IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

Gauthier SIMON
23/07/2025

Outline

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

Executive Summary

- Summary of methodologies
- Summary of all results

Introduction

- Project background and context
- Problems you want to find answers

Section 1

Methodology

Methodology

Executive Summary

- Data collection methodology:
  - Describe how data was collected
- Perform data wrangling
  - Describe how data was processed
- Perform exploratory data analysis (EDA) using visualization and SQL
- Perform interactive visual analytics using Folium and Plotly Dash
- Perform predictive analysis using classification models
  - How to build, tune, evaluate classification models

Data Collection

- Data sourced from IBM Skills Network datasets
- Spacex Rest API calls for mission data
- Web Scraping (Html tables)
- Data loaded into Pandas DataFrames for analysis

Data Collection – SpaceX API

- Used requests library to access SpaceX API endpoints.
- Retrieved launch data, payloads, and outcomes.
- Example: df = pd.read_csv('https://cf-courses-data.s3.us.cloud-object-storage.appdomain.cloud/IBM-DS0321EN-SkillsNetwork/datasets/spacex_launch_dash.csv')

Start
-> Request SpaceX API Endpoint
-> Receive JSON/CSV response
-> Parse and convert to dataframe
-> End

Data Collection - Scraping

- Scraped HTML tables containing SpaceX launch records.
- Used pandas' read_html() or BeautifulSoup to parse tables.
- Extracted mission dates, payloads, launch sites, customers, outcomes.

Start
->Download HTML Page
-> Parse HTML for table
-> Extract rows/columns
-> Store in DataFrame
-> End

Data Wrangling

- Data cleaning: null value removal, type conversion.
- Feature engineering (e.g., extracting year from date).
- Label encoding/categorical feature transformation.
- Normalization/scaling of numerical features.

EDA with Data Visualization

- Pie chart: Success vs. Failure rate per launch site.
- Scatter plot: Payload Mass vs. Launch Success, colored by booster version.
- Catplot (Seaborn): Class distribution by Launch Site, Orbit, Flight Number.
- Line plot: Average success rate over years.

Why these charts?

- Pie charts visualize categorical proportions (success/failure).
- Scatter plots explore correlation between payload and success.
- Catplots reveal categorical variable distributions and relationships.
- Line plots show time trends (success rate evolution).

EDA with SQL

- Created table from cleaned records:
  CREATE TABLE spacextable AS SELECT * FROM SPACEXTBL WHERE Date IS NOT NULL;
- Counted successful and failed missions:
  SELECT COUNT(CASE WHEN mission_outcome LIKE 'Success%' THEN 1 END) AS Total_No_of_Success, ...
- Summed payload for NASA customers:
  SELECT SUM(payload_mass__kg_) AS total_payload FROM SPACEXTBL WHERE customer LIKE 'NASA%';
- Average payload by booster version:
  SELECT AVG(payload_mass__kg_) AS avg_payload FROM SPACEXTBL WHERE Booster_Version LIKE 'F9 v1.1%';
- Listed all launch sites:
  SELECT DISTINCT Launch_Site FROM SPACEXTBL;
- Filtered launches by site:
  SELECT * FROM SPACEXTBL WHERE Launch_Site LIKE 'CCAFS SLC-40';
- Maximum payload with subquery:
  SELECT Booster_Version, PAYLOAD_MASS__KG_ FROM SPACEXTBL WHERE ...
- Grouped landing outcomes:
  SELECT Landing_Outcome, COUNT(*) FROM SPACEXTBL GROUP BY Landing_Outcome ORDER BY Outcome_Count DESC;

Build an Interactive Map with Folium

- Markers for each launch site (latitude/longitude).
- Circles for launch site clusters.
- Lines possibly for flight trajectories.

Why?

- Markers pinpoint launch sites for geographic context.
- Circles highlight site density or cluster zones.
- Lines (if used) visualize flight paths.

Build a Dashboard with Plotly Dash

- Dropdown for launch site selection.
- Pie chart for success rates (all sites/single site).
- Range slider for payload mass selection.
- Scatter chart for payload-success correlation.

Interactions:

- Select site to filter plots dynamically.
- Adjust payload slider to zoom in on specific payload ranges.

Why?

- Interactive controls allow users to drill down on specific launch sites and payloads, enhancing data exploration and insight.

Predictive Analysis (Classification)

- Multiple models built: Logistic Regression, SVM, Decision Tree, KNN.
- Data split into train/test sets.
- Feature scaling/normalization applied.
- Hyperparameter tuning with GridSearchCV.
- Model evaluation via accuracy, confusion matrix.
- Best performing model selected by cross-validated score.

[Wrangled Data] -> [Feature Scaling & Train-Test Split] -> [Model Training: Logistic Regression / SVM / Decision Tree / KNN] -> [Hyperparameter Tuning (GridSearchCV)] ->[Model Evaluation (Accuracy, Confusion Matrix)] ->[Select Best Model] ->[Predict & Analyze Results]

Results

Exploratory Data Analysis Results

- Most launches were successful, with site-specific variation.
- Payload mass and booster version impact success rates.
- NASA and SpaceX were frequent customers with high payloads.

Predictive Analysis Results

- Decision Tree and KNN provided best accuracy after tuning.
- Best hyperparameters found via GridSearchCV.
- Predictive model could classify future launches with confidence scores up to 87.5% (Decision Tree).

