IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

Vijay Ananth Karunanithi
16.08.2023

Outline

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

Executive Summary

This executive summary provides a concise overview of the data analysis and predictive modeling conducted on a dataset related to Space-X's rocket launches. The analysis aims to explore patterns, predict outcomes, and evaluate the performance of various machine learning algorithms.

The dataset includes information on rocket launches, including launch site details, rocket characteristics, and landing outcomes. The analysis involves data preprocessing, exploratory data analysis (EDA), and predictive modeling using multiple algorithms.

Key Findings:

- Exploratory Data Analysis (EDA) revealed insights into launch sites, payload masses, and landing outcomes.
  A significant portion of launches originated from specific launch sites.
- Launch sites "CCAFS SLC 40" and "VAFB SLC 4E" are the most frequently used, with a higher number of successful landings at "CCAFS SLC 40."
- The analysis highlighted variations in payload masses, indicating that payloads vary widely across launches.
- A predictive analysis was conducted using Logistic Regression, Support Vector Machine (SVM), Decision Tree, and K-Nearest Neighbors (KNN) algorithms.

Introduction

Space exploration is at the forefront of technological advancement, with SpaceX leading the charge. Now, a new contender emerges – Space-Y. Leveraging SpaceX's launch data, this project aims to empower Space-Y's mission planning and success through predictive analysis.

Problem & Objective

Space-Y aims to optimize rocket launches for mission success. The challenge is identifying key factors that impact outcomes. Our goal is to use SpaceX's launch data to predict success, guiding Space-Y's decision-making.

Scope & Impact

Focused on historical SpaceX data, we'll conduct exploratory analysis and develop predictive models. Insights will empower Space-Y to make informed launch decisions, reducing risks and costs, while advancing the frontier of space exploration.

Section 1

Methodology

Methodology

- Data collection methodology:

  Acquire SpaceX's launch data, including launch sites, booster versions, payload mass, outcomes, and more.

- Perform data wrangling:

  Cleanse the data by handling missing values, outliers, and inconsistencies.

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

- Perform predictive analysis using classification models

  Choose suitable algorithms for predictive analysis, including Logistic Regression, SVM, Decision Tree, and KNN. Split data into training and validation sets to assess model performance. Utilize metrics such as accuracy, precision, recall, and F1-score to evaluate models.

Data Collection

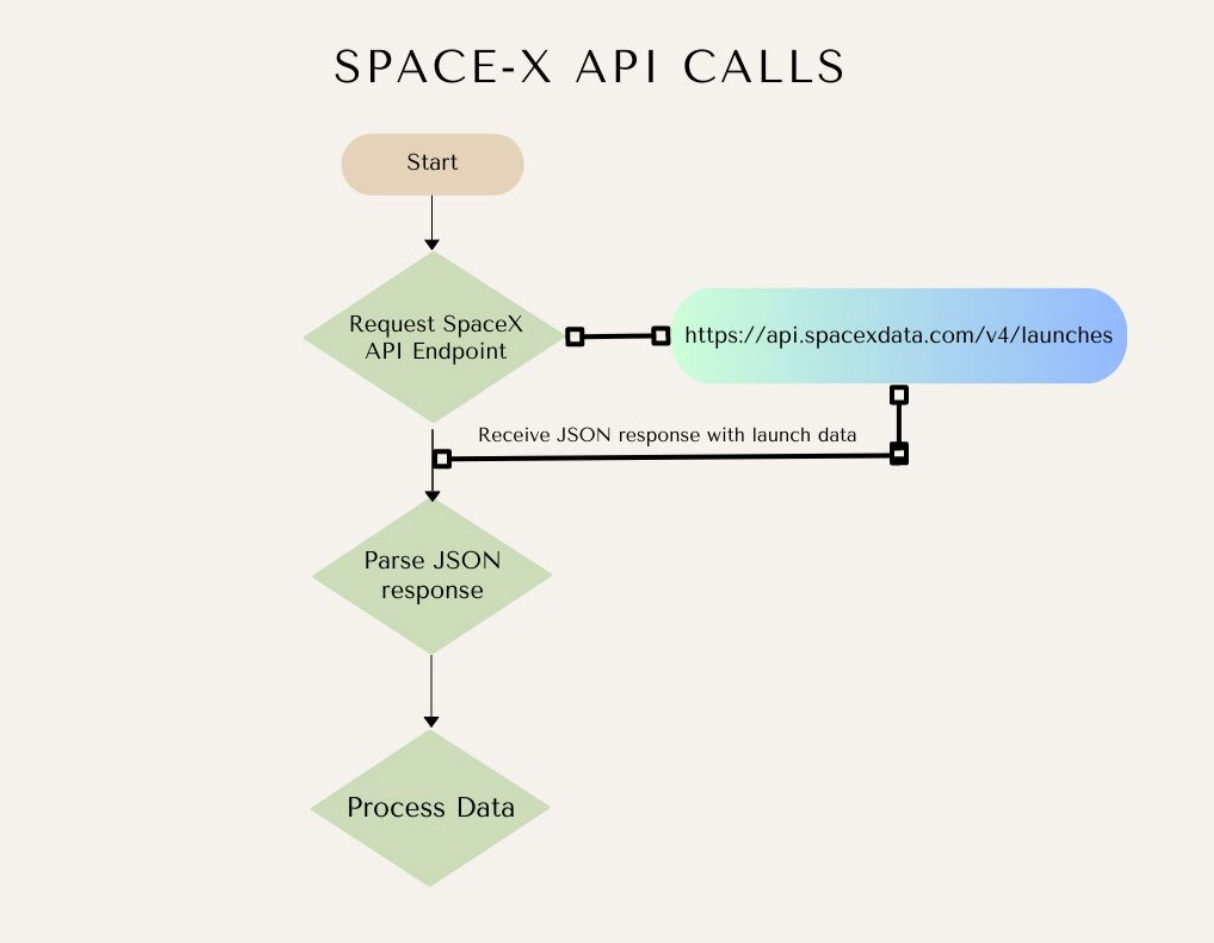
SpaceX provides an Application Programming Interface (API) that allows developers to access their launch data programmatically.

The SpaceX API allows us to retrieve detailed information about SpaceX's launch history, including launch details, payloads, launch sites, outcomes, and more.

The next slide specifies how SpaceX API was used to obtain the launch data used for the project :

Data Collection – SpaceX API

- More on: https://github.com/Vijay190899/Project/blob/main/1_Data_Wrangling.ipynb



Data Wrangling

- The processes involved (post Collection) were
- **Data Inspection:**
  - Explore the raw data to understand its structure, attributes, and potential issues.
  - Check for missing values, outliers, and inconsistencies.
- **Data Transformation:**
  - Standardize numerical data (scaling, normalization) to ensure uniformity.
  - Encode categorical data using techniques like one-hot encoding.
- **Data Aggregation:**
  - Group data by specific attributes and perform aggregations (sum, average, count) to create summaries.

