IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

Aayush Chaurasia
March 12, 2025

Outline

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

Executive Summary

- For analysis on SpaceX past launches along with creating a predictive model and an interactive dashboard involves the following steps:
  - Data collection using SpaceX API and web scraping from Falcon9 wiki page.
  - Data wrangling and visualization using pandas and seaborn respectively. SQL queries also helped to understand the data in great detail. Folium helped to understand the geospatial data of SpaceX launch sites.
  - Creating classification models using Logistic Regression, SVM, Decision Tree and KNN.
  - Created a dashboard using 'plotly dash' to give the data insights a presentable view to the target audience.
- Based on the analysis and creating predictive model we can say that the chance of first stage rockets to land depends on their payload mass, orbit, launch site which will help to attain optimal decision thereby controlling the cost of project.

Introduction

- The commercial space age is here and companies are investing in space projects to make space travel convenient and affordable to the audience. There are several companies like Blue Orbital, Virgin Galactic and Rocket Lab working hard to make satellite providers and reusable rockets.
- However, SpaceX founded by billionaire industrialist Elon Musk is one of the top companies among them who have a steady growth in their space missions with one of their popular projects called Falcon 9.
- Our company SpaceY is trying to analyze the landing success probability of first stage rockets based on the data gathered from SpaceX sites in order to take a decision for future space projects for our company.
- To solve this problem, we need to address the following questions:
  - How many landing sites are there in this project?
  - Which landing sites have the most successful landings?
  - How factors like boosters, payload mass affects the landing success?

Section 1

Methodology

Methodology

Executive Summary

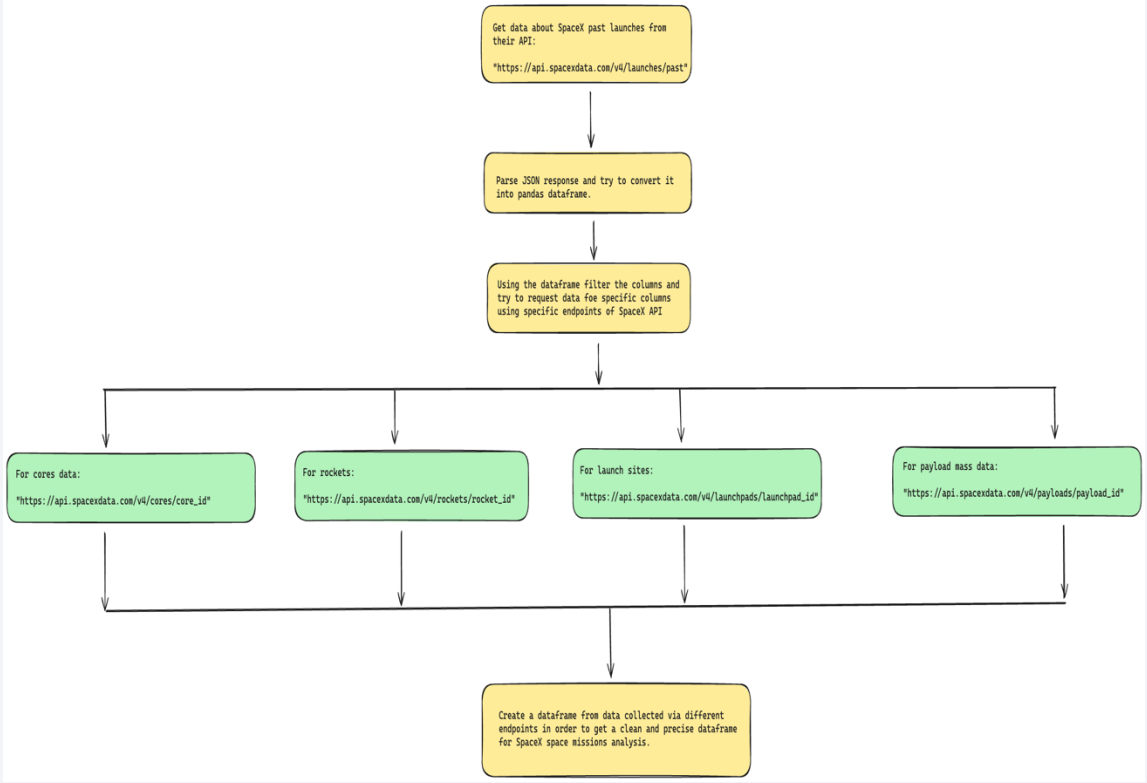
- Data collection methodology:
  - For collection of data we used SpaceX API endpoint (api.spacexdata.com/v4/) with data about cores, rockets and their past launches via web scraping tools.
- Perform data wrangling
  - The collected data consists Flight Number, Date, Booster Version, Payload Mass, Orbit, Launch Site and Outcome. We will convert Outcome as target binary class data (0 or 1). The remaining data will be transformed as numerical data with standardization.
- Perform exploratory data analysis (EDA) using visualization and SQL
- Perform interactive visual analytics using Folium and Plotly Dash
- Perform predictive analysis using classification models
  - We used various classification models including Logistic Regression, Support Vector Classifier, Decision Tree Classifier and KNN with best hyperparameters using Grid Search CV to pick the best model among them.

Data Collection

- For data collection, we used SpaceX REST API endpoint along with web scraping libraries like BeautifulSoup to collect data on rockets, past launches and other details to get insights on Falcon9 space missions.
- The API had several endpoints:
  - /cores: Details about the rocket cores.
  - /rockets: Details about their rockets, with booster types and versions.
  - /past: This contained data about their past launches with their corresponding dates.
- The data was scraped as HTML format which was later converted as pandas dataframe format which is a 2-dimensional tabular data.

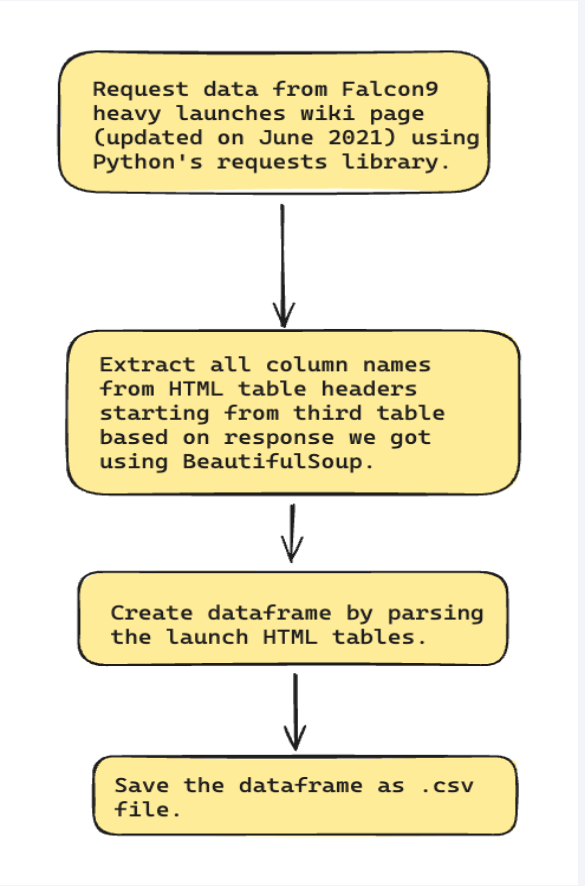
Data Collection – SpaceX API

- First we got data on past launches from /past endpoint.
- Then using the parsed data we'll request data from different endpoints by passing the unique id that was there in the parsed data.
- We will parse data from:
  - /cores
  - /payloads
  - /rockets
  - /launchpads
- For reference, check out the jupyter notebook on this GitHub repo.



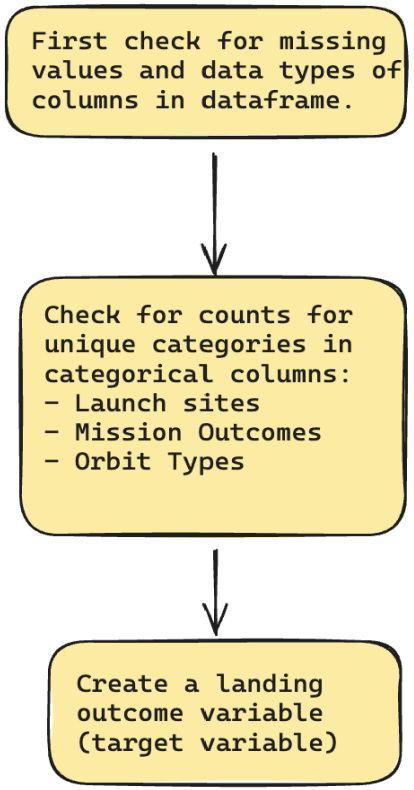
Data Collection - Scraping

- Using requests library get the data from wiki page and scrape the text response using beautifulsoup.
- Then extract the column names from table headers starting from the third table.
- For complete process of data collection via scraping, checkout the GitHub repo.



Data Wrangling

- First, we will check the missing values and data types of columns in the dataframe.
- Check for counts for the following:
  - Number of launches on each site.
  - Number and occurrence of each orbit.
  - Number of occurrence of mission outcomes of the orbits.
- Create a landing outcome label from Outcome column which will be our target variable.
- For further details, go to my repo.



EDA with Data Visualization

- For visualizing relationship between two variables, we used scatter plots for:
  - Payload Mass vs Flight Number
  - Launch Site vs Flight Number
  - Launch Site vs Payload Mass
  - Orbit vs Flight Number
  - Orbit vs Payload Mass
- For relationship between success rate and orbit, bar graph was used.
- To visualize launch success yearly trend we used line plot.
- For reference, checkout the GitHub repo.

