IBM Developer
SKILLS NETWORK

# Winning Space Race with Data Science

Austing Dong
2023.10.22

# Outline

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

# Executive Summary

- Summary of methodologies
  - Data Collection via API
  - Data Collection via Web Scraping
  - Data Wrangling
  - Exploratory Data Analysis (EDA) with Data Visualization
  - Exploratory Data Analysis (EDA) with SQL
  - Building interactive map with Folium & Dash
  - Prediction
- Summary of all results
  - EDA result
  - Machine Learning result

# Introduction

- Project background and context
  - Target: improve SpaceX Falcon 9 rockets' successful rate
- Problems
  - Find factors influence landing process
  - Find relations between variables and their contribution to the influence
  - Find the best conditions that will cause a successful landing

Section 1

# Methodology

# Methodology

## Executive Summary

- Data collection methodology:
  - Make get request from SpaceX API
  - Perform Web Scraping from Wikipedia
- Data Wrangling
- 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

- API

Use get request → Transform to JSON → Convert to Pandas Data frame

- Web Scraping

Send request → Use Beautiful Soup to decode → Convert to Pandas Data frame

# Data Collection – SpaceX API


```
In [6]: spacex_url="https://api.spacexdata.com/v4/launches/past"

In [7]: response = requests.get(spacex_url)
```

```
In [17]: # Use json_normalize meethod to convert the json result into a dataframe
         response.json()
         data = pd.json_normalize(response.json())
```

Using the dataframe `data` print the first 5 rows

```
In [18]: # Get the head of the dataframe
         data.head()
```

[Link](#)

# Data Collection - Scraping


Link

In [11]:
```
# use requests.get() method with the provided static_url
# assign the response to a object
response = requests.get(static_url)
```

```
soup = BeautifulSoup(response.text)
```

Print the page title to verify if the `BeautifulSoup` object was created properly

In [18]:
```
# Use soup.title attribute
soup.title
```

In [57]:
```
# df= pd.DataFrame({ key:pd.Series(value) for key, value in launch_dict.items() })
headings = []
for key,values in dict(launch_dict).items():
    if key not in headings:
        headings.append(key)
    if values is None:
        del launch_dict[key]

def pad_dict_list(dict_list, padel):
    lmax = 0
    for lname in dict_list.keys():
        lmax = max(lmax, len(dict_list[lname]))
    for lname in dict_list.keys():
        ll = len(dict_list[lname])
        if  ll < lmax:
            dict_list[lname] += [padel] * (lmax - ll)
    return dict_list

pad_dict_list(launch_dict,0)


df=pd.DataFrame(launch_dict)

df.head()
```

# Data Wrangling

Launch number for each site → Number and occurrence each orbit → Mission outcome each orbit → Labeling landing outcome → Export

```
[5]: # Apply value_counts() on column LaunchSite
df["LaunchSite"].value_counts()

[5]: CCAFS SLC 40    55
KSC LC 39A      22
VAFB SLC 4E     13
Name: LaunchSite, dtype: int64
```

```
[8]: # landing_outcomes = values on Outcome column
landing_outcomes = df["Outcome"].value_counts()
landing_outcomes

[8]: True ASDS      41
None None      19
True RTLS      14
False ASDS      6
True Ocean      5
False Ocean     2
None ASDS       2
False RTLS      1
Name: Outcome, dtype: int64
```

```
df.to_csv("dataset_part_2.csv", index=False)
```

```
[6]: # Apply value_counts on Orbit column
df["Orbit"].value_counts()

[6]: GTO      27
ISS      21
VLEO     14
PO        9
LEO       7
SSO       5
MEO       3
ES-L1     1
HEO       1
SO        1
GEO       1
Name: Orbit, dtype: int64
```

```
[16]: # landing_class = 0 if bad_outcome
# landing_class = 1 otherwise
landing_class = []
for outcome in df["Outcome"]:
    if outcome in bad_outcomes:
        landing_class.append(0)
    else:
        landing_class.append(1)
```

# EDA with Data Visualization

- Scatter Graph – visualize correlations between variables
  - Flight Number vs. Launch Site
  - Payload Mass vs. Launch Site
  - Success Rate vs. Orbit Type
  - Flight Number vs. Orbit Type
  - Payload Mass vs. Orbit Type
- Line Graph – visualize trends
  - Launch Success Yearly Trend
- Bar Graph – relationship between numerical and categorical variables
  - Orbit Type vs. Success Rate
- Link

