IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

Name: Macheing Nyany
Date: 30-09-2021

Outline

- Executive Summary

- Introduction

- Methodology

- Results

- Conclusion

- Appendix

Executive Summary

The goal of this data science project is all about determining the reusability of SpaceX's Falcon 9 first stage with certainty so as the cost of space exploration could be minimized. Many EDA, data visualization, predictive methodologies including pandas, numpy, SQL, matplotlib, seaborn, folium, plotly dash, logistic regression, SVM, decision tree, and KNN were employed to determine to reusability falcon-9's first stage.

The overall reusability of falcon-9's first stage is 66.67%. KSC LC-39A is the only launch site with higher success ratio of 76.9%, and the booster version with higher success ratio is FT. The reusability success rate increased gradually between year 2010 and 2020, and it is likely to continue to increase in the nearer future.

Introduction

This project aimed to analyze SpaceX's Falcon 9 rocket launches.

The goal is to predict if Falcon-9's first stage would land successfully after being launched into space.

By determining the successful landing of Falcon-9's stage, we could determine the launching cost of the rocket.

By reusing the first stage of Falcon 9, the launching cost per rocket would surely reduce from 165 million to approximately 63 millions.

This finding made SpaceX's space program relatively cheaper compared to other providers.
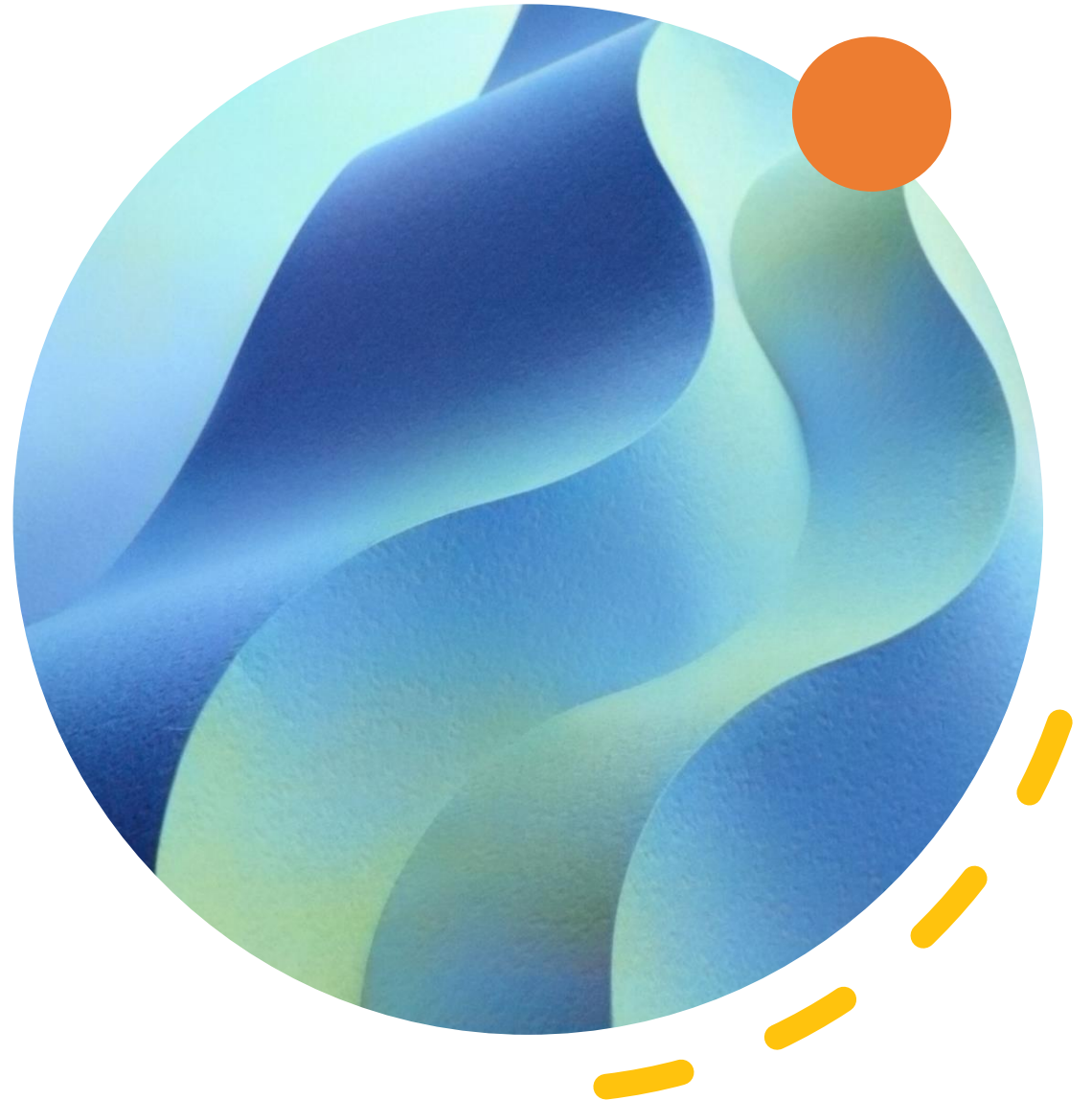
Section 1

Methodology

Methodology

Executive Summary

Data collection methodology:

- Request SpaceX's Restful API to obtain launch data, and applied data cleaning techniques.

Perform data wrangling:

- Finding data patterns using Pandas and Numpy python libraries.

Perform exploratory data analysis (EDA) using visualization and SQL

Perform interactive visual analytics using Folium and Plotly Dash

Perform predictive analysis using logistic regression, SVM, decision tree, KNN classification models.

- Organize data into classes; class 1 for successful landing, and class 0 for unsuccessful landing.
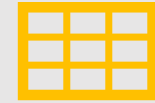- Standardize and fit the data, split the data into training set and testing set.

Data Collection

Request SpaceX's Restful API to obtain rocket launches data.

Apply data cleaning techniques to clean and format the requested data.

Normalizes the data and turn them into data frame.

Filter out data of Falcon 9 from the rest of rocket launches.

Apply basic data wrangling techniques to grasp the nature of data.

Save the normalized data into a file for the later uses.

Data Collection – SpaceX API

- Present API Call:

- spacex_url="https://api.spacexdata.com/v4/launches/past"

- response = requests.get(spacex_url)

- data = pd.json_normalize(response.json())

- GitHub URL link:
 https://github.com/Macheing/Spacex_ds_project/blob/main/Spacex_data_collection.ipynb
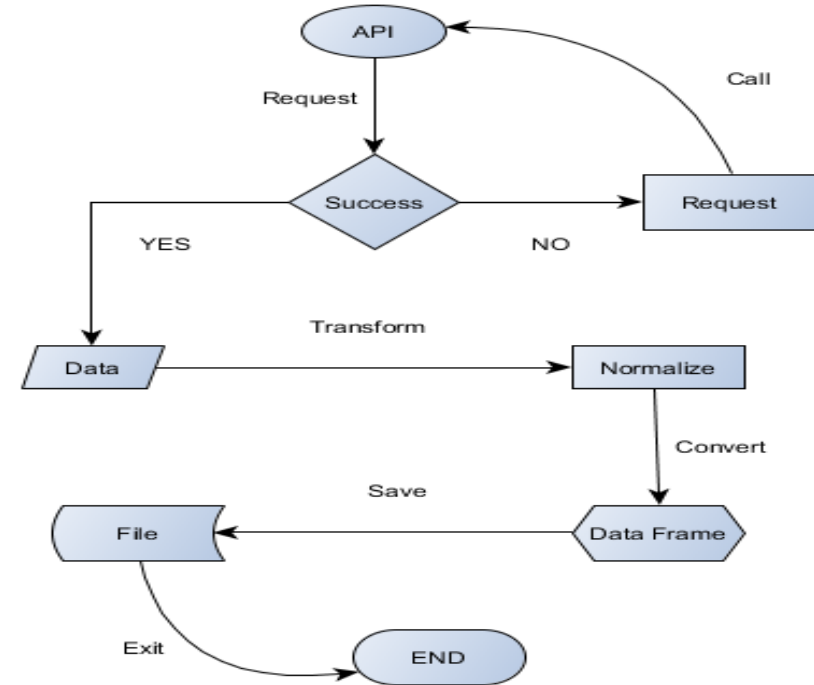
Data Collection - Scraping

- ➤ Extract Booster version data from rockets API URL.

- ➤ Extract Launch site data from launchpads API URL.

- ➤ Extract Payload data from payloads API URL.

- ➤ Extract Core data from cores API URL.

- ➤ Extract Launch data from launches past API URL.

- ➤ Map all the extracted data into dictionary of dataframe.

- ➤ Display dataframe and save the data into file.

