IBM DATA SCIENCE CAPSTONE PROJECT

ROCKET LAUNCH SUCCESS PREDICTION THROUGH MODEL DEVELOPMENT

Fredrick Irungu Njuguna
31/12/2023

GIT HUB Link: IBM Capstone Project

Outline

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

Executive Summary

- Main objective: To predict if the Falcon 9 first stage will land successfully
- Methodologies:
  - Data collection: Link to REST API and Webscraping
  - Data cleaning: Link to data Wrangling
  - Exploratory data analysis (EDA) using SQL, Pandas and Matplotlib: Link to SQL
  - Data visualization using visual analytics and dashboards: Links to static visuals and dashboards
  - Predictive analysis using machine learning: Link to predictive model
- Results
  - Different Launch sites had different success rates with CCAFS LC-40 having success rate of 60 %, while KSC LC-39A and VAFB SLC 4E had a success rate of 77 %.
  - Payload had a huge influence on the launch success rate
  - Different Orbit types had different success rate
  - LEO orbit the Success appears related to the number of flights
  - KNN model had the highest accuracy and score

Introduction

- **Project background and context**
  - ❖ SpaceX has gained worldwide attention for a series of historic milestones.
  - ❖ It is the only private company ever to return a spacecraft from low-earth orbit
  - ❖ SpaceX accomplished the first spacecraft from low-earth orbit in 2010.
  - ❖ SpaceX advertises Falcon 9 rocket launches on its website with a cost of 62 million dollars.
  - ❖ Other rocket providers cost upward of 165 million dollars each
  - ❖ Space X can reuse the first stage and hence saving on cost.
- **Problems statement**
  - ❖ The cost of launching rockets is very high with an upward cost of 165 million dollars
  - ❖ Reusing the first stage can save on launching cost.
  - ❖ We build a predictive model to determine if the first stage will land successfully.
  - ❖ This information can be used to bid for a rocket launch.

Section 1

Methodology

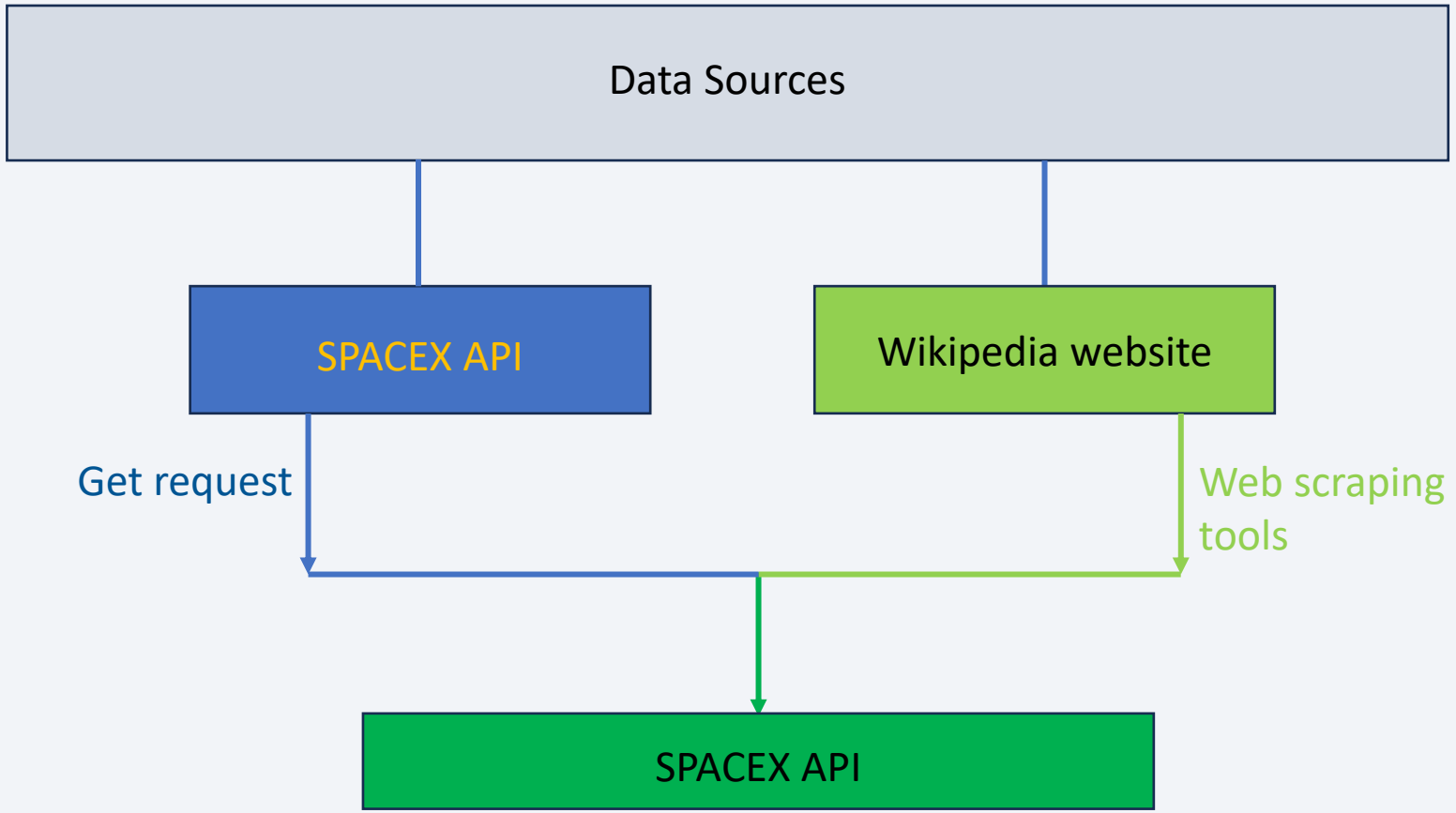
Methodology

Executive Summary

- Data collection:
  - ❖REST API : SPACEX
  - ❖Web scraping: Wikipedia website
- Performed data wrangling
- Performed exploratory data analysis (EDA) using visualization and SQL
- Performed interactive visual analytics using Folium and Plotly Dash
- Performed predictive analysis using classification models