# EDA with SQL

- SQL queries:
  - Display the names of the unique launch sites in the space mission
  - Display 5 records where launch sites begin with the string 'CCA'
  - Display the total payload mass carried by boosters launched by NASA (CRS)
  - Display average payload mass carried by booster version F9 v1.1
  - List the date when the first successful landing outcome in ground pad was achieved.
  - List the names of the boosters which have success in drone ship and have payload mass greater than 4000 but less than 6000
  - List the total number of successful and failure mission outcomes
  - List the names of the booster_versions which have carried the maximum payload mass. Use a subquery
  - List the records which will display the month names, failure landing_outcomes in drone ship ,booster versions, launch_site for the months in year 2015.
  - 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.
- Link

# Build an Interactive Map with Folium

- Using red circle to locate launch sites’ area, containing
  - Marker clusters: indicating multiple launches in the same place
  - Red Icon: indicating a failed launch
  - Green Icon: indicating a successful launch
  - Markers showing distance between key locations

- It shows more details on the location, number, distances.

- Link

# Build a Dashboard with Plotly Dash

- Charts:
  - Pie chart of launch success count for all sites
  - Pie chart for the launch site with highest launch success ratio
  - Payload vs. Launch Outcome scatter plot for all sites

- Link

# Predictive Analysis (Classification)

- Classification Models
  - Logistic Regression
  - SVM
  - Decision Tree
  - KNN

- Link

# Results

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

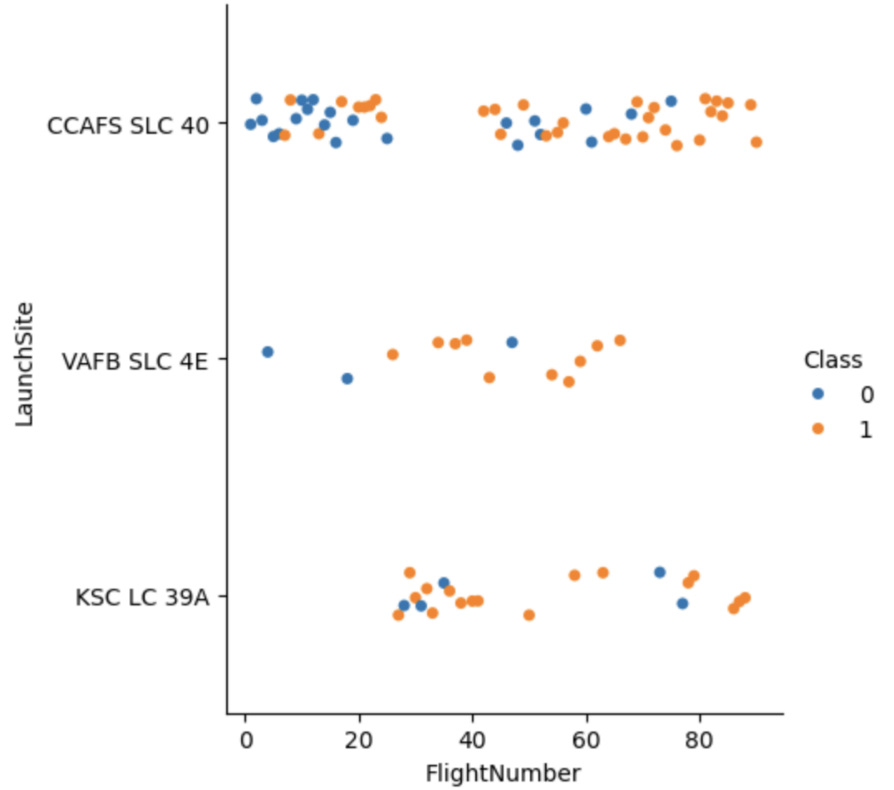
Section 2

# Insights drawn from EDA

# Flight Number vs. Launch Site

- We find that:
  - As flight number increase, each launch site tend to success
  - CCAFS SLC 40 has the most frequent launch samples
  - VAFB SLC 4E tend to have a higher successful rate