More on : https://github.com/Vijay190899/Project/blob/main/1_Data_Wrangling.ipynb

Data Wrangling

TASK 4: Create a landing outcome label from Outcome column

Using the `Outcome`, create a list where the element is zero if the corresponding row in `Outcome` is in the set `bad_outcome`; otherwise, it's one. Then assign it to the variable `landing_class`:

```
# landing_class = 0 if bad_outcome
# landing_class = 1 otherwise

landing_class = [0 if outcome in bad_outcomes else 1 for outcome in df['Outcome']]
```

This variable will represent the classification variable that represents the outcome of each launch. If the value is zero, the first stage did not land successfully; one means the first stage landed Successfully

```
df['Class']=landing_class
df[['Class']].head(8)
```

	Class
0	0
1	0
2	0
3	0
4	0
5	0
6	1
7	1

Identify and calculate the percentage of the missing values in each attribute

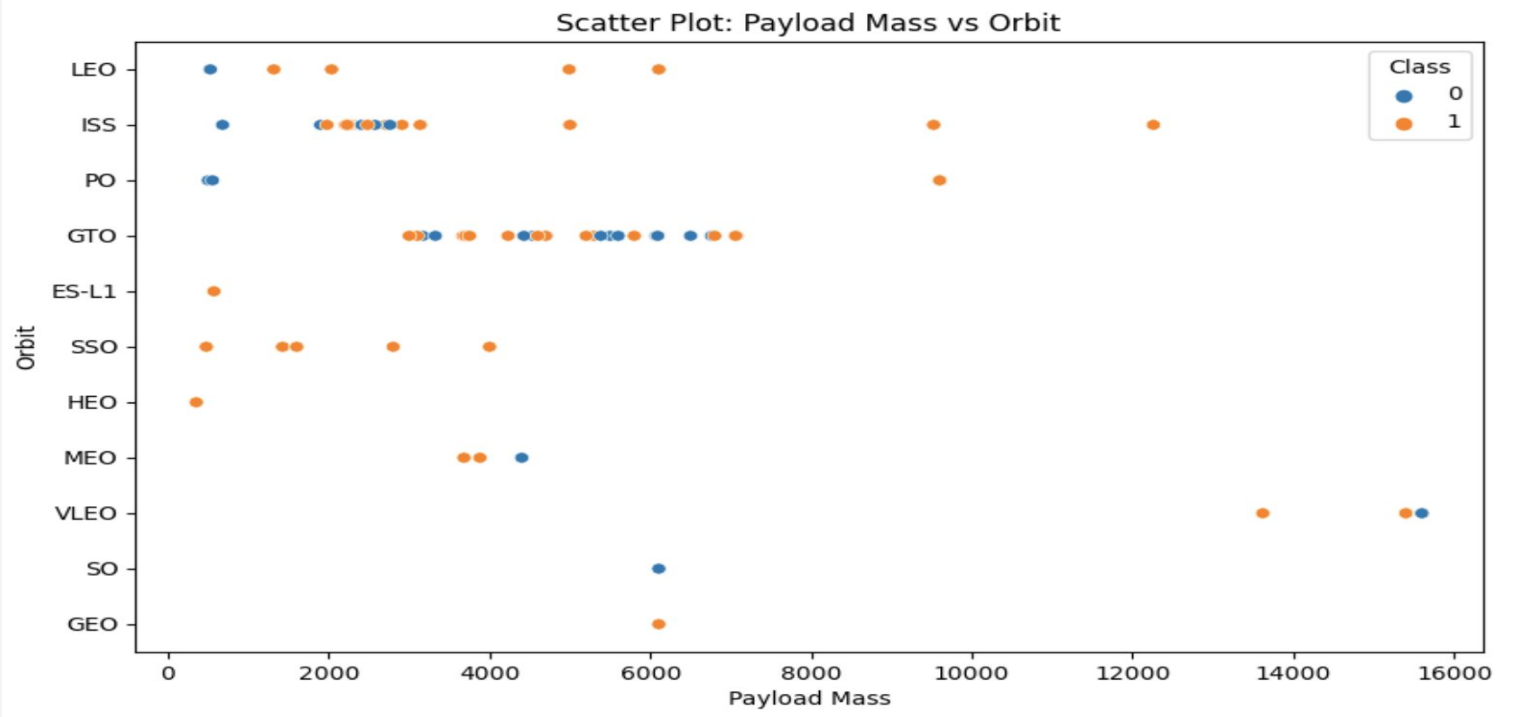
```
In [5]: df.isnull().sum()/df.count()*100
```

```
Out[5]: FlightNumber      0.000
        Date              0.000
        BoosterVersion    0.000
        PayloadMass       0.000
        Orbit             0.000
        LaunchSite        0.000
        Outcome           0.000
        Flights           0.000
        GridFins          0.000
        Reused            0.000
        Legs              0.000
        LandingPad       40.625
        Block             0.000
        ReusedCount       0.000
        Serial            0.000
        Longitude         0.000
        Latitude          0.000
        dtype: float64
```

More on : https://github.com/Vijay190899/Project/blob/main/1_Data_Wrangling.ipynb

EDA with Data Visualization

- Scatterplots, Line Charts and Histograms were used to visualise correlations between the attributes of the dataset. Shown below is a scatterplot that displays the Payload Mass vs Orbit's correlation on the success of the landing;



More on : https://github.com/Vijay190899/Project/blob/main/2_EDA%2BVisualisation.ipynb

EDA with SQL

The SQL queries that have been used in the analysis can be summarised into the following;

Launch Sites Analysis:

- Explored unique launch site names and identified records starting with 'CCA'.
- Calculated total payload mass carried by boosters launched by NASA (CRS).

Booster Performance:

- Analyzed payload mass trends, including average for booster version F9 v1.1.
- Identified boosters with successful drone ship landings and specific payload criteria.

Mission Outcomes:

- Counted and categorized successful and failure mission outcomes.
- Ranked landing outcomes during a specified time period.

Temporal Analysis:

- Determined the date of the first successful ground pad landing.
- Analyzed mission outcomes, booster versions, and launch sites month-wise for 2015.

More on : https://github.com/Vijay190899/Project/blob/main/2_EDA_WITH_SQL.ipynb

EDA with SQL

Eg : An actual query performed to rank the outcomes between 4th June,2010 and 20th March,2017.

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

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

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

More on : https://github.com/Vijay190899/Project/blob/main/2_EDA_WITH_SQL.ipynb

Predictive Analysis (Classification)

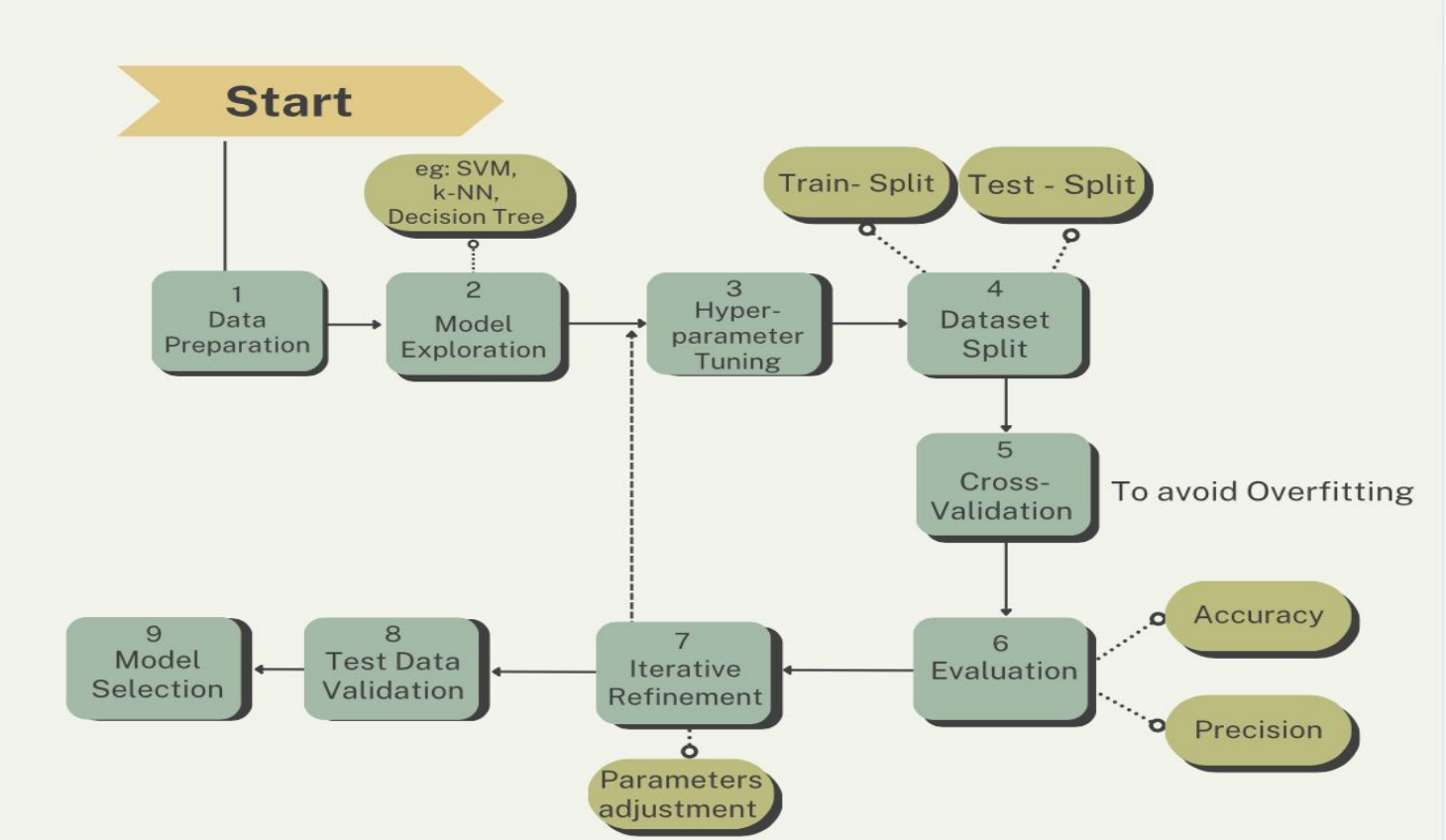
The following methods were used to achieve accurate predictive analysis;