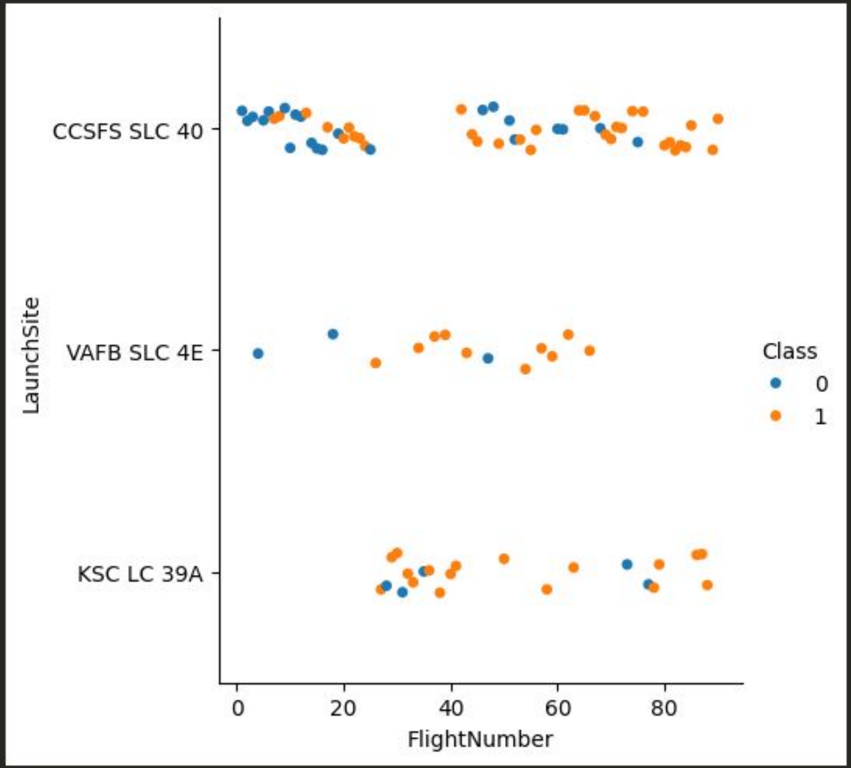
Section 2

Insights drawn from EDA

Flight Number vs. Launch Site

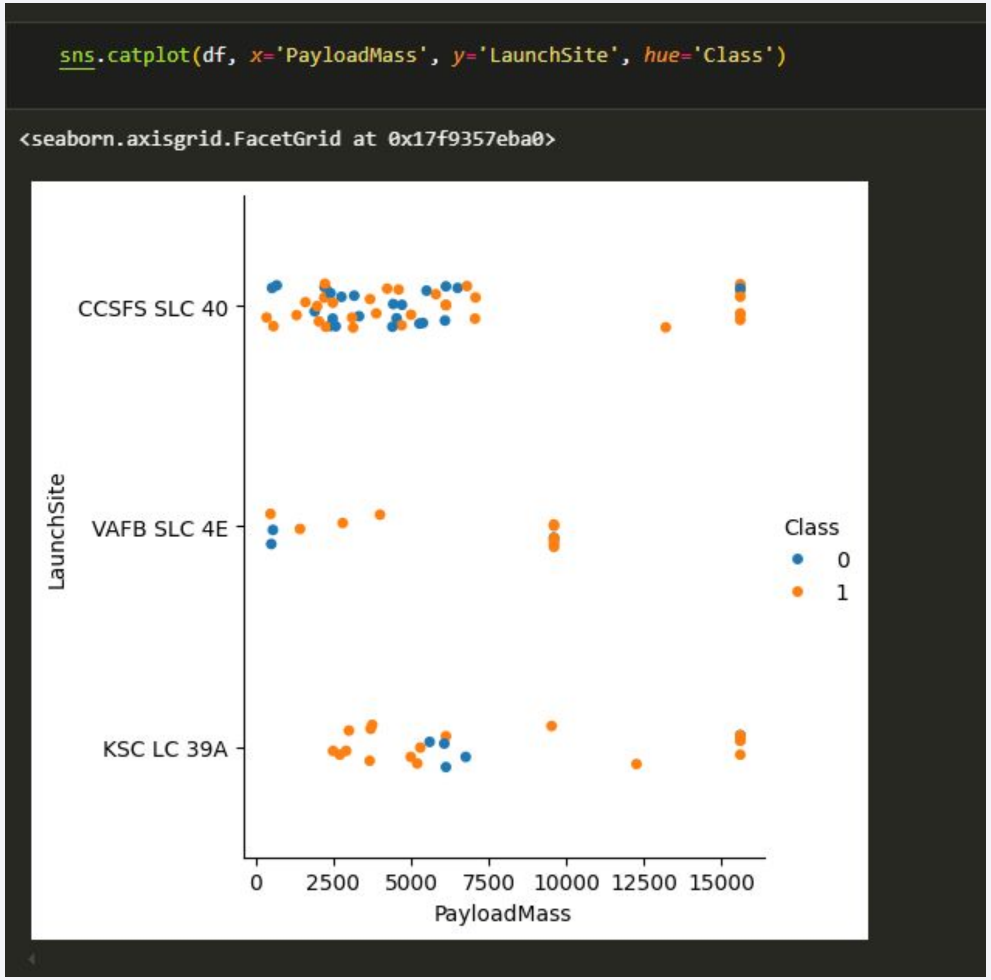
- Show a scatter plot of Flight Number vs. Launch Site

```
sns.catplot(df, x='FlightNumber', y='LaunchSite', hue='Class')
plt.show()
```