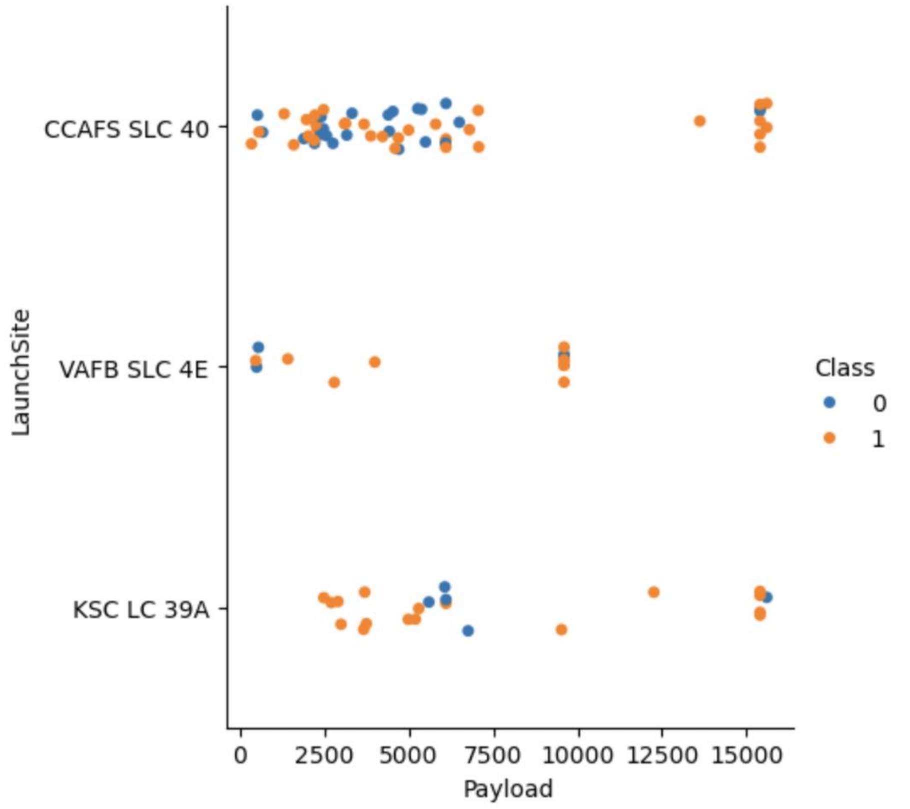
```
[7]: ### TASK 1: Visualize the relationship between Flight Number and Launch Site
sns.catplot(x="FlightNumber", y="LaunchSite", data=df, hue="Class")
plt.xlabel("FlightNumber")
plt.ylabel("LaunchSite")
plt.show()
```



# Payload vs. Launch Site

- We find that:
  - Higher Payload (>7500) may lead to a better successful rate for these Launch Sites
  - However, too much Payload like 15000 may lead to some failures

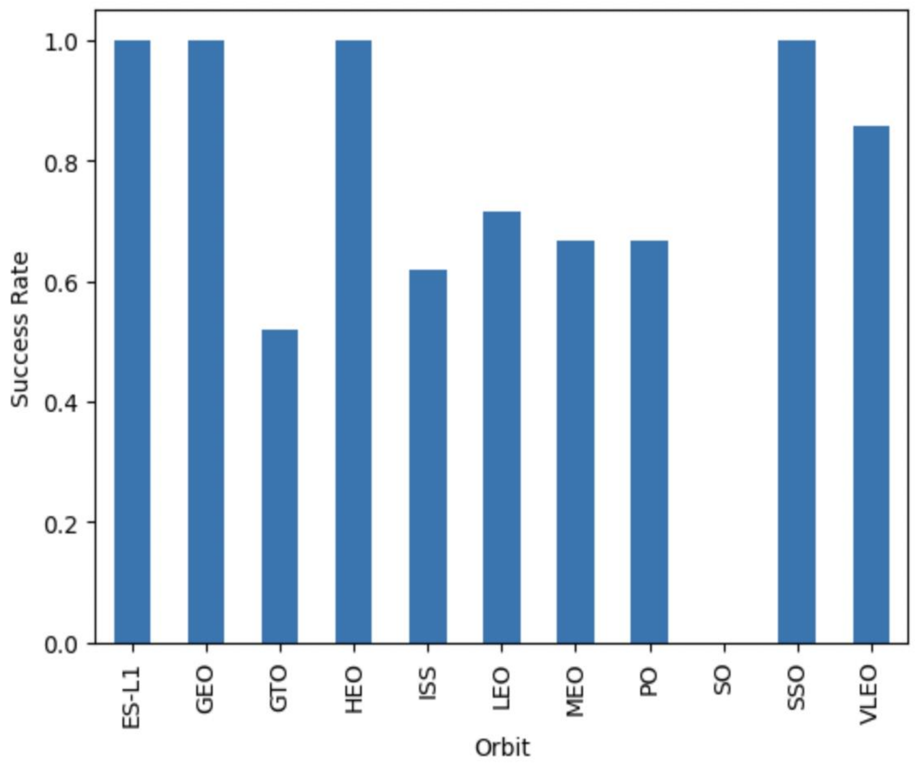
[9]:
```python
### TASK 2: Visualize the relationship between Payload and Launch Site
sns.catplot(x="PayloadMass", y="LaunchSite", data=df, hue="Class")
plt.xlabel("Payload")
plt.ylabel("LaunchSite")
plt.show()
```