EDA with SQL

- Displayed the names of unique launch sites in the space mission.
- Displayed the first five launch sites starting with 'CCA'.
- Displayed total and average payload mass carried by boosters launched by NASA (CRS).
- Listed the date when first successful landing was achieved.
- Listed the names of boosters which have success in drone ship and have payload mass between 4000 and 6000.
- Listed total number of successful and failed mission outcomes.
- Displayed the names of booster versions which have carried maximum payload mass.
- Displayed monthly failed missions in the year 2015.
- Displayed the count of failure (drone ship) and success (ground pad) between dates 2010-06-04 and 2017-03-20 in descending order.
- Reference - GitHub Repo

Build an Interactive Map with Folium

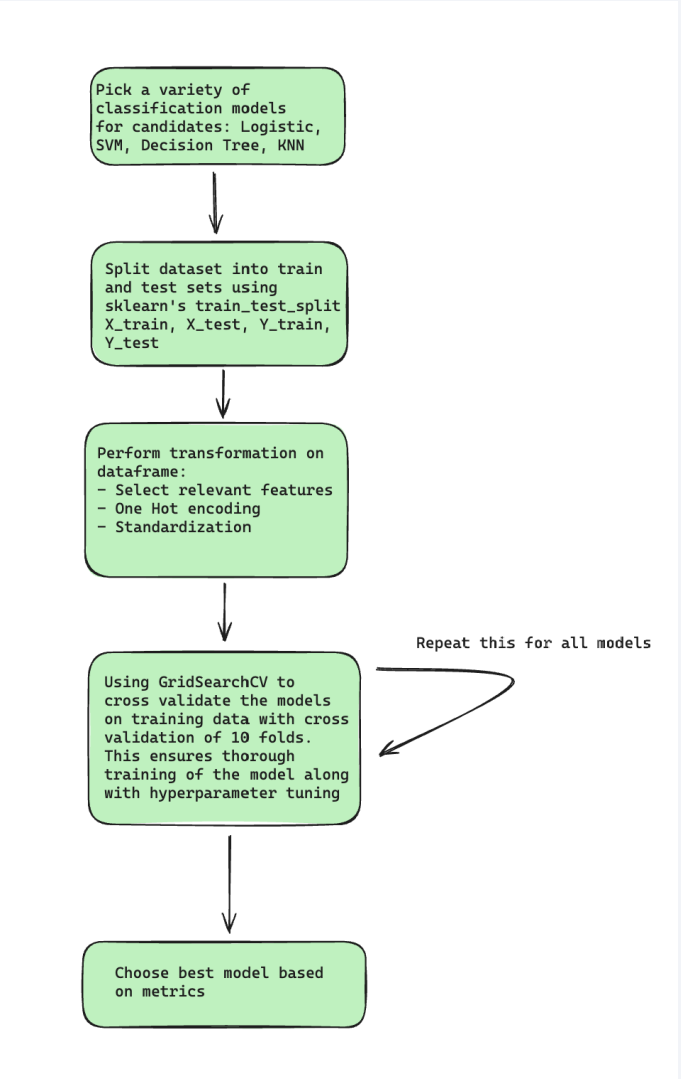
- First we created a Folium Map object followed by markers to display the launch sites where the space missions took place in the past on the map.
- Then, to display the successful and failed missions, we created a marker cluster object and based on their corresponding color (success = green, failure = red) we marked their locations on the map using marker object.
- Then we calculated the distance of launch site from nearest railway, coastline, city and highway to know the proximities of the site.
- Reference - Repo

Build a Dashboard with Plotly Dash

- To build a dashboard we used the following components and graphs:
  - Dropdown
  - Pie chart
  - Range Slider
  - Scatter plot
- The dropdown has a list of launch sites from which we can choose and render the pie charts and scatter plots based on that.
- The Range slider controls the payload mass range from 0 (min) to 10000 (max).
- The pie chart displayed the success rates of various sites based on option selected from dropdown.
- The scatter plot displayed relation between outcome (class) and payload mass based on the launch site and payload range selected.
- GitHub Repo

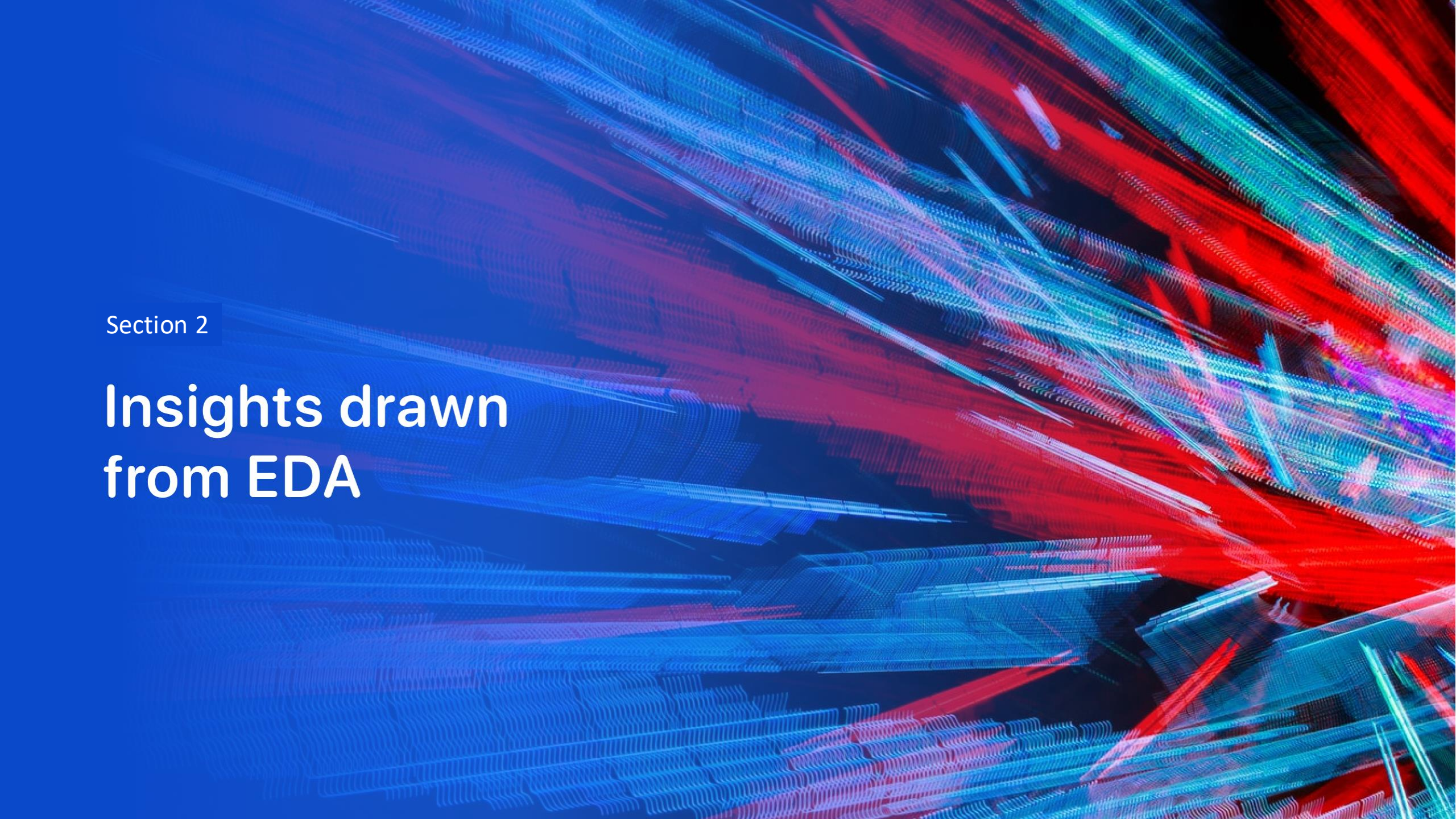
Predictive Analysis (Classification)

- First we considered a set of algorithms for classification which includes: Logistic Regression, SVM, Decision Tree and KNN classifier.
- This was followed by splitting the dataset into training and testing sets.
- Then we passed the transformed train data after selecting features, one hot encoding categorical and standardizing it to the model and evaluated their metrics one by one.
- Using GridSearchCV we performed hyperparameter tuning of the models in order to pick best among them.
- The model which had the best metrics was considered as the best classifier model.
- Reference – GitHub Repo