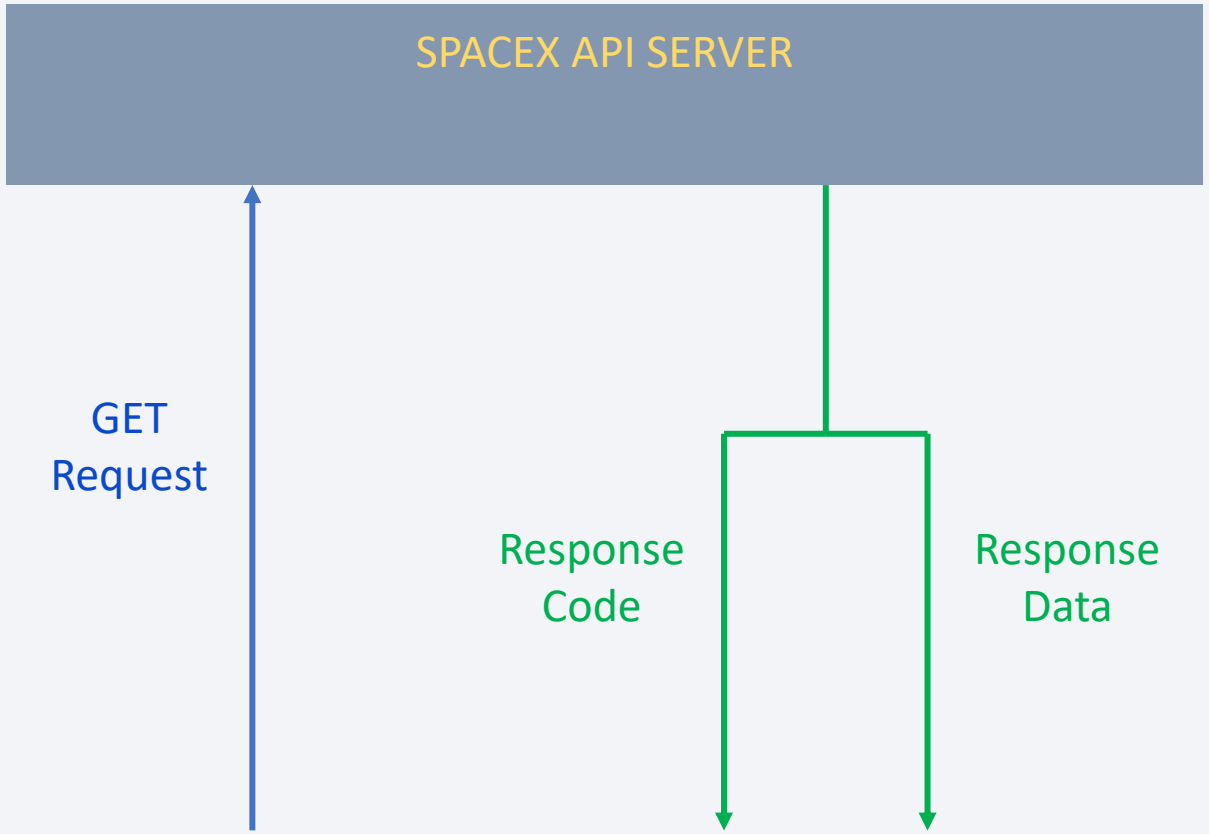
Data Collection

- Data was collected from two sources as shown in the flow chart.



Data Collection – SpaceX API

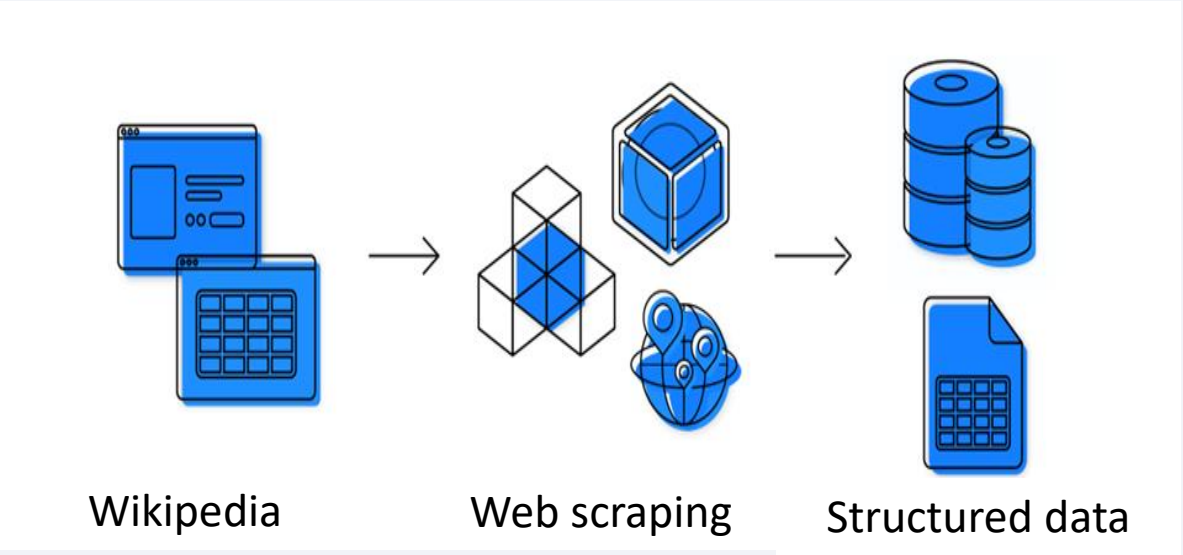
- Data collection
  - ❖*get* request to the *SpaceX API*
  - ❖HTTP request from https://api.spacexdata.com/v4/rockets
  - ❖Function: getBoosterVersion, getLaunchSite, getPayloadData and getCoreData
  - ❖Created dataframe from data obtained.
  - ❖Convertdataframe to .csv file



GIT HUB Link: REST API

Data Collection – Web Scraping

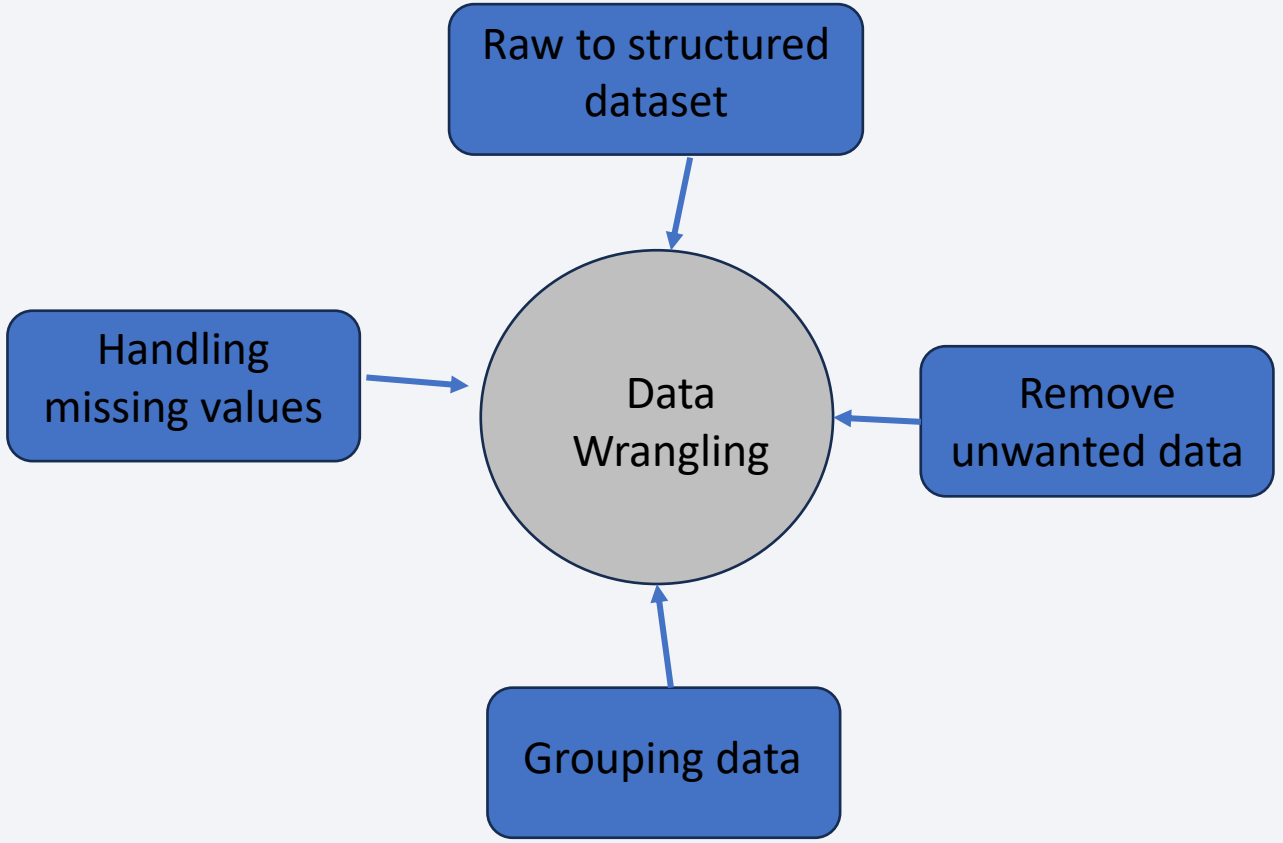
- Data source: Wikipedia website.
- Perfomed web scraping
- Used Beautifulsoup for parsing HTML data
- Extracted table columns and rows to form Pandas dataframe
- Converted Pandas dataframe to .csv file



Webscraping link

Data Wrangling

- Raw data was processed into a structured and analyzable format.
- Raw data were then cleaned and standardized to rectify inconsistencies
- Cleaning involved dealing with missing values such as replacing NAN values and removing the unwanted columns.
- Feature engineering phase was implemented to extract relevant variables and create new informative features.



Data wrangling Link

EDA with Data Visualization

- Following charts were plotted:

Scatter Plots:

✓ To explore relationships between numerical variables.

✓ They revealed patterns, trends, or correlations between variables

Bar Charts:

✓ To display the distribution of categorical variables.

✓ They illustrated the frequency or proportion of each category within a categorical variable, offering insights into its composition.