- **Model Exploration & Selection:**
  - Explored Logistic Regression, SVM, Decision Tree, KNN.
  - **Hyperparameter Tuning:** Used GridSearchCV for fine-tuning.
- **Evaluation & Comparison:**
  - Trained models with cross-validation.
  - Evaluated using metrics (accuracy, precision, recall, F1-score).
  - Chose model with balanced performance metrics.
- **Iterative Refinement & Testing:**
  - Iterated for better results (hyperparameters, features).
  - Final model evaluated on independent test data.

More on : https://github.com/Vijay190899/Project/blob/main/4_PDA_with_ML.ipynb

Predictive Analysis (Classification)

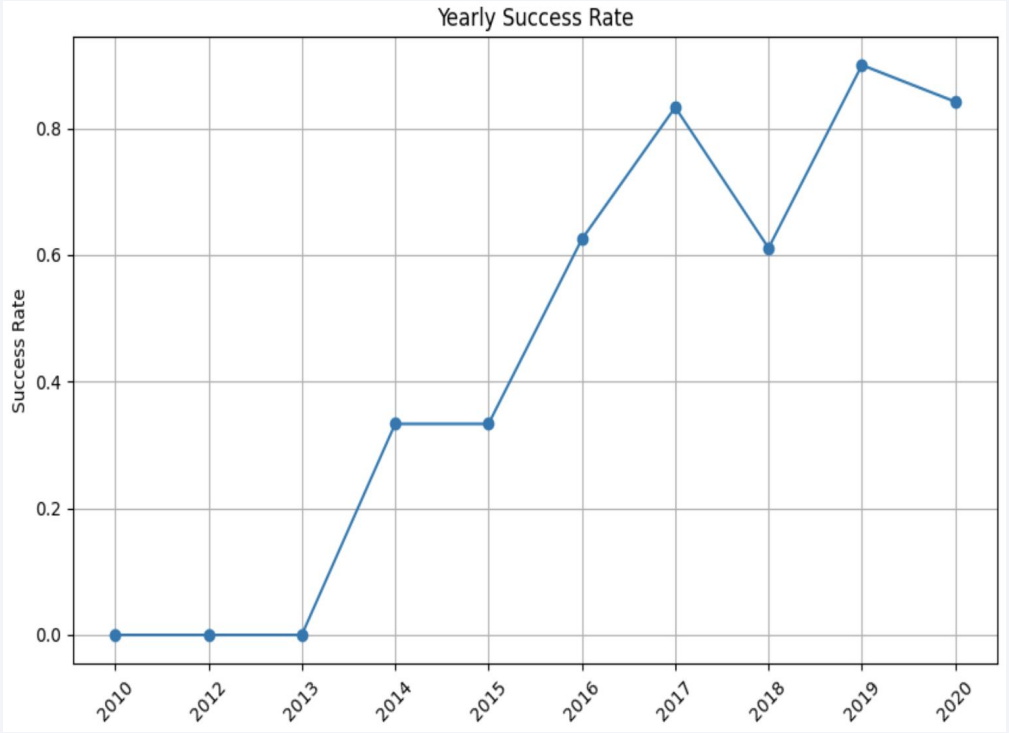
The flowchart below explains the process;



More on : https://github.com/Vijay190899/Project/blob/main/4_PDA_with_ML.ipynb

Results

- Exploratory data analysis showed that the success rate of landings kept increasing since 2013 till 2020.
- Predictive analysis showed that Decision Tree was the best performing model.



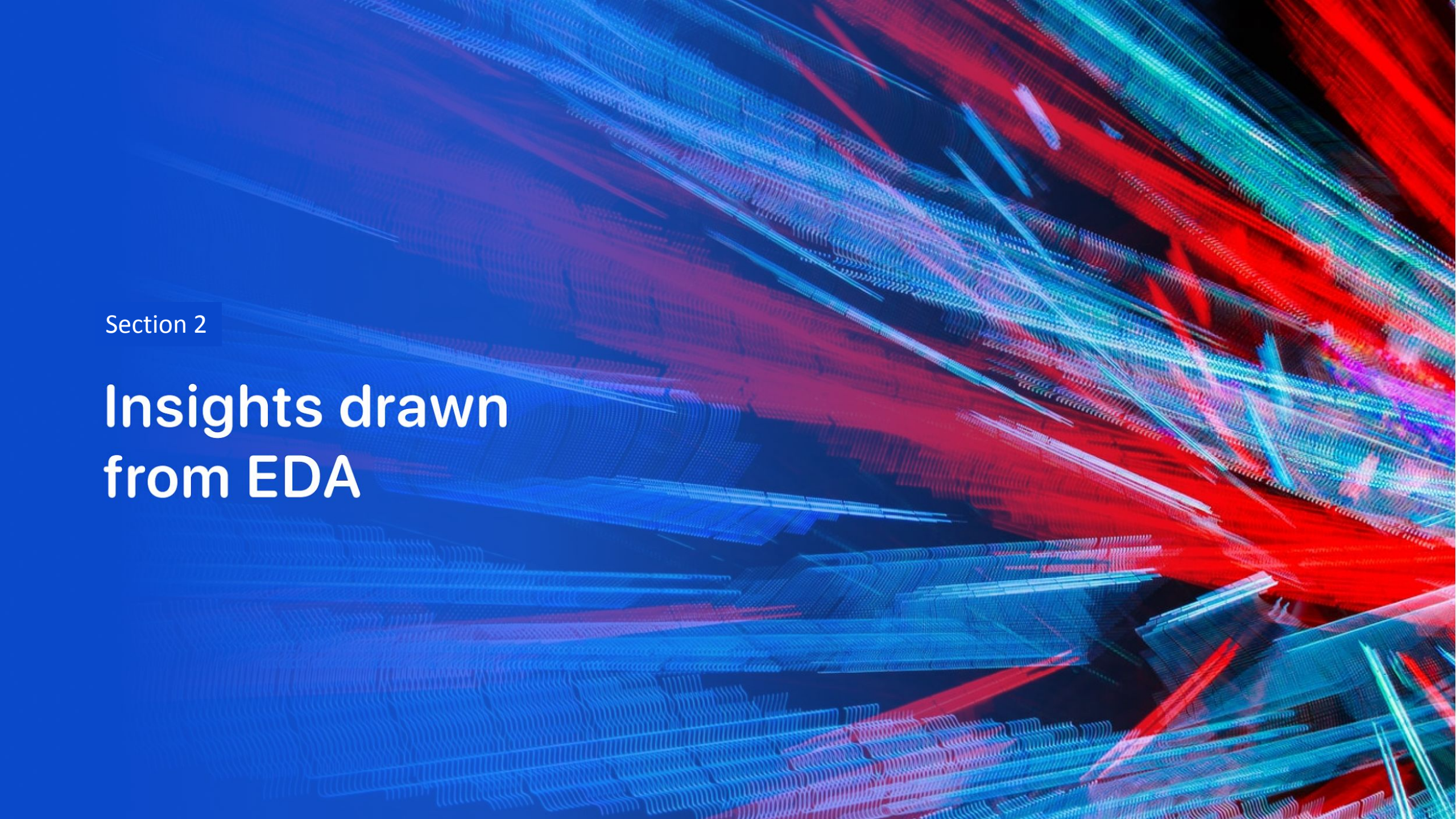

```
best_scores = {
    'Logistic Regression': logreg_cv.best_score_,
    'SVM': svm_cv.best_score_,
    'Decision Tree': tree_cv.best_score_,
    'KNN': knn_cv.best_score_
}
best_method = max(best_scores, key=best_scores.get)

print("Best performing method:", best_method)
print("Best score:", best_scores[best_method])
```