Payload vs. Launch Site

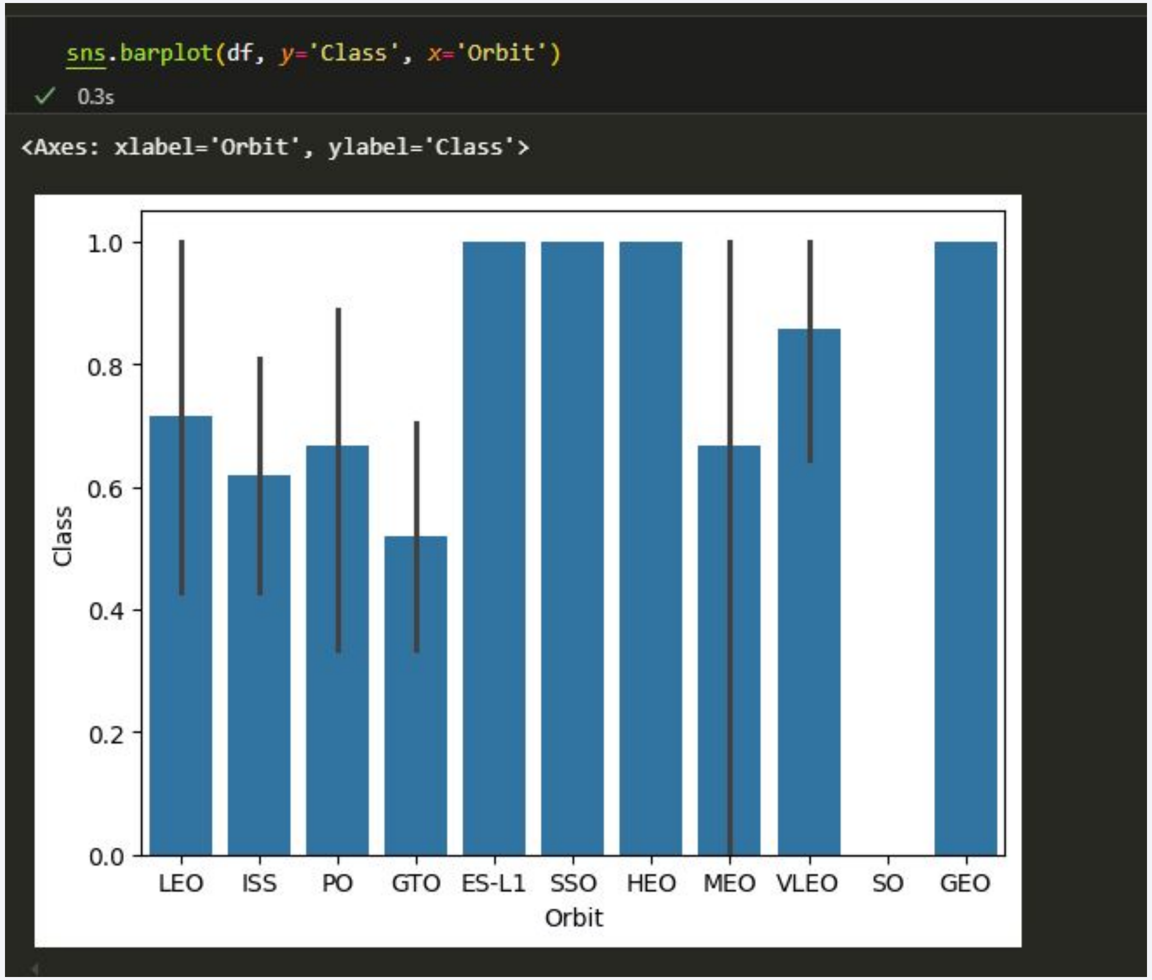
- Show a scatter plot
  of Payload vs. Launch Site