Line plots

- To visualize trends in ordered categories.
- They helped to reveal patterns and trends over ordered sequence, providing insights into sequential variations.

Git Hub Link: EDA Data Visulaization

EDA with SQL

- I performed the following SQL queries
  - %sql SELECT DISTINCT LAUNCH_SITE FROM SPACEXTBL to retrive unique values
  - %sql SELECT * FROM SPACEXTBL WHERE LAUNCH_SITE LIKE 'CCA%' LIMIT 5; to display records say 5
  - %sql SELECT SUM(PAYLOAD_MASS__KG_) AS Total_Payload_Mass FROM SPACEXTBL WHERE CUSTOMER = 'NASA (CRS)'; to display total values
  - %sql SELECT AVG(PAYLOAD_MASS__KG_) AS Average_Payload_Mass FROM SPACEXTBL WHERE BOOSTER_VERSION = 'F9 v1.1'; to calculate and display average values
  - %sql SELECT MISSION_OUTCOME, COUNT(*) AS Total_Count FROM SPACEXTBL GROUP BY MISSION_OUTCOME; for total values
  - %sql SELECT BOOSTER_VERSION FROM SPACEXTBL WHERE PAYLOAD_MASS__KG_ = (SELECT MAX(PAYLOAD_MASS__KG_) FROM SPACEXTBL); for sub querry
  - %sql SELECT LANDING_OUTCOME, COUNT(*) AS Outcome_Count FROM SPACEXTBL WHERE Date BETWEEN '2010-06-04' AND '2017-03-20' GROUP BY LANDING_OUTCOME ORDER BY Outcome_Count DESC; for counts

GitHub Link: EDA with SQL

Interactive Map with Folium

Map objects were as follows:

Markers:

- ❖AS Placed markers on locations of interest.
- ❖Markers were used to highlight key points on the map such as launch site

Circles:

- ❖Used to highlight a circular area with a text on a specific coordinate

PolyLine:

- ❖Illustrated connections or routes between points such as launch site to the selected coastline. They can used to determine distance

Mouse-clicks:

- ❖Displayed additional information when clicking on markers or other objects .

GitHub Link: Interactive Maps with Folium

Dashboard with Plotly Dash

The following plots were considered for dashboards

Pie Charts:

- ❖Represented the composition of a whole in terms of percentages.
- ❖Pie charts were used to show the success rate for the launching sites and the total success launches for a particular site

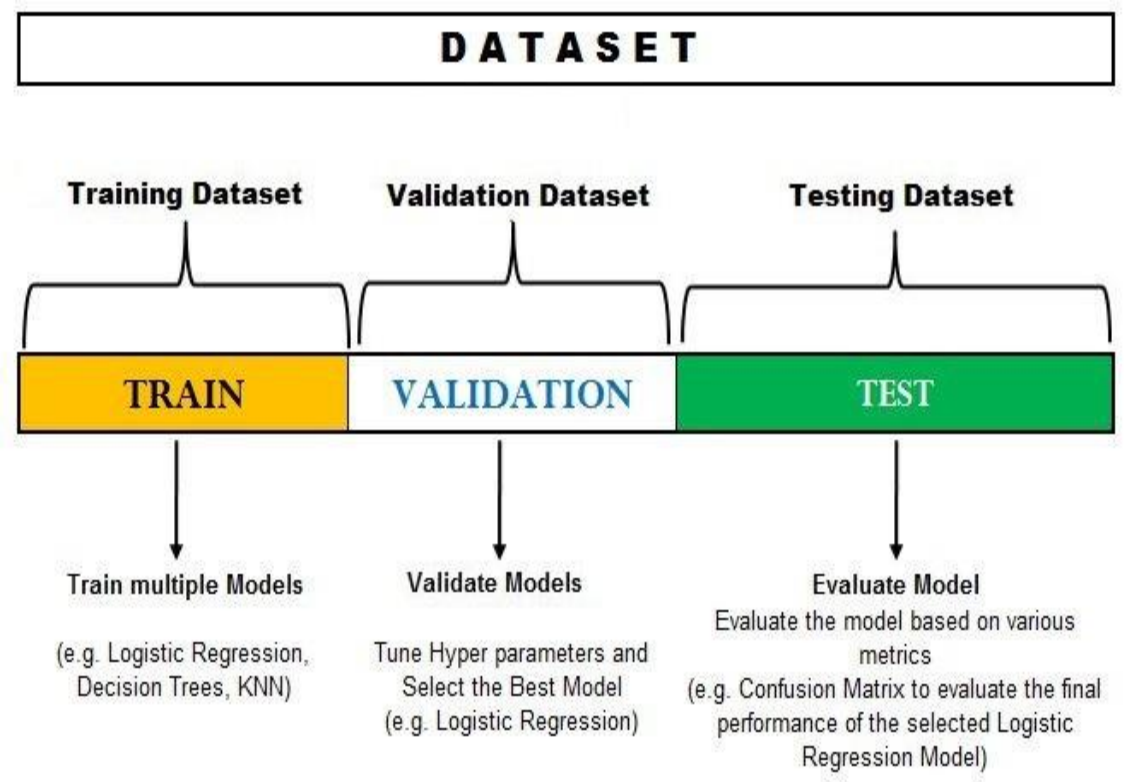
Scatter Plots:

- ❖Were used to explore correlation between two numerical variables.
- ❖Scatter plots were added to show how payload was correlated to mission outcome and identify some visual patterns

GitHub Link: Interactive dashboards

Predictive Analysis (Classification)

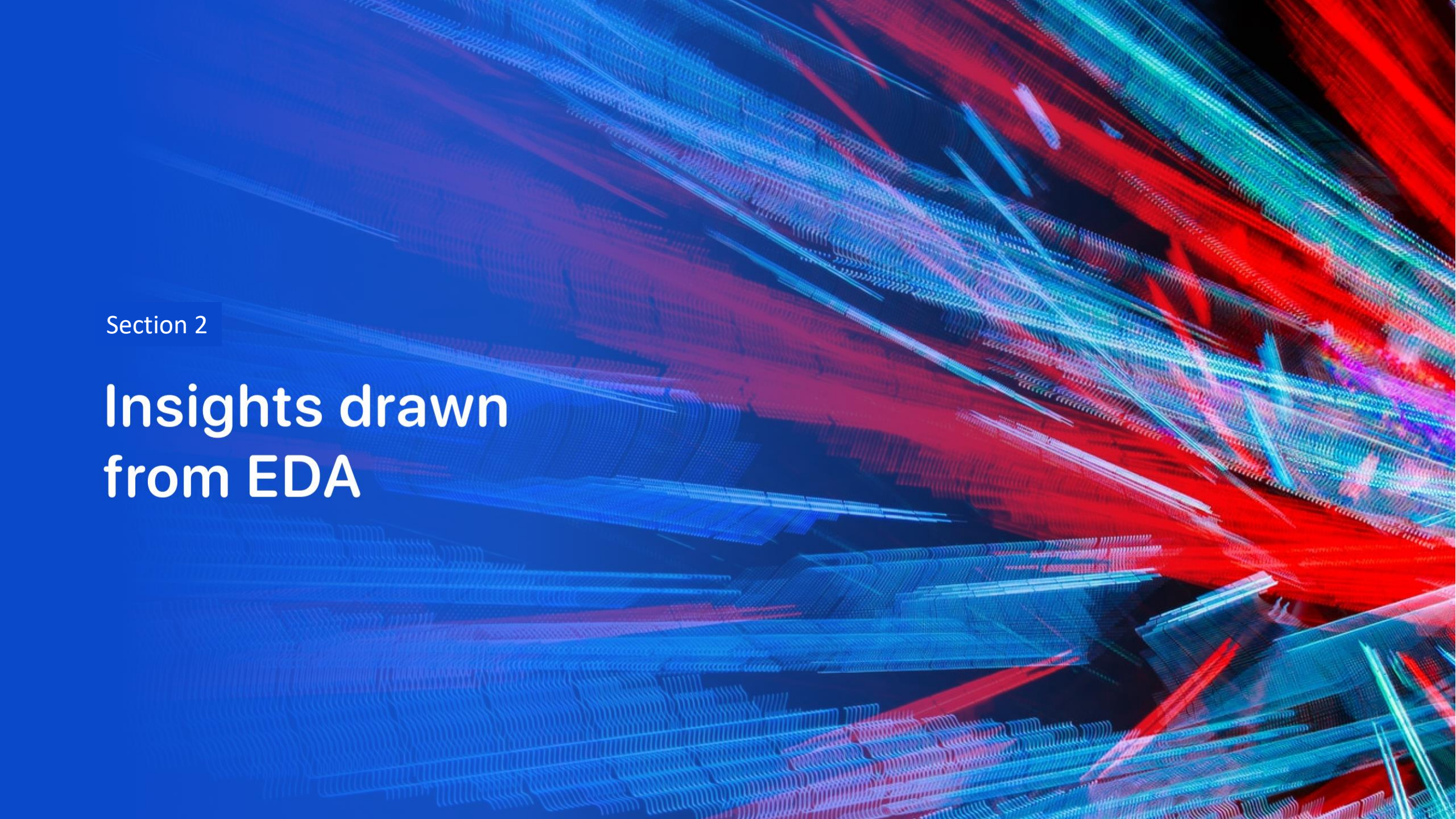
- Data normalization/standardization using .StandardScaler() function.
- Data was split into training and testing data using train_test_split function.
- The training data was divided into validation data, a second set was used for training data.
- The models were trained and hyperparameters are selected using the function GridSearchCV
- Parameters were displayed using best_params_ and the accuracy on the validation data using best_score_