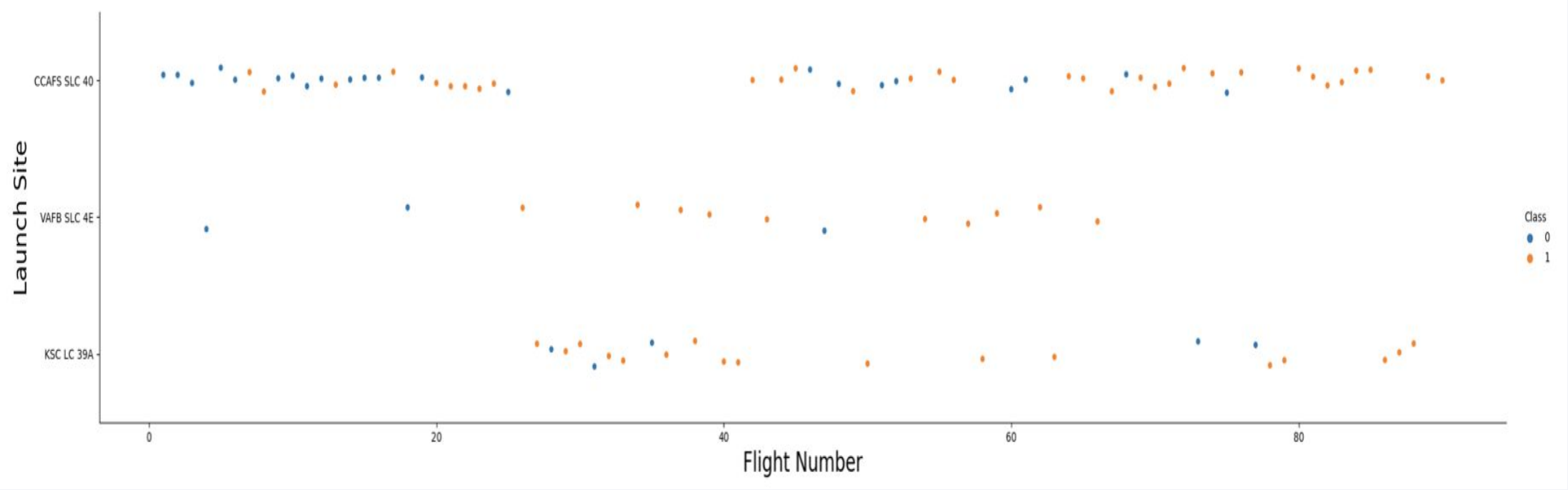
```
Best performing method: Decision Tree
Best score: 0.8767857142857143
```