Success Rate vs. Orbit Type

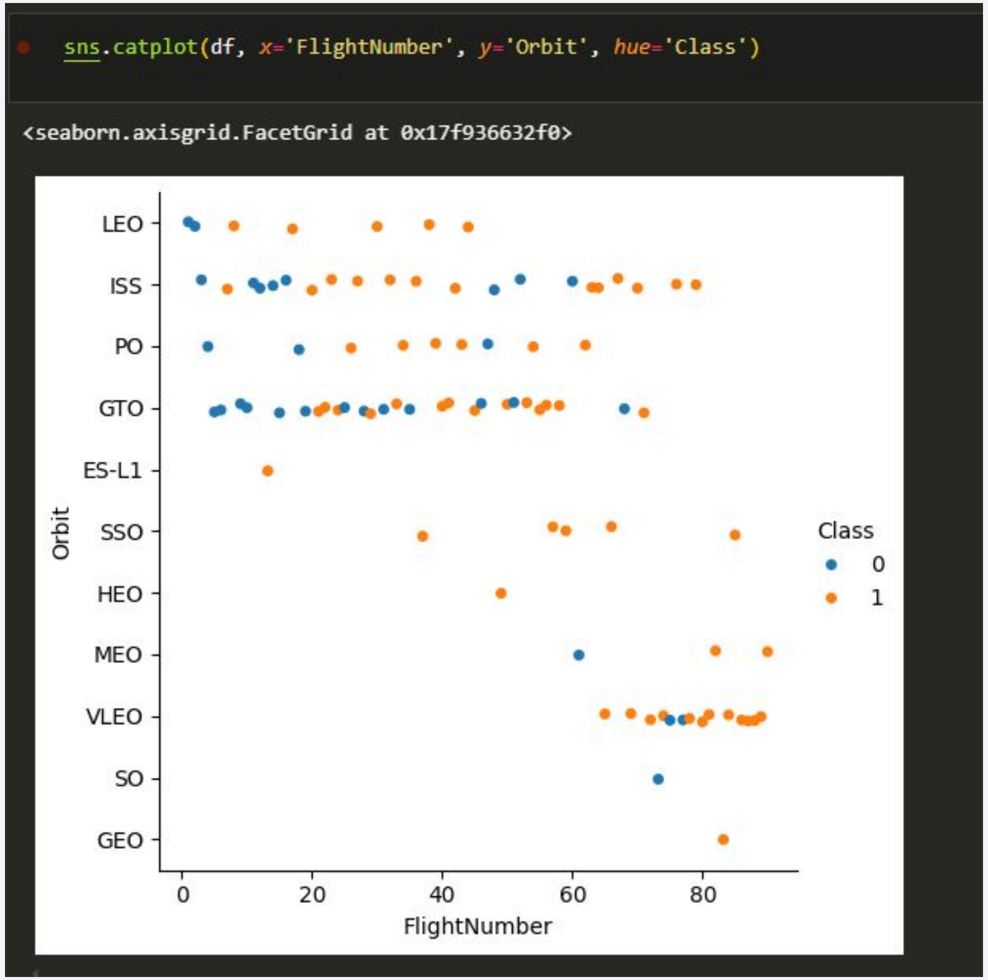
- Show a bar chart for the success rate of each orbit type

- In percentage of Success by Orbit type



Flight Number vs. Orbit Type

- Show a scatter point of Flight number vs. Orbit type
- Show the screenshot of the scatter plot with explanations



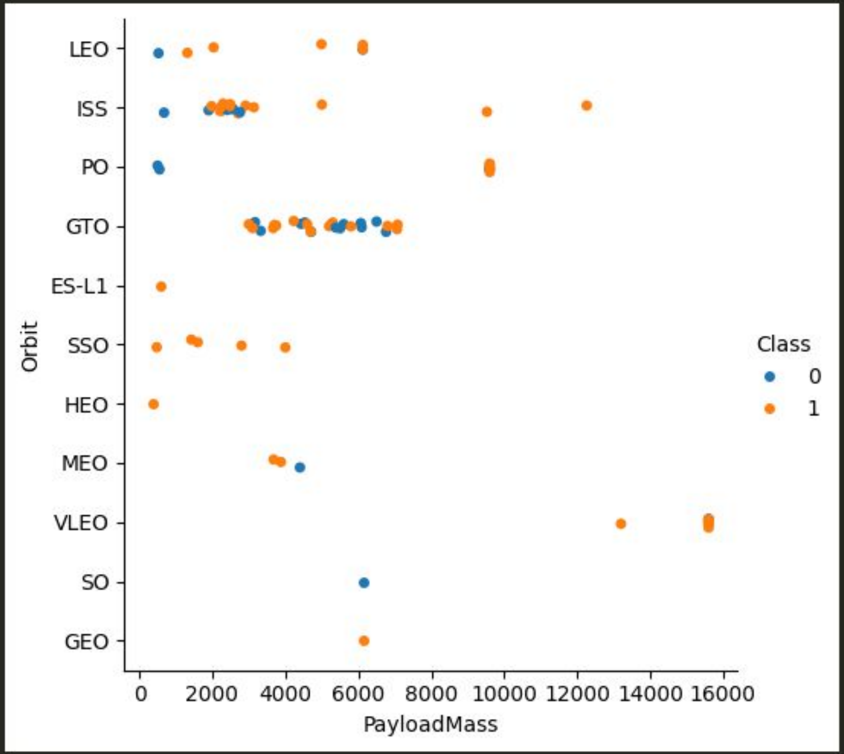
Payload vs. Orbit Type

- Show a scatter point of payload vs. orbit type

- Show the screenshot of the scatter plot with explanations

```
sns.catplot(df, x='PayloadMass', y='Orbit', hue='Class')
```