# Success Rate vs. Orbit Type

- We find that:
  - ES-L1, GEO, HEO, and SSO have the best Successful Rate

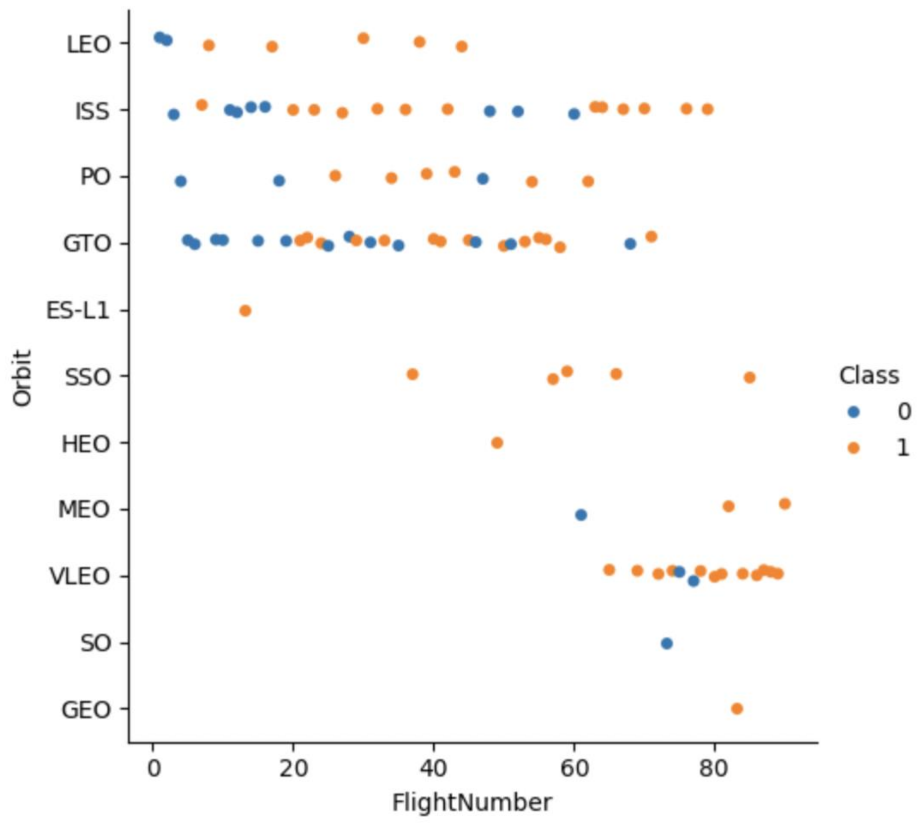
```
[33]: ### TASK  3: Visualize the relationship between success rate of each orbit type
df_ob = df.groupby(["Orbit"])["Class"].mean()
df_ob.plot(kind="bar")
plt.xlabel("Orbit")
plt.ylabel("Success Rate")
plt.show()
```