Section 2

Insights drawn from EDA

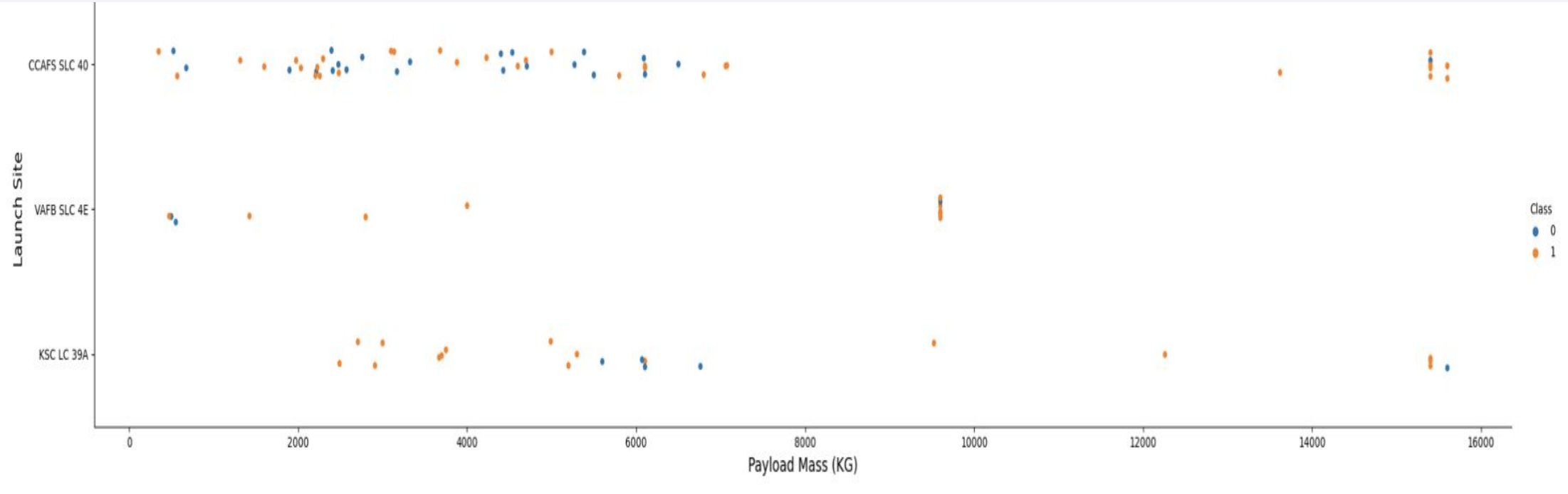


Flight Number vs. Launch Site



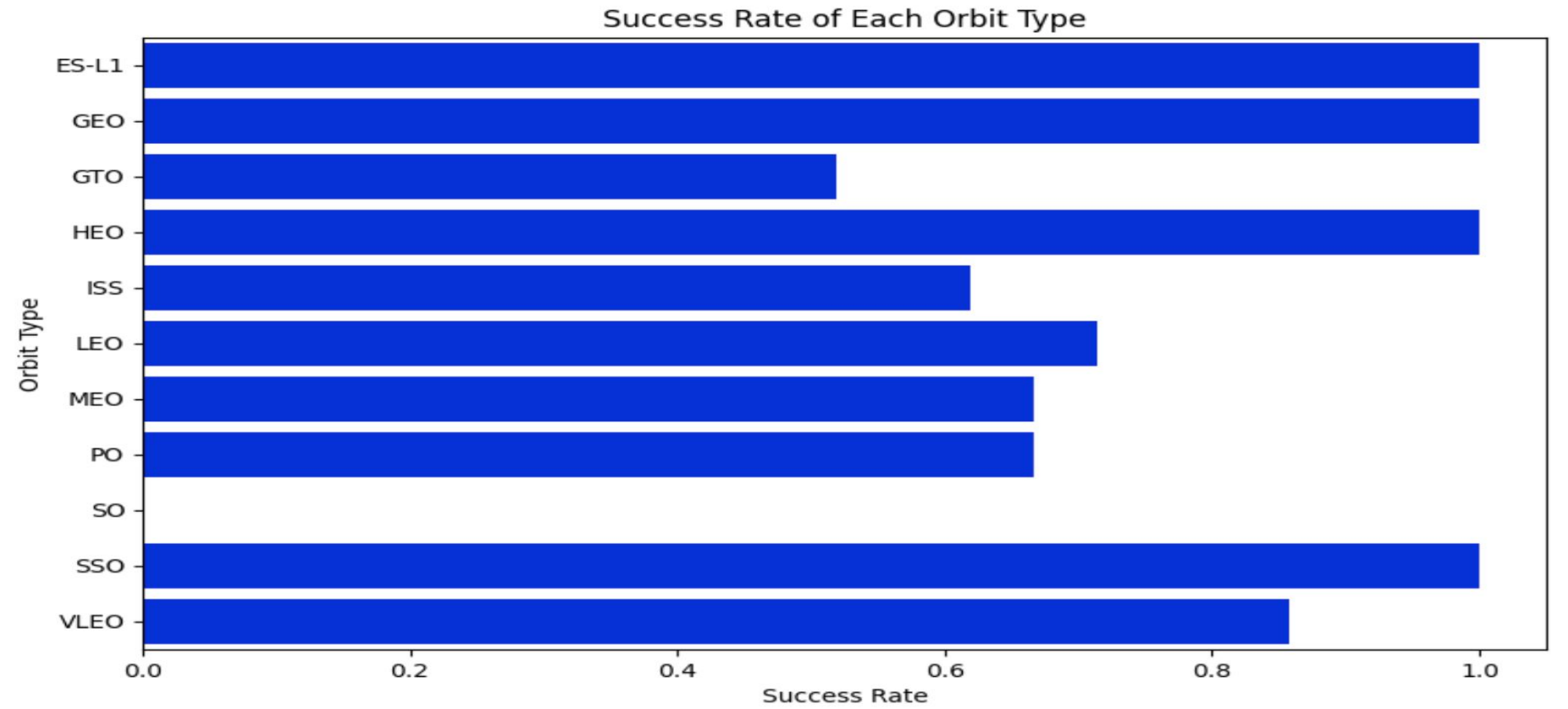
CCAFS SLC 40 has had more successful launches on the flights later on.

Payload vs. Launch Site



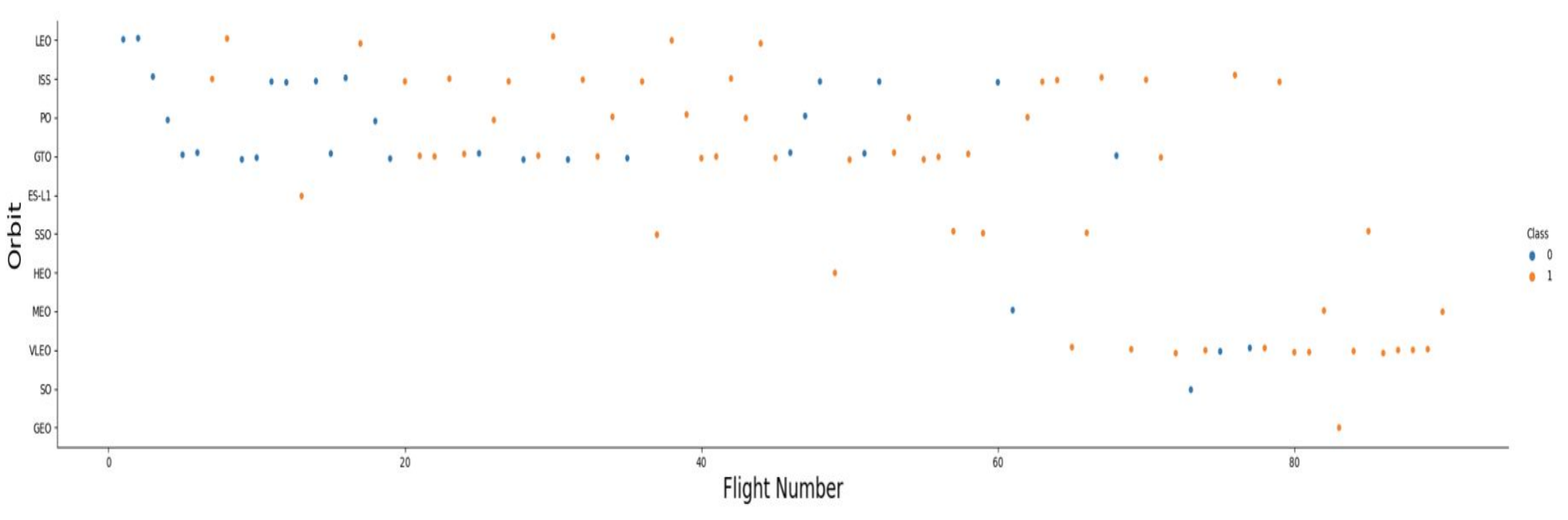
VAFB-SLC launchsite there are no rockets launched for heavy-payload mass(greater than 10000).

