time. Also, locations of the launch sites have been plotted on a world map to gain more insight. After exploring the data, we needed to choose a model to predict our outcome. Since our output is a categorical one, four different models have been selected. One-hot encoding has been used to convert the categorical variables into numerical ones for model to use. All models have been trained and their best hyperparameters have been found. Their accuracy has been compared and the best model has been selected.

Executive Summary

- After we have done all these, it's been found out that variables, Orbit Type, Payload Mass, Flight Number, Launch Site, Booster Version Category all have relations to a successful landing.
- Models SVM, KNN, Logistic Regression, and Decision Tree have been tested, their best hyper-parameters have been found and all underwent a 10-fold cross validation. Among those, the best accuracy was found on the Decision Tree and that model was decided to be used for future predictions.

Introduction

- Mankind is on the brink of the Space Age. We've been sending rockets, satellites, autonomous vehicles for exploration, observation, and communication.
- This venture is not a cheap one. Among many companies one called SpaceX has the edge by having *much* lower costs for sending rockets. They retain this edge by reusing a part of the rocket(first stage) for future launches.
- To be able to reuse the part of the rocket, it needs to safely land back to the earth. It is this technology which allows them to send rockets for cheaper.

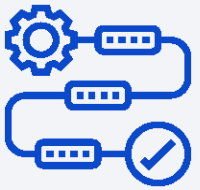
Introduction

- It is apparent that landing this first stage successfully is of extreme importance. Thus, as a data scientist the following questions come to mind:
  - Are there any relations to the successful landings with some of the parameters?(Launch Site, Payload Mass, ...)
  - How can we determine the chances of first stage landing successfully using these parameters?

Section 1

Methodology

Methodology



Executive Summary

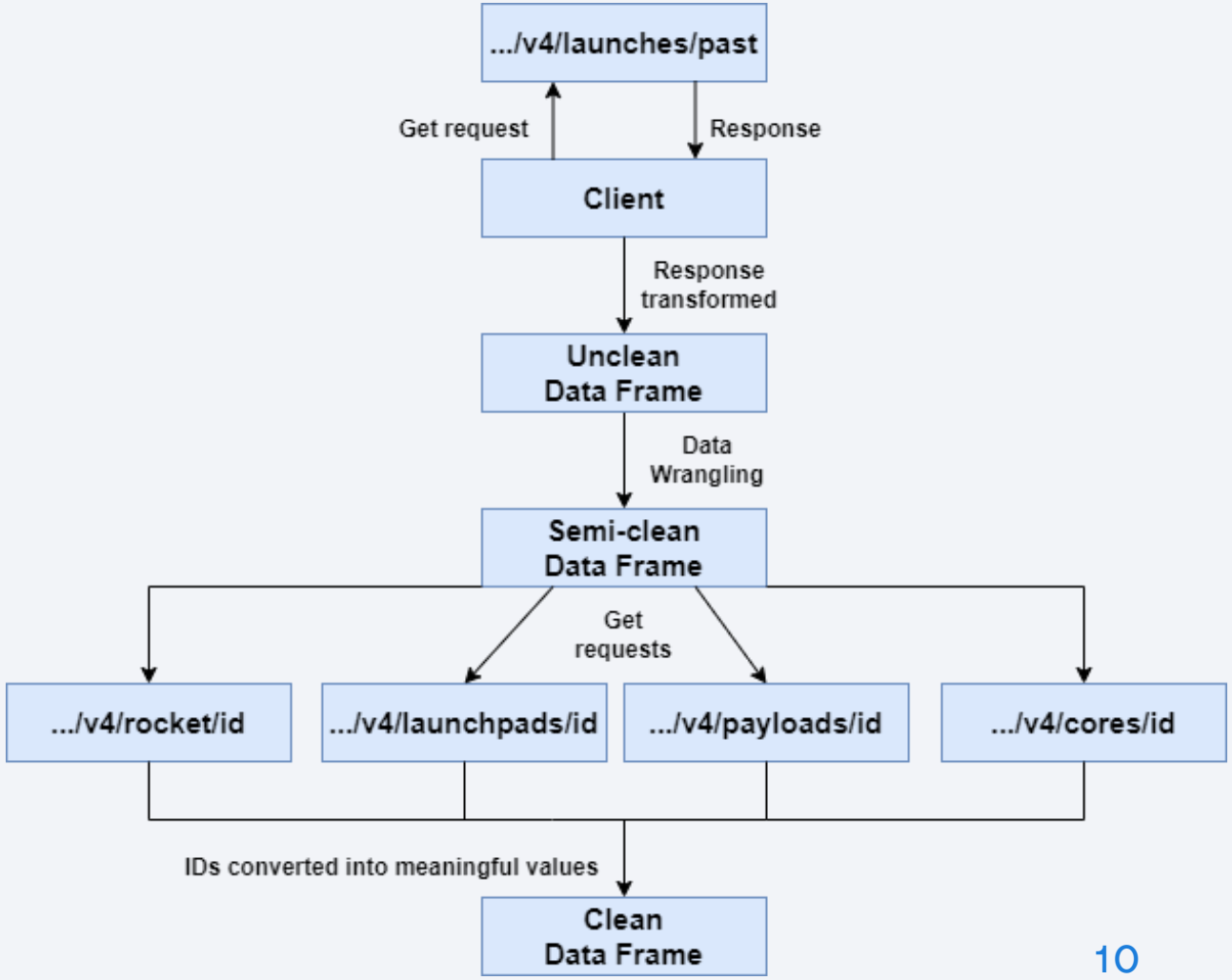
- Data collection methodology:
  - To obtain the data web scraping was performed and public data was downloaded from SpaceX API.
- Perform data wrangling
  - One-hot encoding was used, replaced missing values, turned meaningless values into meaningful ones through the SpaceX API, unnecessary features were dropped.
- 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 classification algorithms were used for prediction, grid search along with cross validation has been performed to get the best hyperparameters. Then, to compare the models we have checked their accuracy.

Data Collection

- We needed records of SpaceX rocket launches, with the help of the SpaceX API, data relating to the launches were acquired.
- Some of the columns contained IDs instead of meaningful values. So, through the API these values for the related columns were acquired.
- Next, we have scraped tables from a wikipedia page relating to the Falcon 9 launches. We made a get request to the html code of this site. After that, html code was parsed and the required tables were located. Then, a dictionary was made out of the values of the tables. This dictionary got converted into a pandas data frame.

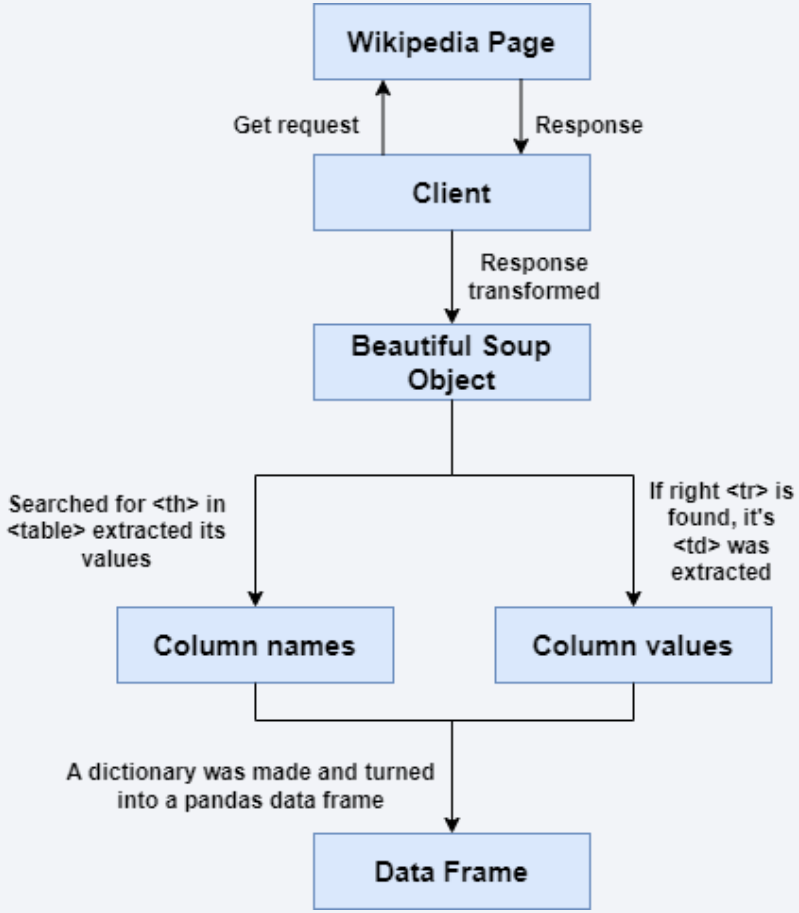
Data Collection – SpaceX API

- To retrieve our data we have used SpaceX REST API
- https://api.spacexdata.com/is the base URL here for the API.("…" is the base URL on the flowchart)
- Here, "id" on the API calls are the related IDs for the call.
- To check the process you can ctrl+click on the GitHub URL below
- Data Collection with API



Data Collection - Scraping

- Wikipedia Page
- Beautiful Soup was used to parse and extract data from the html
- To check the process ctrl+click on the link below
- Data Collection with Webscraping



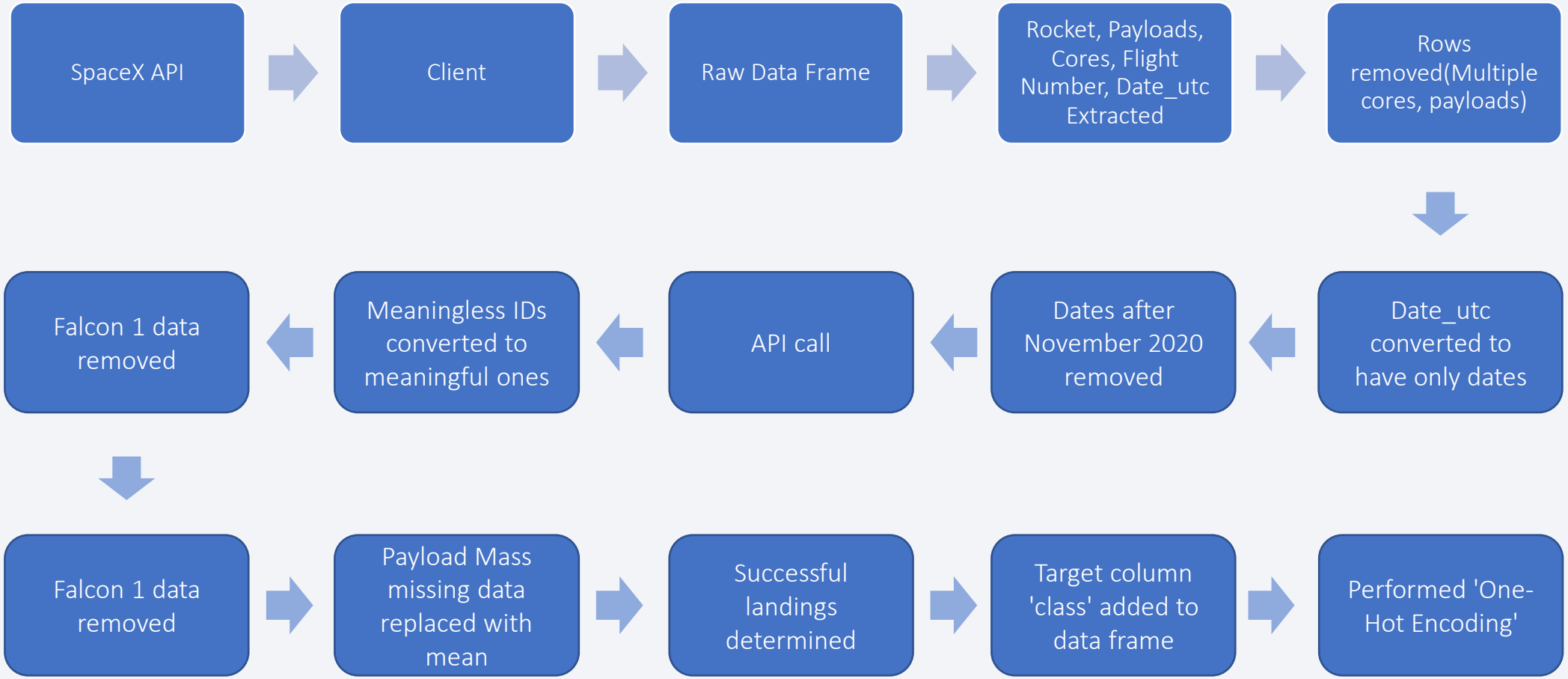
Data Wrangling

- The data we got from the SpaceX API needed some adjustments. First off, we separated the columns: Rocket, Payloads, Cores, Flight Number, Date_utc to form a new dataframe. These are the relevant data we need.
- Then, we have removed the rows with multiple cores and payloads. After that, we removed the time component in the date_utc column and kept the dates. Rows before the date November 2020 were also dropped.
- Now, the data we had acquired contained some meaningless IDs, we have used the appropriate API calls to convert them into meaningful values. Made a dictionary out of them and turned it into a data frame. We only wanted the Falcon 9 Launches so we drop the rows containing Falcon 1.

Data Wrangling

- We checked if we had any missing data/none values on our data frame. On the Launch Pad and Payload Mass columns we had some missing data. We decided to keep the none values we had on Launch Pad to show that no Launch Pad was used on those specific launches. On the other hand, the missing values on Payload Mass were replaced with the average of the Payload Mass column.
- We are interested in predicting the outcome. Thus, we need to take a look at the categorical Outcome column, decide which of the labels indicate success and which do not. According to this classification, we have added a new column (target variable) called 'class' and for each successful landing entries of 1, and for each unsuccessful landing entries of 0 were made. Also, categorical independent variables were transformed with one-hot encoding.