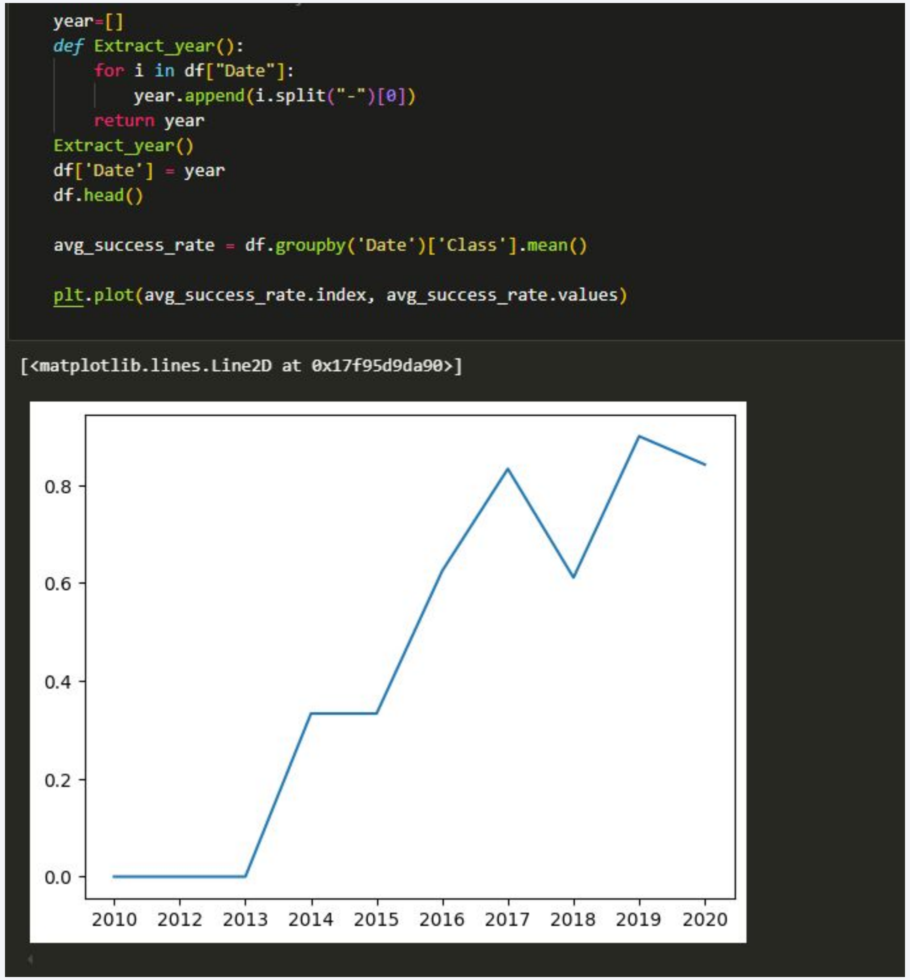
```
<seaborn.axisgrid.FacetGrid at 0x17f959fb7d0>
```



Launch Success Yearly Trend

- Show a line chart of yearly average success rate

- Show the screenshot of the scatter plot with explanations



All Launch Site Names

- SQL search to get all the launch site names so : CCAFS LC-40, VAFB SLC - 4E, CCAFS SLC-40, KSC LC-39A

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

```
 * 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'

```
%sql select * from SPACEXTBL where Launch_Site like 'CCA%' LIMIT 5;
```

✓ 0.0s

* sqlite:///my_data1.db
Done.

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

Total Payload Mass

- Calculate the total payload carried by boosters from NASA

```
%sql select sum(payload_mass__kg_) as total_payload from SPACEXTBL where customer like 'NASA%';
```

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

total_payload
99980

Average Payload Mass by F9 v1.1

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

```
%sql select avg(payload_mass__kg_) as avg_payload from SPACEXTBL where Booster_Version like 'F9 v1.1%';
```

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

avg_payload
2534.6666666666665

First Successful Ground Landing Date

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

```
%sql select  min(date(Date)) as Date from SPACEXTBL where Mission_Outcome like 'Success';
```

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

Date
2010-06-04

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 Payload as Booster_name from SPACEXTBL where mission_outcome like 'Success' and (payload_mass__kg_ >4000 and payload_mass__kg_ >6000);
```

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

Booster_name
Iridium NEXT 1
Inmarsat-5 F4
Iridium NEXT 2
Intelsat 35e
Iridium NEXT 3
Iridium NEXT 4
Hispasat 30W-6 PODSat
Iridium NEXT 5
Iridium NEXT 6 GRACE-FO 1, 2
Telstar 19V
Iridium NEXT-7
Telstar 18V / Apstar-5C
Iridium NEXT-8
Crew Dragon Demo-1, SpaceX CRS-17
Starlink v0.9, RADARSAT Constellation
AMOS-17, Starlink 1 v1.0
Starlink 1 v1.0, SpaceX CRS-19
JCSat-18 / Kacific 1, Starlink 2 v1.0
Starlink 2 v1.0, Crew Dragon in-flight abort test
Crew Dragon in-flight abort test, Starlink 3 v1.0
Starlink 3 v1.0, Starlink 4 v1.0
Starlink 4 v1.0, SpaceX CRS-20
Starlink 5 v1.0, Starlink 6 v1.0
Starlink 6 v1.0, Crew Dragon Demo-2
Crew Dragon Demo-2, Starlink 7 v1.0
Starlink 7 v1.0, Starlink 8 v1.0
Starlink 8 v1.0, SkySats-16, -17, -18, GPS III-03
Starlink 9 v1.0, SXRS-1, Starlink 10 v1.0
Starlink 10 v1.0, SkySat-19, -20, -21, SAOCOM 1B
Starlink 11 v1.0, Starlink 12 v1.0
Starlink 12 v1.0, Starlink 13 v1.0
Starlink 13 v1.0, Starlink 14 v1.0
Starlink 14 v1.0, GPS III-04
Crew-1, Sentinel-6 Michael Freilich
Starlink 15 v1.0, SpaceX CRS-21

Total Number of Successful and Failure Mission Outcomes

- Calculate the total number of successful and failure mission outcomes

```
%sql SELECT COUNT(CASE WHEN mission_outcome LIKE 'Success%' THEN 1 END) AS Total_No_of_Success, COUNT(CASE WHEN mission_outcome not like 'Success' THEN 1 END) AS Total_No_of_Fail FROM SPACEXTBL;
```