Success Rate vs. Orbit Type



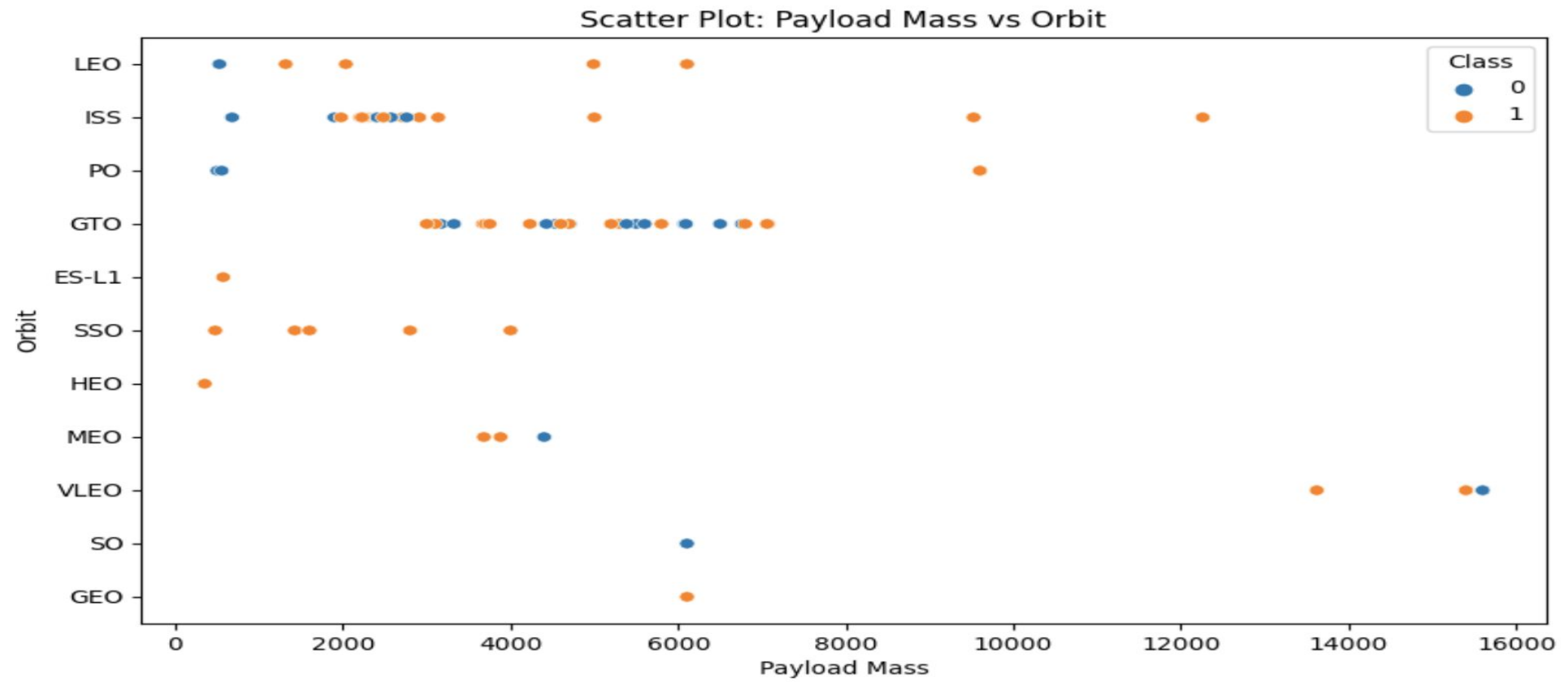
ES-L1,Geo-Synchronous Orbit (GEO),Highly Elliptical Orbit (HEO) and Sun-Synchronous Orbit (SSO) have the highest success rates.

Flight Number vs. Orbit Type



- Low Earth Orbit's (LEO) success is directly proportional to the number of flights.
- Geostationary Transfer Orbit (GTO) has no observable patterns.

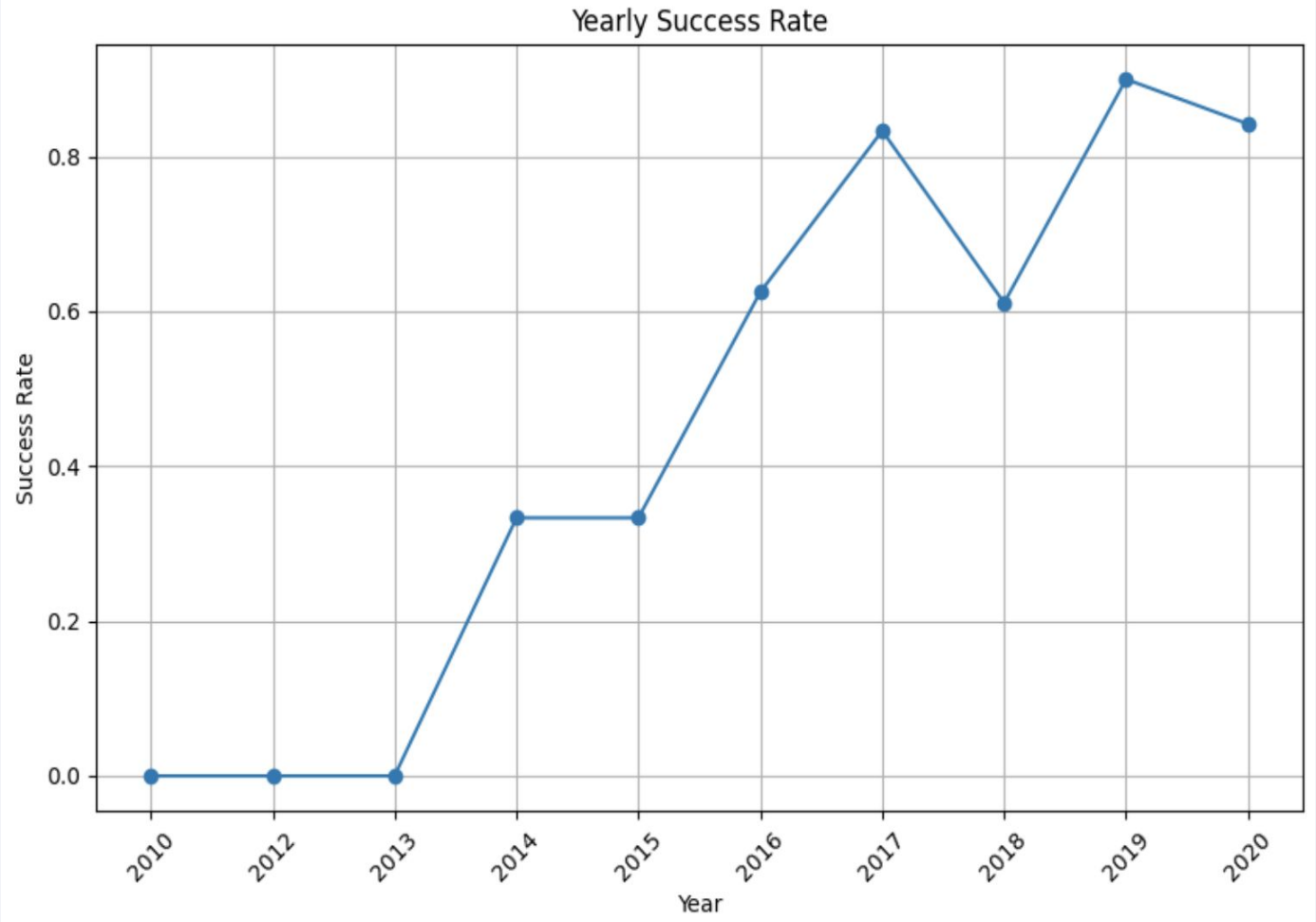
Payload vs. Orbit Type



- Polar, LEO and ISS have more successful landings with heavier payloads.
- GTO,once again, has no observable patterns.

Launch Success Yearly Trend

The tend shows that the success rate has been increasing on an uptrend since 2013 to 2020.



All Launch Site Names

- The query was made to output all unique launch sites' names, which outputs the following names;
  - CCAFS LC-40
  - VAFB SLC-4E
  - KSC LC-39A
  - CCAFS SLC-40

```
%%sql select distinct Launch_Site
from SPACEXTABLE;
```

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

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

Launch Site Names Begin with 'CCA'

- The following query was made to output all the details of launch sites whose names begin with ‘CCA’.
- The like clause was used to make a comparison to the strings inside ‘% %’.
- The limit clause is used to limit the number of outputs to 5.

```
%%sql
select *
from SPACEXTABLE
where Launch_Site like '%CCA%'
limit 5;
```

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

Date	Time (UTC)	Booster_Version	Launch_Site	Payload	PAYLOAD_MASS__KG_	Orbit	Customer	Mission_Outcome	Landing_
2010-04-06	18:45:00	F9 v1.0 B0003	CCAFS LC-40	Dragon Spacecraft Qualification Unit	0	LEO	SpaceX	Success	Failure (p
2010-08-12	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	07:44:00	F9 v1.0 B0005	CCAFS LC-40	Dragon demo flight C2	525	LEO (ISS)	NASA (COTS)	Success	N
2012-08-10	00:35:00	F9 v1.0 B0006	CCAFS LC-40	SpaceX CRS-1	500	LEO (ISS)	NASA (CRS)	Success	N
2013-01-	15:10:00	F9 v1.0 B0007	CCAFS LC-40	SpaceX CRS-2	677	LEO (ISS)	NASA (CRS)	Success	N

Total Payload Mass

- The following query was made to output the total payload mass carried by the boosters launched by NASA.