Data Wrangling



Data Wrangling

- You can check the performed data wrangling in the following Jupyter Notebooks
- Reference 1: Data Wrangling on Data Collection API
- Reference 2: Data Wrangling

EDA with Data Visualization

- First, we wanted to see how Flight Number and Payload Mass affected the success of the landing outcome. So, we did a scatter plot showing that.
- After that, we wanted to examine the Flight Number and Launch Site's effect on the success of the landing outcome. A scatter plot was plotted for that.
- A scatter plot of Flight Number vs Launch Sites were also made with success outcome colour coded.
- Rockets are sent to different orbits. For each orbit the success rate of landing differs. We wanted to visualize that and plotted a bar chart orbit vs success rate.
- For each orbit, we wanted to see if there's a relation with the flight number. Therefore, we have plotted a scatter plot with variables Flight Number and Orbit Type, and success was colour coded.

EDA with Data Visualization

- Similarly, we wanted to reveal the relationship between Payload Mass and Orbit type. Once again, a scatter plot has been drawn with success colour coded.
- Finally, we wanted to scrutinize on the success rate as years passed by. To depict this, a line plot of success rate vs years were drawn.
- You can ctrl+click on the reference to go to the github page.
- Reference to Jupyter Notebook: EDA with Data Visualization

EDA with SQL

- Retrieved the unique launch site names:
  - Select DISTINCT("LAUNCH_SITE") from SPACEXTBL;
- Displayed 5 records of launch sites beginning with 'CCA':
  - SELECT * from SPACEXTBL WHERE "LAUNCH_SITE" LIKE "CCA%" LIMIT 5
- Total payload mass carried by boosters launched by NASA (CRS):
  - SELECT SUM(PAYLOAD_MASS__KG_) from SPACEXTBL WHERE "Customer" = "NASA (CRS)";
- Average payload mass carried by booster version F9 v1.1:
  - SELECT AVG("PAYLOAD_MASS__KG_") FROM SPACEXTBL WHERE "Booster_Version" = "F9 v1.1";

EDA with SQL

- Date the first successful landing outcome in ground pad was achieved:
  - SELECT min("Date") AS "DATE" from SPACEXTBL WHERE "Landing _Outcome" = "Success (ground pad)";
- Names of the boosters which have success in drone ship and have payload mass greater than 4000 but less than 6000:
  - SELECT "BOOSTER_VERSION" from SPACEXTBL WHERE "LANDING _OUTCOME" = "Success (drone ship)" AND "PAYLOAD_MASS__KG_" BETWEEN 4000 and 6000;
- Total number of successful and failure mission outcomes:
  - SELECT "MISSION_OUTCOME", COUNT("MISSION_OUTCOME") from SPACEXTBL GROUP BY "MISSION_OUTCOME"
- Names of the booster versions which have carried maximum payload mass using subquery:
  - SELECT DISTINCT("booster_version") FROM SPACEXTBL WHERE "PAYLOAD_MASS__KG_" = (SELECT max("PAYLOAD_MASS__KG_") from SPACEXTBL);

EDA with SQL

- Records displaying the month names, failure landing_outcomes in drone ship ,booster versions, launch_site for the months in year 2015:
    - SELECT substr(Date, 4, 2) AS "Month", "LANDING _OUTCOME", "Booster_version", "Launch_site" from SPACEXTBL where "LANDING _OUTCOME" = "Failure (drone ship)" and substr(Date,7,4)='2015'
- Count of successful landing_outcomes between the date 04-06-2010 and 20-03-2017 in descending order:
    - SELECT "Landing _Outcome", count("Landing _Outcome") AS "Count" from SPACEXTBL GROUP BY "Landing _Outcome" HAVING ("DATE" BETWEEN "04-06-2010" AND "20-03-2017") AND ("Landing _Outcome" LIKE "suc%") ORDER BY count("Landing _Outcome") DESC
- Link to the Jupyter Notebook on GitHub: EDA with SQL

Build an Interactive Map with Folium

- On our interactive map we have placed folium objects on the folium map such as: circles, icons, markers, marker clusters, a mouse position object and a polyline.
- We have added circles to indicate the launch sites, centers were placed at the launch sites and a perimeter was established. Markers were added for various reasons, one of which was to indicate the launch site names, other was to show the distances between the launch sites and some important infrastructure, and places.
- Since we had so many launch records, their markers would get on top of the other making it hard to distinguish. For that reason, we have added marker clusters, the launches made on specific launch sites were clustered together also they were associated with an icon, in which green icons were indicative of successful landings, and red ones were for unsuccessful ones. There was also a popup attached to them indicating success or failure.
- We wanted to know about the distance between the launch sites and some important infrastructure and geographical entities such as coastline. To that end, we have placed a mouse position object which showed us the coordinates of the tip of the mouse when we hovered on the map. Using this the distance between the entities and launch sites were calculated. After that, a polyline object was made one end on the entity and the other on the launch site and attached to this polyline was a marker which indicated distance in km.
- You can ctrl+click on the links below to go to the sites.
- Reference for the Jupyter Notebook : Visual Analytics with Folium
- If you can't see the maps created on the notebook in github here's an alternative link: Rendered Notebook on nbviewer

Build a Dashboard with Plotly Dash

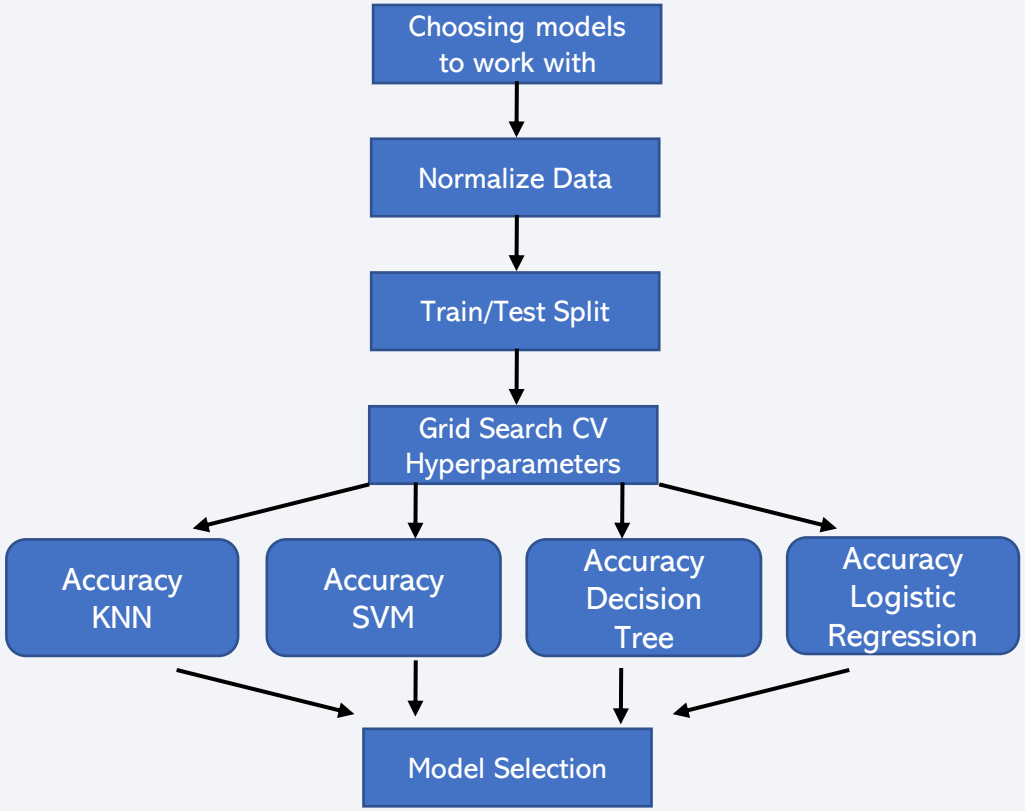
- An interactive pie chart showing the success rates of launch sites, and a scatter plot showing whether landing was successful depending on the payload mass has been made, also booster version was colour coded in the scatter plot. A dropdown menu, and a range slider also has been made. This dropdown menu controls launch site variable in the pie chart and scatter plot, in its default state it shows all sites together. But it's possible to choose a specific launch site. The range slider allows us to control the payload mass variable for the scatter plot it can be changed between 0 and 10000 kg with a step size of 1000kg.

Build a Dashboard with Plotly Dash

- Pie chart has been added so that we can compare different launch site's success rates compared to one another. When all sites are selected it shows which percent of the total landing success is attributed to each site. When a specific site is selected it shows that site's success and failure percentages. That's why the drop down menu was added so that we can change that variable in real time and compare the data more efficiently. This drop down menu also affects the scatter plot. It changes according to which site is selected.
- Scatter plot was made to compare the success/failure of the landings on different payload masses and launch sites. It also shows us the booster version since it's colour coded. We can see which booster version was more effective on specific ranges and sites.
- Using the range slider we can limit the data shown on the scatter plot. So that, we can zero-in on specific ranges and derive conclusions effectively.
- Python Code for the Plotly Dash app: Interactive App Python Code

Predictive Analysis (Classification)

- Our model will be trying to predict the outcome of the rocket landing success. Since we have two values 0 (failure) and 1 (success) we need to choose a categorical model. Among the categorical models we have chosen to work with, Logistic Regression, Decision Tree, K Nearest Neighbours, and Support Vector Machine.
- First thing that needed to be done was normalizing our data set with standard scaling. After that's done, we have split our data set into train(80%) and test(20%) data sets. To find the best model with the best parameters we have used grid search with cross validation. Grid search builds the model with the given hyperparameters, and performs cross validation(in our case we used 10-fold CV). Then, outputs the best parameters among those given by us and gives us the cross validation accuracy. Each model was trained using this method, then they were compared according to their cross validation accuracies to choose the best performing model.
- GitHub link to the Notebook: Machine Learning Prediciton



Results

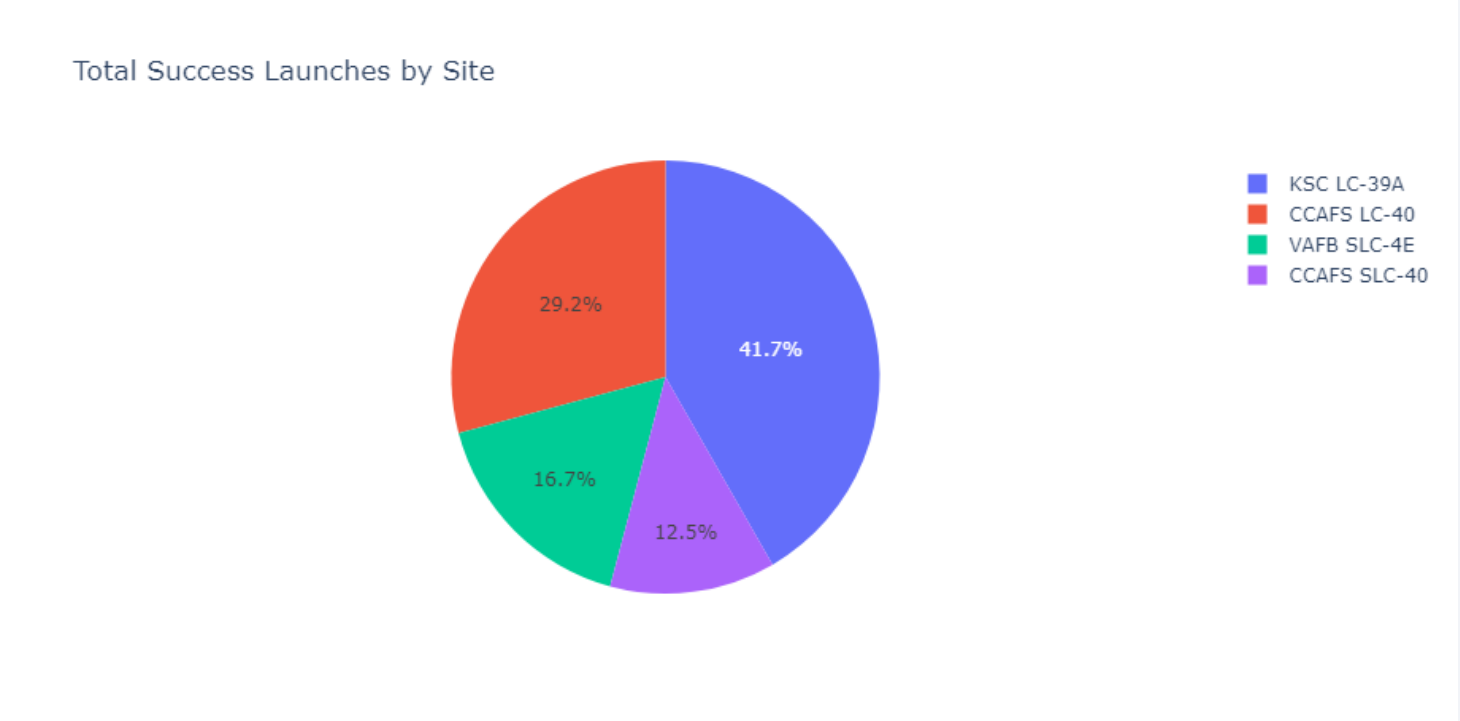
Exploratory Data Analysis

- After our exploratory data analysis we have seen that different payload masses sent to different orbits, and launched from different launch sites have different outcomes. In addition to that as the number of launches increase throughout the years success rates goes up. What we have seen is that all of these parameters contribute to the successful landing of stage one of the rocket.