GitHub Link: Model development

Summary

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


Section 2
Insights drawn from EDA

EDA with Visuals

Through EDA on the data from APi and Wiki, we found insights on :

- ✓ Flight number and Launch Sites
- ✓ Payload and Launch Sites
- ✓ Success rate and Orbit type
- ✓ Flight number and Orbit Type
- ✓ Payload and Orbit type
- ✓ Trend of success rate over the years .

The following slides present some results of visual presntations

GIT HUB link: EDA visuals

Flight Number vs. Launch Site

The following figure shows scatter plot for the flight number and Launch site

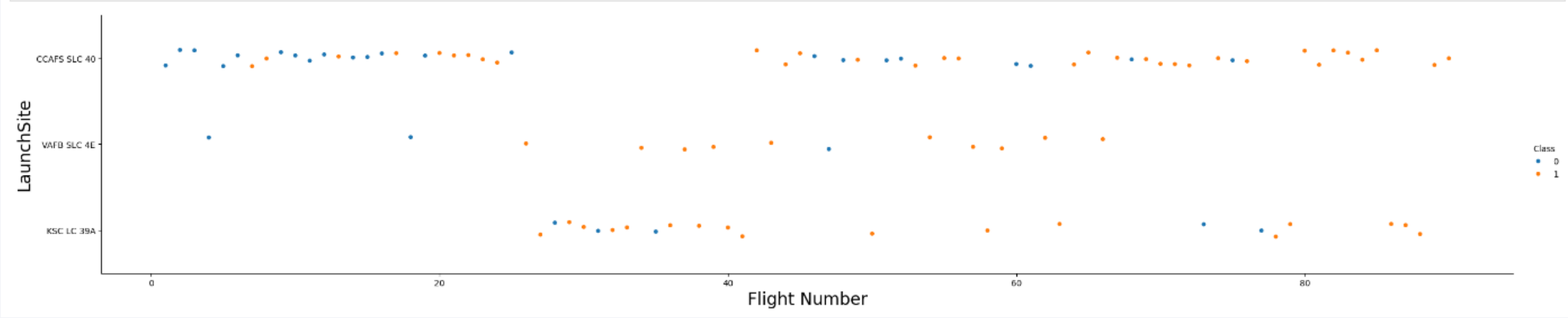


Fig: Correlation between Flight number and Launch site

From this plot, it was found that:

- The earlier flights launch were from CCAFS-SLC-40 site
- That most Launches are Launched from CCAFS-SLC-40 and the fewest were from VAFB SLC 4E site
- That the VAFB SLC 4E had greater success rate from launches conducted later

GIT HUB link: EDA visuals

Payload mass vs. Launch Site

The figure shows scatter plot for the Payload mass and Launch site

It can be conclude that:

- There were no rockets for the CCAFS-SLC-40 site launched for heavy payload mass greater than 10,000
- Launches with heavy payload mass were mostly successful

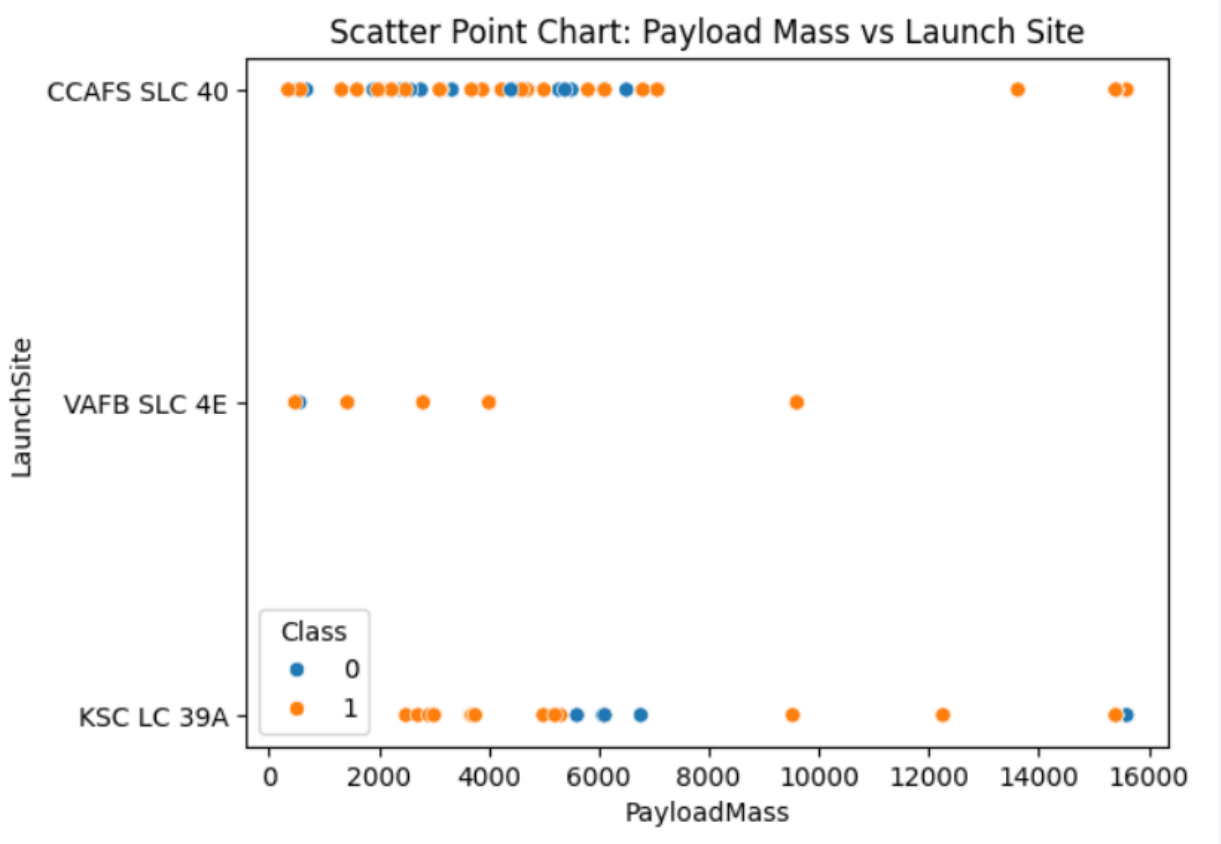


Fig: Correlation between Payload mass and Launch site

GIT HUB link: EDA visuals

Success Rate vs. Orbit Type

The figure shows bar graph for the launch success rate of different Orbit type.