- The sum() function was used to calculate the sum of all the payload masses.

```
%%sql
select customer,sum(PAYLOAD_MASS__KG_) as TotalPayloadMass
from SPACEXTABLE
where CUSTOMER = 'NASA (CRS)';
```

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

Customer	TotalPayloadMass
NASA (CRS)	45596

Average Payload Mass by F9 v1.1

- The following query was made to calculate the average payload mass carried by booster version F9 v1.1.

- The avg() function was used to calculate the mean value of the payload masses.

```
%%sql
select Booster_version,avg(PAYLOAD_MASS__KG_) as AveragePayloadMass
from SPACEXTABLE
where BOOSTER_VERSION = 'F9 v1.1';
```

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

Booster_Version	AveragePayloadMass
F9 v1.1	2928.4

First Successful Ground Landing Date

- The following query was made to find the date of the first successful landing outcome on ground pad.
- The min() function was used to find the earliest date.
- The and clause was used to combine two conditions.

```
%%sql
select min(DATE) as FirstSuccessfulLandingDate, Landing_outcome
from SPACEXTABLE
where Mission_outcome = 'Success' AND landing_outcome like '%ground%';
```

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

FirstSuccessfulLandingDate	Landing_Outcome
2015-12-22	Success (ground pad)

Successful Drone Ship Landing with Payload between 4000 and 6000

- The following query was made to list the names of boosters which have successfully landed on drone ship and had payload mass greater than 4000 but less than 6000.

- The < and > operators were used in the conditions to set a limit.

```
%%sql
SELECT BOOSTER_VERSION, LANDING_OUTCOME
FROM SPACEXTABLE
WHERE LANDING_OUTCOME = 'Success (drone ship)'
    AND PAYLOAD_MASS__KG_ > 4000
    AND PAYLOAD_MASS__KG_ < 6000;
```

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

Booster_Version	Landing_Outcome
F9 FT B1022	Success (drone ship)
F9 FT B1026	Success (drone ship)
F9 FT B1021.2	Success (drone ship)
F9 FT B1031.2	Success (drone ship)

Total Number of Successful and Failure Mission Outcomes

- The following query was made to calculate the total number of successful and failure mission outcomes.
- The count() function was to display the total number of entries in the dataset.
- the group by clause was used to list the rows according to the “Mission_Outcome” column.

```
%%sql
SELECT MISSION_OUTCOME, COUNT(*) AS Count
FROM SPACEXTABLE
GROUP BY MISSION_OUTCOME;
```

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

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

Boosters Carried Maximum Payload

- The following query was used to list the names of the booster which have carried the maximum payload mass.
- A subquery was used to specify the maximum payload mass.

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

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

Booster_Version	PAYLOAD_MASS__KG_
F9 B5 B1048.4	15600
F9 B5 B1049.4	15600
F9 B5 B1051.3	15600
F9 B5 B1056.4	15600
F9 B5 B1048.5	15600
F9 B5 B1051.4	15600
F9 B5 B1049.5	15600
F9 B5 B1060.2	15600
F9 B5 B1058.3	15600
F9 B5 B1051.6	15600
F9 B5 B1060.3	15600
F9 B5 B1049.7	15600

2015 Launch Records

- The following query was made to list the failed landings in drone ship, their booster versions, and launch site names for the year 2015.

- Case expression was used to perform conditional logic, paired with when clause.

```
%%sql
SELECT
    CASE
        WHEN substr('DATE', 4, 2) = '01' THEN 'January'
        WHEN substr('DATE', 4, 2) = '02' THEN 'February'
        WHEN substr('DATE', 4, 2) = '03' THEN 'March'
        WHEN substr('DATE', 4, 2) = '04' THEN 'April'
        WHEN substr('DATE', 4, 2) = '05' THEN 'May'
        WHEN substr('DATE', 4, 2) = '06' THEN 'June'
        WHEN substr('DATE', 4, 2) = '07' THEN 'July'
        WHEN substr('DATE', 4, 2) = '08' THEN 'August'
        WHEN substr('DATE', 4, 2) = '09' THEN 'September'
        WHEN substr('DATE', 4, 2) = '10' THEN 'October'
        WHEN substr('DATE', 4, 2) = '11' THEN 'November'
        WHEN substr('DATE', 4, 2) = '12' THEN 'December'
    END AS MonthName,
    LANDING_OUTCOME,
    BOOSTER_VERSION,
    LAUNCH_SITE
FROM SPACEXTABLE
WHERE substr('DATE', 7, 4) = '2015'
    AND LANDING_OUTCOME = 'Failure (drone ship)';
```

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

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

- The following query was made to rank the count of landing outcomes [such as Failure (drone ship) or Success (ground pad)] between 4th June,2010 and 20th March,2017,in descending order.

- The order by clause (along with desc clause) was used to list the count in descending order.