Results

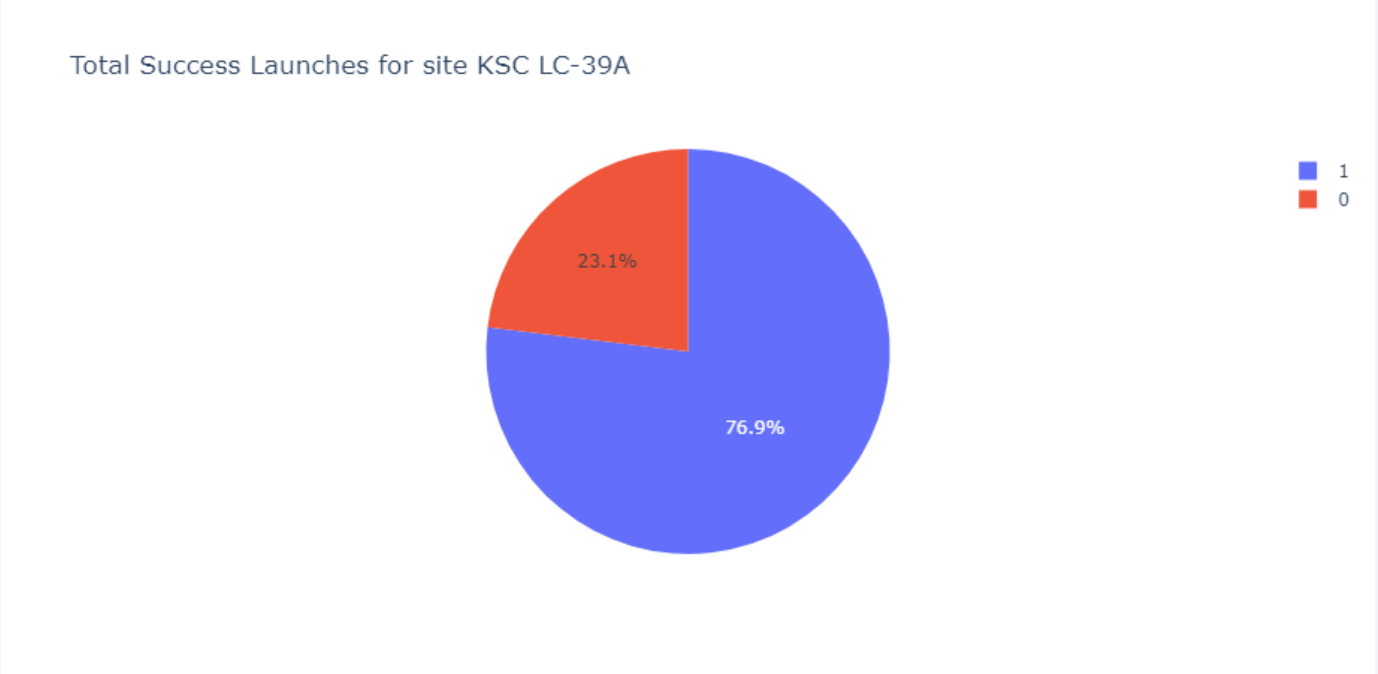
Dashboard Demo with Screenshots

- As we can see, among the launch sites some are responsible for more of the total success



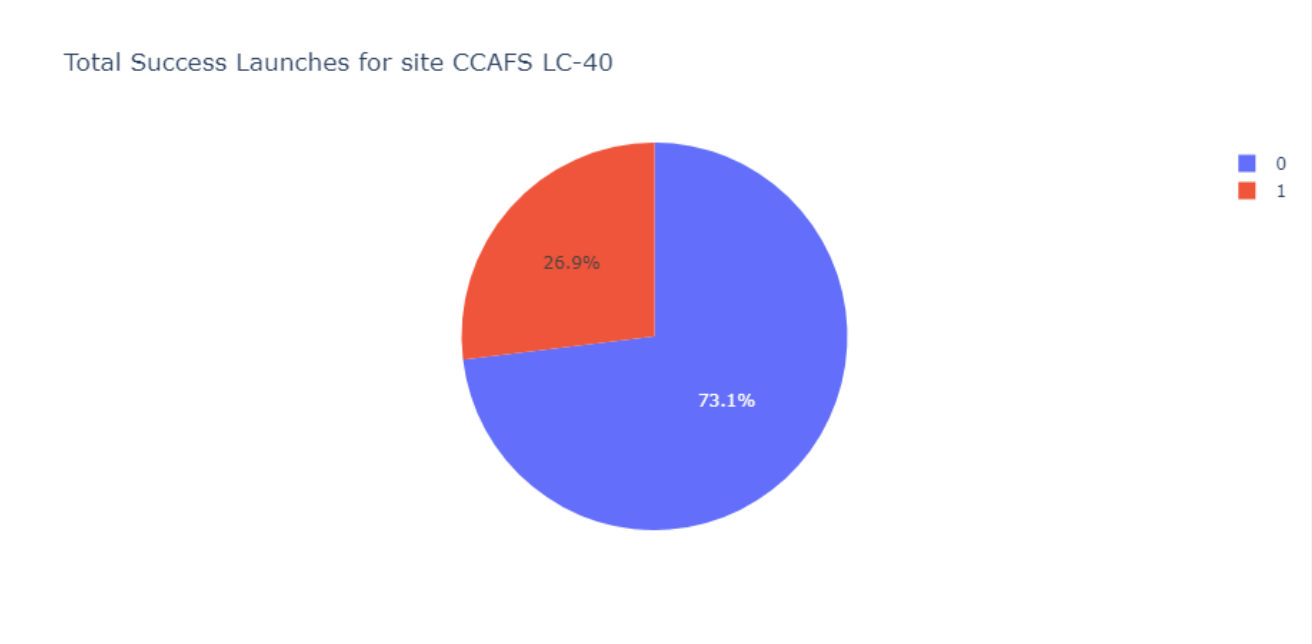
Results

- The launch site responsible for most success has great odds to produce a successful landing.

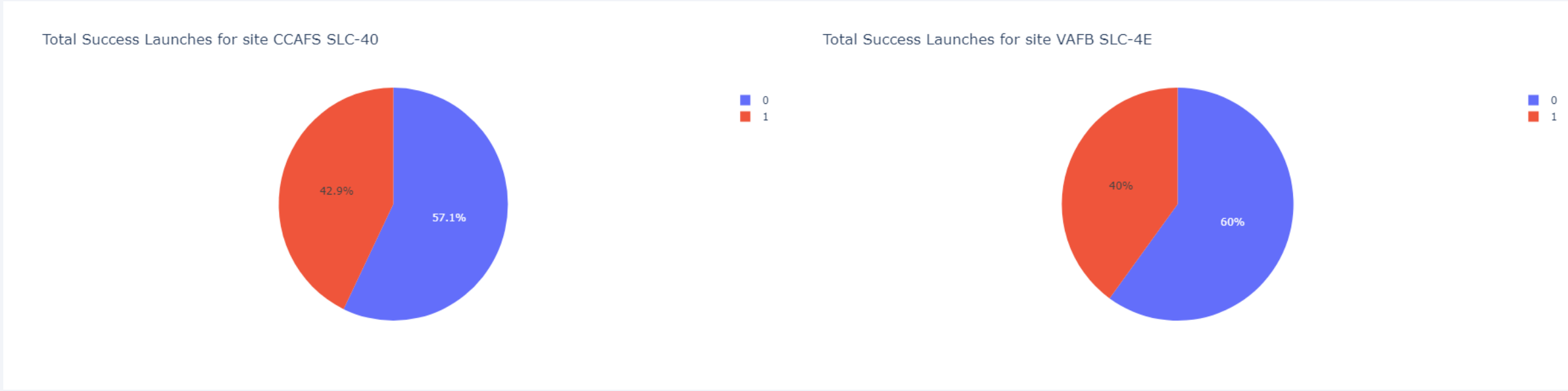


Results

- But, the launch site with the second number of success has a terrible percentage to produce a successful landing.



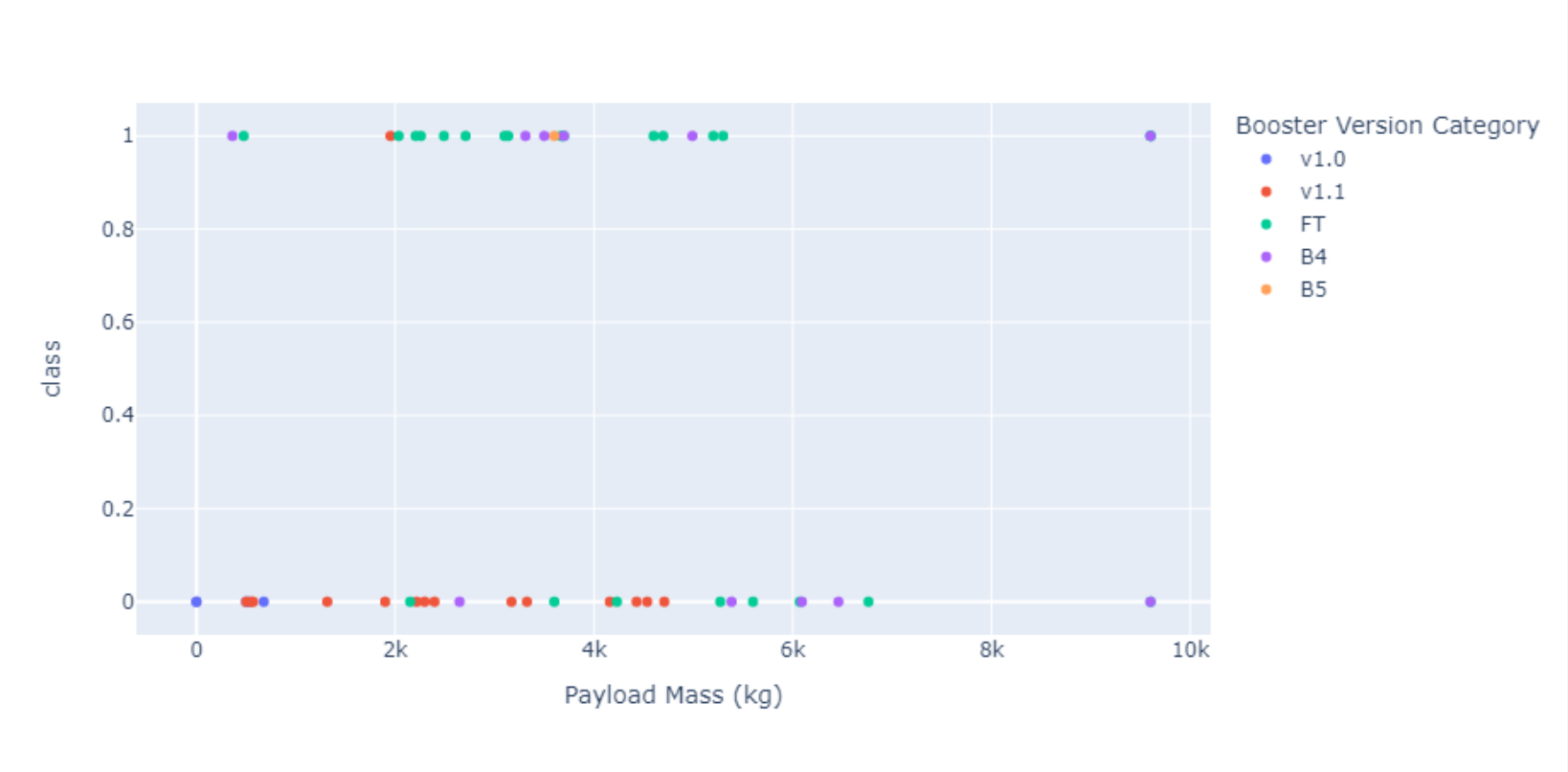
Results



- As we can see even the launch sites responsible for the worst number of successful landings have better percentages for a successful landing.