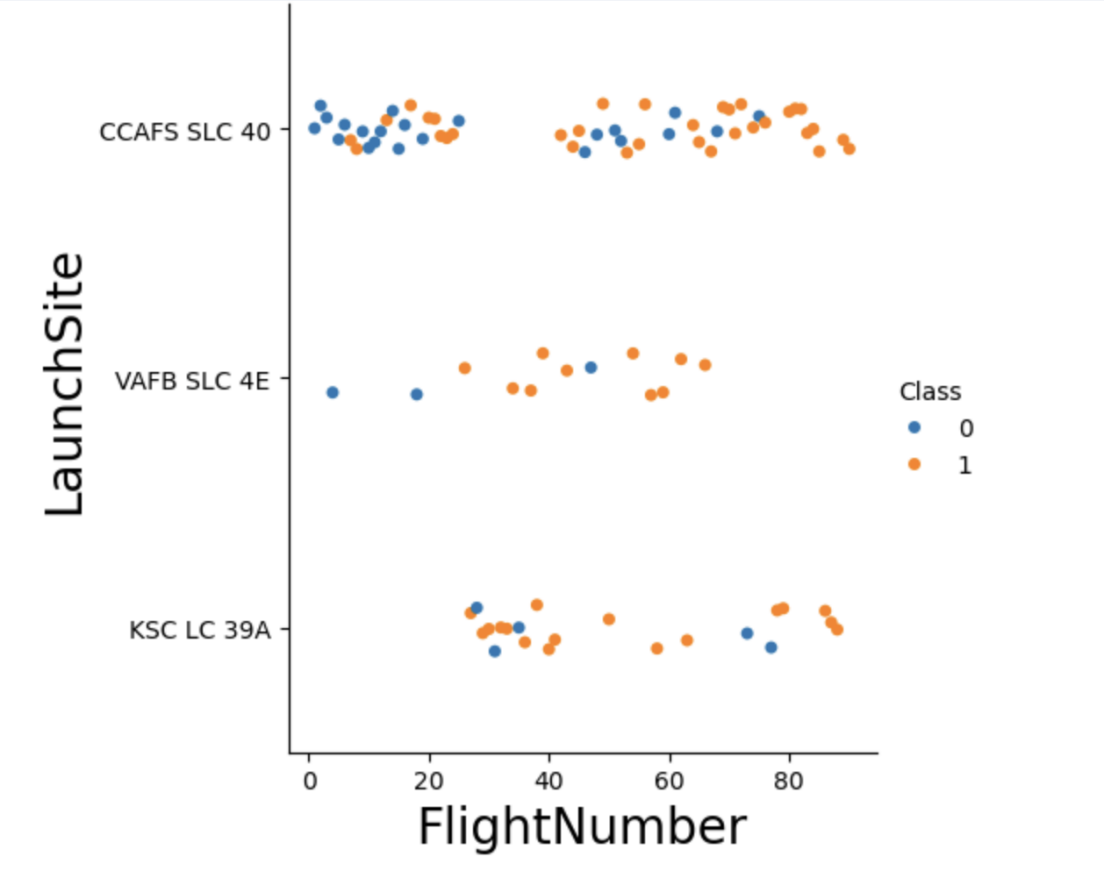
Results

- After doing EDA, it was evident that:
  - With time progression, success rates increased at a steady rate since 2013.
  - Only particular orbits like GEO, PO and ISS showed an increase in first stage phase success when payload was increased.
  - Landing site like KSC LC-39A had highest success rate among all landing sites which could be one of the most preferred locations for first stage launch missions.
- The predictions calculated by the model shows that:
  - Model is able to obtain high precision (high amount of true positives) and god recall (quite a few false positives), that shows that the model is able to grasp the patterns in the data.
  - Since there were multiple candidates for best model. Keeping simplicity in mind, it was best to go with Logistic Regression as the data was already transformed with no missing and no outlier values. Decision Tree might be good, however it was overfitting on the train data making it a less preferred choice. One may requiring using pruning techniques to make it a strong candidate for the final model.


Section 2
Insights drawn from EDA

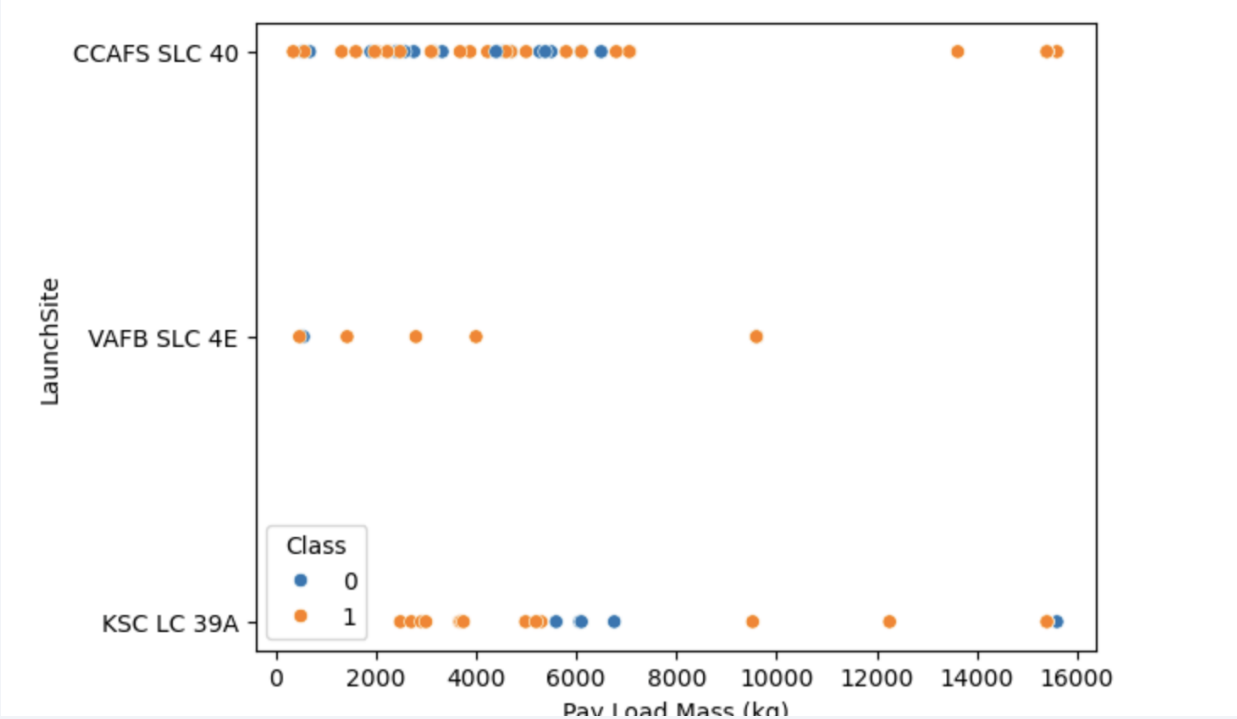
Flight Number vs. Launch Site

- The scatter plot on the right demonstrates the relation between LaunchSite and FlightNumber.
- Here the scatterplot is colored on the basis of mission outcome (0 for failure and 1 for success).
- As we can see the patterns in the plot shows that the chances of successful landing gets increased as the flight number increases.
- This is because due to failed missions in the past, they are trying to mitigate the drawbacks that have happened in the past space missions in their first stages.



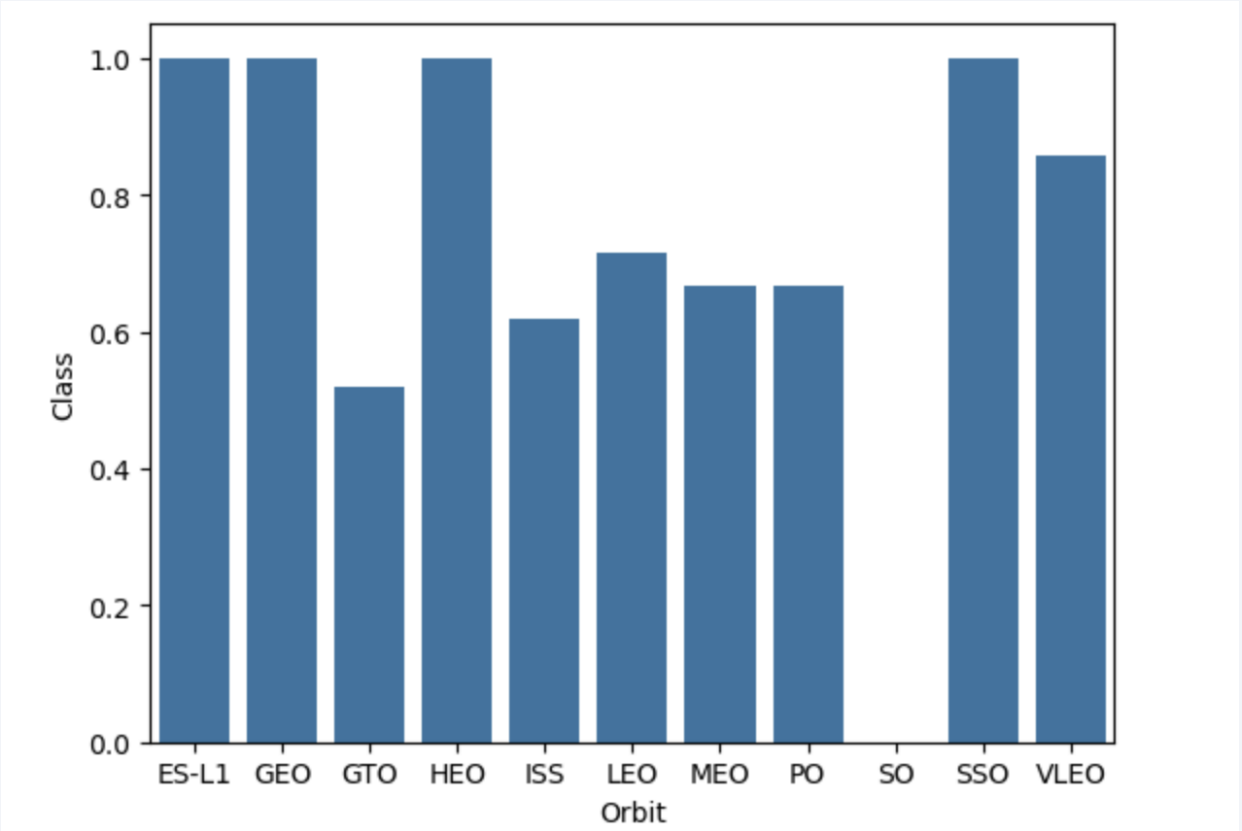
Payload vs. Launch Site

- The scatter plot on the right demonstrates the relation between LaunchSite and Payload Mass (kg).
- In this, we can see that for launch site VAFB SLC 4E, there were no heavy rockets with payload mass greater than 10000.
- For other launch sites the payload mass has gone up to 16000.