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

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

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

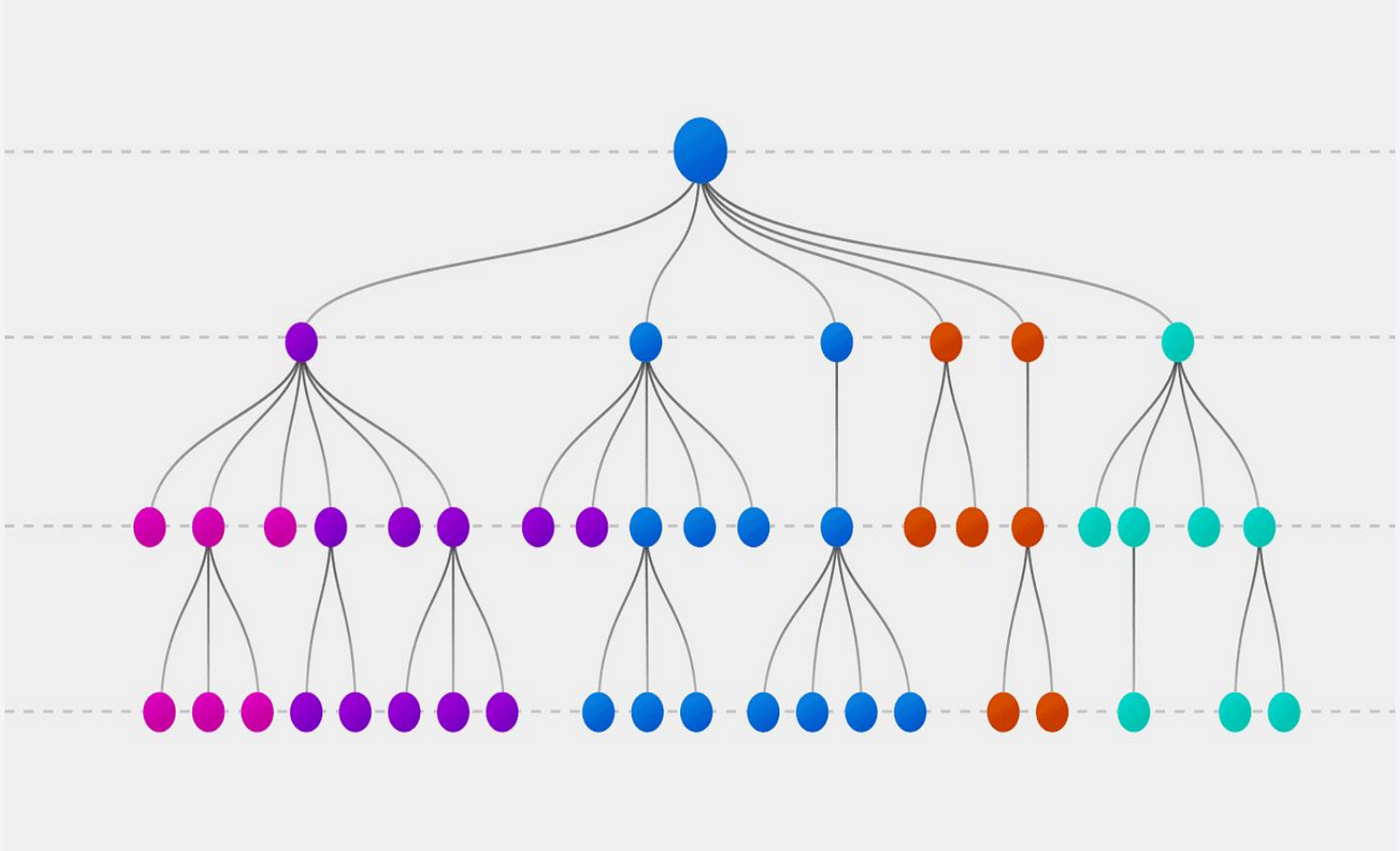
Section 5

Predictive Analysis (Classification)

Classification Accuracy

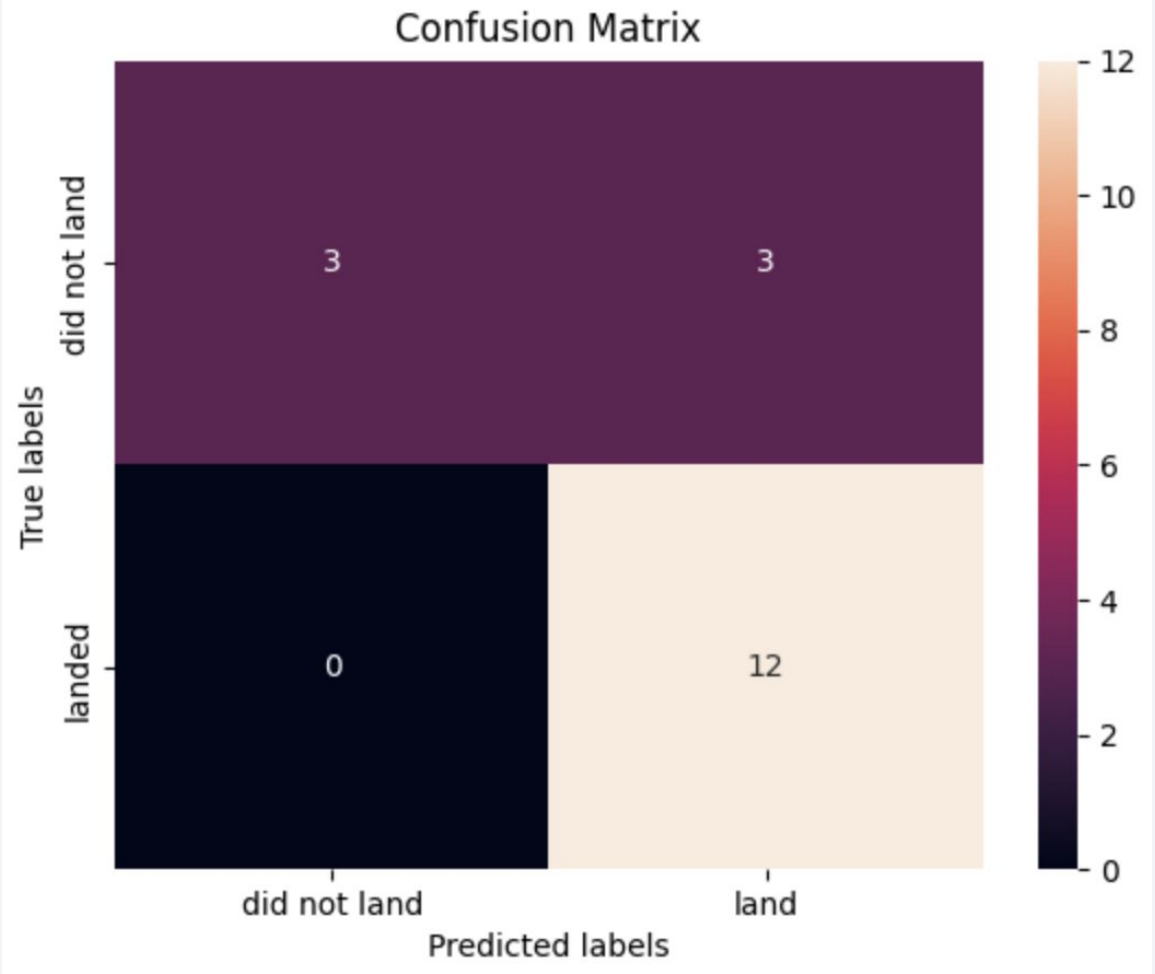
The Decision Tree classifier has come out the most accurate amongst all the evaluated models with an accuracy of 87.67%.

More on: https://github.com/Vijay190899/Project/blob/main/4_PDA_with_ML.ipynb



Confusion Matrix

- As evident, the matrix displays;
  - A very high amount of True Positives.
  - Doesn't predict any False Negatives.



Conclusions

The analysis of SpaceX launches and the subsequent predictive modeling process have provided valuable insights into the performance of rocket launches, factors influencing successful landings, and the accuracy of classification models.

Throughout this project, we have journeyed through several stages, from data exploration to model evaluation, to arrive at actionable conclusions.

Key Findings

Exploratory Data Analysis (EDA):

- Our EDA highlighted significant trends and patterns in the SpaceX launch data.
- We observed that a majority of launches were from the Cape Canaveral Air Force Station (CCAFS), reflecting SpaceX's strategic partnership with the location.

Landing Outcomes and Success Rates:

- We identified that successful landings were more frequent on ground pads than on drone ships.
- The success rate was higher for launches using the Falcon 9 v1.1 booster version.

Predictive Analysis:

- We employed classification models to predict the outcome of SpaceX launches.
- The Decision Tree model exhibited the highest accuracy, successfully classifying launch outcomes based on features such as booster version, launch site, and payload mass.

Appendix

Data Sources

SpaceX Launch Data:

- Source: SpaceX API
- Description: The primary dataset containing details of SpaceX launches, including launch site, outcome, booster version, and more.
- Format: JSON

Exploratory Data Analysis (EDA)

Launch Sites Analysis:

- Overview of launch site distribution and its correlation with landing outcomes.
- Visualizations: Bar charts, Scatterplots.

Landing Outcomes Analysis:

- Analysis of landing outcomes, including success rates on ground pads and drone ships.
- Visualizations: Line Charts, Scatterplots.

Booster Version Analysis:

- Exploration of the relationship between booster versions and landing outcomes.
- Visualizations: Scatterplots, histograms.

Predictive Modeling

Data Preprocessing:

- Feature engineering, data transformation, handling missing values.
- Techniques: One-hot encoding, standard scaling.

Model Selection:

- Evaluation of multiple classification models, including Logistic Regression, Support Vector Machine, Decision Tree, and K-Nearest Neighbors.
- Metrics: Accuracy, confusion matrices.

Decision Tree Model:

- Detailed analysis of the best-performing Decision Tree model.

Tools

Data Collection:

- Python code for fetching SpaceX launch data using the SpaceX API.

Data Wrangling:

- Pandas for cleaning, transforming, and preparing the data for analysis.

Exploratory Data Analysis:

- Pandas, Matplotlib,Seaborn and SQL for generating visualizations and conducting exploratory analysis.

Predictive Modeling:

- Python(Scikit-learn) for implementing and evaluating classification models.

Thank you!