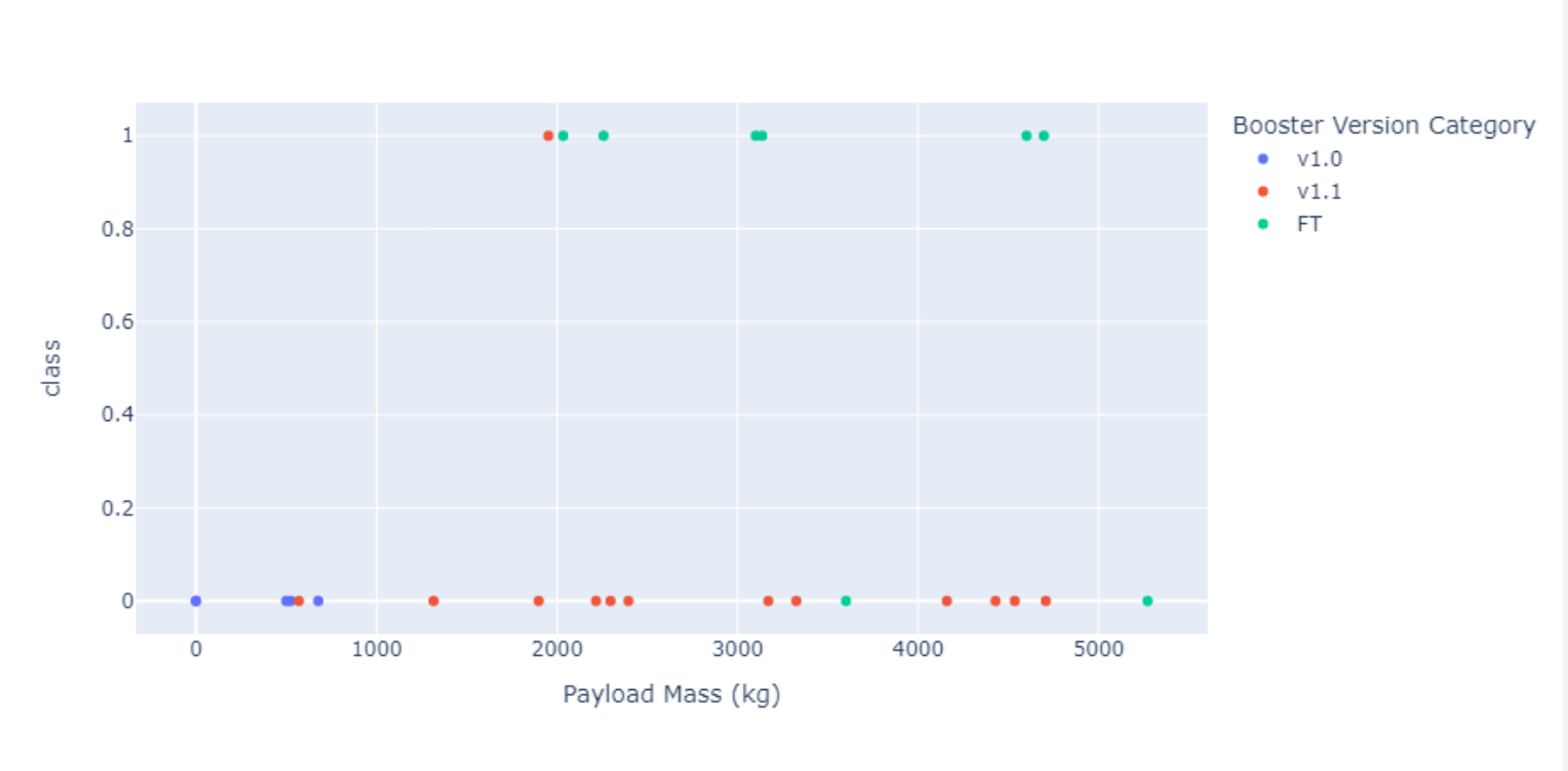
Results

- For all sites



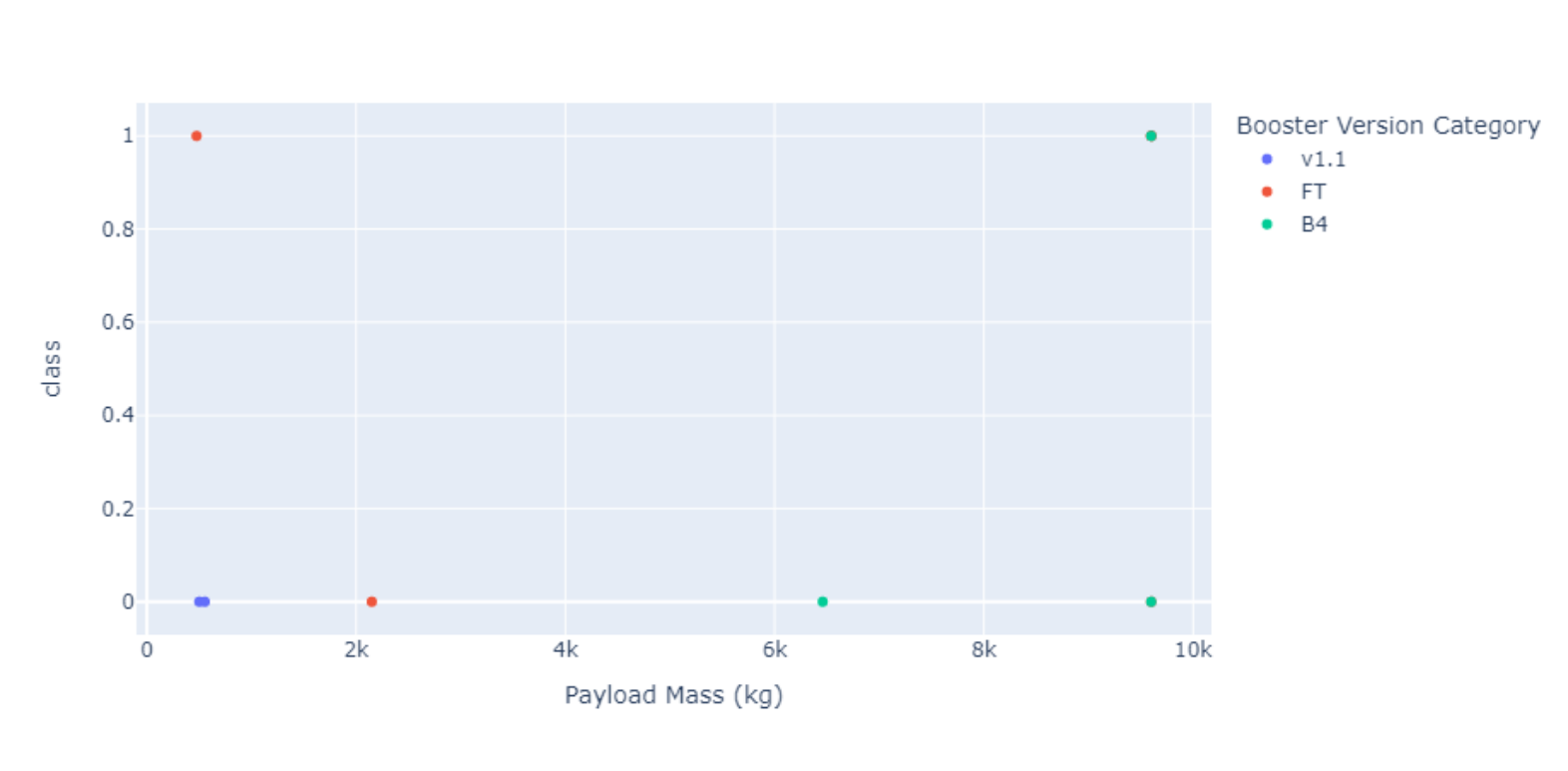
Results

- For CCAFS LC-40 site



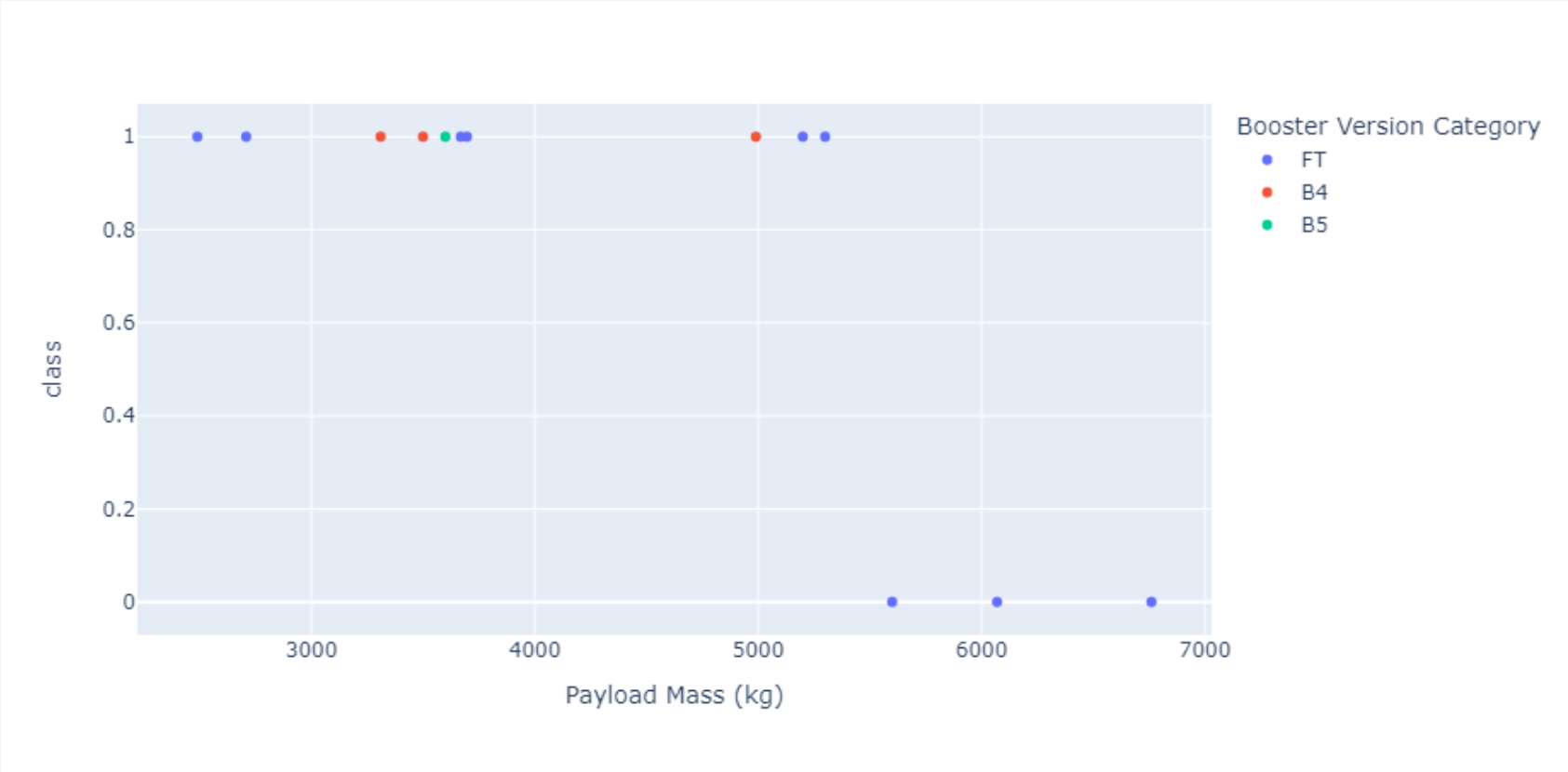
Results

- For VAAFB SLC-4E site



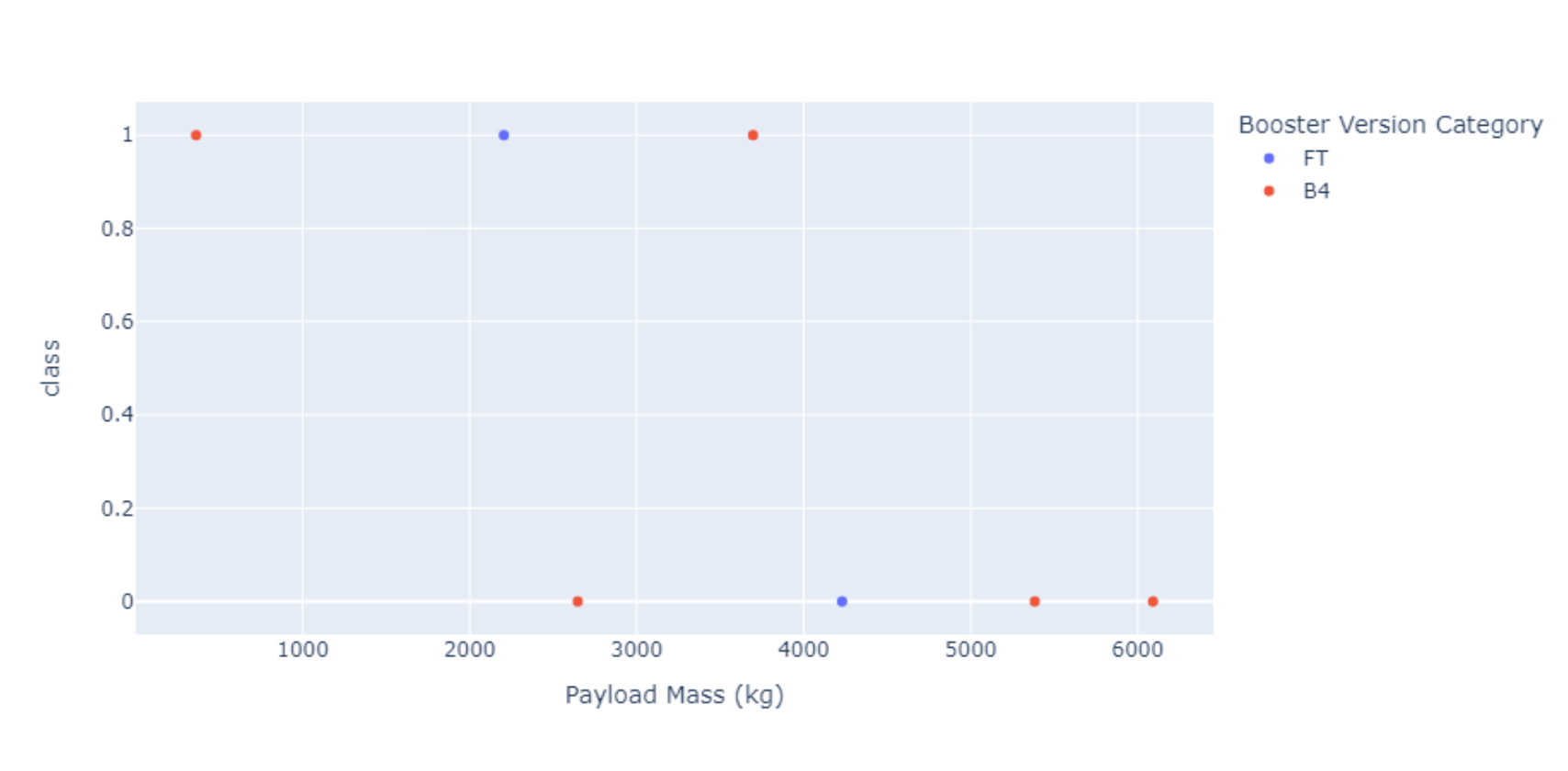
Results

- For KSC LC-39A site



Results

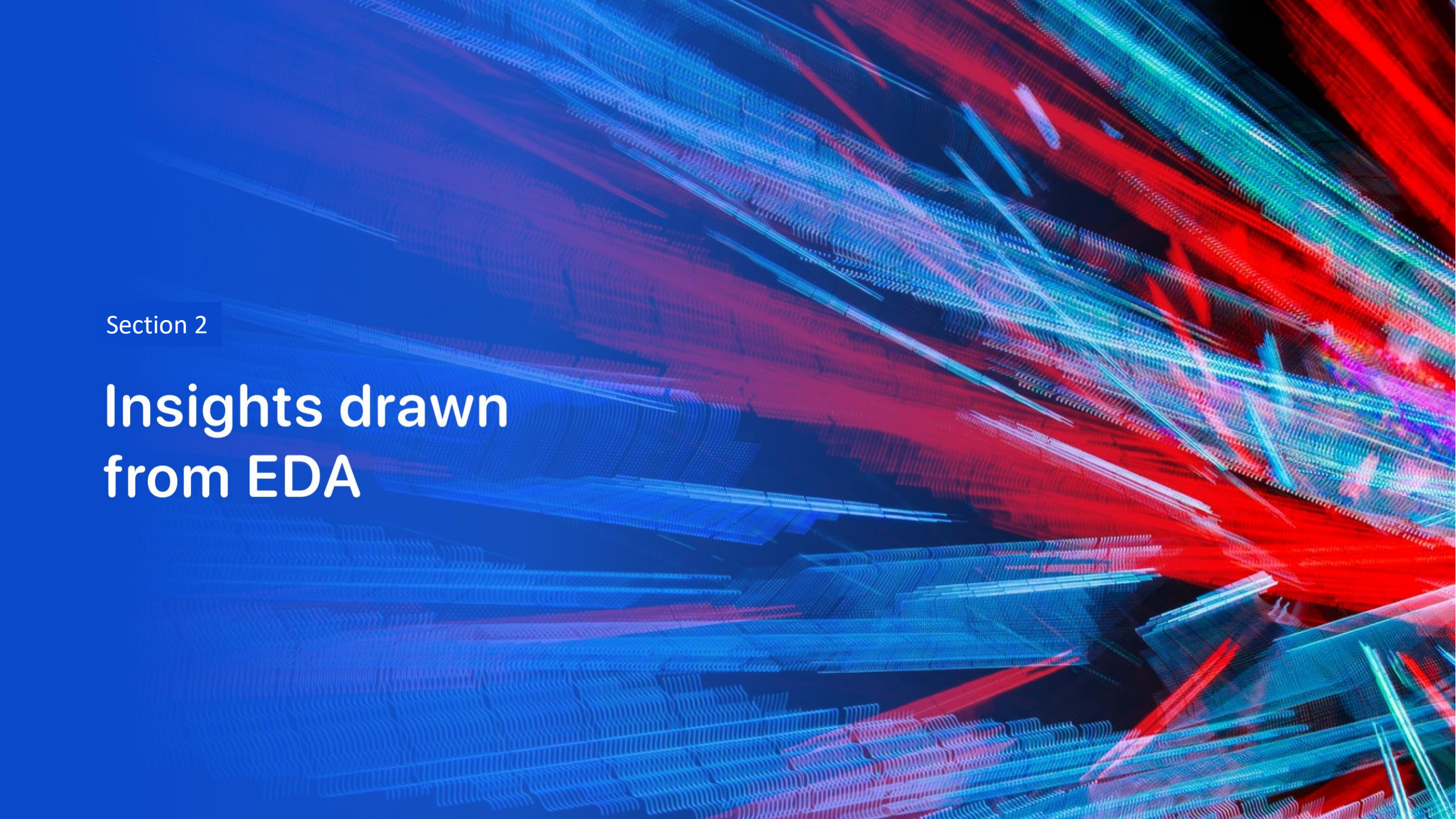
- For CCAFS SLC-40 site



Results

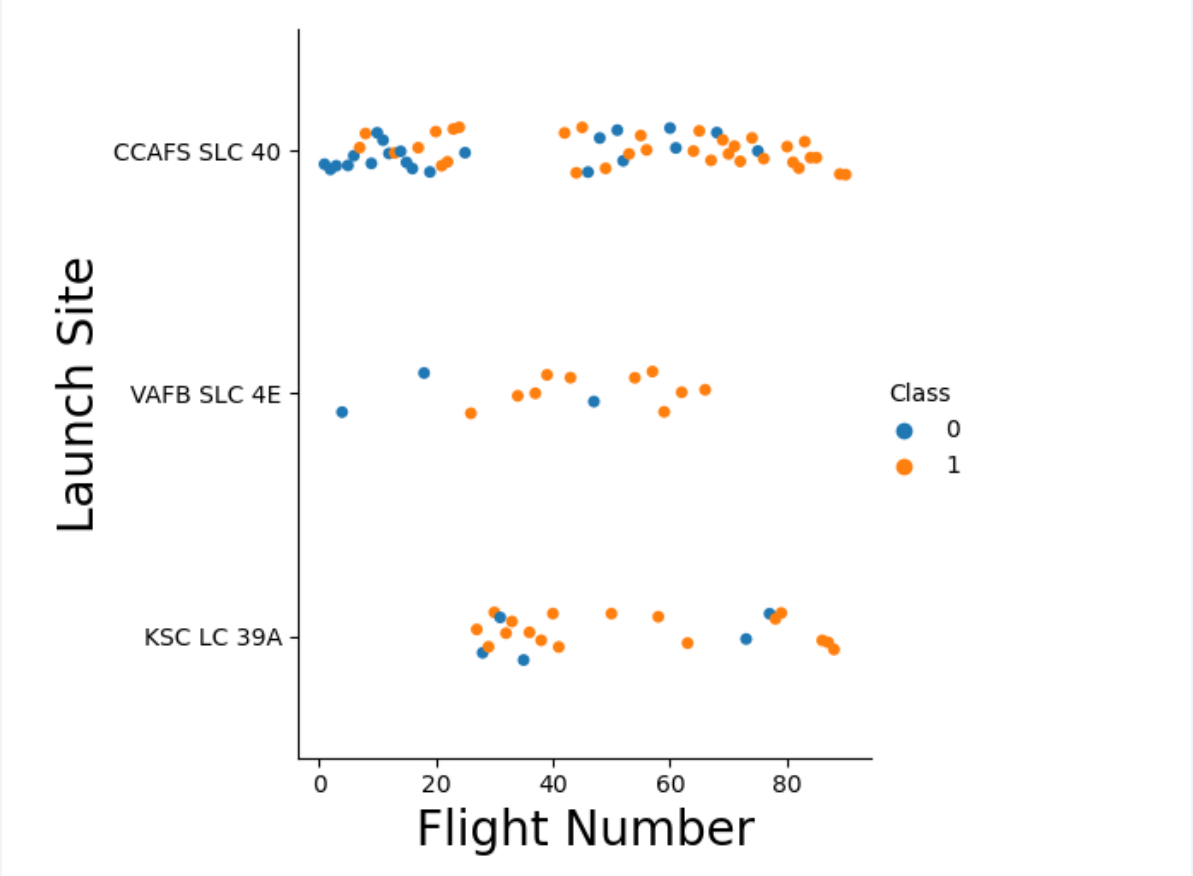
Predictive analysis

- On our predictive analysis we have tested four predictive models, K-Nearest Neighbours, Decision Tree, Logistic Regression and Support Vector Machine.
- When accuracies checked for these models using 10-fold cross validation their accuracies were approximately the same except for the decision tree. However, when the code ran multiple times to calculate accuracy, decision tree's accuracy flactuated. Sometimes it was below, the same and better than the other models. Therefore, I wouldn't choose the Decision Tree model for the prediction. But, other models can be used depending on preference.


Section 2
Insights drawn from EDA

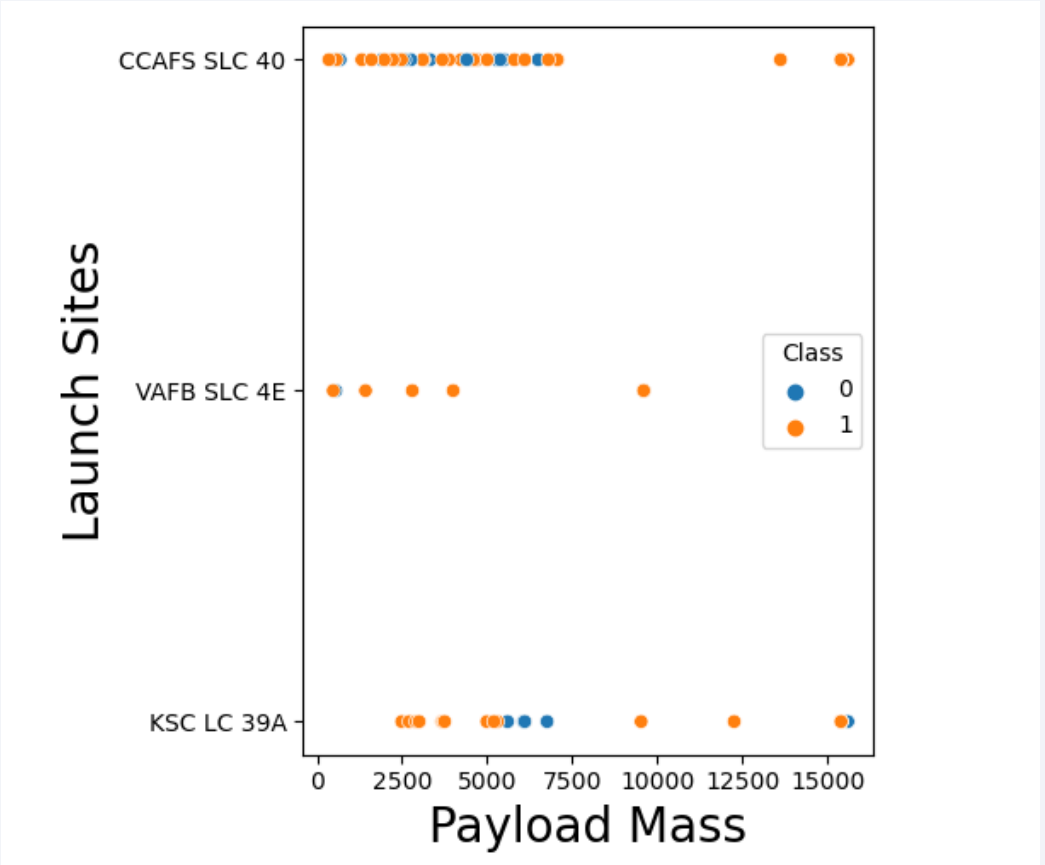
Flight Number vs. Launch Site