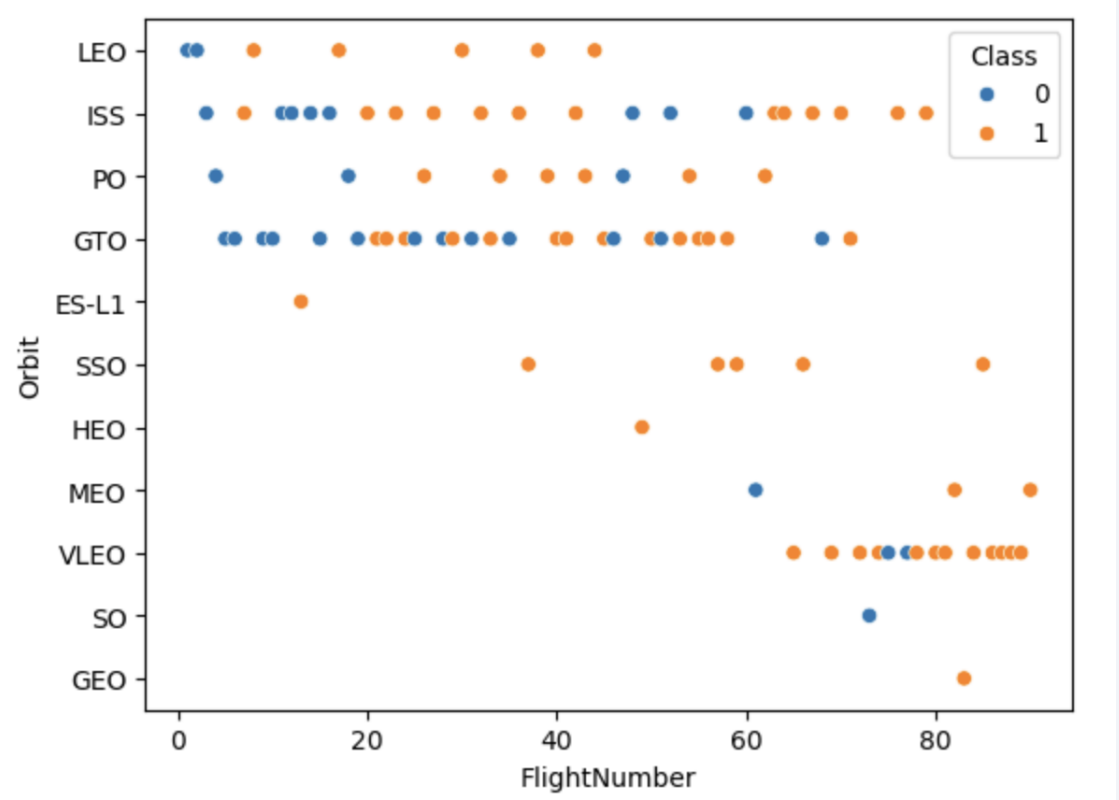
Success Rate vs. Orbit Type

- The plot on the right demonstrates the relation between Landing Outcome and Orbit.
- As we can see, the following orbits have the highest success rates:
  - ES-L1
  - GEO
  - HEO
  - SSO
- GTO has the lowest success rate.



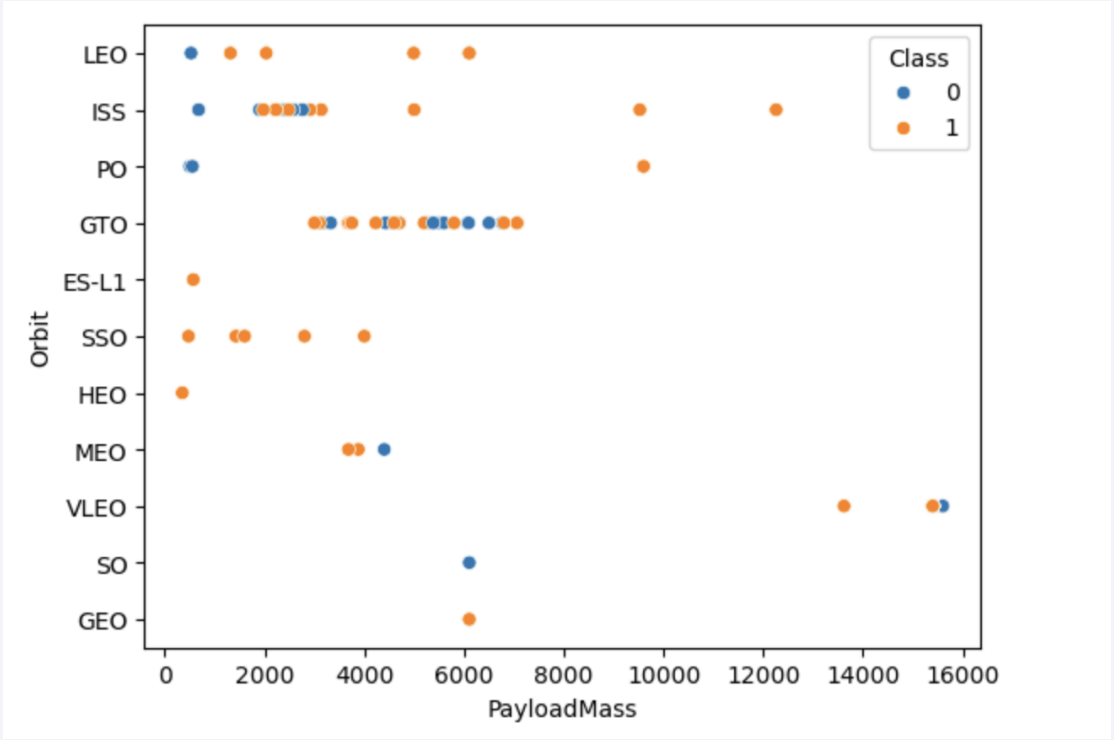
Flight Number vs. Orbit Type

- The plot on the right shows relation between Orbit and FlightNumber.
- Here, in case of LEO orbit, success rate seems to be related to the number of flights.
- On the other hand, in case of GTO there is no relation.



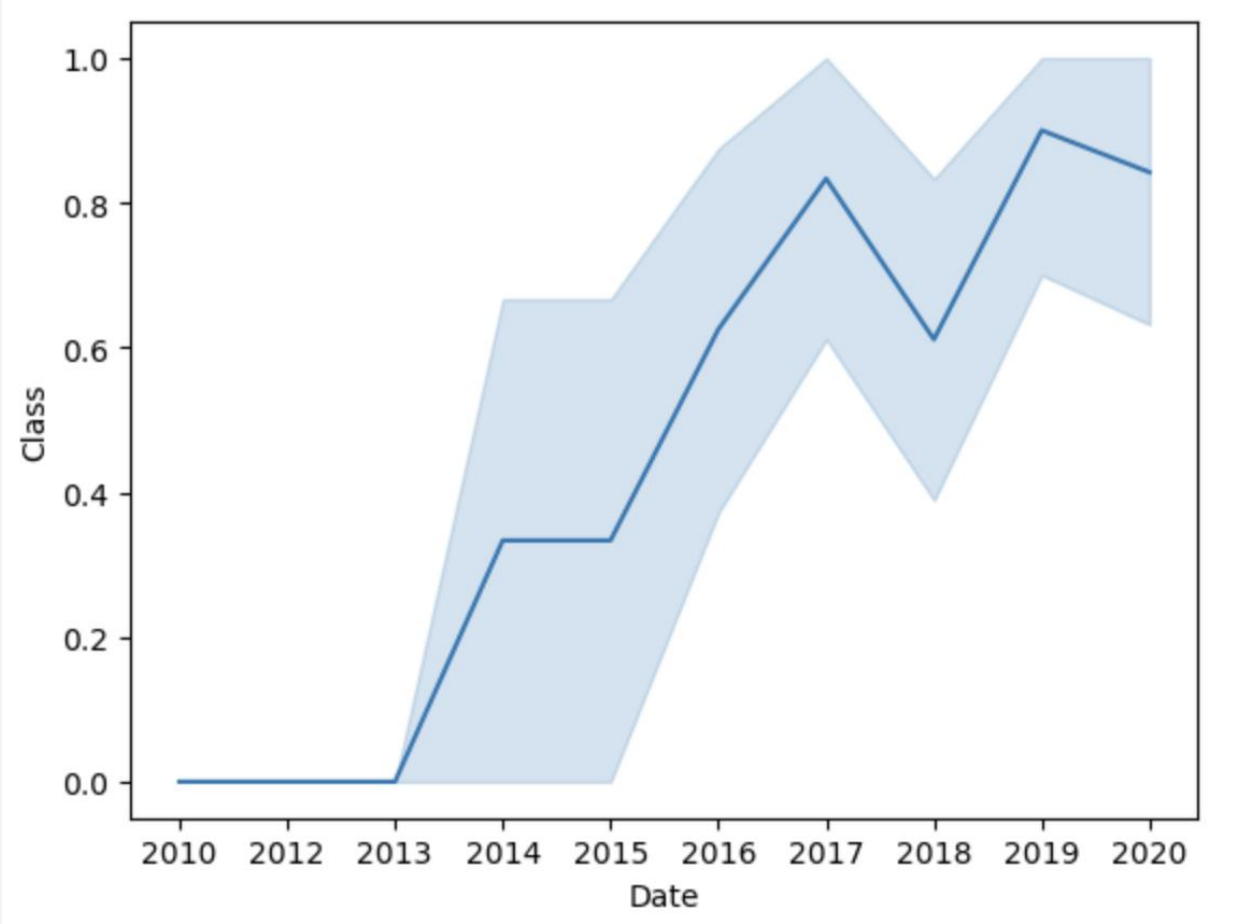
Payload vs. Orbit Type

- The plot shows the relation between Orbit and Payload Mass.
- As we can see, heavier payload mass rockets have more chances of being successful in case of LEO, PO and ISS orbits.
- Orbits SO and GEO have the least number of rockets with only one rocket with payload mass of 6000.