GitHub URL link:
https://github.com/Macheing/Spacex_ds_project/blob/main/Spacex_data_collection.ipynb
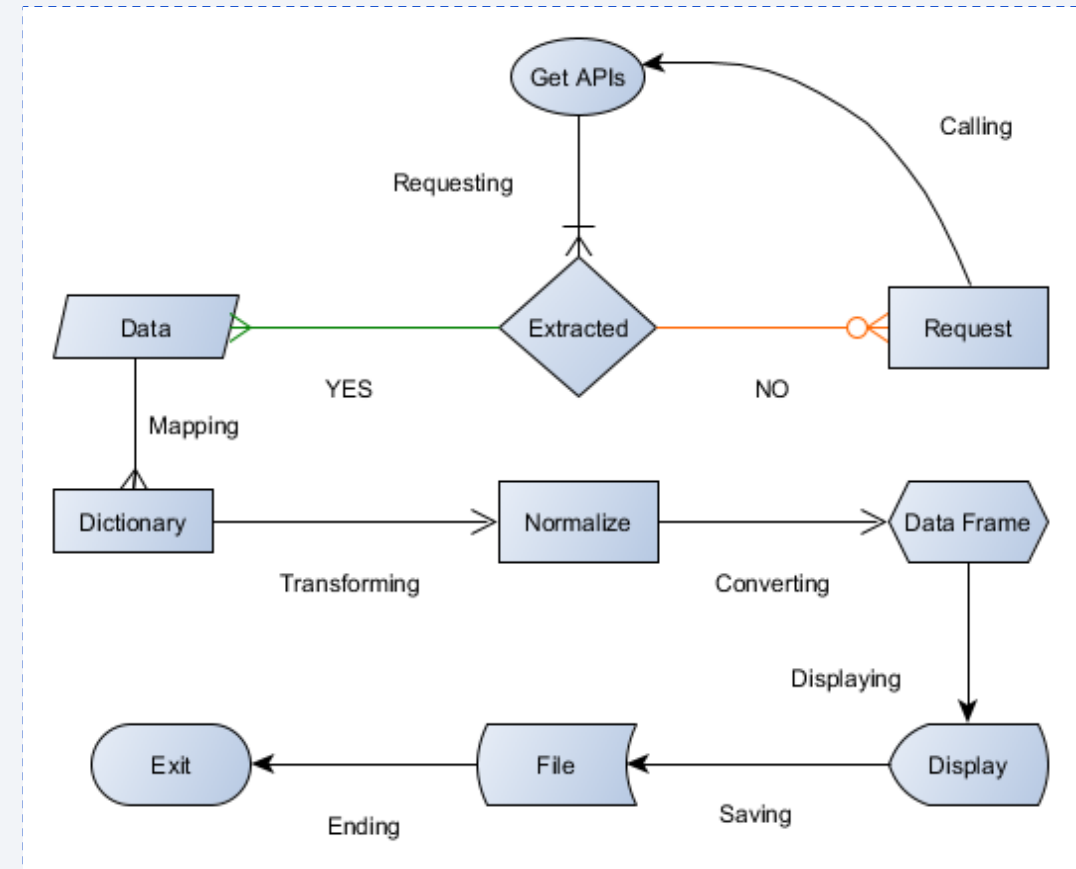
Data Wrangling

- ➢ From data, count the number of launches per launch site.

- ➢ From data, count the number of launches into each orbit.

- ➢ From data, count the number of landing outcome from each launch.

- ➢ Filter landing outcomes into successful or failed landing.

- ➢ Create a class of landing outcomes by mapping 1 for successful landing, and 0 for failed landing.

- ➢ Create a new column of dataframe named class and save the file.

- • GitHub URL link: https://github.com/Macheing/Spacex_ds_project/blob/main/Spacex_data_wrangling.ipynb
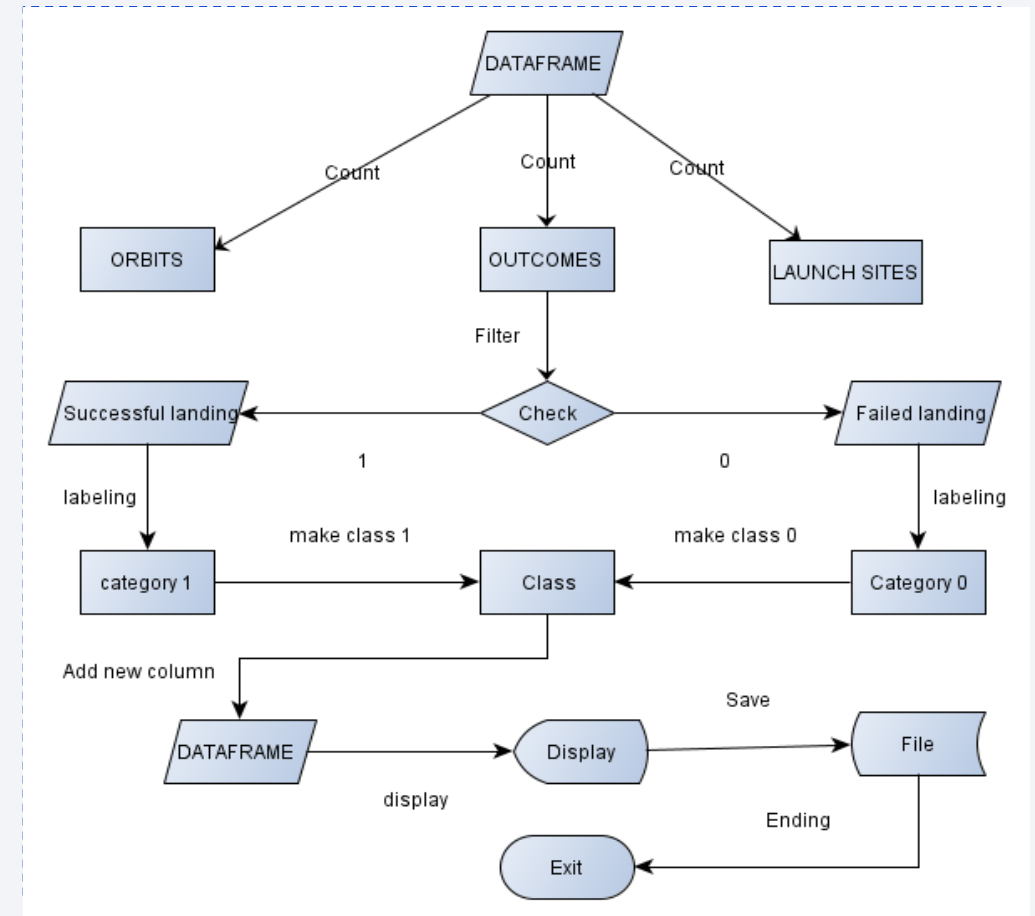
EDA with Data Visualization

- Plot Flightnumber vs Payloadmass using scatter.

- Plot Flightnumber vs Launchsite using scatter.

- Plot Payloadmass vs Launchsite using scatter.

- Plot Orbit mean vs Class using bar chart.

- Plot Flightnumber vs Orbit, and Payloadmass vs Orbit using scatter plot.

- Plot Year mean vs Class using line plot.

- GitHub URL link: https://github.com/Macheing/Spacex_ds_project/blob/main/SpaceX_data_visualization.ipynb
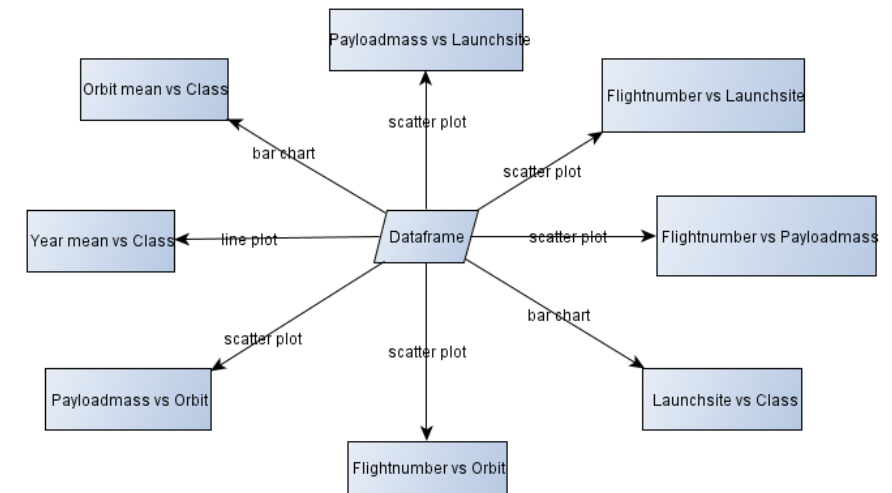
EDA with SQL

- Load csv data into db2 database.

- Connect database with notebook and execute SQL magic queries.

- Select unique launch sites, total and average payload per launch.

- Select successful ground landing outcome.

- Select payload between 4000 and 6000 in drone ship landing.

- Select booster version carrying maximum payload into space.

- Count total landing outcomes between year 2010 to 2017.

- GitHub URL link:
https://github.com/Macheing/Spacex_ds_project/blob/main/Spacex_data_analysis.ipynb
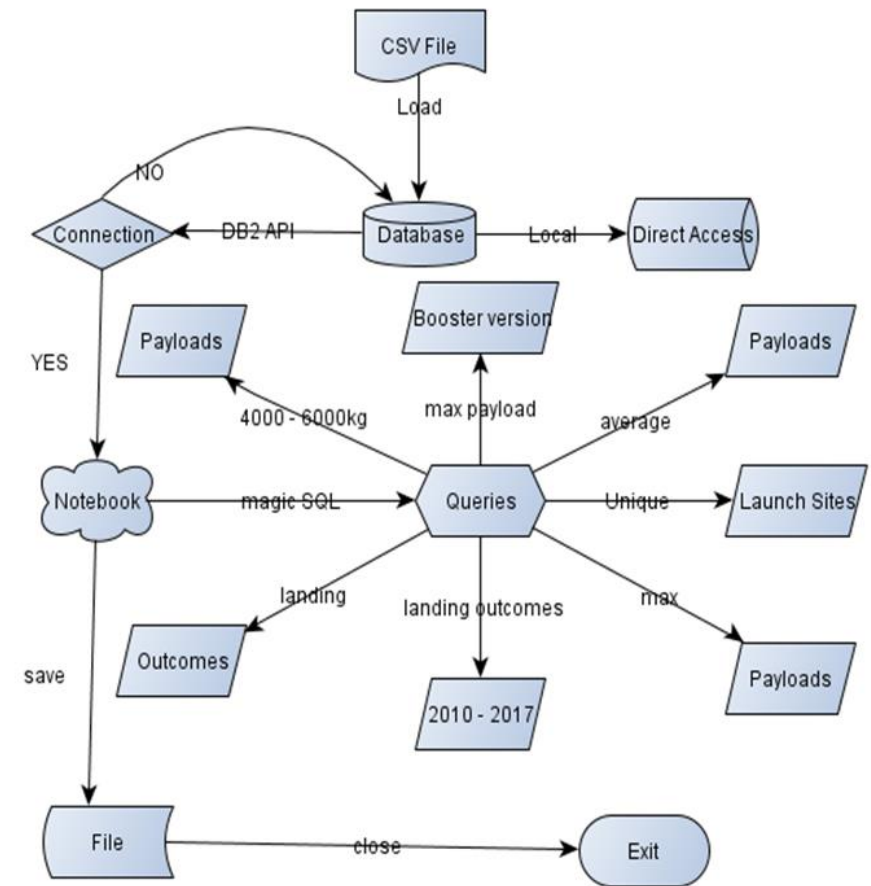
Build an Interactive Map with Folium

- Marks all launch sites on global map.