# Flight Number vs. Orbit Type

- We find that:
  - As Flight Number increases, LEO, PO, VLEO are more likely to success
  - GTO tends to have no relationships between Flight Number and Success

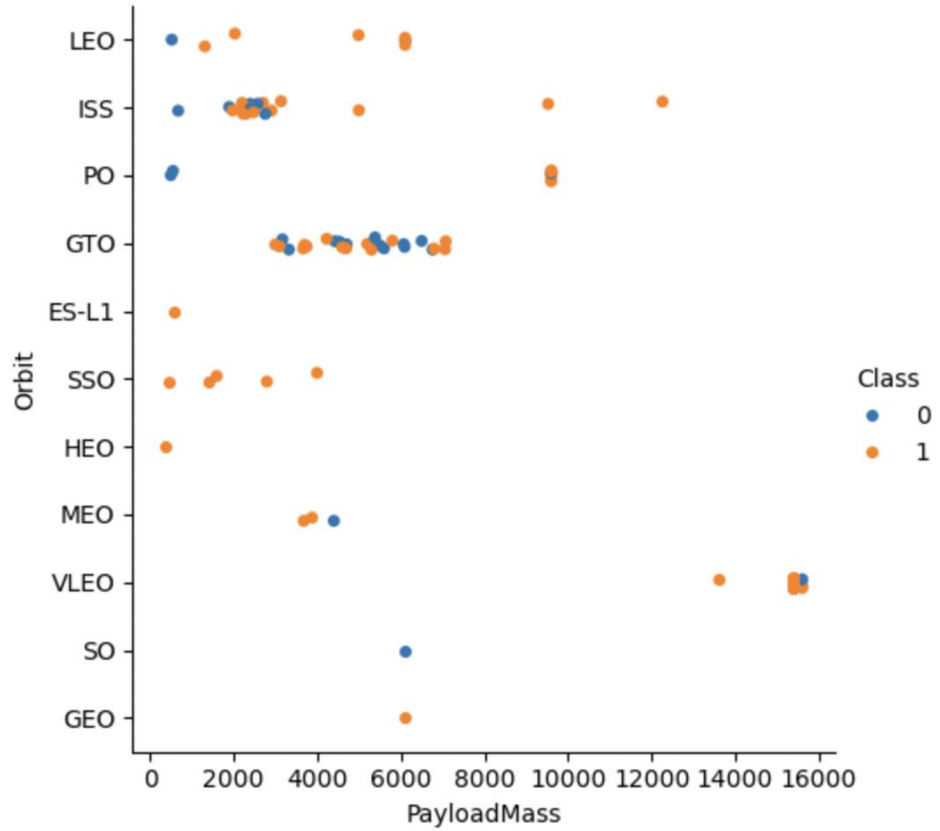
```
[34]: ### TASK  4: Visualize the relationship between FlightNumber and Orbit type
sns.catplot(x="FlightNumber", y="Orbit", hue="Class", data=df)
plt.show()
```



# Payload vs. Orbit Type

- We find that:
  - For most Orbits, a higher Payload Mass are more possible to success
  - However, for GTO, we cannot find a significant relationship between Payload Mass and success