Launch Success Yearly Trend

- Shows relationship between Landing Outcome and Date of launch.
- This shows that the success rate kept increasing since 2013 till 2020.
- There is a sudden rise in success rate after 2013.



All Launch Site Names

In [13]:
```
%%sql
SELECT DISTINCT("Launch_Site") FROM SPACEXTABLE;
```

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

Out[13]:

Launch_Site
CCAFS LC-40
VAFB SLC-4E
KSC LC-39A
CCAFS SLC-40

- This query displays the unique names of launch sites from the SPACEXTABLE.
- The result shows total 4 unique launch sites.

Launch Site Names Begin with 'CCA'

In [16]:
```
%%sql
SELECT * FROM SPACEXTABLE
    WHERE "Launch_Site" LIKE 'CCA%' LIMIT 5;
```

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

Out[16]:

Date	Time (UTC)	Booster_Version	Launch_Site	Payload	PAYLOAD_MASS__KG_	Orbit	Customer	Mission_Outcome	Landing_
2010-06-04	18:45:00	F9 v1.0 B0003	CCAFS LC-40	Dragon Spacecraft Qualification Unit	0	LEO	SpaceX	Success	Failure (p
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 (p
2012-05-22	7:44:00	F9 v1.0 B0005	CCAFS LC-40	Dragon demo flight C2	525	LEO (ISS)	NASA (COTS)	Success	N
2012-10-08	0:35:00	F9 v1.0 B0006	CCAFS LC-40	SpaceX CRS-1	500	LEO (ISS)	NASA (CRS)	Success	N
2013-03-01	15:10:00	F9 v1.0 B0007	CCAFS LC-40	SpaceX CRS-2	677	LEO (ISS)	NASA (CRS)	Success	N

- This query displays the top 5 rows from the SPACEXTABLE where launch site starts with ‘CCA’.
- The result shows two launches in 2010, two in 2012 and one in 2013.

Total Payload Mass

```
In [18]: %%sql
         SELECT SUM("PAYLOAD_MASS__KG_") AS TOTAL_PAYLOAD_MASS FROM SPACEXTABLE
             WHERE Customer = "NASA (CRS)";
 * sqlite:///my_data1.db
Done.
Out[18]: TOTAL_PAYLOAD_MASS
         ------------------
                    45596
```

- This query sums all the payload masses from the SPACEXTABLE, where Customer is 'NASA (CRS)'.
- The total payload mass is 45596 kg.

Average Payload Mass by F9 v1.1

In [19]:
```
%%sql
SELECT AVG("PAYLOAD_MASS__KG_") AS AVG_PAYLOAD_MASS FROM SPACEXTABLE
    WHERE "Booster_Version" = "F9 v1.1";
```

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

Out[19]:

AVG_PAYLOAD_MASS
2928.4

- Here, we try to calculate mean of payload mass with an alias as AVG_PAYLOAD_MASS from SPACEXTABLE for booster version F9 v1.1.
- The average comes around 2928.4 kg.

First Successful Ground Landing Date

```
In [21]:
%%sql
SELECT Date FROM SPACEXTABLE
    WHERE "Landing_Outcome" = "Success (ground pad)" LIMIT 1;
```

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

Out[21]:

Date
2015-12-22

- Displays the top 1 row where landing outcome is success (ground pad).
- The first successful ground landing date was on 22nd Dec., 2015.

Successful Drone Ship Landing with Payload between 4000 and 6000

In [22]:
```
%%sql
SELECT "Booster_Version" FROM SPACEXTABLE
    WHERE "Landing_Outcome" = "Success (drone ship)"
    AND "PAYLOAD_MASS__KG_" BETWEEN 4000 AND 6000;
```

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

Out[22]:

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

- The booster versions: F9 FT B1022, F9 FT B1026, F9 FT B1021.2 and F9 FT B1031.2 have successful drone ship landing with payload between 4000 and 6000.

Total Number of Successful and Failure Mission Outcomes

In [31]:
```
%%sql
SELECT "Mission_Outcome", COUNT("Mission_Outcome") AS Freq FROM SPACEXTABLE
    GROUP BY "Mission_Outcome";
```

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

Out[31]:

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

- First, we will group the table on the basis of mission outcome followed by displaying mission outcome and their corresponding counts.
- We can say that no. successes = 100 and failure = 1.

Boosters Carried Maximum Payload

In [30]:
```
%%sql
SELECT "Booster_Version" FROM SPACEXTABLE
    WHERE "PAYLOAD_MASS__KG_" = (
        SELECT MAX("PAYLOAD_MASS__KG_") FROM SPACEXTABLE
    );
```

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

Out[30]:

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

- There are several boosters which have carried maximum payload mass in their space missions.

2015 Launch Records

In [34]:
```
%%sql
SELECT substr(Date, 6, 2) AS Month, "Landing_Outcome", "Booster_Version", "Launch_Site" FROM SPACEXTABLE
    WHERE substr(Date, 0, 5) = "2015" AND "Landing_Outcome" = "Failure (drone ship)";
```

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

Out[34]:

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

- Here it is evident that in the year 2015, there were two failed drone ship missions happened in the months of January and April.
- These two failures happened at the same launch site.

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

In [40]:
```
%%sql
SELECT "Landing_Outcome", COUNT(*) AS Freq FROM SPACEXTABLE
    GROUP BY "Landing_Outcome" HAVING "Landing_Outcome" IN ("Failure (drone ship)", "Success (ground pad)")
    AND Date BETWEEN "2010-06-04" AND "2017-03-20"
    ORDER BY Freq DESC;
```

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

Out[40]:

Landing_Outcome	Freq
Success (ground pad)	9
Failure (drone ship)	5

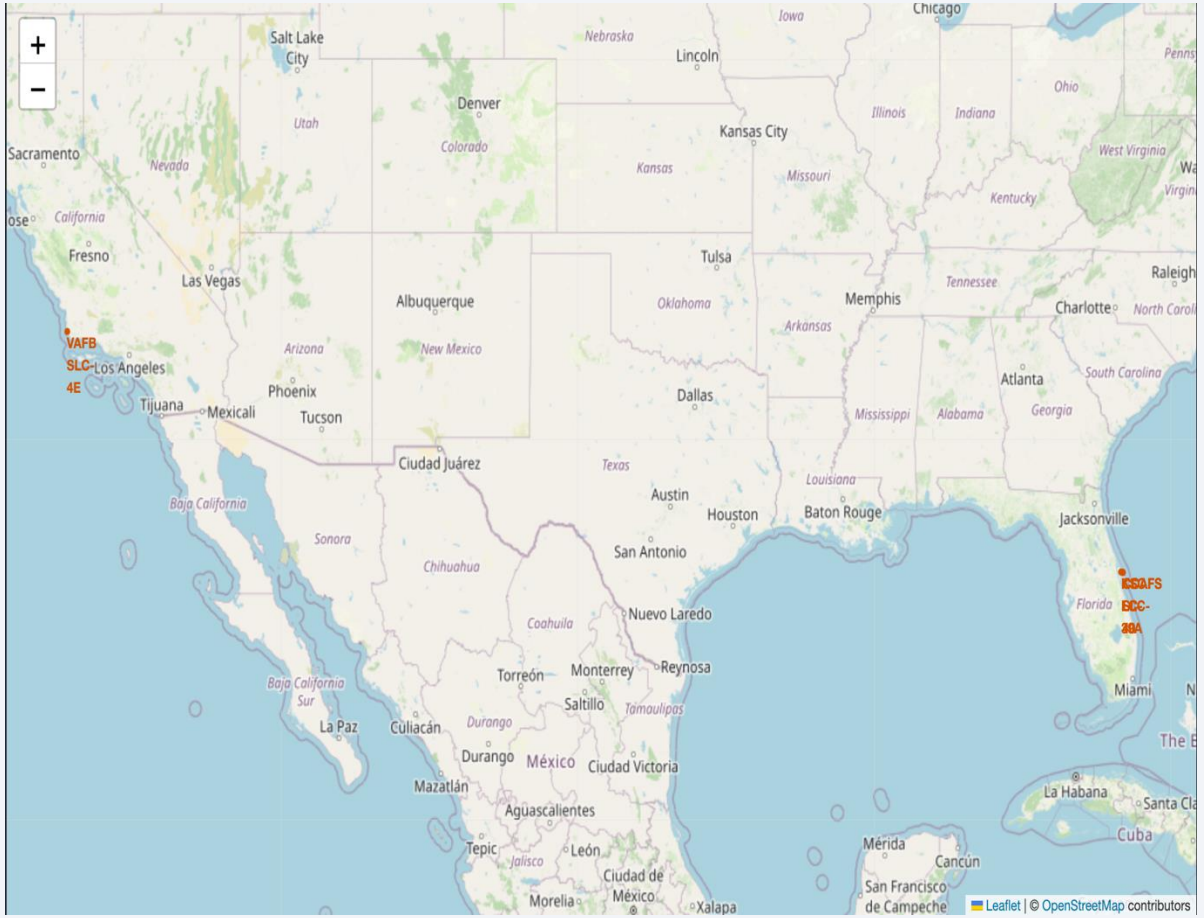
- This shows that the success landing on ground pad has the highest frequency followed by failure of drone ships.