- Display success vs failure launches per each launch site on map.

- Calculate distances between launch site to its closer proximity.

- Add mouse position to display the coordinates.

- GitHub URL link: https://github.com/Macheing/Spacex_ds_project/blob/main/Spacex_interactive_visualization.ipynb
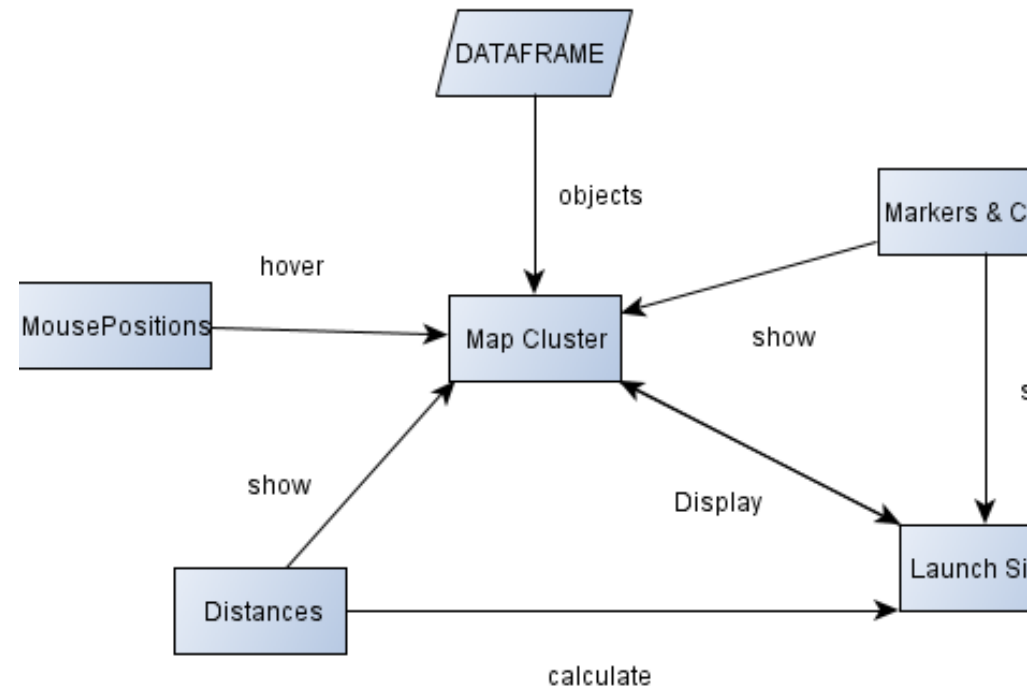
Build a Dashboard with Plotly Dash

➢ Add title of dashboard, dropdown containing list of launch sites.

➢ Add callback function for dropdown to display pie chart if all sites or specific launch site is selected.

➢ Calculate percentage of successful launches of all sites or each site based on dropdown input value.

➢ Add range-slider callbacks function to display scatter plot output if all sites, specific site is selected, or when specific range-slider is pick.

➢ Calculate scatter output based on the range-slider and dropdown inputs of selected site.

• GitHub URL link for basic dashboard app:
https://github.com/Macheing/Spacex_ds_project/blob/main/spacex_dash_app1.py

• GitHub URL link for modified dashboard app:

https://github.com/Macheing/Spacex_ds_project/blob/main/spacex_dash_app2.py
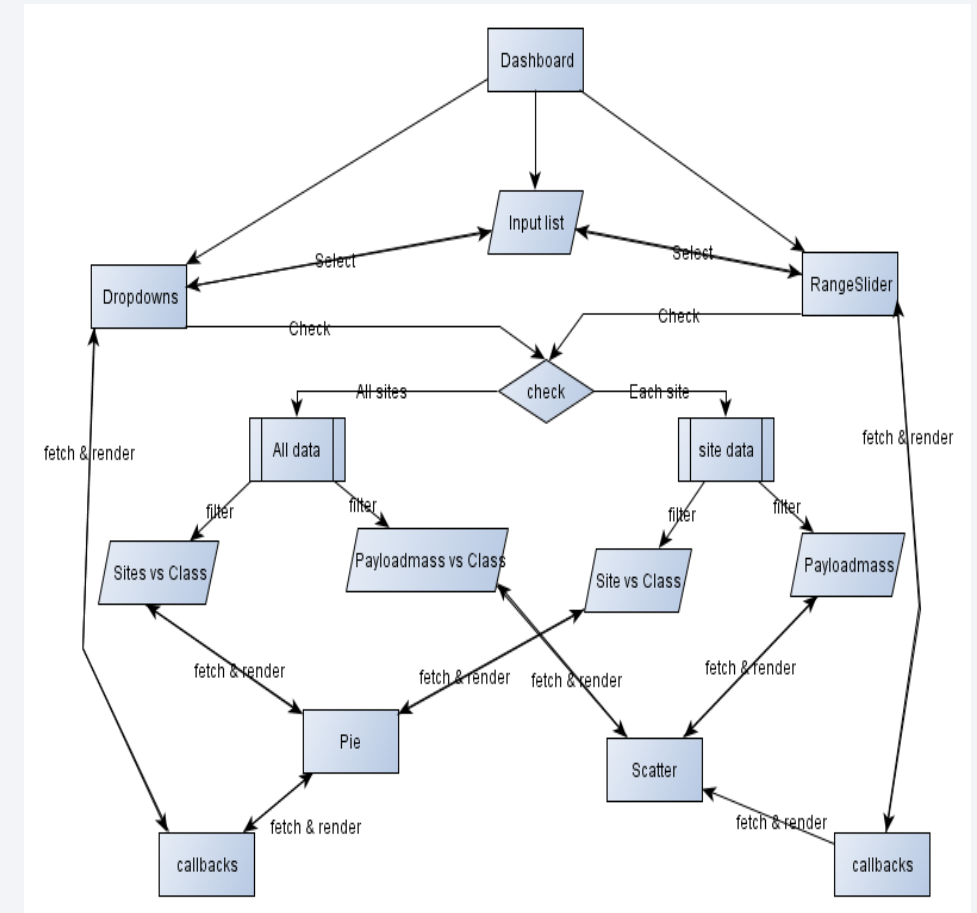
Predictive Analysis (Classification)

➢ Split data into X and Y data sets, fit and transform X data set.

➢ Split X data into x-train and x-test, and split Y data into y-train and y-test.

➢ Build logistic regression, pass it to GridSearchCV, train model using training datasets, and test mode using testing datasets.

➢ Build SVM, pass it to GridSearchCV, train model using training datasets, and test mode using testing datasets.

➢ Build decision tree, pass it to GridSearchCV, train model using training datasets, and test mode using testing datasets.

➢ Build KNN, pass it to GridSearchCV, train model using training datasets, and test mode using testing datasets.

• GitHub URL link:
https://github.com/Macheing/Spacex_ds_project/blob/main/Spacex_ML_Prediction.ipynb
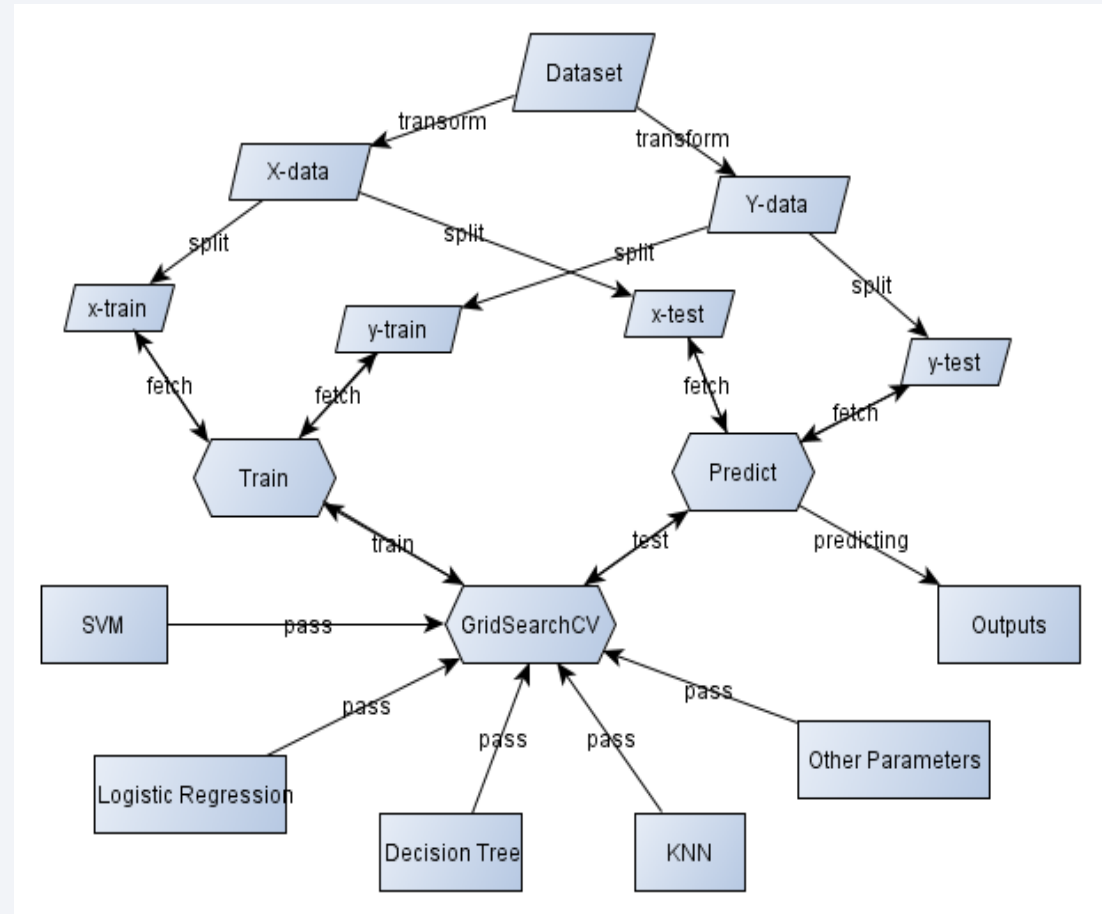
Results

➢ The average of reusability success rate is 66.67%.