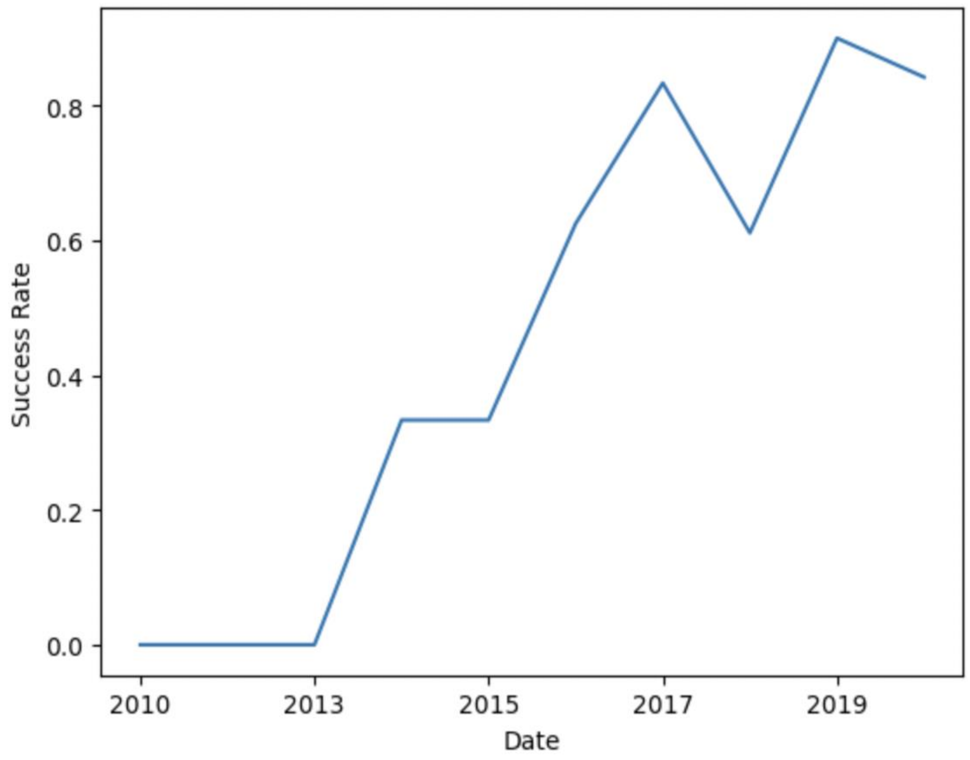
[35]:
```
### TASK  5: Visualize the relationship between Payload and Orbit type
sns.catplot(x="PayloadMass", y="Orbit", hue="Class", data=df)
plt.show()
```



# Launch Success Yearly Trend

- We find that:
  - From 2010-2020, the Success Rate tend to increase over years.
  - There is a small decrease of Success Rate in 2018

```
[43]: # Plot a line chart with x axis to be the extracted year and y axis to be the success rate
df_year = df.groupby("Date")["Class"].mean()
df_year.plot(kind="line")
plt.ylabel("Success Rate")
plt.show()
```



# All Launch Site Names

- SQL query:
  - SELECT DISTINCT launch_site FROM SPACEXTBL
- Explanation:
  - DISTINCT: make launch_site unique

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

# Launch Site Names Begin with 'CCA'

- SQL query:
  - SELECT * FROM SPACEXTBL WHERE launch_site LIKE "CCA%" LIMIT 5

| Date | Time (UTC) | Booster_Version | Launch_Site | Payload | PAYLOAD_MASS__KG_ | Orbit | Customer | Mission_Outcome | Landing_Outcon |
|---|---|---|---|---|---|---|---|---|---|
| 2010-04-06 | 18:45:00 | F9 v1.0 B0003 | CCAFS LC-40 | Dragon Spacecraft Qualification Unit | 0 | LEO | SpaceX | Success | Failure (parachut |
| 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 (parachut |
| 2012-05-22 | 07:44:00 | F9 v1.0 B0005 | CCAFS LC-40 | Dragon demo flight C2 | 525 | LEO (ISS) | NASA (COTS) | Success | No attem |
| 2012-08-10 | 00:35:00 | F9 v1.0 B0006 | CCAFS LC-40 | SpaceX CRS-1 | 500 | LEO (ISS) | NASA (CRS) | Success | No attem |
| 2013-01-03 | 15:10:00 | F9 v1.0 B0007 | CCAFS LC-40 | SpaceX CRS-2 | 677 | LEO (ISS) | NASA (CRS) | Success | No attem |

- Explanation:
  - WHERE launch_site LIKE “CCA%” implies the launch_site need to start with “CCA”
  - LIMIT 5 indicates that the result contains not more than 5 elements

# Total Payload Mass

- SQL query
  - SELECT SUM(PAYLOAD_MASS__KG_) FROM SPACEXTBL WHERE customer="NASA (CRS)"

| SUM(PAYLOAD_MASS__KG_) |
|---:|
| 45596 |

- Explanation:
  - SUM(PAYLOAD_MASS__KG_) calculates the total sum of PAYLOAD_MASS__KG_
  - WHERE customer="NASA (CRS)” implies that we only select “NASA (CRS)” customers