Section 3

Launch Sites Proximities Analysis

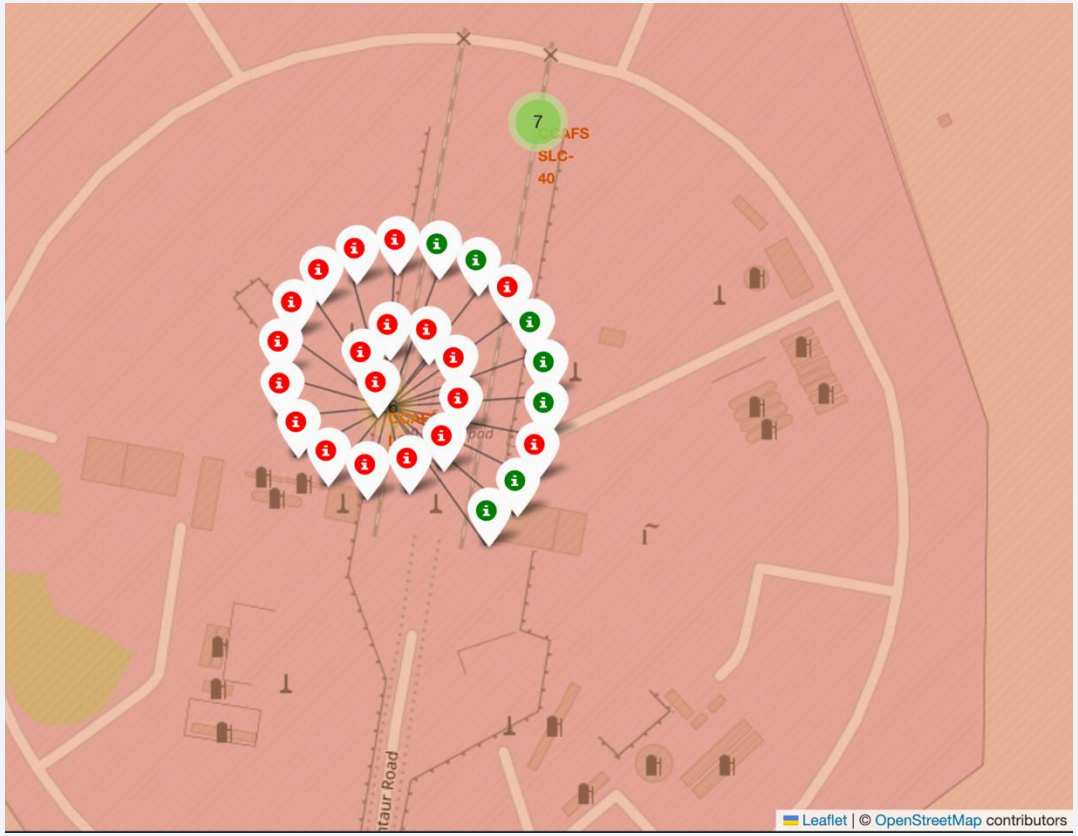
Launch Sites Location on Global Map

- The screenshot shows the launch sites as marker objects where the mission took place.
- First a folium map was created, followed by adding markers as a child object on the Folium Map object.
- Marker color is red with a circle of radius 1000 units.



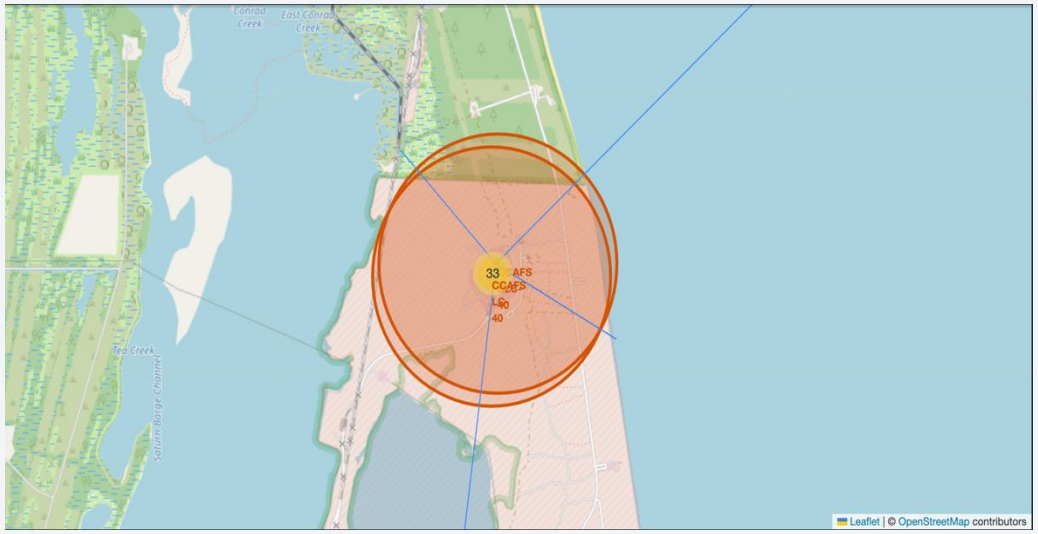
Success Rate Launch Outcome Markers

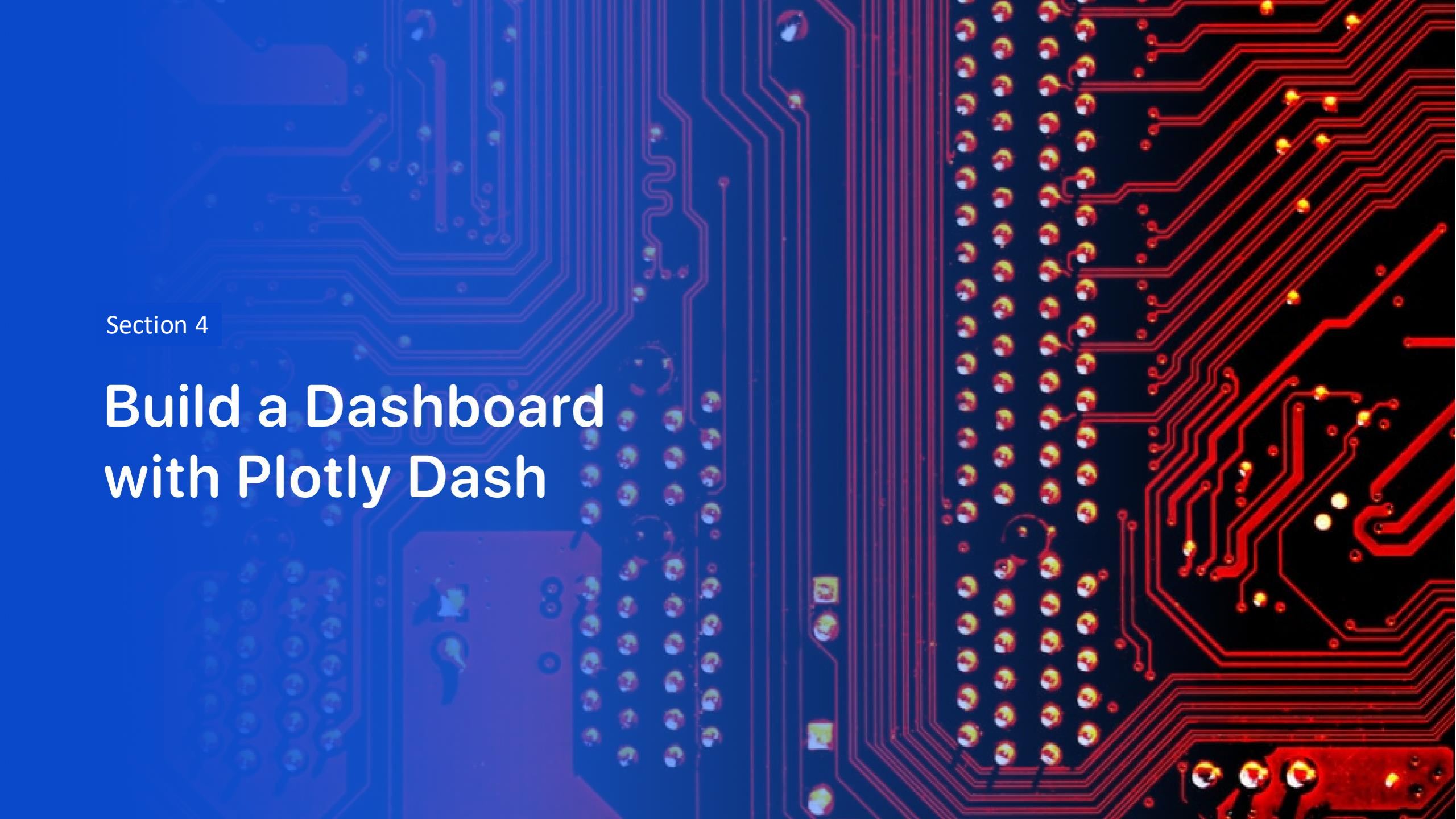
- This screenshot demonstrates the successful and failed launch outcomes of past missions shown as markers with color code (green for success, red for failure) at a specific launch site.
- Based on the number of success and failures we can determine which sites are prone to failure for landing during first stage phase.