➢ Site KSC LC 39A has launches success rate of 77% followed by CCAFS SLC-40 with 43% success rate.

➢ Site VAFB SLC-4E has success rate of 40% followed by CCAFS LC-40 with success rate of 27%.

➢ Much of reusability success rate occurred to rockets having payload mass between 1950 -5200 kg.

➢ Much of reusability success rate occurred to rockets having booster version of FT.

➢ Much of reusability success rate occurred in ES-LI, GEO, HEO, and SEO orbits.

➢ Much of reusability success rate increased between 2013 to 2020.

➢ All launch sites are located closer to Equator, they have closest proximities to coastlines, and they're quite from cities

➢ Logistic regression has a reusability prediction best-score of 84.64% and accuracy of 83.33%.

➢ Support Vector machine has a reusability prediction best-score of 84.82% and accuracy of 83.33%..

➢ Decision tree has a reusability prediction best-score of best-score of 87.68% and accuracy of 66.67%.

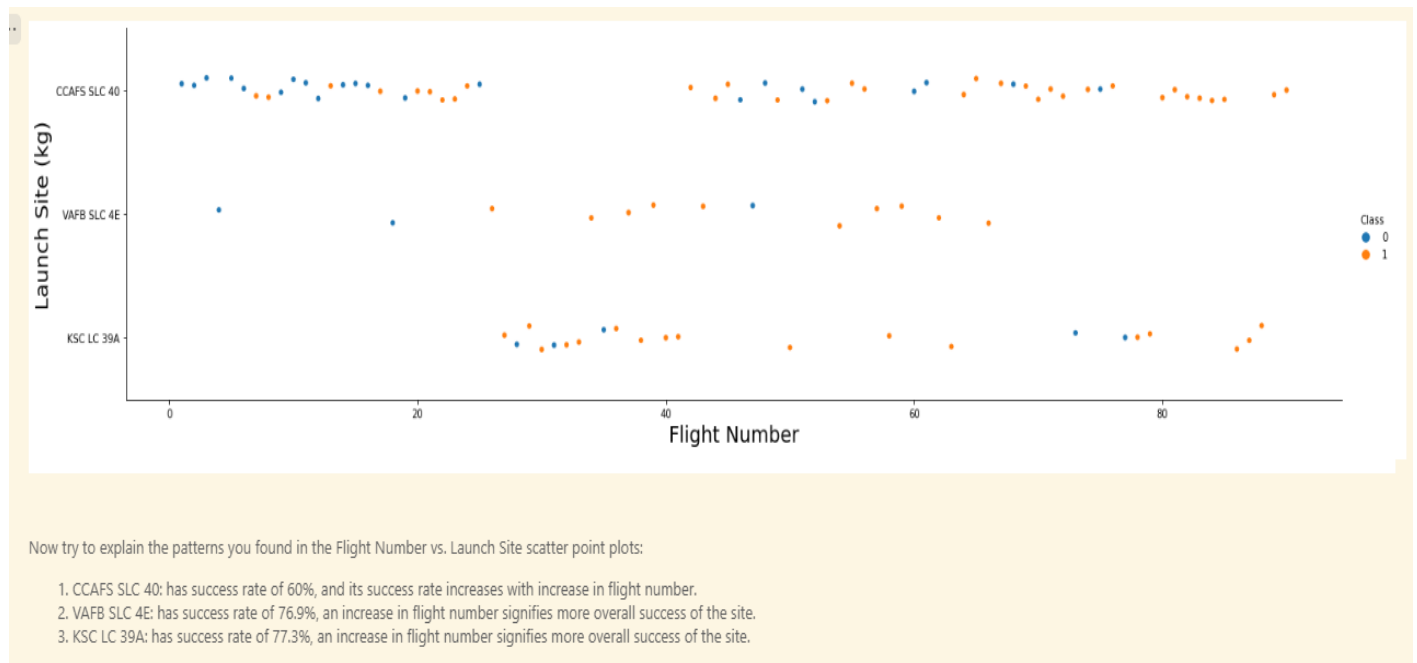➢ K nearest neighbors has a reusability prediction best-score of 84.82% and accuracy of 83.33%.

Section 2

Insights drawn from EDA

Flight Number vs. Launch Site

- Show a scatter plot of Flight Number vs. Launch Site:

sns.catplot(y="LaunchSite", x="FlightNumber", hue="Class", data=df, aspect = 4)

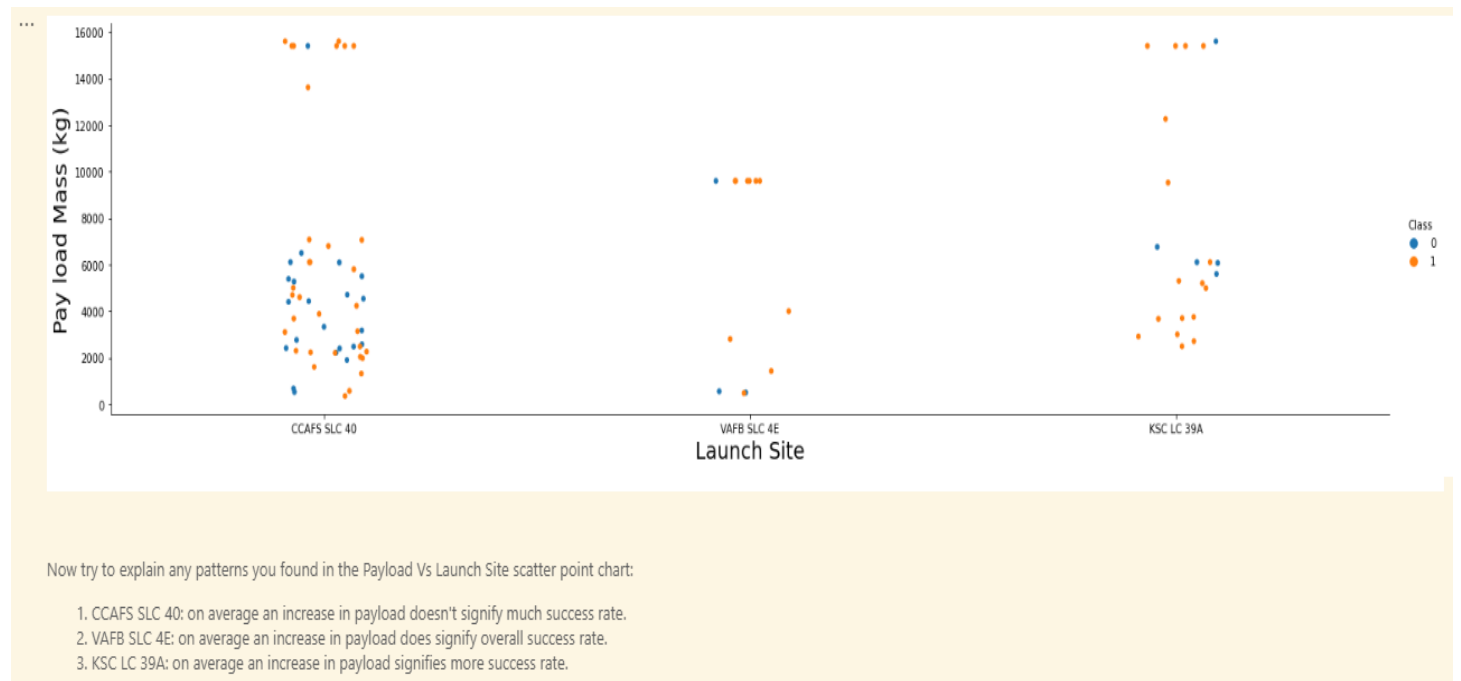- Show the screenshot of the scatter plot with explanations:

Now try to explain the patterns you found in the Flight Number vs. Launch Site scatter point plots:

1. CCAFS SLC 40: has success rate of 60%, and its success rate increases with increase in flight number.
2. VAFB SLC 4E: has success rate of 76.9%, an increase in flight number signifies more overall success of the site.
3. KSC LC 39A: has success rate of 77.3%, an increase in flight number signifies more overall success of the site.

Payload vs. Launch Site

- Show a scatter plot of Payload vs. Launch Site:

  sns.catplot(y="PayloadMass", x="LaunchSite", hue="Class", data=df, aspect = 4)

- Show the screenshot of the scatter plot with explanations:


Now try to explain any patterns you found in the Payload Vs Launch Site scatter point chart:

1. CCAFS SLC 40: on average an increase in payload doesn't signify much success rate.
2. VAFB SLC 4E: on average an increase in payload does signify overall success rate.
3. KSC LC 39A: on average an increase in payload signifies more success rate.