It can be conclude that:

- ❖ GEO,HEO, ES-L1 and SSO had the highest launch success rate
- ❖ GTO orbit had the lowest success rate

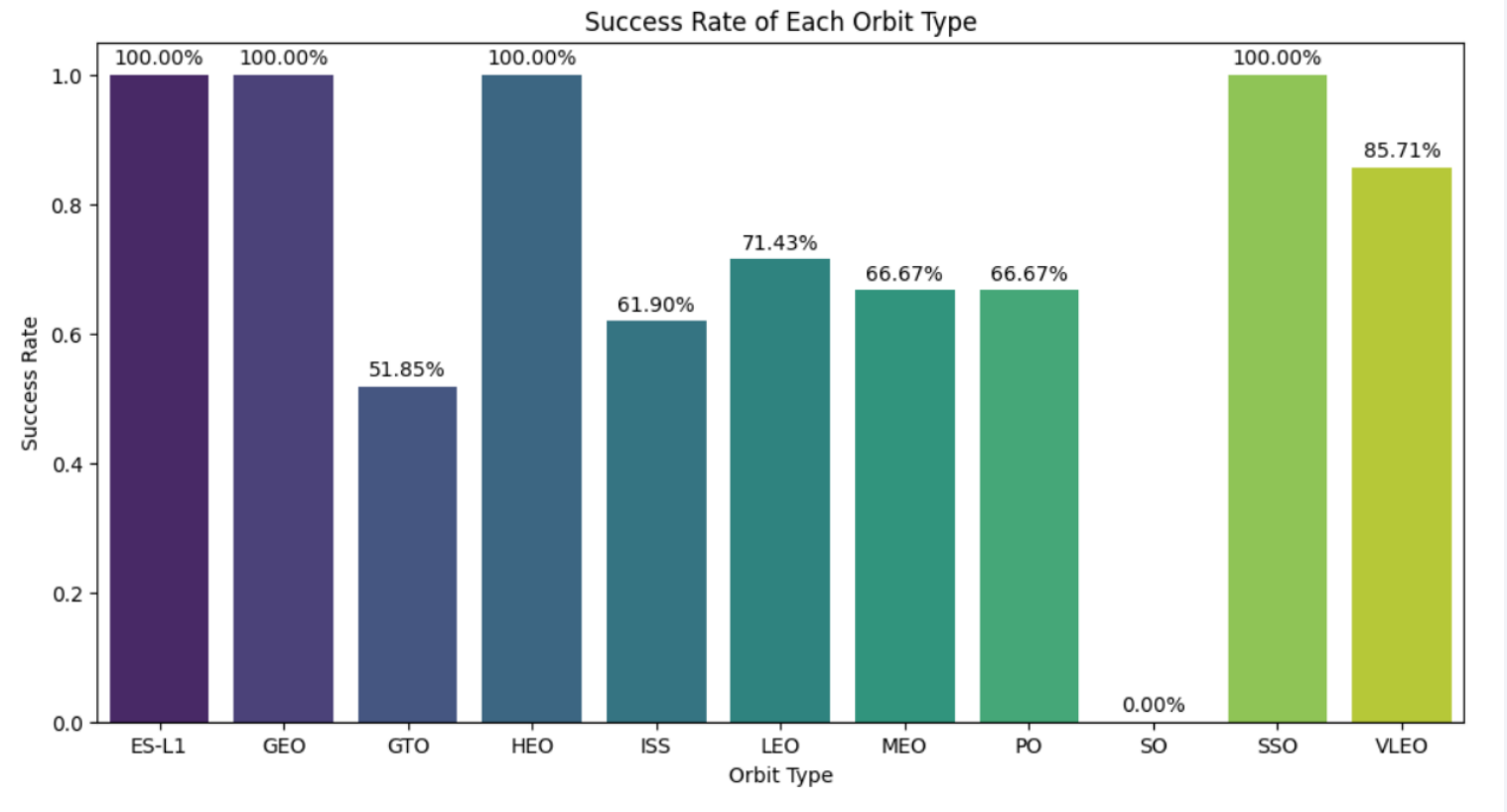


Fig: Launch success rate for various Orbits

GIT HUB link: EDA visuals

Flight Number vs. Orbit Type

The figure shows scatter plot for the Orbit type and the flight number.

It can be deduced that:

- The success launching of the LEO orbit is directly related to the number of flights.
- Launch success of the GTO orbit was not related to the fight number.

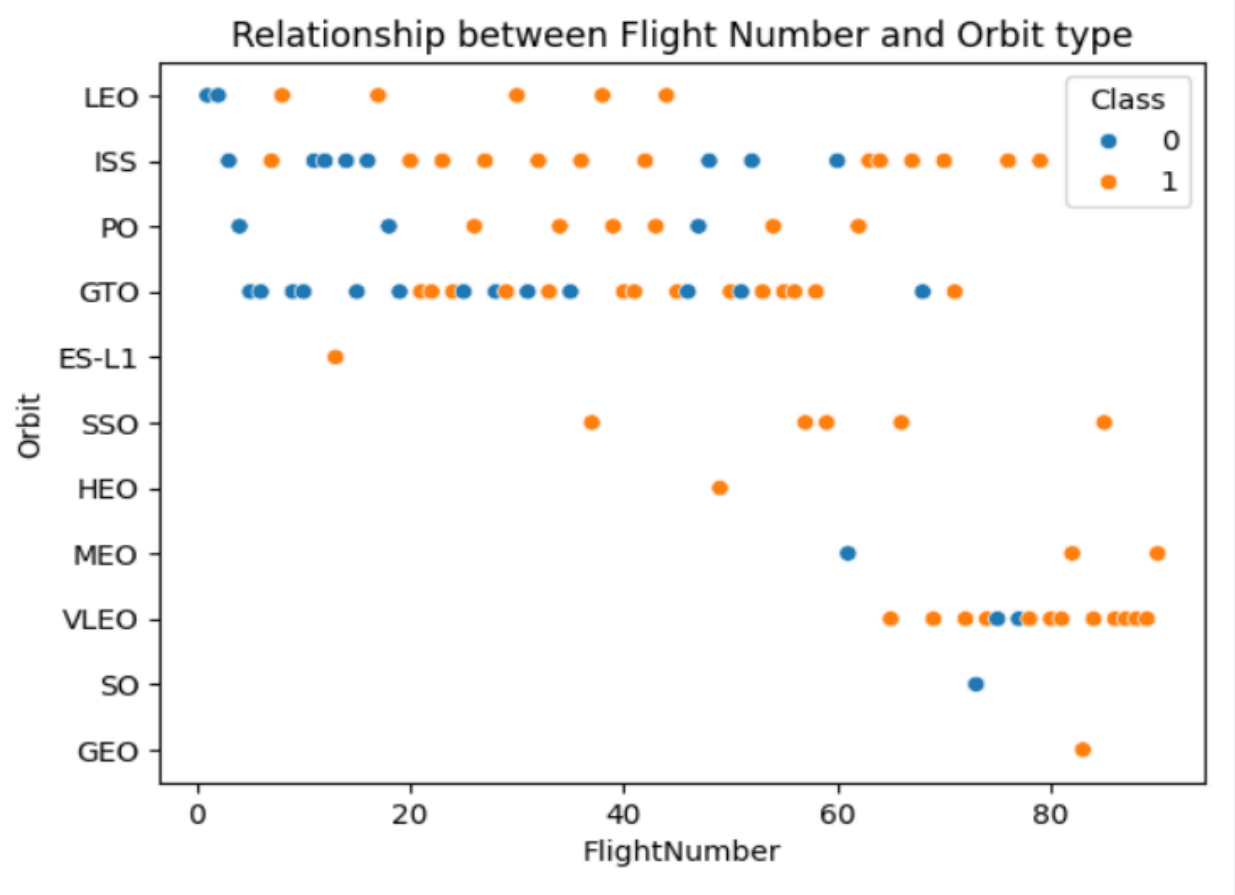


Fig: Flight number vs Orbit type

GIT HUB link: EDA visuals

Payload vs. Orbit Type

The figure shows a scatter plot for the launch success rate of different Orbit type wrt the payload mass..

It can be conclude that:

- Successful landing for LEO and ISS were more with heavy payload mass.
- It is not possible to distinguish where the successful landing were for GTO but it was mostly launched for average payload mass.