- From the plot it is inferred that site CCAFS SLC 40 hosts bulk of the launches. As the flight number increases, successful landings also increase on all of the Launch sites. KSC LC 39A's, and VAFB SLC 4E's success rate seems to be good. VAFB SLC 4E is used occasionally for launches.



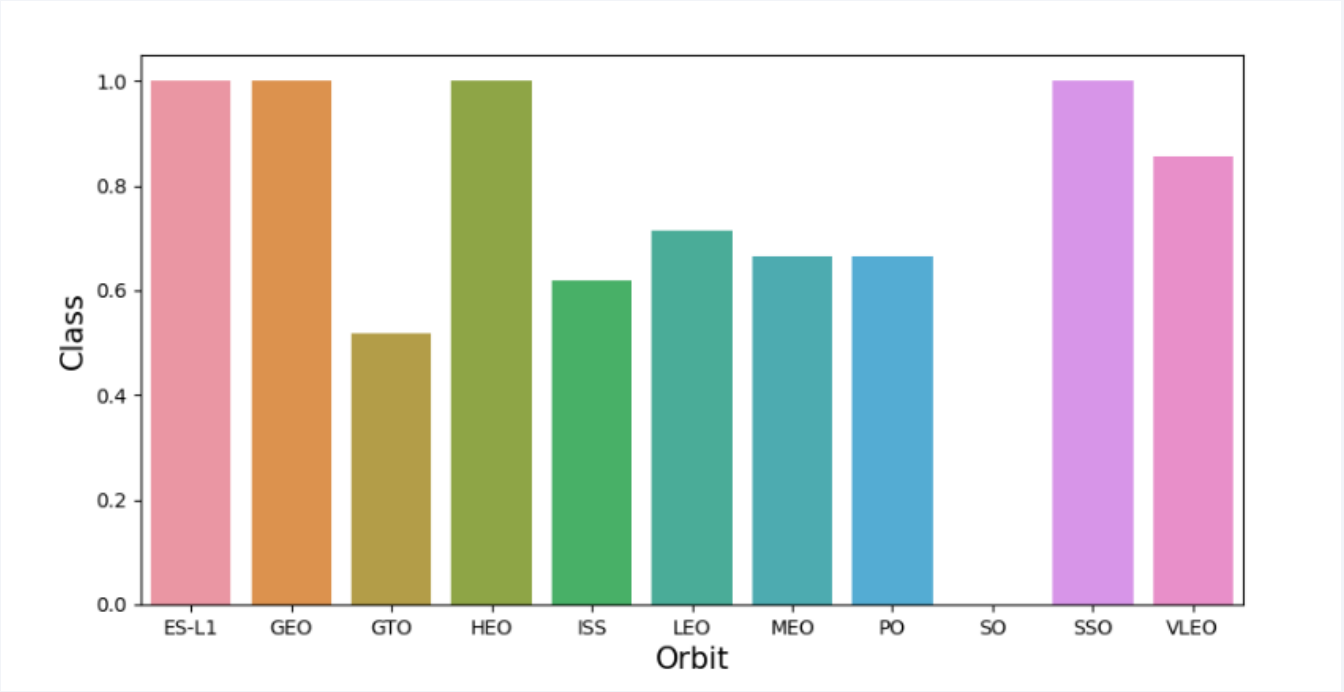
Payload vs. Launch Site

- For VAFB SLC, it is observed that very heavy payloads haven't been launched (>10000kg), usually low massed payloads have been launched here.
- For CCAFS SLC, we can see bulk of the launches are between 0 and 7500kg in that range there doesn’t seem to be a pattern for success. But, for very heavy loads even though very few all landings are a success.
- For KSC LC, payloads lower than 5000kg achieve successful landings. Also, there are no rocket launches with light payloads (<2000kg)



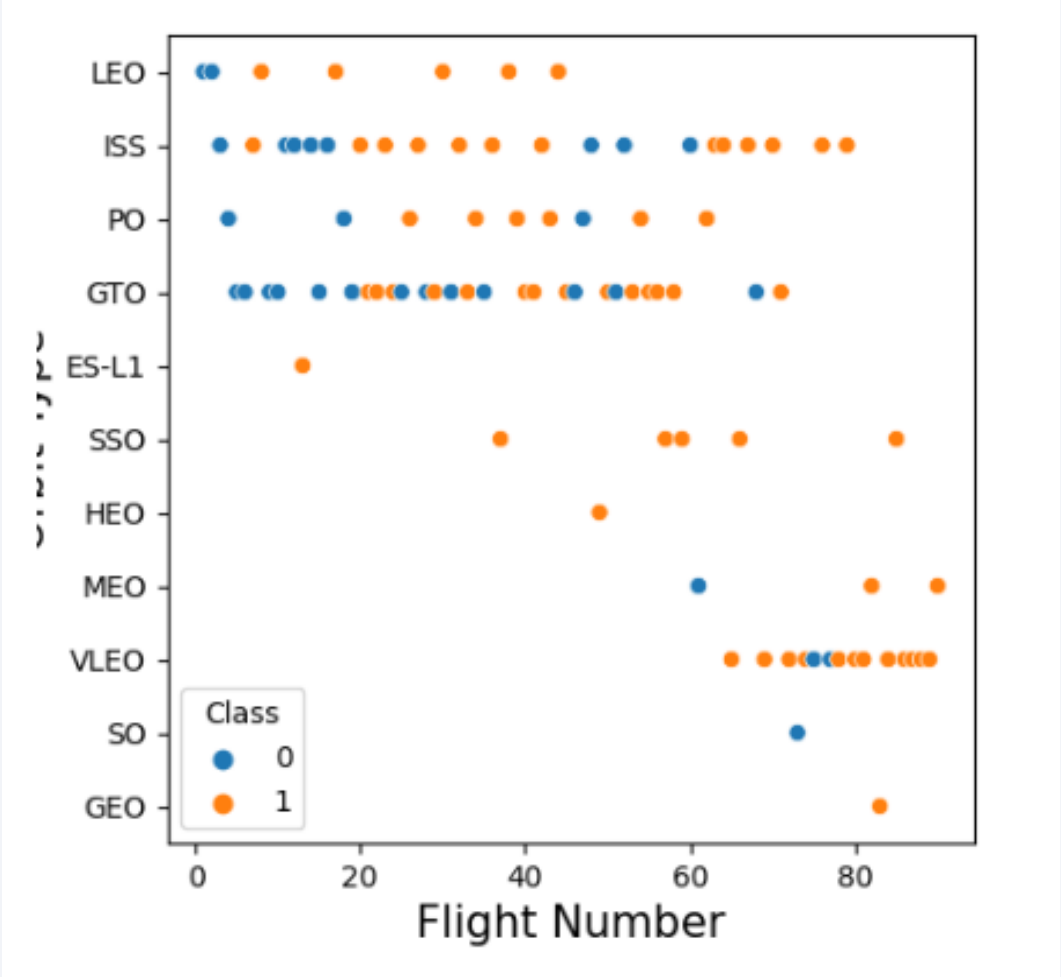
Success Rate vs. Orbit Type

- ES-L1, GEO, HEO, SSO orbits have perfect successful landings. VLEO also has a great success rate.
- Meanwhile, SO has not seen success at all, GTO has success at 50%, and the rest all have around 60% success rates.