Success Rate vs. Orbit Type

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

  df.groupby(['Orbit']).mean()['Class'].plot(kind='bar')

- Show the screenshot of the scatter plot with explanations:
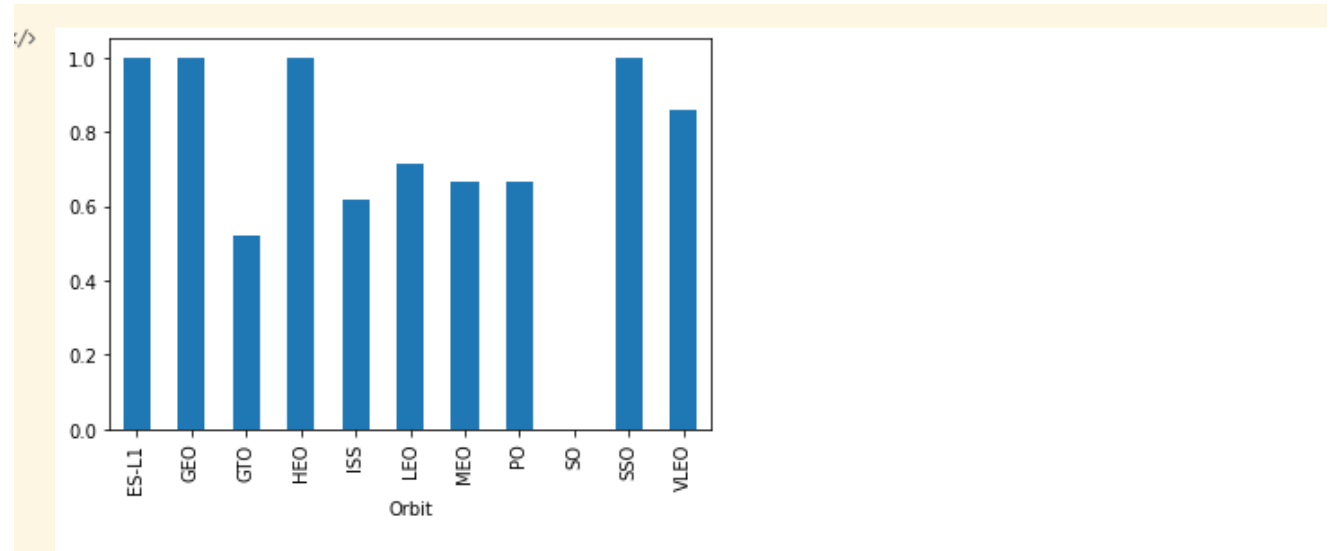
Analyze the ploted bar chart try to find which orbits have high sucess rate.

1. Orbits ES-LI, GEO, HEO, and SEO orbit have higher success rate of 100%.
2. Orbits VLEO AND LEO have success rate of 80% and 70% respectively.
3. Orbits MEO, PO have success rate of nearly 65% respectively.
4. Orbits ISS, GTO, and SO have success rate of 60%, 50%, 0% respectively.

Flight Number vs. Orbit Type

- Show a scatter point of Flight number vs. Orbit type:

sns.catplot(y="FlightNumber", x="Orbit", hue="Class", data=df, aspect = 4)

- Show the screenshot of the scatter plot with explanations:

You should see that in the LEO orbit the Success appears related to the number of flights; on the other hand, there seems to be no relationship between flight number when in GTO orbit.
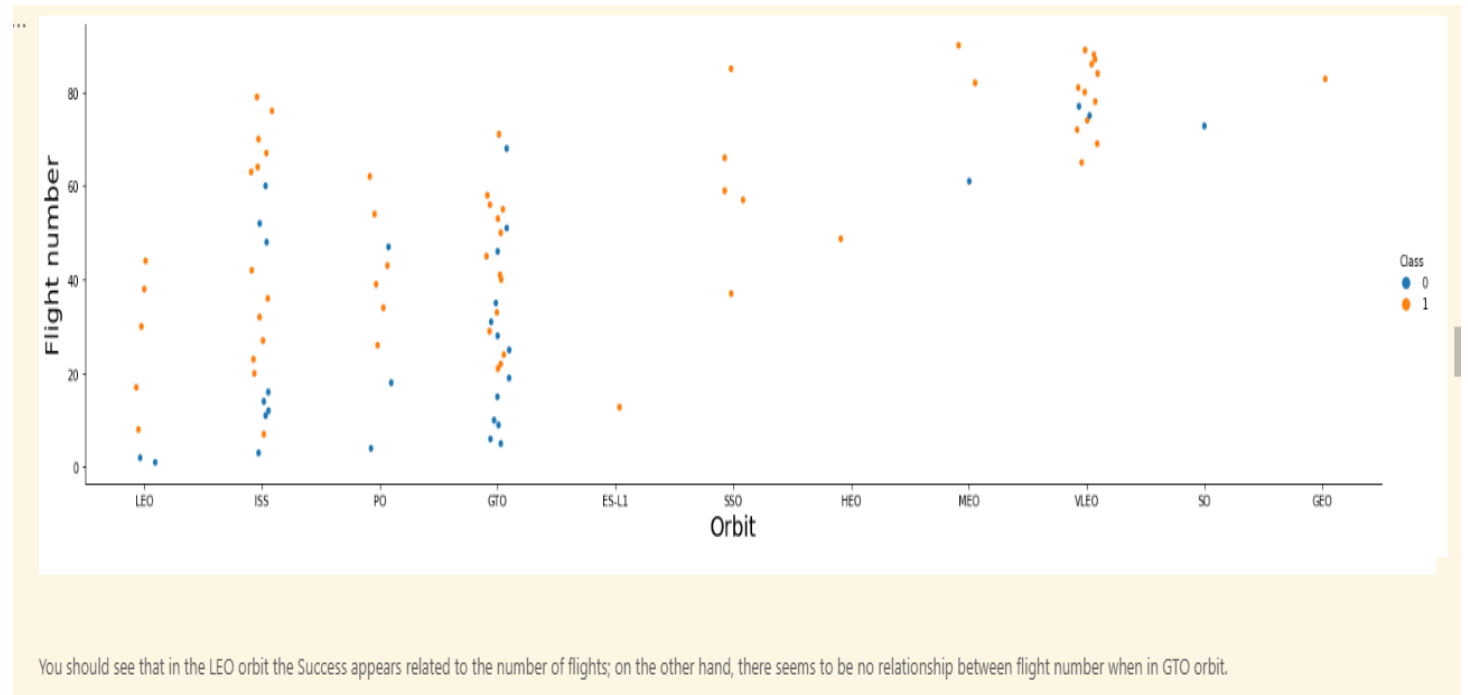
Payload vs. Orbit Type

- Show a scatter point of payload vs. orbit type

sns.catplot(y="PayloadMass", x="Orbit", hue="Class",

data=df, aspect = 5)

- Show the screenshot of the scatter plot with explanations:
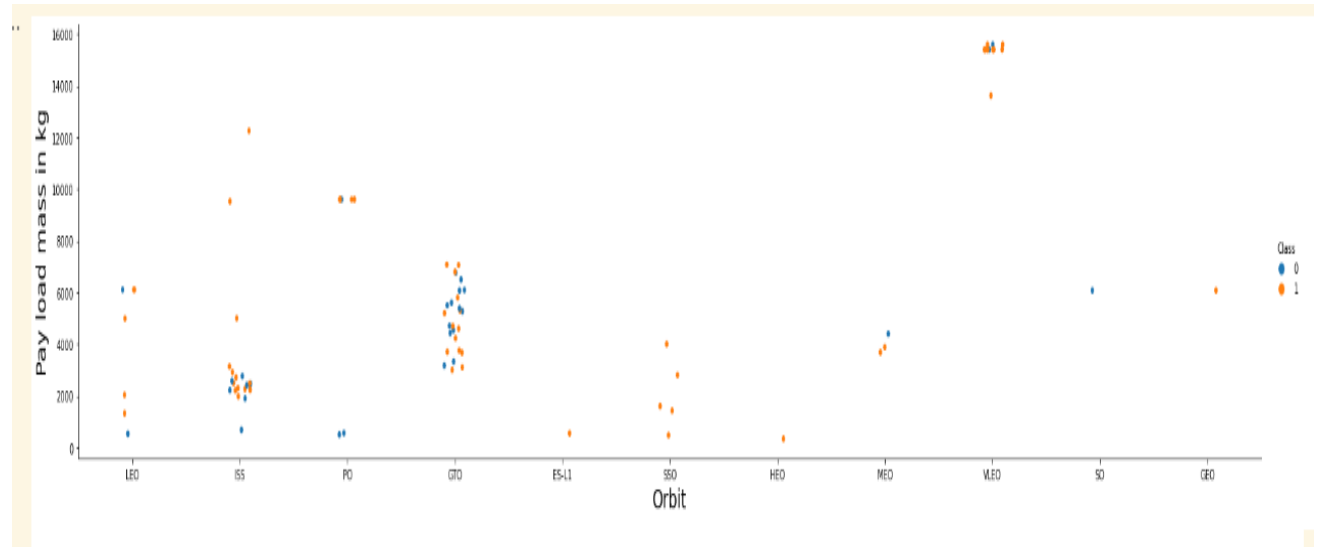
You should observe that Heavy payloads have a negative influence on GTO orbits and positive on GTO and Polar LEO (ISS) orbits.