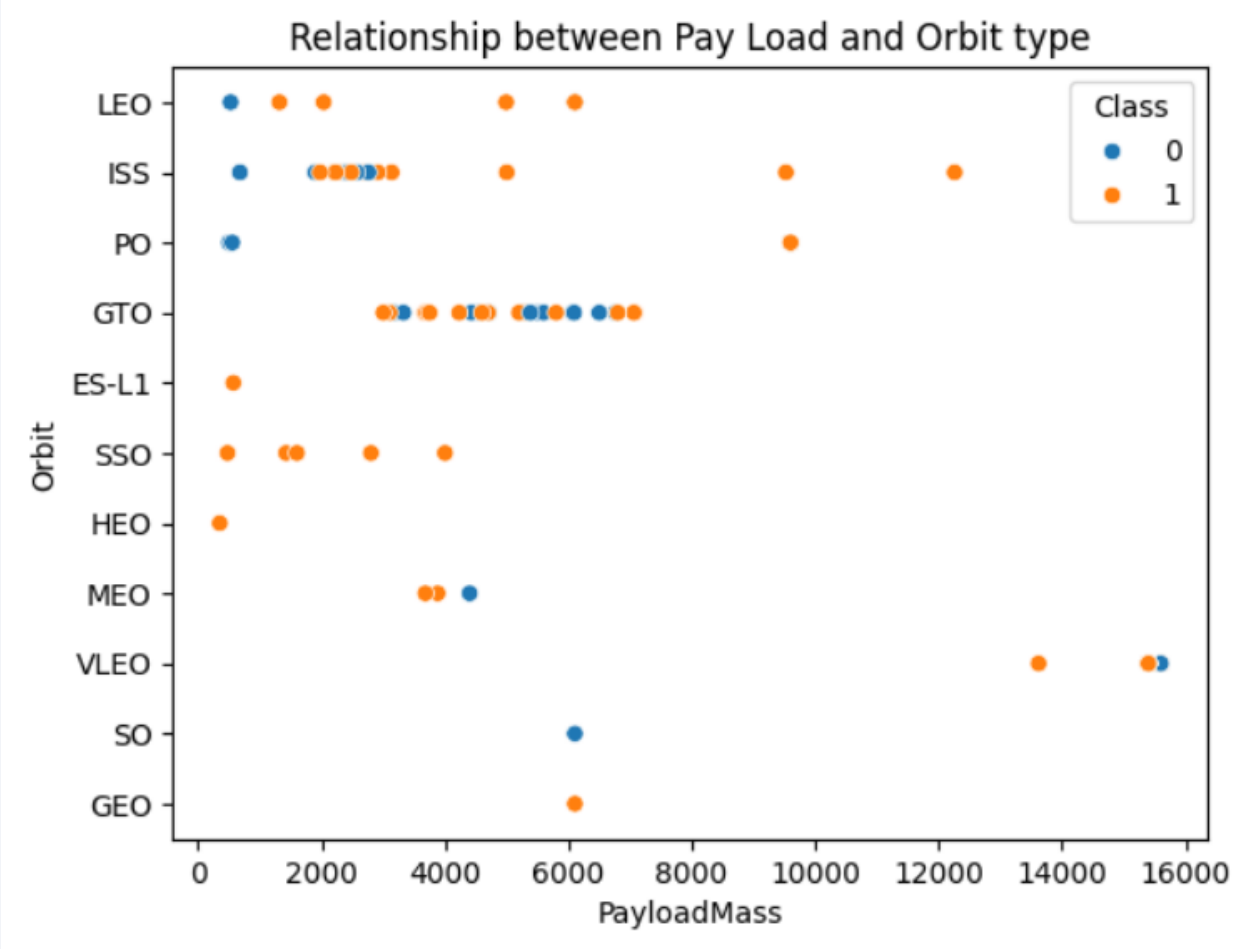


Fig: Payload mass vs Orbit type

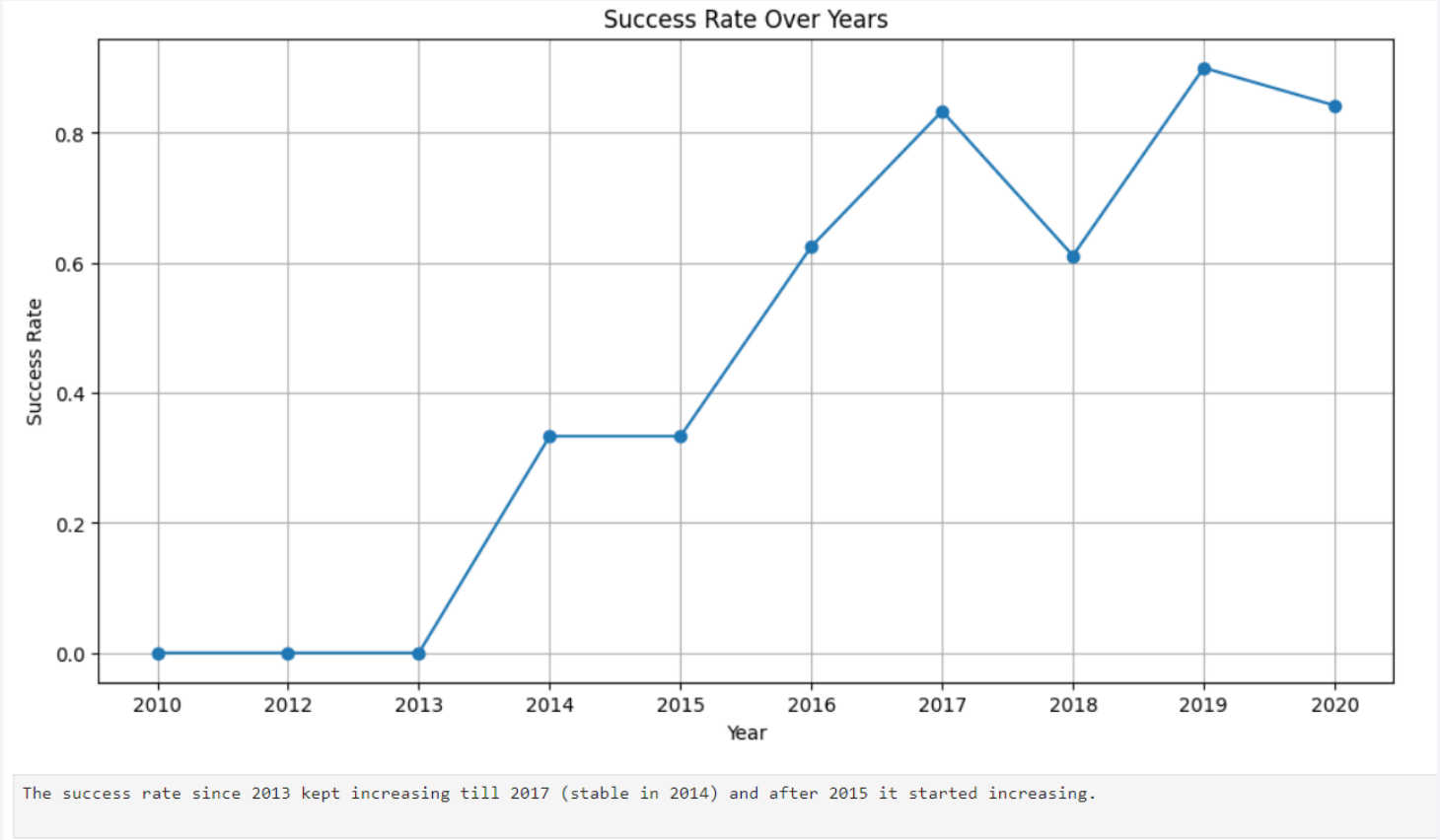
GIT HUB link: EDA visuals

Launch Success Yearly Trend

The figure shows a line plot for the launch success rate at different years.

It can be conclude that:

- Successful landing kept increasing from 2013 until 2017, but with some stagnation for the year 2014.



The success rate since 2013 kept increasing till 2017 (stable in 2014) and after 2015 it started increasing.