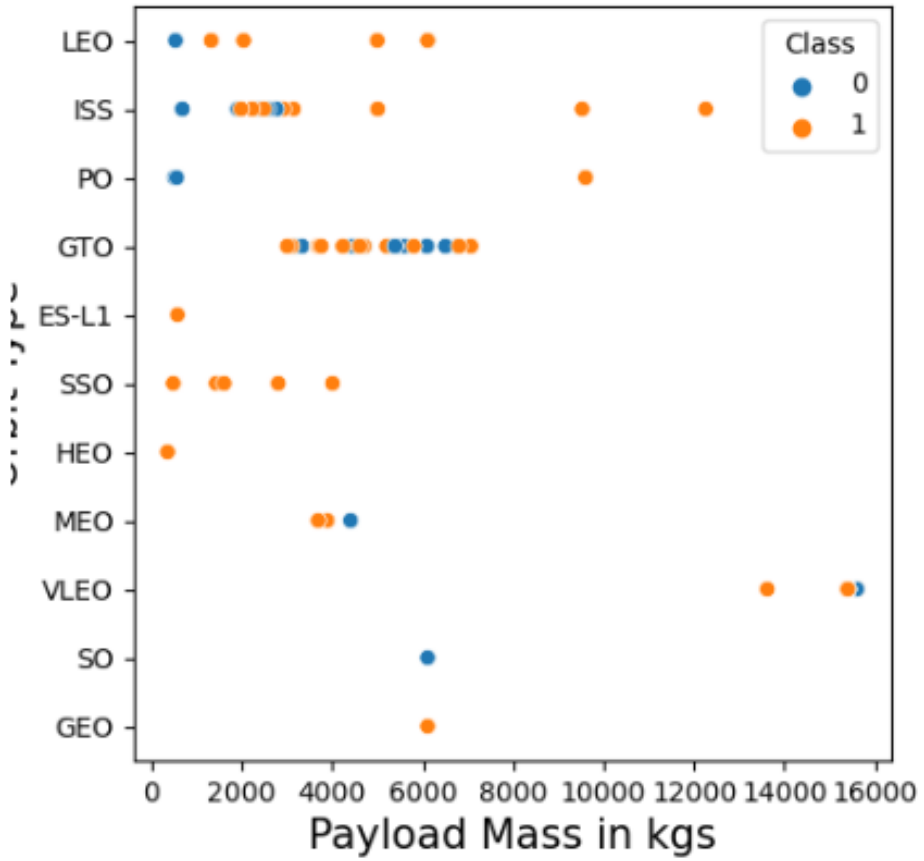
Flight Number vs. Orbit Type

- We can see that for LEO flight number has a relation with success.
- There doesn't seem to be a relation with flight number in GTO.
- Some of the orbits started getting used only after much later such as VLEO with great success.



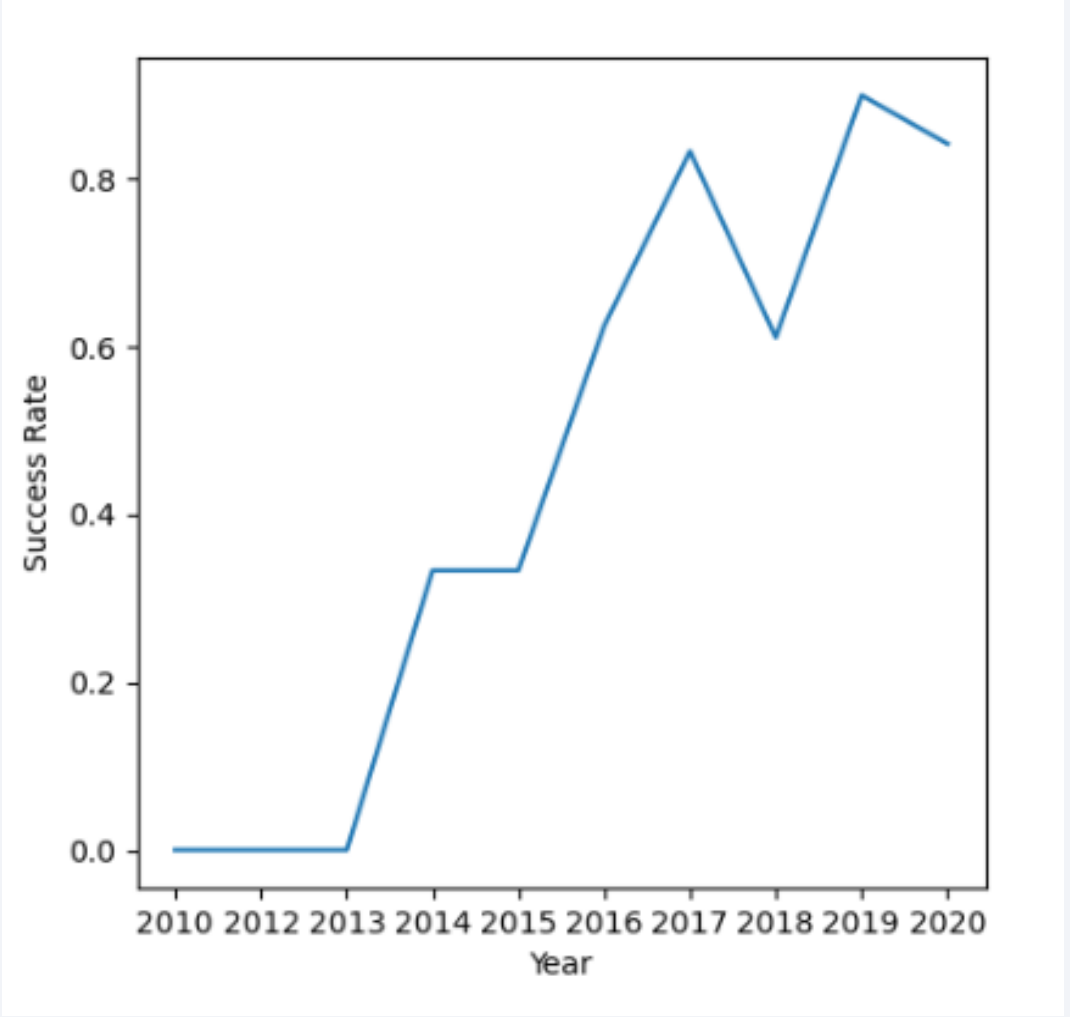
Payload vs. Orbit Type

- Higher success rates are observed for LEO, ISS, PO for heavy payload masses.
- However, we can't observe a relation to success rate on payload mass in GTO.
- Also, on SSO only lighter payloads have been launched and all ended up successful for landing.



Launch Success Yearly Trend

- We can clearly see a trend of increasing success with each passing year.



All Launch Site Names

- Find the names of the unique launch sites
- Query: Select DISTINCT("LAUNCH_SITE") from SPACEXTBL;
- There are only four launch sites in our data.

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`
- Query: SELECT * from SPACEXTBL WHERE "LAUNCH_SITE" LIKE "CCA%" LIMIT 5
- It shows the first 5 rows beginning with 'CCA' on launch sites in the data.

Date	Time (UTC)	Booster_Version	Launch_Site	Payload	PAYLOAD_MASS__KG_	Orbit	Customer	Mission_Outcome	Landing _Outcome
04-06-2010	18:45:00	F9 v1.0 B0003	CCAFS LC-40	Dragon Spacecraft Qualification Unit	0	LEO	SpaceX	Success	Failure (parachute)
08-12-2010	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)
22-05-2012	07:44:00	F9 v1.0 B0005	CCAFS LC-40	Dragon demo flight C2	525	LEO (ISS)	NASA (COTS)	Success	No attempt
08-10-2012	00:35:00	F9 v1.0 B0006	CCAFS LC-40	SpaceX CRS-1	500	LEO (ISS)	NASA (CRS)	Success	No attempt
01-03-2013	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
- Query: SELECT SUM(PAYLOAD_MASS__KG_) from SPACEXTBL WHERE "Customer" = "NASA (CRS)";
- It shows the sum of the payloads carried by the boosters provided from NASA across all the flights.

SUM(PAYLOAD_MASS__KG_)
45596

Average Payload Mass by F9 v1.1

- Calculate the average payload mass carried by booster version F9 v1.1
- Query: SELECT AVG("PAYLOAD_MASS__KG_") FROM SPACEXTBL WHERE "Booster_Version" = "F9 v1.1";
- It shows the average payload mass carried by rockets using booster version F9 v1.1

AVG("PAYLOAD_MASS__KG_")
2928.4

First Successful Ground Landing Date

- Find the dates of the first successful landing outcome on ground pad
- Query: SELECT min("Date") AS "DATE" from SPACEXTBL WHERE "Landing _Outcome" = "Success (ground pad)";
- Among the landing outcomes where the landing went successfully onto a ground pad it selects the earliest date.

DATE
01-05-2017

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
- Query: SELECT "BOOSTER_VERSION" from SPACEXTBL WHERE "LANDING _OUTCOME" = "Success (drone ship)" AND "PAYLOAD_MASS__KG_" BETWEEN 4000 and 6000;
- Among the successful landing outcomes onto drone ship and the payload mass being in between 4000kg and 6000kg, these booster versions have been used.

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
- Query: SELECT "MISSION_OUTCOME", COUNT("MISSION_OUTCOME") from SPACEXTBL GROUP BY "MISSION_OUTCOME"
- It groups all the unique mission outcome names with the frequency it appears.

Mission_Outcome	COUNT("MISSION_OUTCOME")
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
- Query: SELECT DISTINCT("booster_version") FROM SPACEXTBL WHERE "PAYLOAD_MASS__KG_" = (SELECT max("PAYLOAD_MASS__KG_") from SPACEXTBL);
- Among the entries where maximum payload is selected, it shows the unique names appearing in booster_version.

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
- SELECT substr(Date, 4, 2) AS "Month", "LANDING _OUTCOME", "Booster_version", "Launch_site" from SPACEXTBL where "LANDING _OUTCOME" = "Failure (drone ship)" and substr(Date,7,4)='2015'
- Shows the months, landing outcomes, booster versions, launch sites in year 2015 where landing outcomes are failures on drone ships.

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

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
- SELECT "Landing _Outcome", count("Landing _Outcome") AS "Count" from SPACEXTBL GROUP BY "Landing _Outcome" HAVING ("DATE" BETWEEN "04-06-2010" AND "20-03-2017") AND ("Landing _Outcome" LIKE "suc%") ORDER BY count("Landing _Outcome") DESC
- Among the entries between the dates 04/06/2010 and 20/03/2017 and landing outcomes being successful, showing the unique landing outcomes and how many times they appeared in a descending order.