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

Total_No_of_Success	Total_No_of_Fail
100	3

Boosters Carried Maximum Payload

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

```
query = """
SELECT Booster_Version, PAYLOAD_MASS__KG_
FROM SPACEXTBL
WHERE PAYLOAD_MASS__KG_ = (
    SELECT MAX(PAYLOAD_MASS__KG_) FROM SPACEXTBL
)
"""
cur.execute(query)
results = cur.fetchall()
for row in results:
    print(row)
```

```
('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

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

```
query_2015_failures = """
SELECT
    CASE substr(Date, 6, 2)
        WHEN '01' THEN 'January'
        WHEN '02' THEN 'February'
        WHEN '03' THEN 'March'
        WHEN '04' THEN 'April'
        WHEN '05' THEN 'May'
        WHEN '06' THEN 'June'
        WHEN '07' THEN 'July'
        WHEN '08' THEN 'August'
        WHEN '09' THEN 'September'
        WHEN '10' THEN 'October'
        WHEN '11' THEN 'November'
        WHEN '12' THEN 'December'
    END AS Month_Name,
    Landing_Outcome,
    Booster_Version,
    Launch_Site
FROM SPACEXTBL
WHERE substr(Date, 0, 5) = '2015'
  AND Landing_Outcome LIKE '%Failure%'
  AND Landing_Outcome LIKE '%drone ship%'
"""
cur.execute(query_2015_failures)
failure_records_2015 = cur.fetchall()
for record in failure_records_2015:
    print(record)
```

```
('January', 'Failure (drone ship)', 'F9 v1.1 B1012', 'CCAFS LC-40')
('April', 'Failure (drone ship)', 'F9 v1.1 B1015', 'CCAFS LC-40')
```

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

- Rank the count of landing outcomes (such as Failure (drone ship) or Success (ground pad)) between the date 2010-06-04 and 2017-03-20, in descending order

```
query = """
SELECT Landing_Outcome, COUNT(*) as Outcome_Count
FROM SPACEXTBL
GROUP BY Landing_Outcome
ORDER BY Outcome_Count DESC
"""
cur.execute(query)
for row in cur.fetchall():
    print(row)
```