GIT HUB link: EDA visuals

EDA with SQL

I used SQL queries to get insights on the data by performing tasks such as:

- ✓ Listing the names of all the launch sites
- ✓ Displaying the launch site names beginning with 'CCA’
- ✓ Calculating the total payload mass
- ✓ Ranking of the landing outcomes in descending order
- ✓ Extracting dates such as the first successful ground landing date

Some examples of the tasks are presented in the following sides

GIT HUB: EDA with SQL

All Launch Site Names

The following sites were identified using sql magic query

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

The SQL query used was:

%sql SELECT DISTINCT LAUNCH_SITE FROM SPACEXTBL; to display the names of the unique launch sites in the space mission. The results are:

GIT HUB: EDA with SQL

Launch Site Names Begin with 'CCA'

Sites beginning with CCA were identified using sql magic query.

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

The SQL query used was:

%sql SELECT * FROM SPACEXTBL WHERE LAUNCH_SITE LIKE 'CCA%' LIMIT 5; to display 5 records where launch sites begin with the string 'CCA'. They are shown below:

GIT HUB: EDA with SQL

Total Payload Mass

Total payload mass was calculated using the following sql magic query:

%sql SELECT SUM(PAYLOAD_MASS__KG_) AS Total_Payload_Mass FROM SPACEXTBL WHERE CUSTOMER = 'NASA (CRS)';

The tola payload mass is shown below:

Total_Payload_Mass
45596

GIT HUB: EDA with SQL

Average Payload Mass by F9 v1.1

The average payload mass was calculated using the following sql magic query:

%sql SELECT AVG(PAYLOAD_MASS__KG_) AS Average_Payload_Mass FROM SPACEXTBL WHERE BOOSTER_VERSION = 'F9 v1.1';

The average payload mass is shown below:

Average_Payload_Mass
2928.4

GIT HUB: EDA with SQL

First Successful Ground Landing Date

The first date for successful ground landing date, shown below, was extracted using the following SQL query:

%sql SELECT MIN(Date) AS First_Successful_Landing_Date FROM SPACEXTBL WHERE LANDING_OUTCOME = 'Success';

First_Successful_Landing_Date
2018-07-22

GIT HUB: EDA with SQL

Successful Drone Ship Landing with Payload between 4000 and 6000

Booster version, with payload between 4000 and 6000, which successfully landed are as shown.

They were extracted using the following SQL query:

%sql SELECT BOOSTER_VERSION FROM SPACEXTBL WHERE LANDING_OUTCOME = 'Success' AND PAYLOAD_MASS__KG_ > 4000 AND PAYLOAD_MASS__KG_ < 6000;

Booster_Version
F9 B5 B1046.2
F9 B5 B1047.2
F9 B5 B1048.3
F9 B5 B1051.2
F9 B5B1060.1
F9 B5 B1058.2
F9 B5B1062.1

GIT HUB: EDA with SQL

Total Number of Successful and Failure Mission Outcomes

Total number of mission outcomes for failed and successful missions are shown below..

They were extracted using the following SQL query:

%sql SELECT MISSION_OUTCOME, COUNT(*) AS Total_Count FROM SPACEXTBL GROUP BY MISSION_OUTCOME;

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

GIT HUB: EDA with SQL

Boosters Carried Maximum Payload

The list the names of the booster versions which carried the maximum payload mass is as shown.

They were extracted using the following SQL subquery:

```
%sql SELECT BOOSTER_VERSION FROM SPACEXTBL WHERE
PAYLOAD_MASS__KG_ = (SELECT MAX(PAYLOAD_MASS__KG_) FROM
SPACEXTBL);
```

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

GIT HUB: EDA with SQL

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

Ranking of the landing outcomes, in descending order, between 2010-06-04 and 2017-03-20 are listed as show.

They were extracted using the following SQL query:

```
%sql SELECT LANDING_OUTCOME, COUNT(*) AS Outcome_Count FROM SPACEXTBL WHERE Date BETWEEN '2010-06-04' AND '2017-03-20' GROUP BY LANDING_OUTCOME ORDER BY Outcome_Count DESC;
```

Landing_Outcome	Outcome_Count
No attempt	10
Success (drone ship)	5
Failure (drone ship)	5
Success (ground pad)	3
Controlled (ocean)	3
Uncontrolled (ocean)	2
Failure (parachute)	2
Precluded (drone ship)	1

GIT HUB: EDA with SQL

Section 3

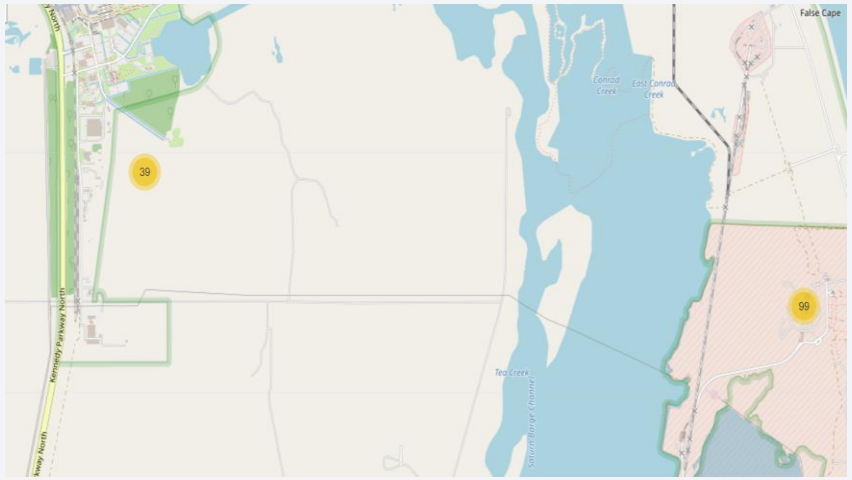
Launch Sites Proximities Analysis

GIT HUB Link: Visualization with Folium maps

Visualization using Folium map

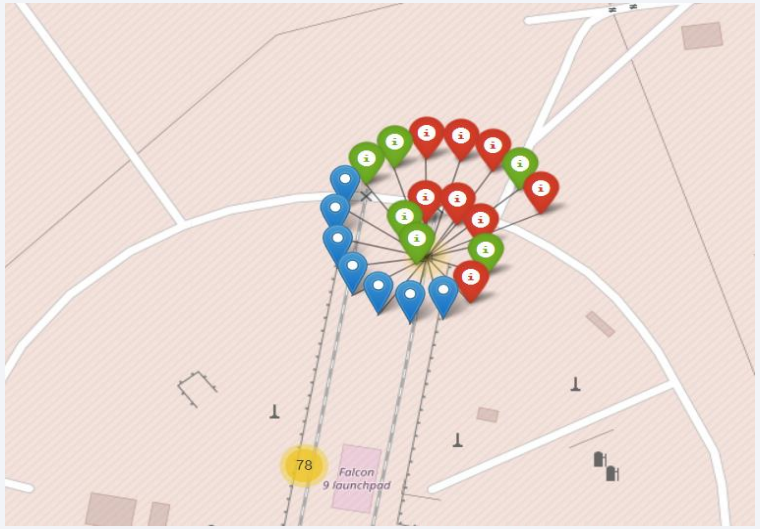
Folium map function in python is a good way to visualize locations using longitude and latitudes

The circular markers shown in the world map below indicates launch site location.



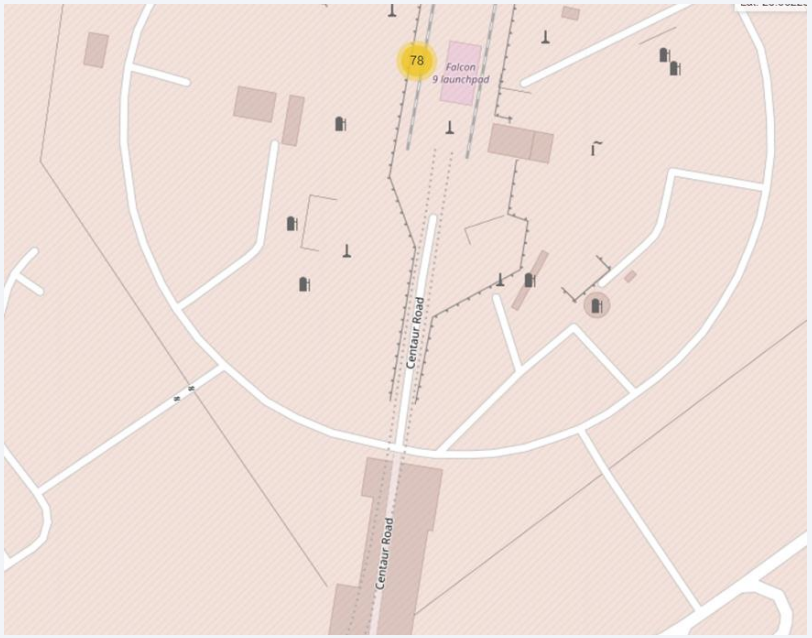
Most launch sites shared the same location (Long and Lat) and hence the markers overlapped.