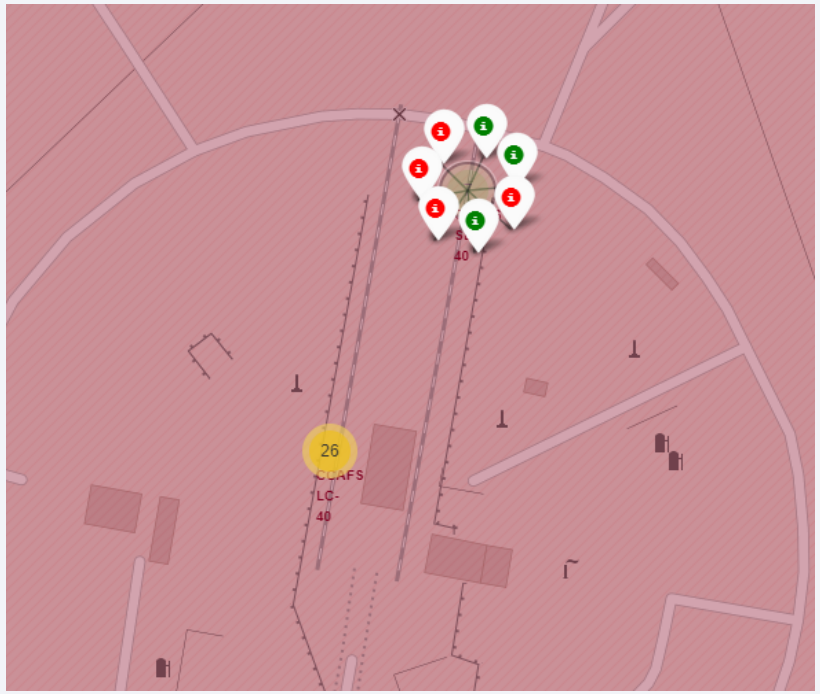
Landing _Outcome	Count
Success (drone ship)	14

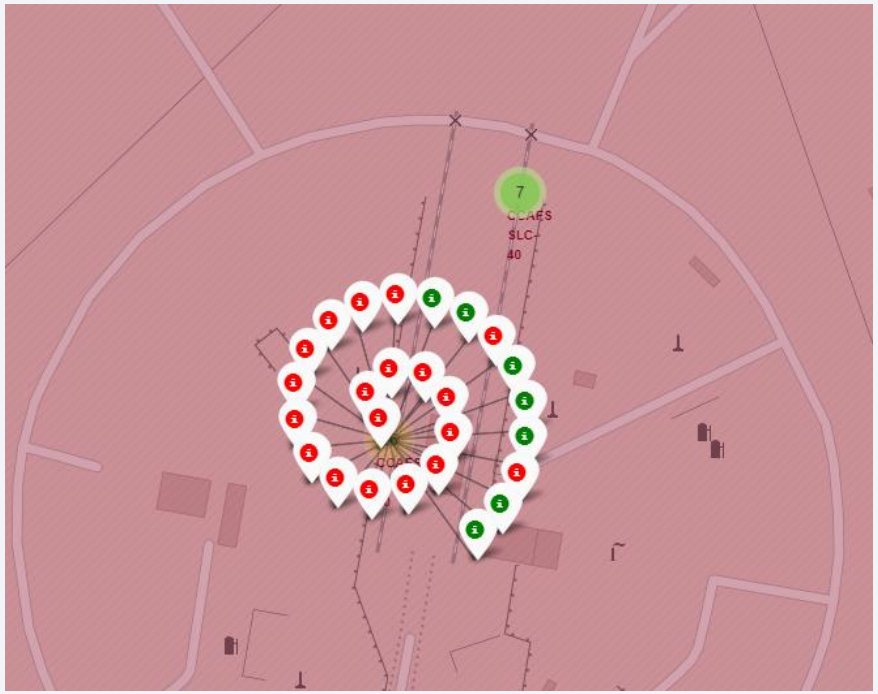

Section 3
Launch Sites
Proximities Analysis

Launch Site Locations

- All launch sites were located on the coasts of USA. What is interesting is three of the launch sites were located in Florida, and they are incredibly close to each other. While the other one is located in California.

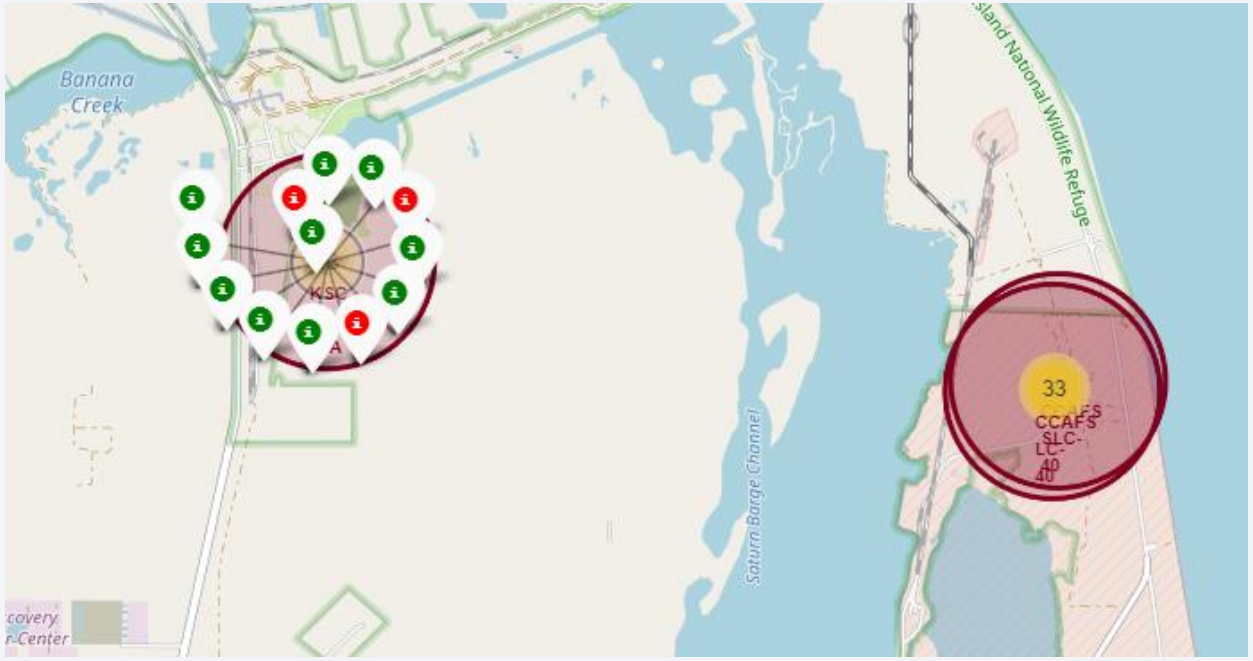
Landing Outcomes by Sites - 1

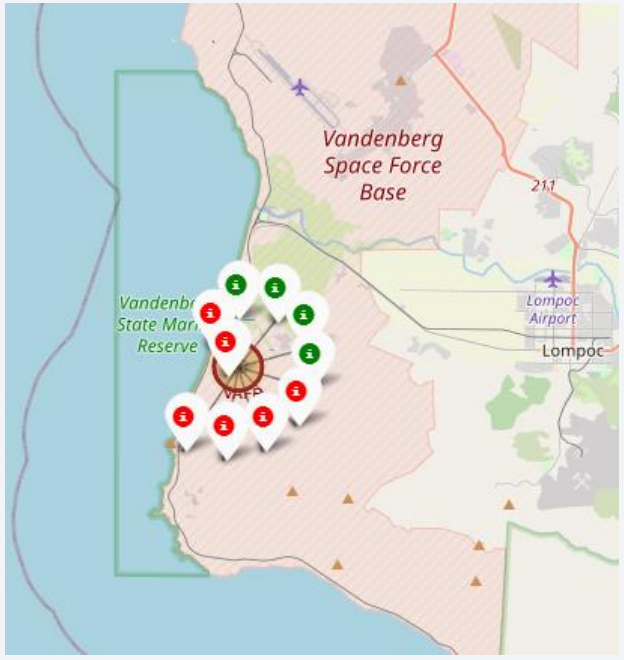




- Here, we see the CCAFS SLC-40 and CCAFS LC-40 launch sites. The red marks indicate failure and green marks indicate success. These two sites are pretty close to each other. We see that the success rate of CCAFS LC-40 is pretty bad, while the other seems decent.

Landing Outcome by Sites - 2

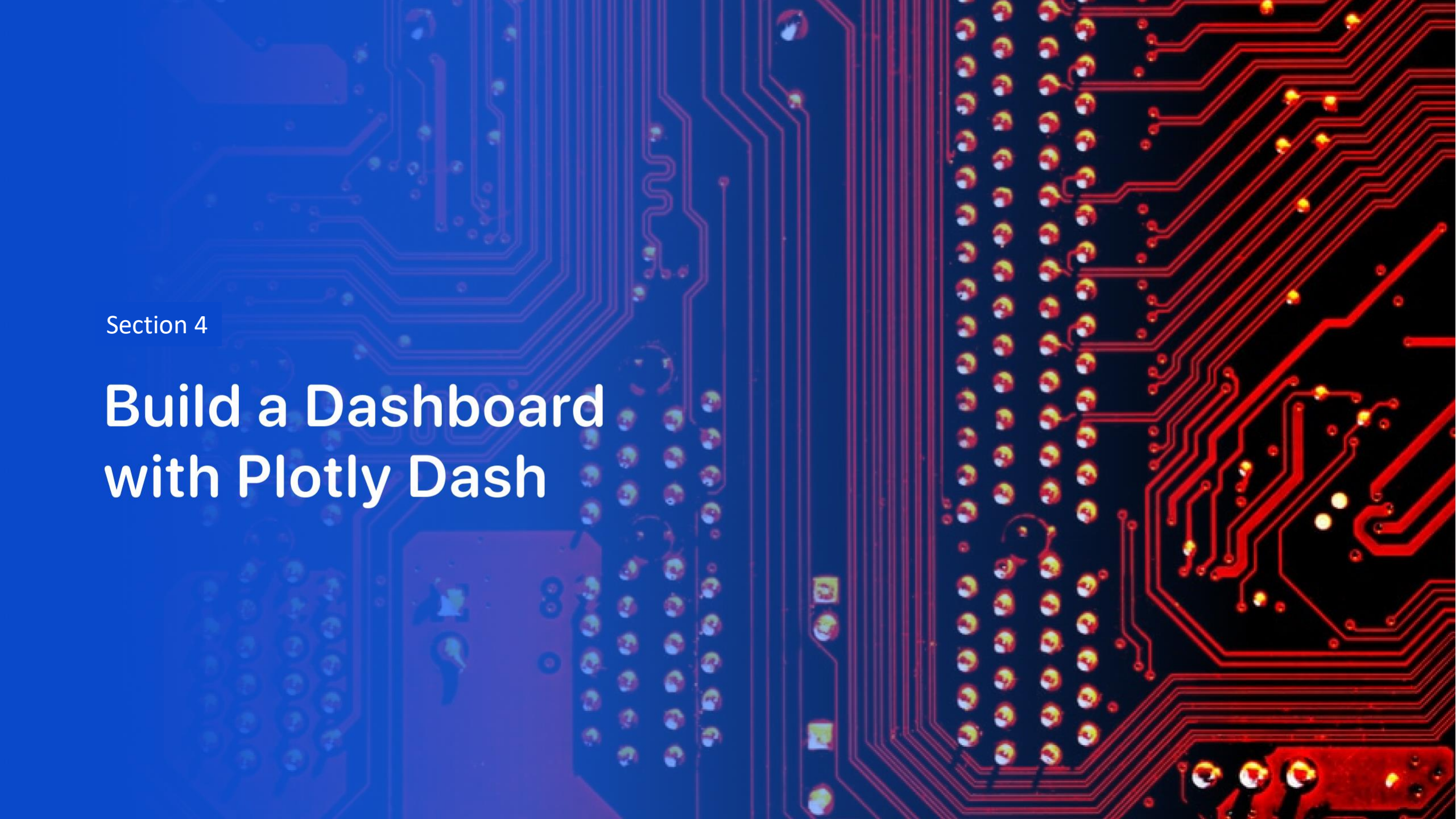




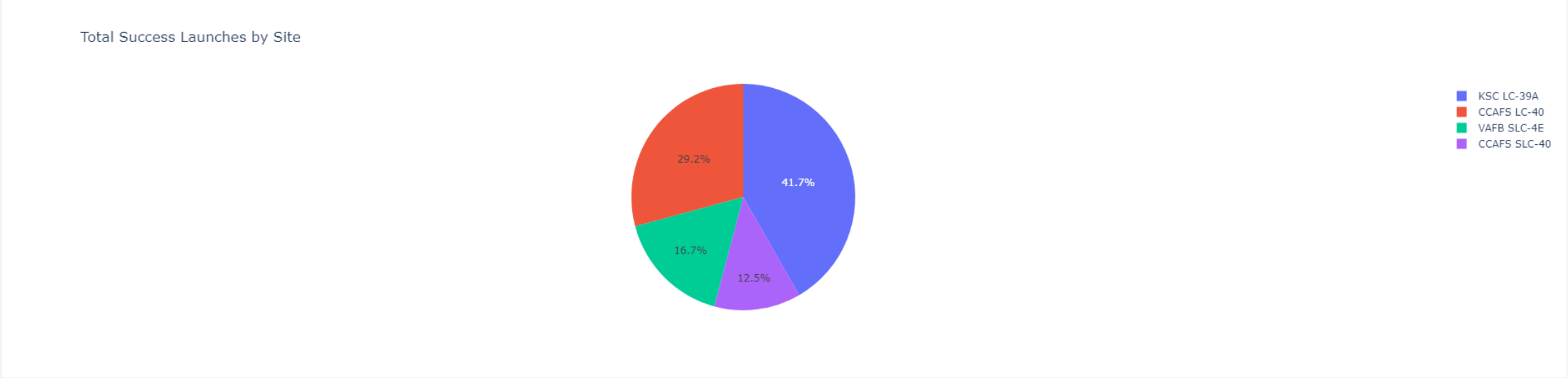
- Here, we see two Launch Sites, KSC LC-39A and VAFB SLC-4E. KSC LC-39A is located really close to the other two and as we see its success rate is amazing compared to the other sites and it's the only one with above 50% rate. Meanwhile, VAFB SLC-4E is located in California and it's success rate is decent.

Distance to Infrastructure

- I have chosen VAFB SLC-4E launch site located in California to show some various distances to infrastructure. It can be seen that coast line and the railway is really close to the site, and the town center and public airport is quite far away. Since it is close to the roads and railways, it provides a logistical advantage. Also, it is far away to city centers and such also close to the coast, in case of a failure in launch, the damage is minimized.