```
('Success', 38)
('No attempt', 21)
('Success (drone ship)', 14)
('Success (ground pad)', 9)
('Failure (drone ship)', 5)
('Controlled (ocean)', 5)
('Failure', 3)
('Uncontrolled (ocean)', 2)
('Failure (parachute)', 2)
('Precluded (drone ship)', 1)
('No attempt ', 1)
```

Section 3

Launch Sites Proximities Analysis

<Folium Map Screenshot 1>

- Replace <Folium map screenshot 1> title with an appropriate title
- Explore the generated folium map and make a proper screenshot to include all launch sites' location markers on a global map
- Explain the important elements and findings on the screenshot

<Folium Map Screenshot 2>

- Replace <Folium map screenshot 2> title with an appropriate title

- Explore the folium map and make a proper screenshot to show the color-labeled launch outcomes on the map

- Explain the important elements and findings on the screenshot

<Folium Map Screenshot 3>

- Replace <Folium map screenshot 3> title with an appropriate title

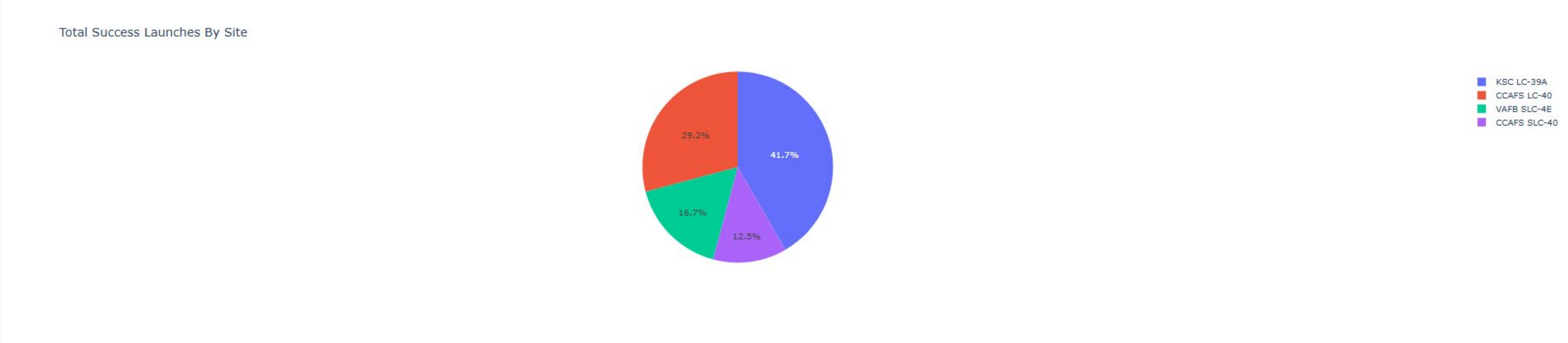
- Explore the generated folium map and show the screenshot of a selected launch site to its proximities such as railway, highway, coastline, with distance calculated and displayed