Launch Success Yearly Trend

- Show a line chart of yearly average success rate:

  df.groupby(['Year']).mean()['Class'].plot(kind='line')

- Show the screenshot of the line plot with explanations:
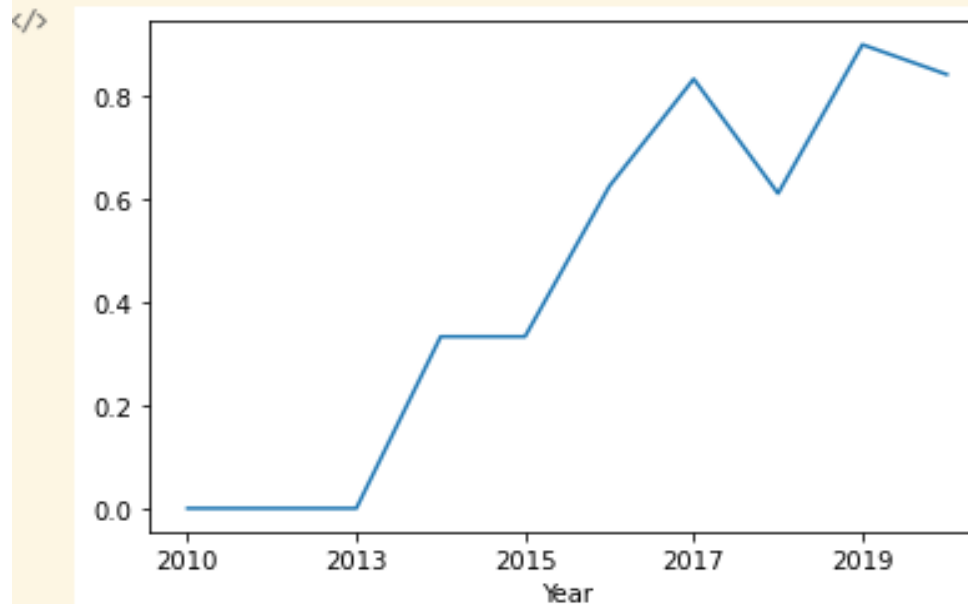
you can observe that the sucess rate since 2013 kept increasing till 2020

All Launch Site Names

- Find the names of the unique launch sites:

  %sql SELECT DISTINCT LAUNCH_SITE FROM VRK0824
       8.SPACEX;

- Present your query result with a short explanation here:

Launch Site Names Begin with 'CCA'

- Find 5 records where launch sites begin with `CCA`:

```sql
%sql SELECT LAUNCH_SITE FROM VRK08248.SPACEX WHERE LAUNCH_SITE LIKE 'CCA%' LIMIT 5;
```

- Present your query result with a short explanation here:

Total Payload Mass

- Calculate the total payload carried by boosters from NASA:

```
%sql select sum(PAYLOAD_MASS__KG_) from VRK08248.SPACEX where CUSTOMER = 'NASA (CRS)';
```

- Present your query result with a short explanation here:

Average Payload Mass by F9 v1.1

- Calculate the average payload mass carried by booster version F9 v1.1:

```sql
%sql select avg(PAYLOAD_MASS__KG_) as average_payload_mass from VRK08248.SPACEX where booster_version = 'F9 v1.1';
```

- Present your query result with a short explanation here:

First Successful Ground Landing Date

- Find the dates of the first successful landing outcome on ground pad:

```
%sql select date,Landing__outcome from VRK08248.SPACEX where LANDING__OUTCOME = 'Success (ground pad)';
```

- Present your query result with a short explanation here:

```
%sql·select·date,Landing__outcome·from·VRK08248.SPACEX·where·LANDING__OUTCOME·=·'Success·(ground·pad)';

... * ibm_db_sa://vrk08248:***@fbd88901-ebdb-4a4f-a32e-9822b9fb237b.c1ogj3sd0tgtu0lqde00.databases.appdomain.cloud:32731/bludb
Done.
```

DATE	landing__outcome
2015-12-22	Success (ground pad)
2016-07-18	Success (ground pad)
2017-02-19	Success (ground pad)
2017-01-05	Success (ground pad)
2017-03-06	Success (ground pad)
2017-08-14	Success (ground pad)
2017-07-09	Success (ground pad)
2017-12-15	Success (ground pad)
2018-08-01	Success (ground pad)

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.

```
%sql select booster_version, PAYLOAD_MASS__KG_ from VRK08248.SPACEX

        where LANDING__OUTCOME = 'Success (drone ship)' AND PAYLOAD_MASS__KG_ between 4000 and 6000;
```

- Present your query result with a short explanation here.

Total Number of Successful and Failure Mission Outcomes

- Calculate the total number of successful and failure mission outcomes:

```
%sql select count(mission_outcome)as count_mission_outcome from VRK08248.SPACEX
        where mission_outcome ='Success' OR mission_outcome = 'Failure';
```

- Present your query result with a short explanation here:

Boosters Carried Maximum Payload

- List the names of the booster which have carried the maximum payload mass:

```
%sql select UNIQUE(booster_version),launch_site, customer, Landing__outcome, PAYLOAD_MASS__KG_,
            (select max(PAYLOAD_MASS__KG_) from VRK08248.SPACEX) as max_payload from VRK08248.SPACEX;
```

- Present your query result with a short explanation here:


```
> %sql·select·UNIQUE(booster_version),launch_site,·customer,·Landing__outcome,·PAYLOAD_MASS__KG_,·(select·max(PAYLOAD_MASS__KG_)·from·VRK08248.SPACEX)·as·max_paylo
```
Python
```
* ibm_db_sa://vrk08248:***@fbd88901-ebdb-4a4f-a32e-9822b9fb237b.c1ogj3sd0tgtu0lqde00.databases.appdomain.cloud:32731/bludb
Done.
```

booster_version	launch_site	customer	landing__outcome	payload_mass__kg_	max_payload
F9 v1.0 B0003	CCAFS LC-40	SpaceX	Failure (parachute)	0	15600
F9 v1.0 B0004	CCAFS LC-40	NASA (COTS) NRO	Failure (parachute)	0	15600
F9 B4 B1045.1	CCAFS SLC-40	NASA (LSP)	Success (drone ship)	362	15600
F9 FT B1038.1	VAFB SLC-4E	NSPO	Success (drone ship)	475	15600
F9 v1.0 B0006	CCAFS LC-40	NASA (CRS)	No attempt	500	15600
F9 v1.1 B1003	VAFB SLC-4E	MDA	Uncontrolled (ocean)	500	15600
F9 v1.0 B0005	CCAFS LC-40	NASA (COTS)	No attempt	525	15600
F9 v1.1 B1017	VAFB SLC-4E	NASA (LSP) NOAA CNES	Failure (drone ship)	553	15600
F9 v1.1 B1013	CCAFS LC-40	U.S. Air Force NASA NOAA	Controlled (ocean)	570	15600
F9 v1.0 B0007	CCAFS LC-40	NASA (CRS)	No attempt	677	15600
F9 B5B1063.1	VAFB SLC-4E	NASA / NOAA / ESA / EUMETSAT	Success	1192	15600
F9 v1.1	CCAFS LC-40	Orbcomm	Controlled (ocean)	1316	15600
F9 v1.1 B1015	CCAFS LC-40	NASA (CRS)	Failure (drone ship)	1898	15600
F9 v1.1 B1018	CCAFS LC-40	NASA (CRS)	Precluded (drone ship)	1952	15600
F9 B5 B1059.2	CCAFS SLC-40	NASA (CRS)	Success	1977	15600

2015 Launch Records

- List the failed landing_outcomes in drone ship, their booster versions, and launch site names for in year 2015:

```sql
%sql select Date, Booster_version, Launch_site, Landing__outcome from VRK08248.SPACEX
        where Landing__outcome = 'Failure (drone ship)' And Date like '%2015%';
```

- Present your query result with a short explanation here:

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:

```
%sql select Landing__outcome, count(landing__outcome) as landing_counts from VRK08248.SPACEX
        where  Date Between '2010-06-05' And '2017-03-20' group by LANDING__OUTCOME;
```

- Present your query result with a short explanation here:

```
%sql select Landing__outcome, count(landing__outcome) as landing_counts from VRK08248.SPACEX
    where  Date Between '2010-06-05' And '2017-03-20' group by LANDING__OUTCOME;

 * ibm_db_sa://vrk08248:***@fbd88901-ebdb-4a4f-a32e-9822b9fb237b.c1ogj3sd0tgtu0lqde00.databases.appdomain.cloud:32731/bludb
Done.
```

landing__outcome	landing_counts
Controlled (ocean)	3
Failure (drone ship)	5
Failure (parachute)	1
No attempt	9
Precluded (drone ship)	1
Success (drone ship)	5
Success (ground pad)	5
Uncontrolled (ocean)	2

Launch Sites Proximities Analysis

Folium Map Sites Locations

- Screenshot of all launch sites' location markers on a global map:

- Important elements and findings on the screenshot:

➢ All launch sites are located near to coastline, far away from populated cities, and closer to Equator.
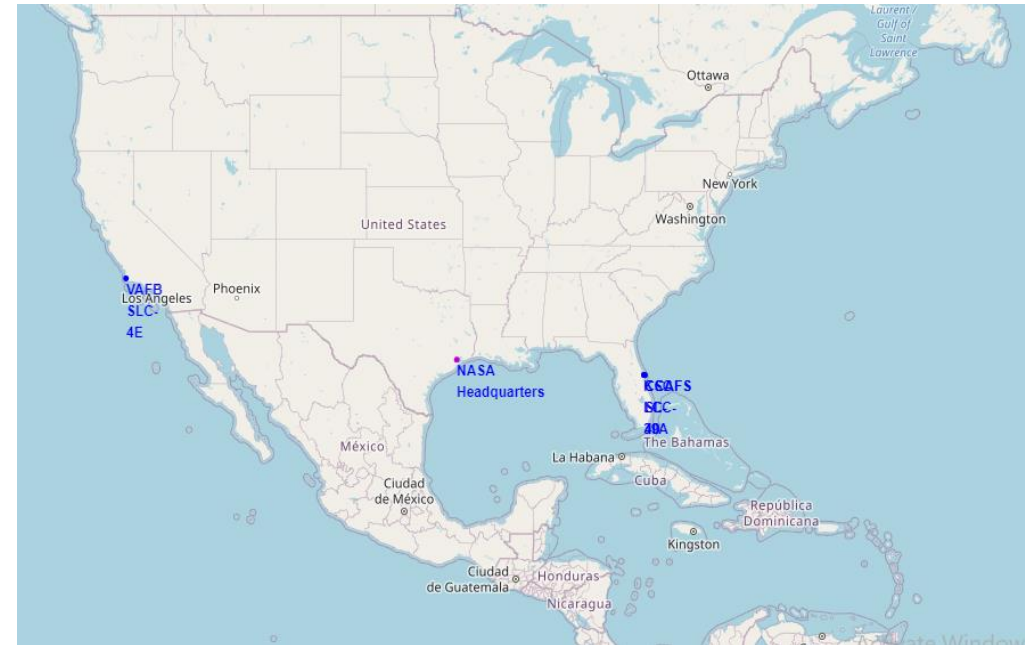
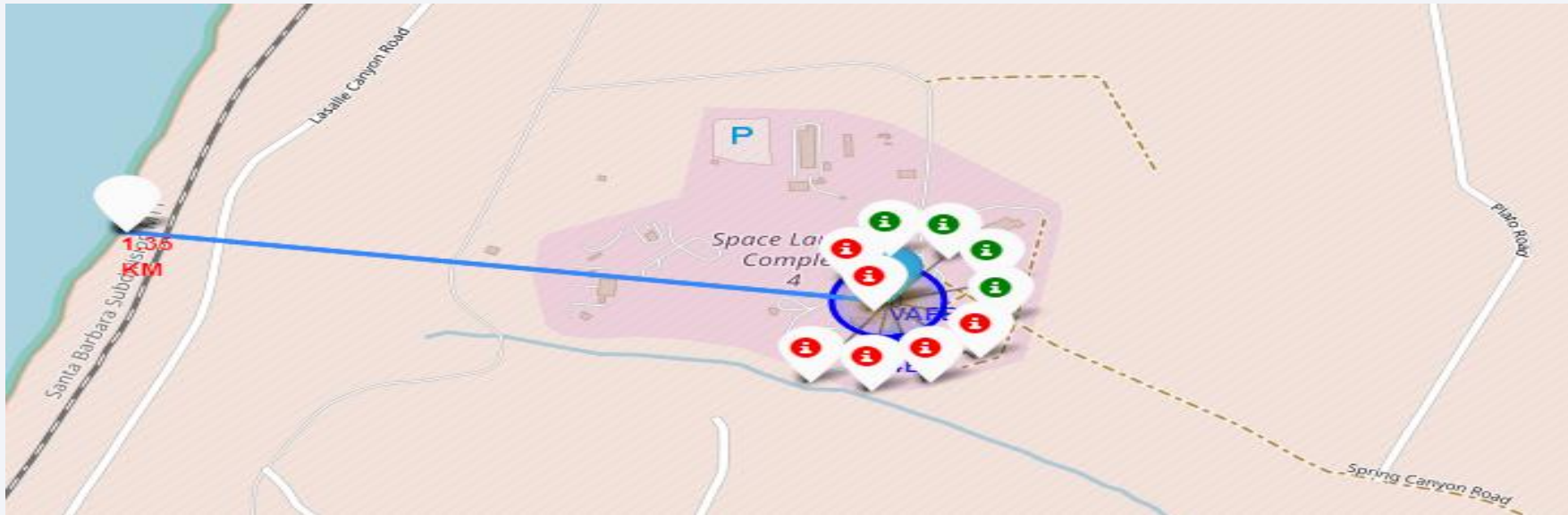# Folium Map Launch Site Outcomes

- Screenshot showing color-labeled launch outcomes on the map:

- Explain the important elements and findings on the screenshot:

➢ Total launches is 56 in all sites, 32 launches failed, and 24 launches succeeded.

Folium Map of Site VAFB SLC-4E

- Screenshot of site VAFB SLC-4E to coastline proximity with distance calculated:

- Explain the important elements and findings on the screenshot:

➢ the distance between launch site and coastline is 1.35 km, it has 6 failed launches and 4 succeeded launches.
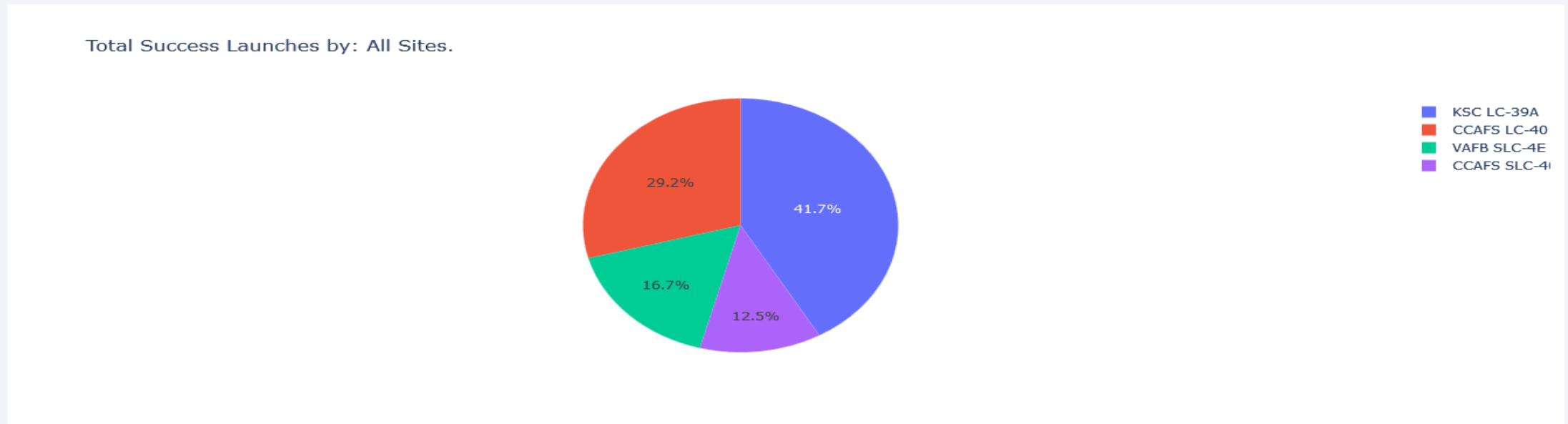
Section 5

Build a Dashboard
with Plotly Dash

Dashboard showing All Launch Sites in Pie chart

- Screenshot of launch success count for all sites in a pie chart
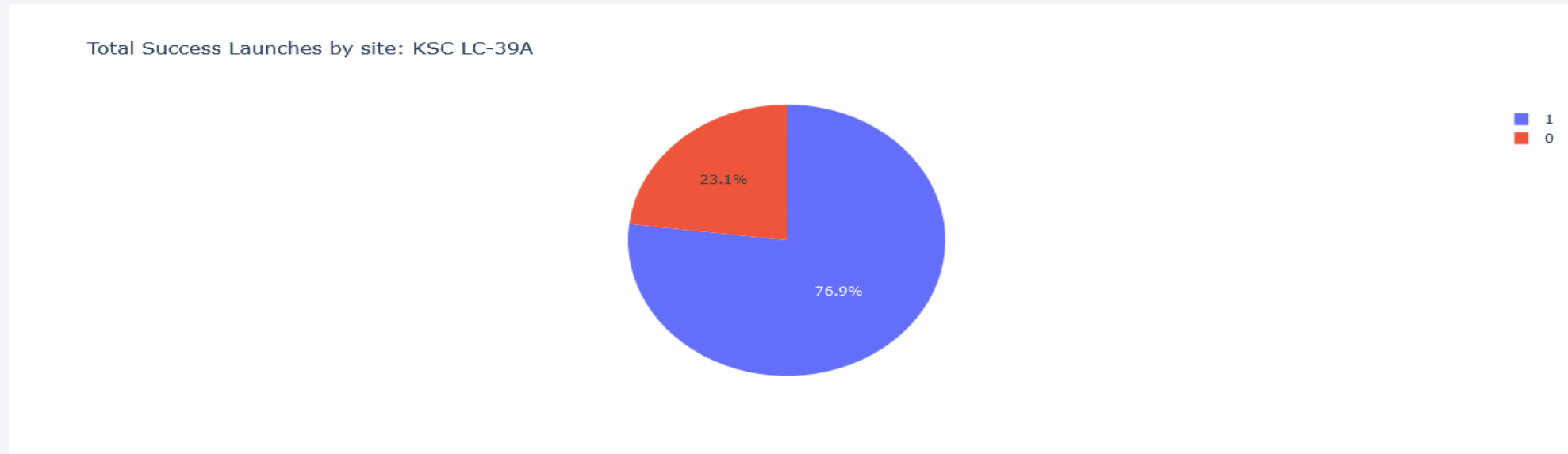

Total Success Launches by: All Sites.

- Important findings on the screenshot:

➢ Site KSC LC-39A has higher success rate followed by CCASF LC-40, VAFB SLC-4E, and CCAFS SLC-40.

Dashboard showing Site KSC LC-39A in Pie chart

- Screenshot of the launching site with highest launch success ratio:

Total Success Launches by site: KSC LC-39A

- Important findings on the screenshot:

➢ Site KSC LC-39A has higher launching success ratio of 76.9% than any other launching site.