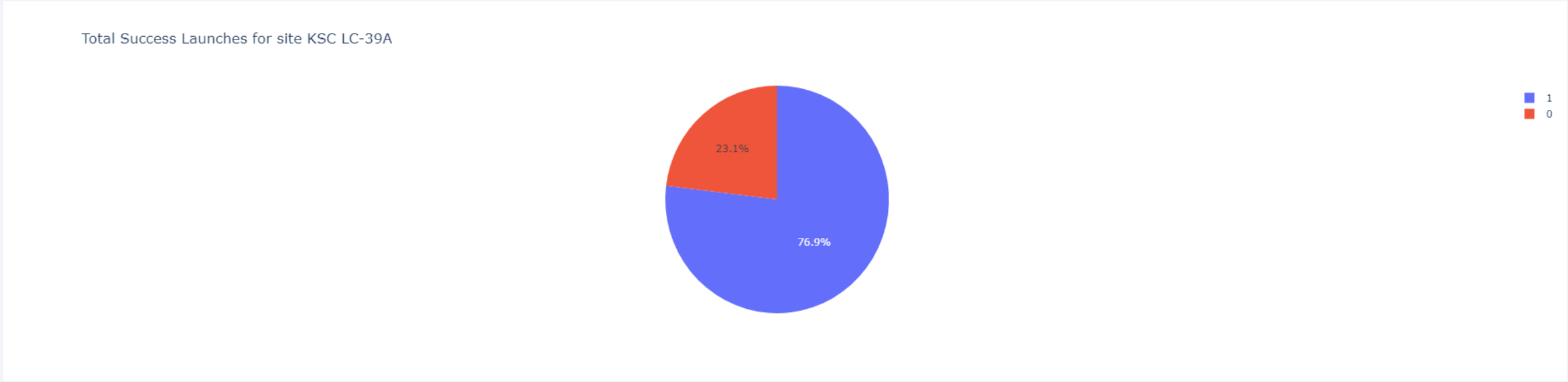

Section 4
Build a Dashboard with Plotly Dash

Successful Launches by Site



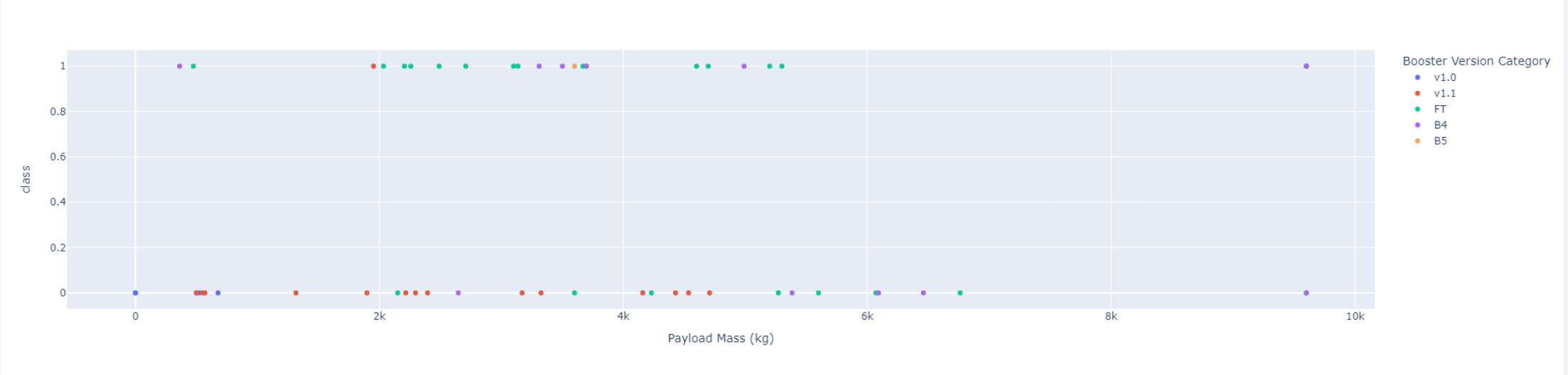
- The blue pie chart is responsible from almost half of the successful launches, red one is also close to it but the launches made from green and purple sites' successful launches are few compared to the others.

Success Rate of KSC LC-39A



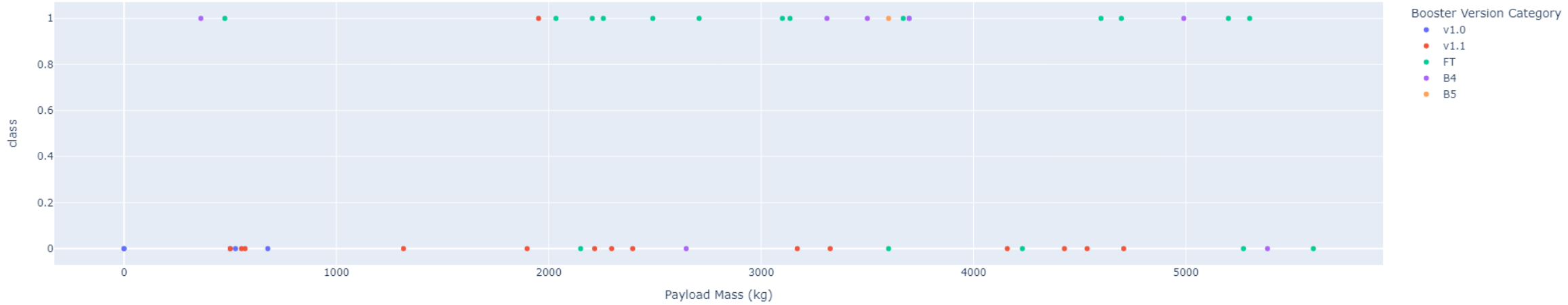
- This launch site was responsible for the bulk of the successful launches on our previous chart. Now, we see on this chart that launch site actually has an amazing success rate.

Effect of Payload Mass, Booster Version on Outcome - 1



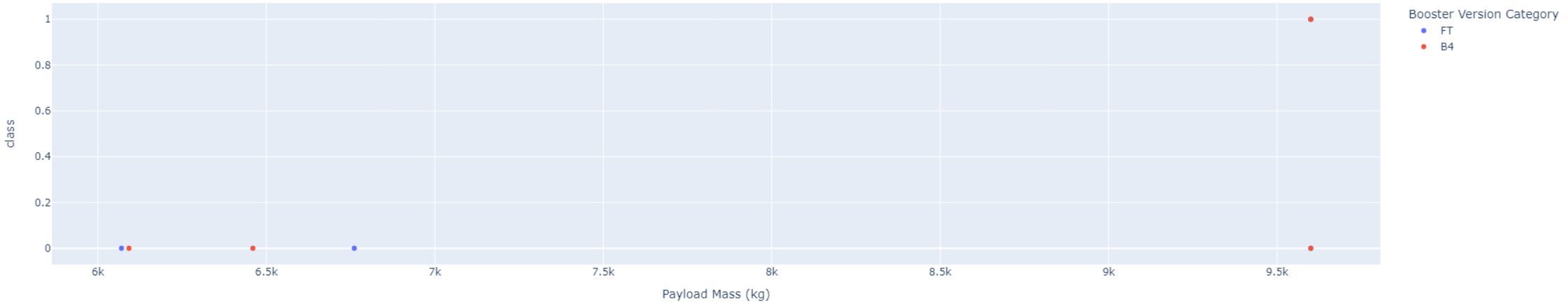
- We can see the effect of payload mass to the outcome of the launch for all sites, and booster version category colour coded.

Effect of Payload Mass, Booster Version on Outcome - 2



- When we limit the payload mass between 0 and 6000kg, we can see that most of the successful launches are done by the FT booster category marked by green, while most of the failures have happened while using v1.1 booster category marked by red. It can also be seen there hasn't been much success with light payloads between 0 and 2000kg. B4 booster category also seems like it has a decent success rate.

Effect of Payload Mass, Booster Version on Outcome - 3



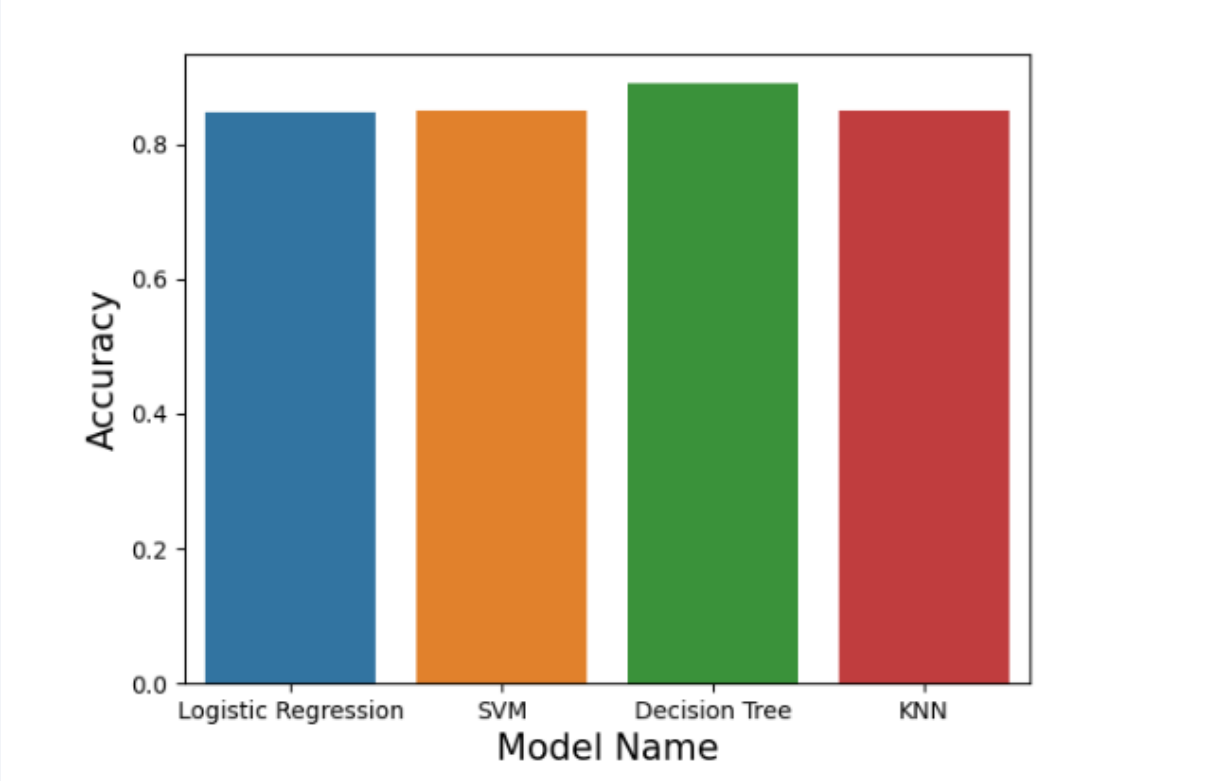
- When we check the payload mass above 6000kg. It can be seen that success drops dramatically, it consists mostly of failures. Also, only two booster version categories were used to launch above 6000kgs and only B4 has seen success only once on heavy load. Not many launches were attempted compared to the lighter payloads.

Section 5

Predictive Analysis (Classification)

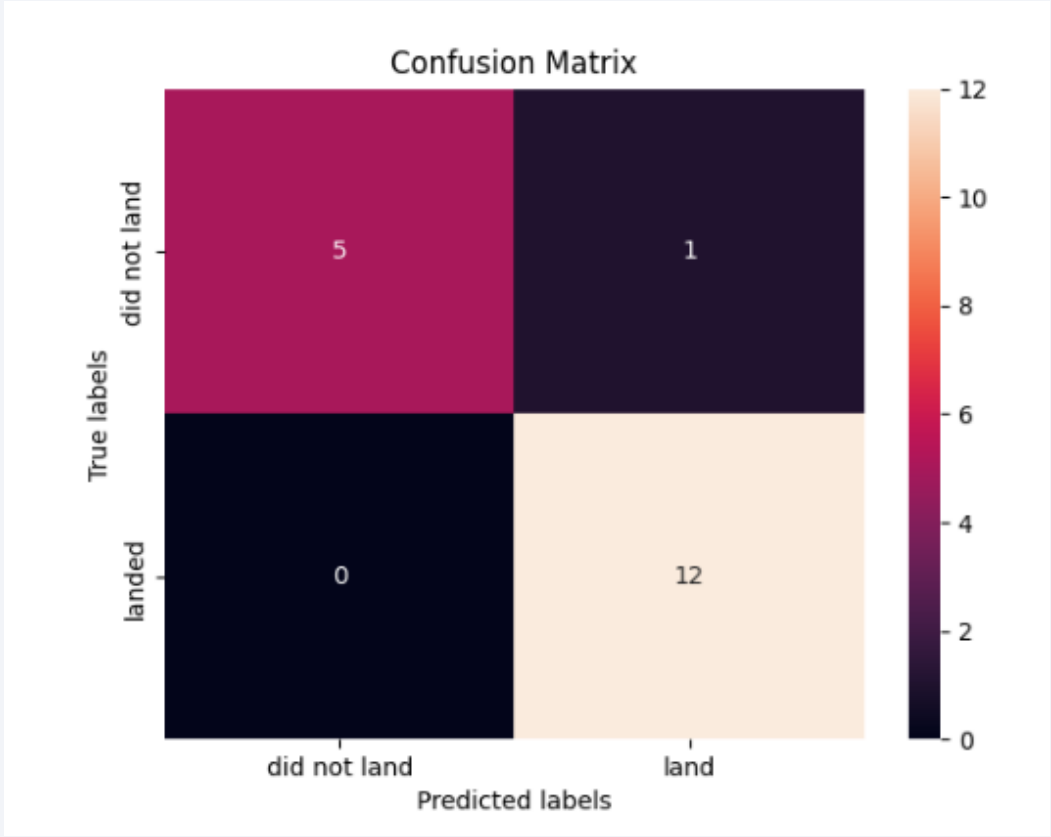
Classification Accuracy

- We can see that Decision Tree has the highest cross validation accuracy, but they are all really close to each other.



Confusion Matrix

- Best performing model was Decision Tree. There are no false negatives as we see all the labels that landed has been correctly classified as landed. There is also only one false positive, the rest of the labels were correctly predicted. Which is pretty impressive. Out of the six labels indicating did not land 5 of them were correctly predicted and out of the 12 labels indicating landed have all been predicted correctly.



Conclusions

- On lighter payloads (up to 6000kg) ISS, SSO, and LEO have the bulk of their launches and are successful, ISS also seems promising for heavier payloads.
- On later flights VLEO has started to be used by great success rates.
- KSC LC 39-A launch site has the best success rate.
- Launch site VAFB SLC 4-E performs great with payloads lighter than 10000kg. While above 10000kg, CCAFS SLC-40 performs great.
- Booster version category FT has great success rate with mass being below 6000kg. While, v1.1 has a terrible one for the same mass range.
- Decision Tree should be used as the predictive model as it has the highest accuracy.

Appendix

- All of the python code, sql queries on the jupyter notebooks for this project can be in my github in this link.

Thank you!