The markers in the map below were colored to indicate successful or failed launch.



The markers are for a particular launch site

The figure below shows a launch site and its proximity to road.



This can be used to calculate the distance between a particular launch site and its proximities such as railway lines, roads, etc.

GIT HUB Link: Visualization with Folium maps

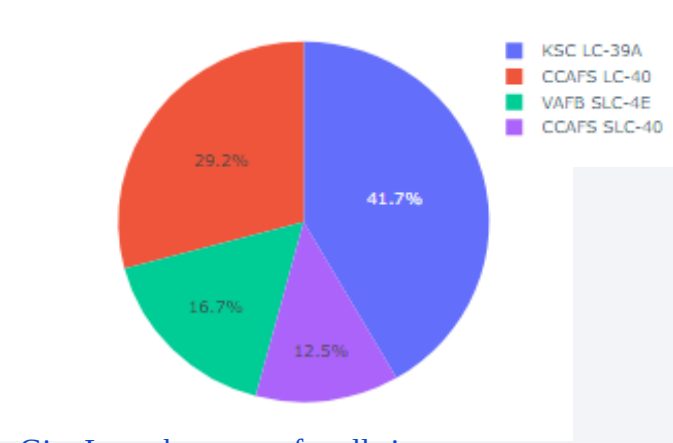
Section 4

Build a Dashboard with Plotly Dash

GIT HUB Link: Dash app

Interactive dashboards

The pie charts show that KSC LC-39A had the highest success rate of 41.7 % while the success rate of CCAFS SLC40 was the lowest at 12.5 %. The success rate for KSC LC-39A launch site was high at 76.9 % while only 23.1 % of the launch failed.



Gig: Launch success for all sites

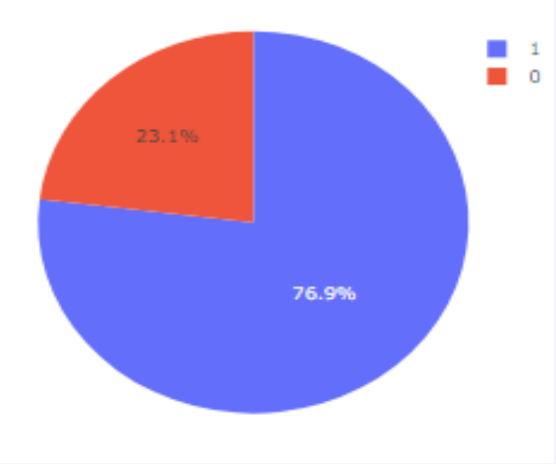
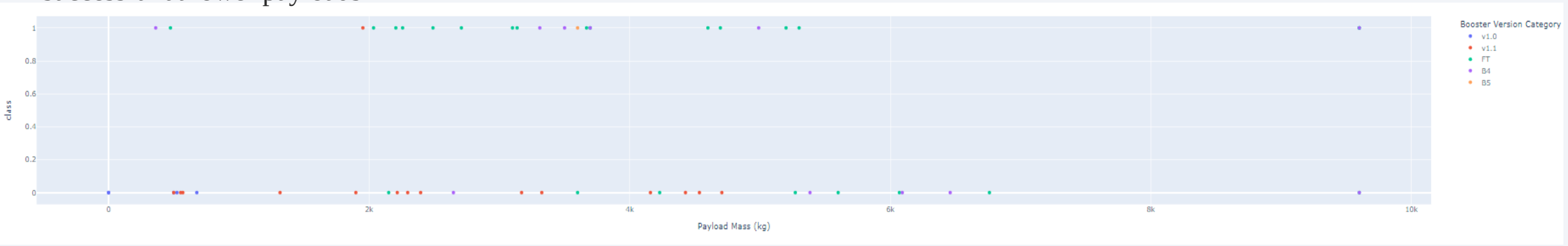


Fig: Success rate for KSC LC-39A Launch site

The scatter plot shows that the booster version FT have the most successful launch at heavy payloads. Booster versions B4 and B5 are mostly successful at lower payloads



[GIT HUB Link: Dash app]

Fig: Payload vs. Launch Outcome scatter plot for all sites

Section 5

Predictive Analysis (Classification)

GIT HUB Link: Model prediction

Model development

- Data was split into training and testing data
- The training data was divided into validation data, a second set was used for training data.
- The models were trained and hyperparameters for comparison
- Predictive model were developed and tuned for best performance.

The models included:

- ❖ K Nearest Neighbors (KNN)
- ❖ Decision Tree
- ❖ Logistic Regression
- ❖ Support Vector Machine

- KNN was the best predictive model with an accuracy of 84% and a score of 83%.

GIT HUB Link: Model prediction

Conclusions

- Data was successfully collected and cleaned
- Different Launch sites had different success rates with CCAFS LC-40 having success rate of 60 %
- KSC LC-39A and VAFB SLC 4E had a success rate of 77 %.
- Payload had a huge influence on the launch success rate
- Predictive models for predicting launching success rate were developed.
- KNN model was the best predictive model with an accuracy of 84% and a score of 83%.

Appendix

Appendix 1: sample dataset

	FlightNumber	Date	BoosterVersion	PayloadMass	Orbit	LaunchSite	Outcome	Flights	GridFins	Reused	Legs	LandingPad	Block	ReusedCount	Serial	
0	1	2010-06-04	Falcon 9	6104.959412	LEO	CCAFS SLC 40	None None	1	False	False	False	NaN	1.0	0	B0003	
1	2	2012-05-22	Falcon 9	525.000000	LEO	CCAFS SLC 40	None None	1	False	False	False	NaN	1.0	0	B0005	
2	3	2013-03-01	Falcon 9	677.000000	ISS	CCAFS SLC 40	None None	1	False	False	False	NaN	1.0	0	B0007	
3	4	2013-09-29	Falcon 9	500.000000	PO	VAFB SLC 4E	False Ocean	1	False	False	False	NaN	1.0	0	B1003	-1
4	5	2013-12-03	Falcon 9	3170.000000	GTO	CCAFS SLC 40	None None	1	False	False	False	NaN	1.0	0	B1004	

Appendix 2: dashboard application

```
# Create a dash application
app = dash.Dash(__name__)

# Create an app layout
app.layout = html.Div(children=[html.H1('SpaceX Launch Records Dashboard',
                                        style={'textAlign': 'center', 'color': '#503D36',
                                               'font-size': 40}),
                                # TASK 1: Add a dropdown list to enable Launch Site selection
                                # The default select value is for ALL sites
                                # dcc.Dropdown(id='site-dropdown',...)
                                dcc.Dropdown(id='site-dropdown',
                                             options=[
                                                      {'label': 'ALL SITES', 'value': 'ALL'},
                                                      {'label': 'CCAFS LC-40', 'value': 'CCAFS LC-40'},
                                                      {'label': 'VAFB SLC-4E', 'value': 'VAFB SLC-4E'},
                                                      {'label': 'KSC LC-39A', 'value': 'KSC LC-39A'},
                                                      {'label': 'CCAFS SLC-40', 'value': 'CCAFS SLC-40'}
                                                     ],
                                             value='ALL',
                                             placeholder="Select a Launch Site here",
                                             searchable=True),
                                html.Br(),
```

Appendix 3: sample sql query

```
%sql CREATE TABLE SPACEXTABLE AS SELECT * FROM SPACEXTBL WHERE Date IS NOT NULL;
```

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

Tasks

Now write and execute SQL queries to solve the assignment tasks.

Note: If the column names are in mixed case enclose it in double quotes For Example "Landing_Outcome"

Task 1

Display the names of the unique launch sites in the space mission

```
%sql SELECT DISTINCT LAUNCH_SITE FROM SPACEXTBL;
```