Dashboard showing All Launch Sites in Scatter Plot

- Screenshot of Payload vs. Launch Outcome scatter plot for all sites, with different payload selected in the range slider:
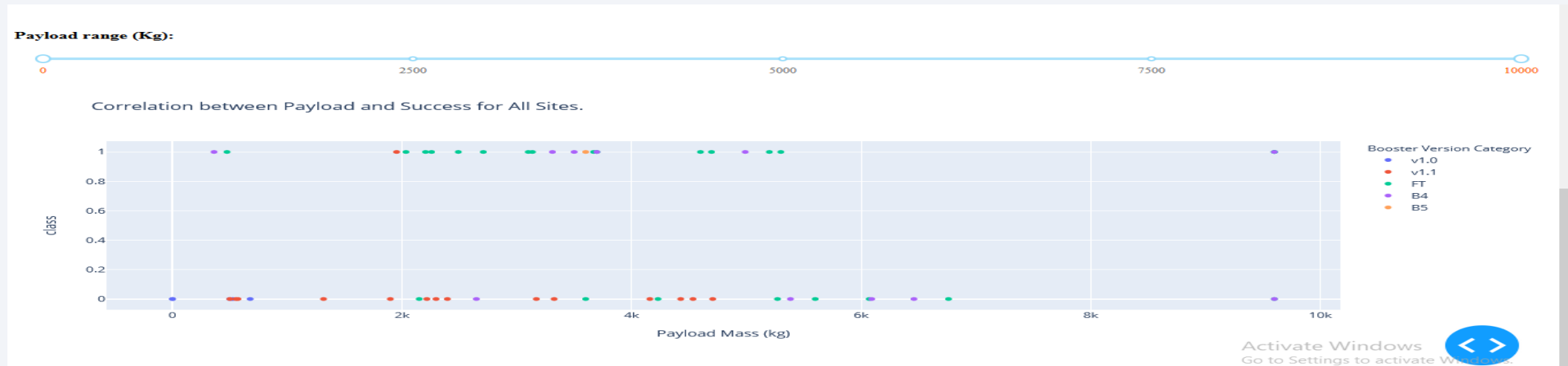
- Important elements and findings on the screenshot:

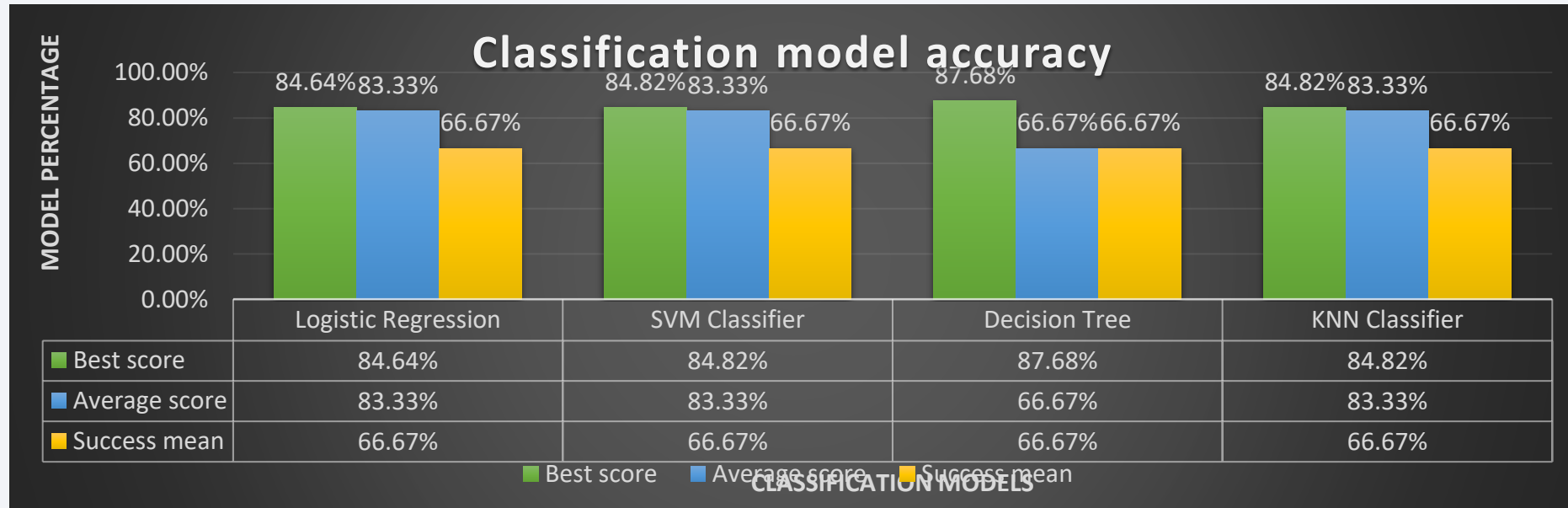➤ Booster version: FT has higher success ratio than any other booster version.

Section 6

Predictive Analysis
(Classification)

Classification Accuracy

- Visualize the built model accuracy for all built classification models, in a bar chart:

	Logistic Regression	SVM Classifier	Decision Tree	KNN Classifier
Best score	84.64%	84.82%	87.68%	84.82%
Average score	83.33%	83.33%	66.67%	83.33%
Success mean	66.67%	66.67%	66.67%	66.67%

- Model with the highest classification accuracy:

➢ Decision Tree classifier has higher best score and lower average score.

Confusion Matrix

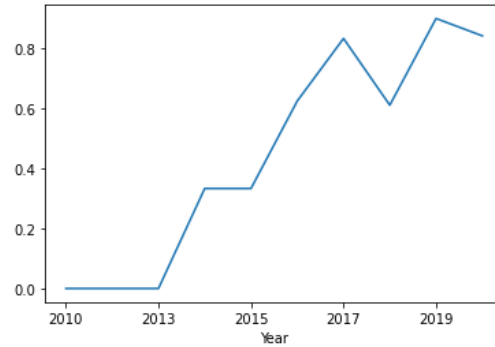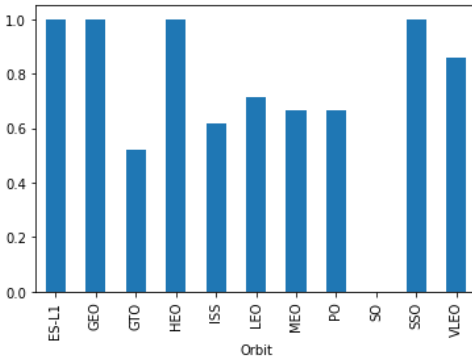- Confusion matrix of the best performing model with an explanation:
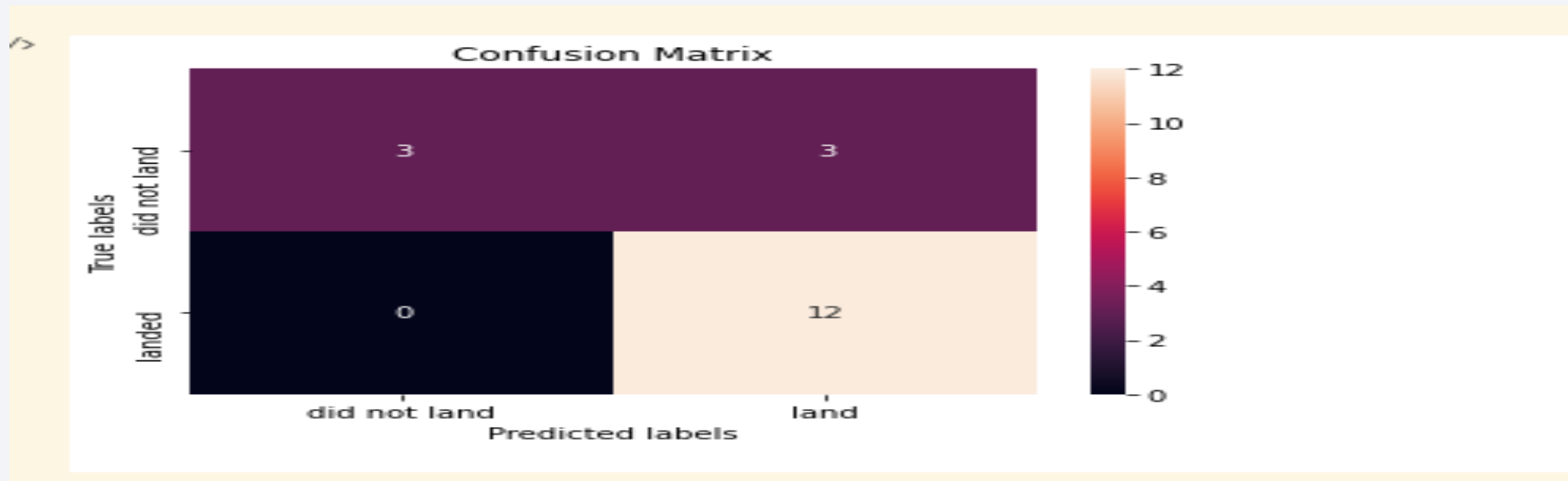
➤ Decision tree classifier predicted that 12 successfully landing outcomes

Conclusions

- The reusability success rate for Falcon 9 is 66.67%.

- More success occurred to rockets launched to ES-LI, GEO, HEO, and SEO orbits.

- The reusability success trends increased between 2013 to 2020.

- More success occurred to rockets having booster version of FT.

- More success occurred to rockets having payload mass between 1950 -5200 kg.

- All launch sites have closer proximities to coastlines, Equator, and farther away from populous cities.

- It is likely that the reusability success rate for the SpaceX's Falcon 9 will drastically increases incoming years.

Appendix

Github repo URL link for whole project:
https://github.com/Macheing/Spacex_ds_project

46

Thank you!