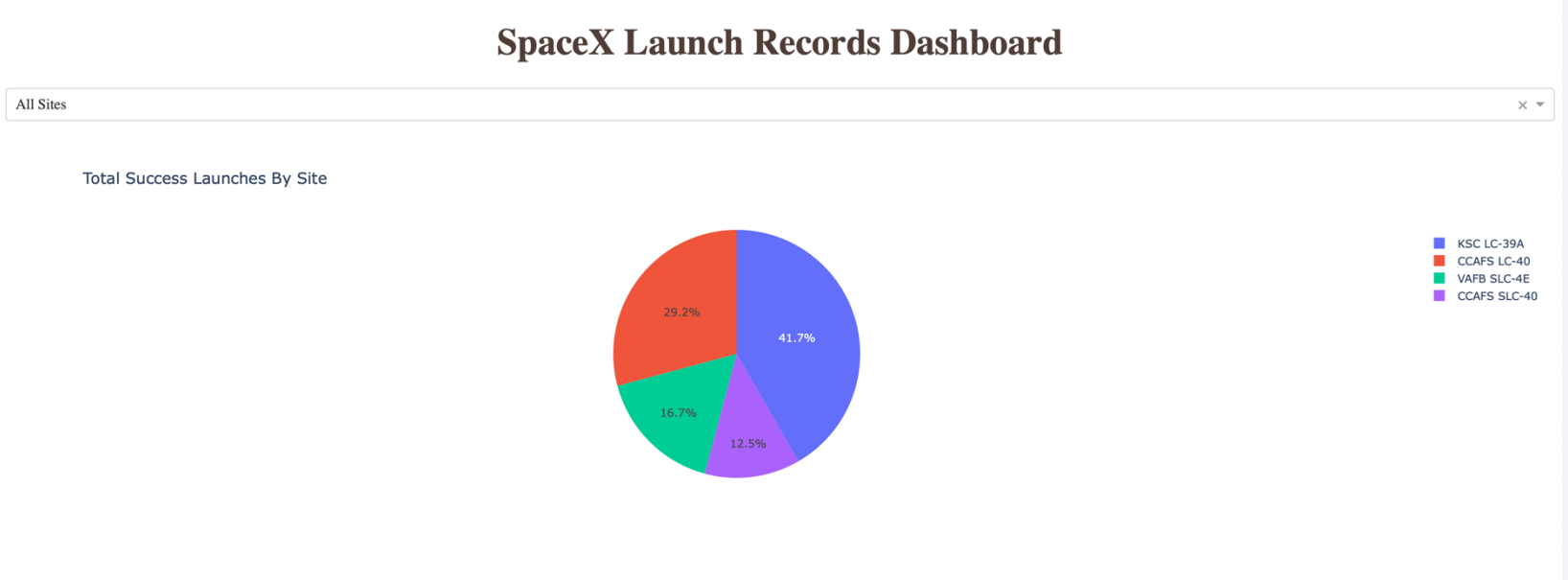
Proximity of nearest civil areas from launch site

- This screenshot shows the proximity of nearest highway, railway, city and coastline from the launch site.
- It is evident from the diagram that the launch site is in proximity with railway, highway and coastline areas with city being the farthest.



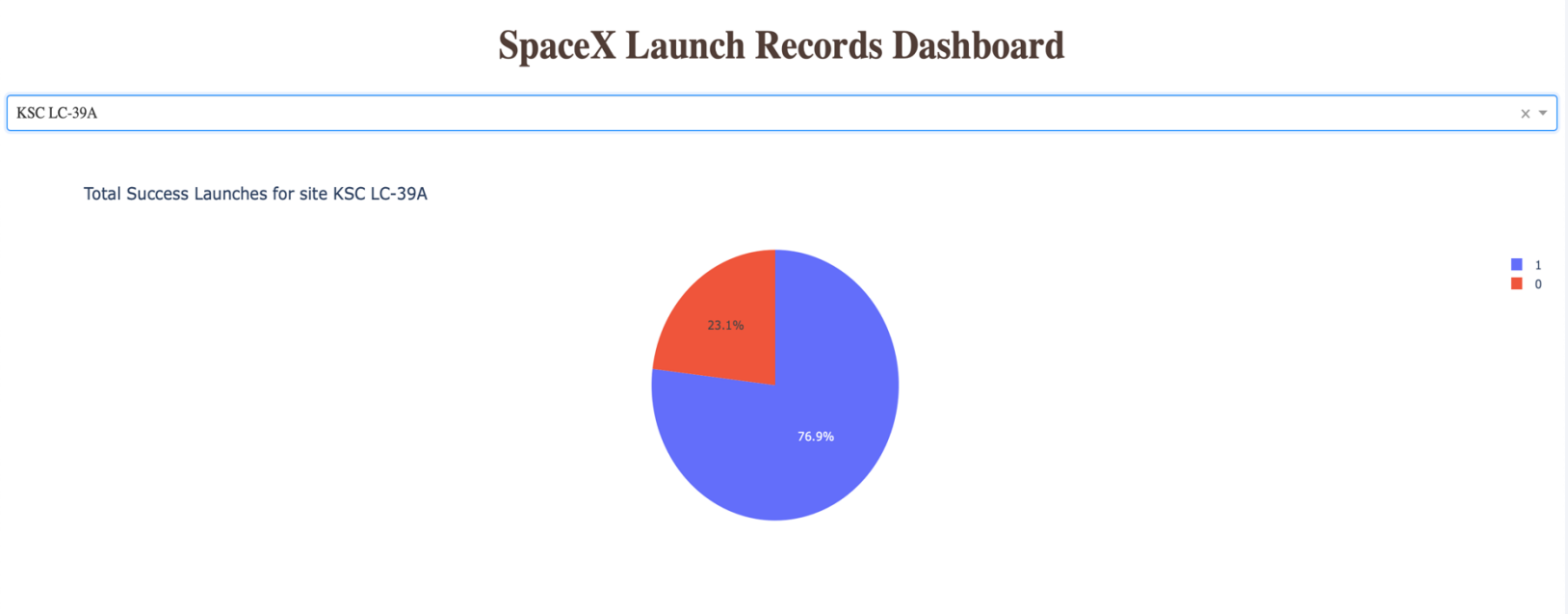

Section 4
Build a Dashboard with Plotly Dash

Total Success Launches By Site



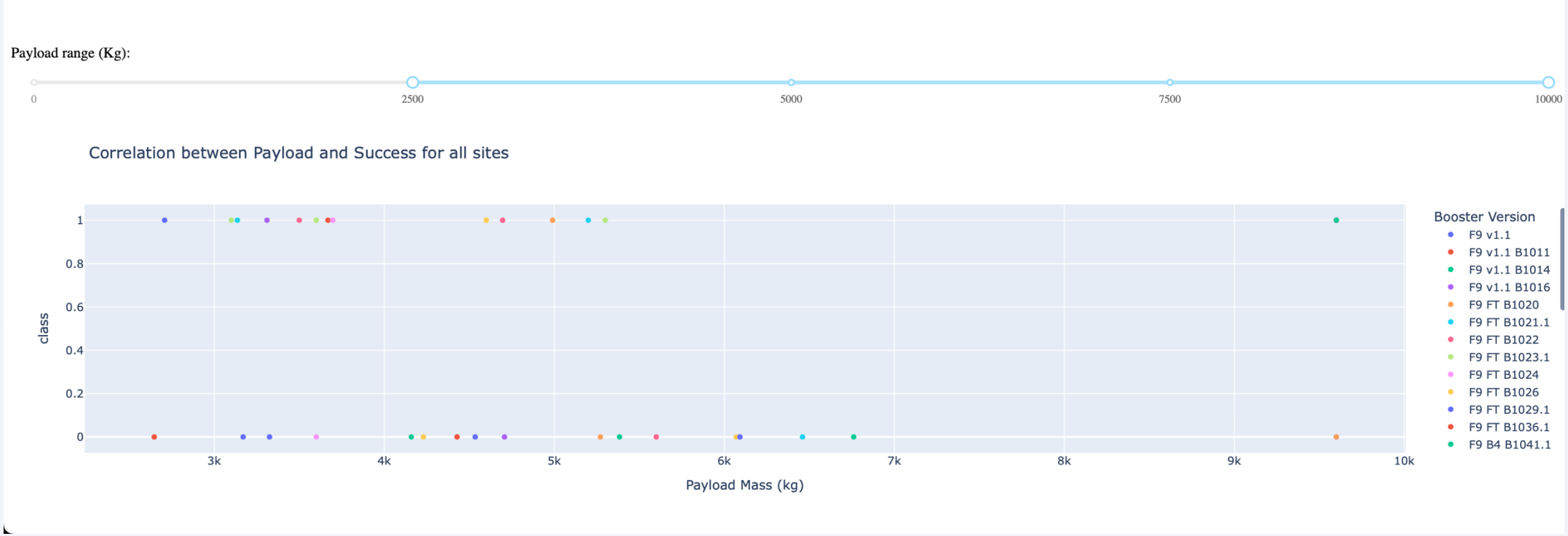
- The pie chart shows the proportion of success rates for all sites.
- The launch site KSC LC-39A has the highest success rate and CCAFS SLC-40 has the lowest.

Success Rate of Launch Site with Highest Success Ratio



- The launch site KSC LC-39A has the highest success ratio with success rate around 77%.
- Failure Rate is 23%.