- Explain the important elements and findings on the screenshot

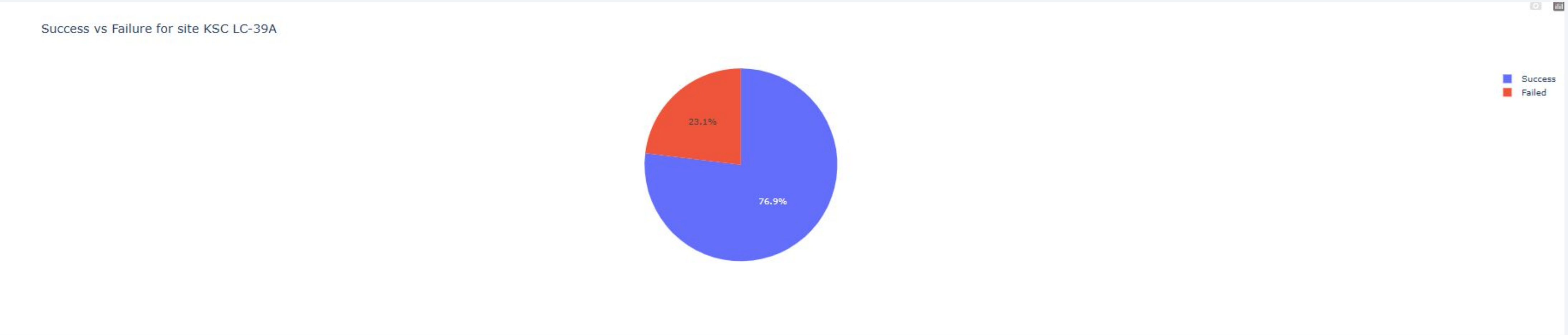
Section 4

Build a Dashboard with Plotly Dash

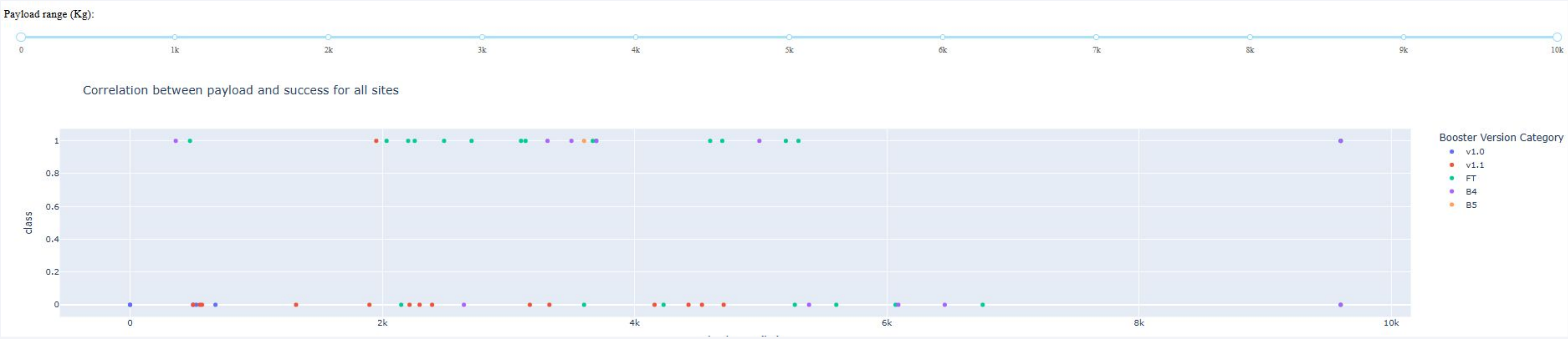
SpaceX Launch Records Dashboard



SpaceX Launch Records Dashboard



SpaceX Launch Records Dashboard



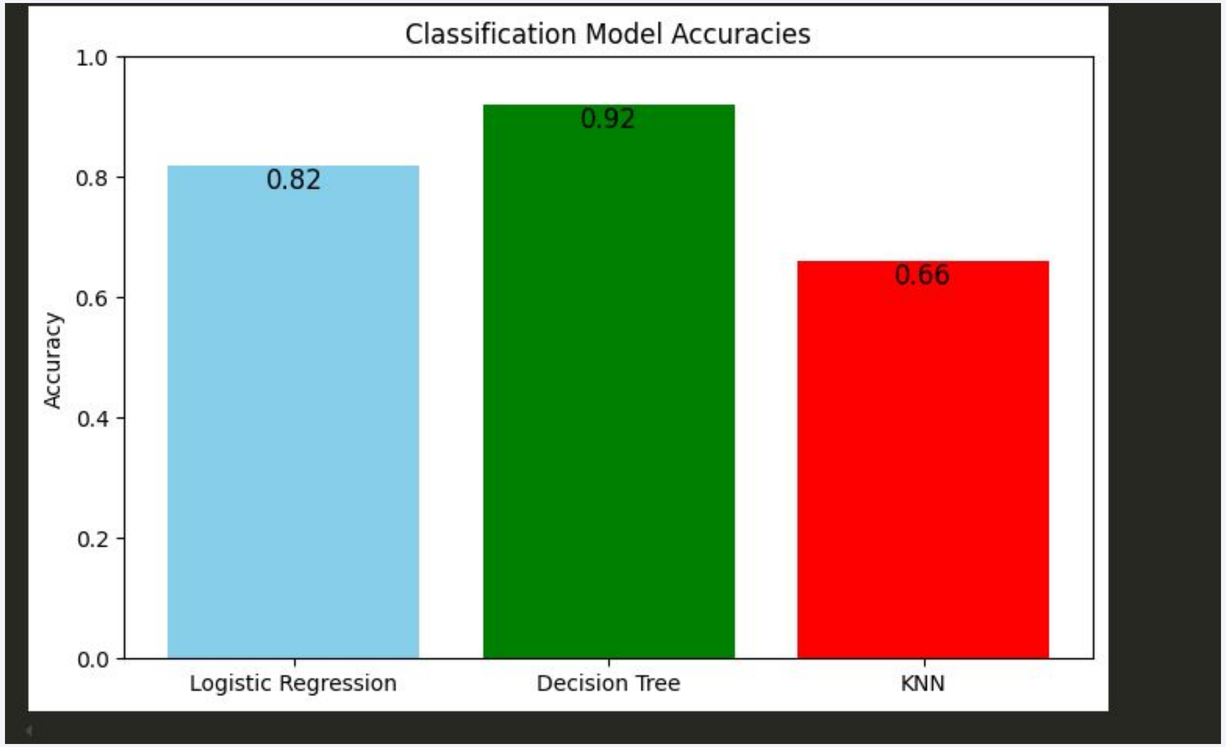
Section 5

Predictive Analysis (Classification)

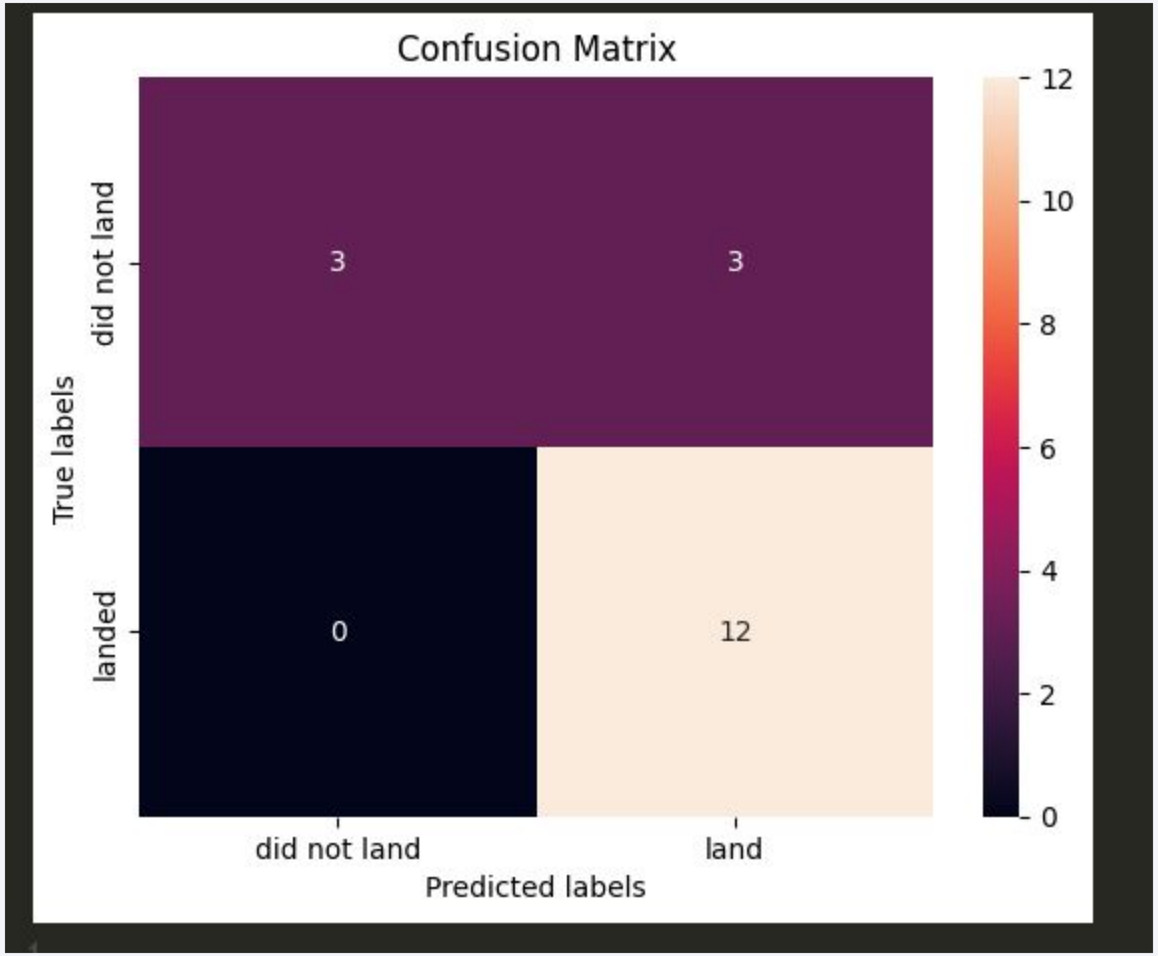
Classification Accuracy

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

- Find which model has the highest classification accuracy



Confusion Matrix



Conclusions

- Among all tested classifiers, the Decision Tree achieved the highest accuracy (0.92), outperforming Logistic Regression, SVM, and KNN.
- The confusion matrix indicates that the Decision Tree model correctly classified the majority of the test samples, making few errors.
- The strong diagonal in the confusion matrix confirms robust predictive capability for both classes.
- Therefore, the Decision Tree model is the best choice for predicting SpaceX mission outcomes in this dataset.

Thank you!