Relation between Payload and Success Rate for all Sites



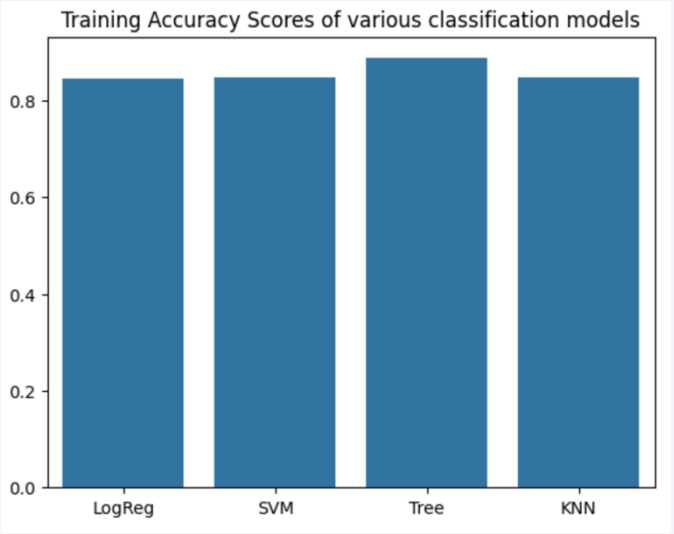
- The graph shows that as the payload mass gets heavier, there is lesser chance of success.

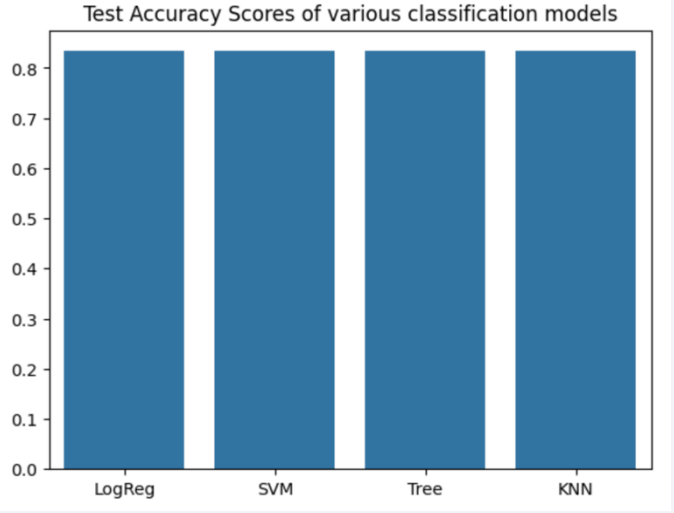
Section 5

Predictive Analysis (Classification)

Classification Accuracy

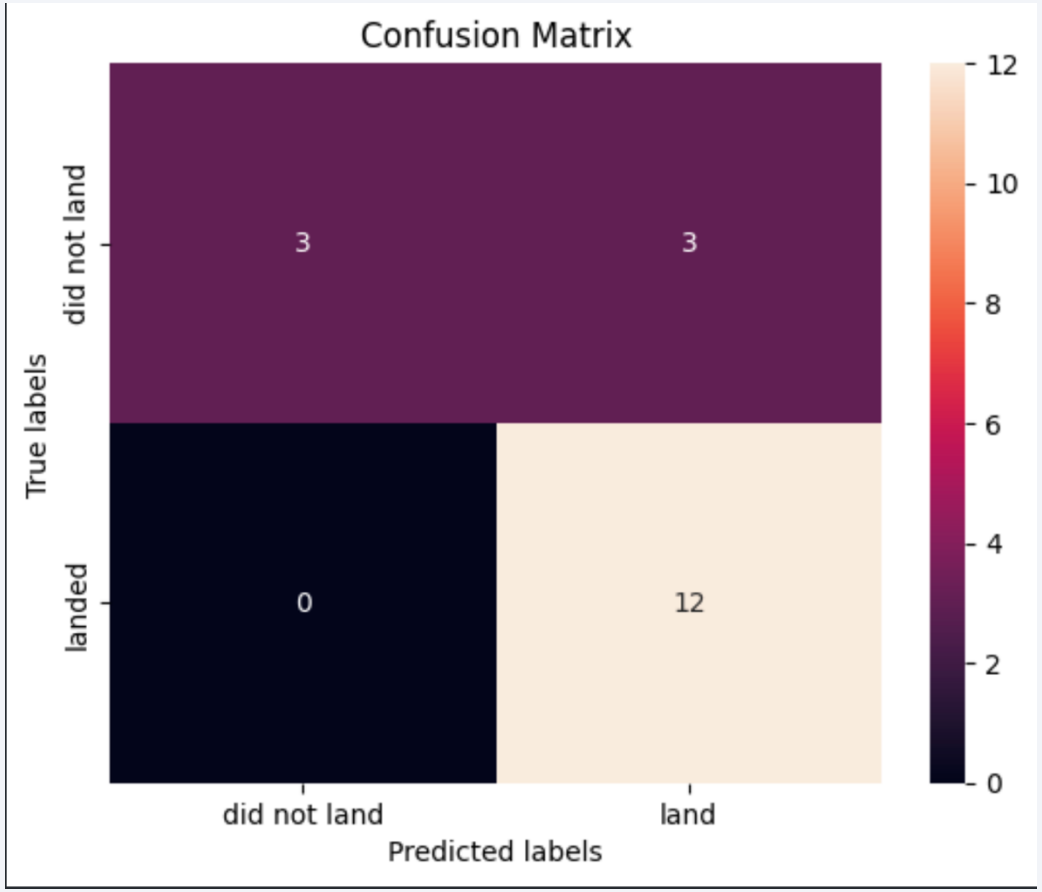
- Based on the bar graph shown on the right, all classification models are getting same training accuracy except Decision Tree Classifier with an accuracy of around 88%.
- Based on the test accuracies, it is evident that except Decision Tree Classifier any other model can be picked up as decision tree is overfitting as there is quite a noticeable difference between train and test accuracies.
- For simplicity we can pick Logistic Regression model as the passed data has been transformed into numeric data with standardization.





Confusion Matrix

- This metric is for logistic regression model.
- The matrix on the right is the confusion matrix with:
  - False Negatives = 0
  - False Positives = 3
- This shows that the model is performing quite well on the training data.
- The training accuracy is around 85% with test accuracy of 83%.



Conclusions

- By doing this analysis we can conclude that based on features like Flight Number, Orbit, Payload Mass, Launch Sites and other such variables we can determine whether the first stage will be successful or not. This is crucial in determining the cost of the project.
- Since the classification models are generalizing quite well on the prior data. It means that they can predict the outcome of the mission with high precision and good recall.
- Though in past the chances of success were low, as the time progressed there was a steady growth in success rates as the techniques were improved based on previous failures.
- The KSC LC-39A launch site has the highest success rate among other sites which give a good advantage to SpaceX in terms of cost management as they can reuse their first stage rockets.

Appendix

Best model training and metrics

Create a logistic regression object then create a GridSearchCV object `logreg_cv` with cv = 10. Fit the object to find the best parameters from the dictionary `parameters`.

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

In [22]:
```
parameters ={"C":[0.01,0.1,1],'penalty':['l2'], 'solver':['lbfgs']}# l1 lasso l2 ridge
lr = LogisticRegression()
grid_search_logreg = GridSearchCV(estimator=lr, param_grid=parameters, cv=10)
logreg_cv = grid_search_logreg.fit(X_train, Y_train)

print(logreg_cv)
```

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

We output the `GridSearchCV` object for logistic regression. We display the best parameters using the data attribute `best_params_` and the accuracy on the validation data using the data attribute `best_score_`.

In [23]:
```
print("tuned hpyerparameters :(best parameters) ",logreg_cv.best_params_)
print("accuracy :",logreg_cv.best_score_)
```

```
tuned hpyerparameters :(best parameters)  {'C': 0.01, 'penalty': 'l2', 'solver': 'lbfgs'}
accuracy : 0.8464285714285713
```

TASK 5

File display

Calculate the accuracy on the test data using the method `score`:

In [24]:
```
logreg_cv.score(X_test, Y_test)
```

Out[24]: 0.8333333333333334

Transformed data for model

In [33]:
```
# HINT: Use get_dummies() function on the categorical columns
features_one_hot = pd.get_dummies(data=features, columns=['Orbit', 'LaunchSite', 'LandingPad', 'Serial'], drop_fi
features_one_hot
```

Out[33]:

	FlightNumber	PayloadMass	Flights	GridFins	Reused	Legs	Block	ReusedCount	Orbit_ES-L1	Orbit_GEO	...	Serial_B1048
0	1	6104.959412	1	False	False	False	1.0	0	0	0	...	0
1	2	525.000000	1	False	False	False	1.0	0	0	0	...	0
2	3	677.000000	1	False	False	False	1.0	0	0	0	...	0
3	4	500.000000	1	False	False	False	1.0	0	0	0	...	0
4	5	3170.000000	1	False	False	False	1.0	0	0	0	...	0
...	...	...	...	...	...	...	...	...	...	...	...	...
85	86	15400.000000	2	True	True	True	5.0	2	0	0	...	0
86	87	15400.000000	3	True	True	True	5.0	2	0	0	...	0
87	88	15400.000000	6	True	True	True	5.0	5	0	0	...	0
88	89	15400.000000	3	True	True	True	5.0	2	0	0	...	0
89	90	3681.000000	1	True	False	True	5.0	0	0	0	...	0

90 rows × 80 columns

Thank you!