# Average Payload Mass by F9 v1.1

- SQL query
  - SELECT AVG(PAYLOAD_MASS__KG_) FROM SPACEXTBL WHERE Booster_Version LIKE "F9 v1.1%"

| AVG(PAYLOAD_MASS__KG_) |
|---|
| 2534.6666666666665 |

- Explanation:
  - AVG(PAYLOAD_MASS__KG_) calculates the average of PAYLOAD_MASS__KG_
  - WHERE Booster_Version LIKE "F9 v1.1%" selects the Booster_Version start with "F9 v1.1%"

# First Successful Ground Landing Date

- SQL query:
  - SELECT MIN(Date) FROM SPACEXTBL WHERE Landing_Outcome LIKE "%Success%ground%pad%"

| MIN(Date) |
|---|
| 2015-12-22 |

- Explanation:
  - LIKE "%Success%ground%pad%" selects the Landing outcome containing "Success", "ground" "pad"

# Successful Drone Ship Landing with Payload between 4000 and 6000

- SQL query:
  - SELECT DISTINCT Booster_Version FROM SPACEXTBL WHERE Landing_Outcome LIKE "%Success%drone%ship%" AND PAYLOAD_MASS__KG_ > 4000 AND PAYLOAD_MASS__KG_ < 6000

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

- Explanation:
  - LIKE “%Success%drone%ship%” filters the successful landing outcome on drone ship
  - Use AND to connect the conditions “>4000”, “<6000” to show that payload mass is in range (4000, 6000)

# Total Number of Successful and Failure Mission Outcomes

- SQL query:

```
[43]: %%sql
SELECT category, outcome
FROM (
  SELECT 'Success' AS category, COUNT(Mission_Outcome) AS outcome
  FROM SPACEXTBL
  WHERE Mission_Outcome LIKE "%Success%"

  UNION ALL

  SELECT 'Failure' AS category, COUNT(Mission_Outcome) AS outcome
  FROM SPACEXTBL
  WHERE Mission_Outcome LIKE "%Failure%"
)
```

| category | outcome |
|---|---|
| Success | 100 |
| Failure | 1 |

- Explanation:
  - Select Success and Failure samples separately
  - Use UNION to connect them

# Boosters Carried Maximum Payload

- SQL query:

```
[46]: %%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 |

- Explanation:
  - Use another SELECT query to get max payload mass

# 2015 Launch Records

- SQL query:

```
%%sql
SELECT SUBSTR(Date, 6, 2) AS month_name, Landing_Outcome, Booster_version, Launch_site
FROM SPACEXTBL
WHERE Landing_Outcome LIKE "%Failure%drone%ship%"
AND SUBSTR(DATE, 0, 5)='2015'
```

| month_name | Landing_Outcome | Booster_Version | Launch_Site |
|---|---|---|---|
| 10 | Failure (drone ship) | F9 v1.1 B1012 | CCAFS LC-40 |
| 04 | Failure (drone ship) | F9 v1.1 B1015 | CCAFS LC-40 |

- Explanation:
  - Use SUBSTR to get month and year

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

- SQL query:

```
%%sql
SELECT Landing_Outcome, COUNT(Landing_Outcome) FROM SPACEXTBL
WHERE Date >= '2010-06-04' AND Date <= '2017-03-20'
GROUP BY (Landing_Outcome) ORDER BY COUNT(Landing_Outcome) DESC
```

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

| Landing_Outcome | COUNT(Landing_Outcome) |
|---|---|
| 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 |

- Explanation:
  - Use ORDER BY to rank by count of landing outcome
  - Use DESC to rank in descending order

Section 3

# Launch Sites Proximities Analysis

# Folium Map – All launch sites

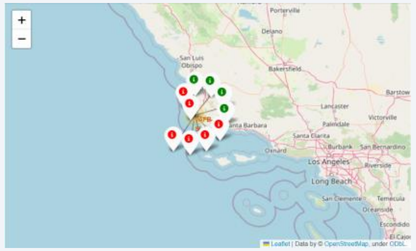
- All launch sites' location markers on a global map



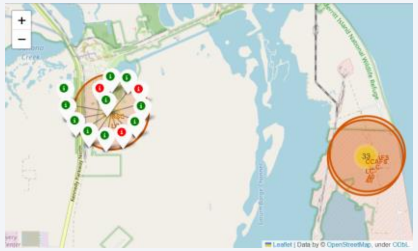
- It shows that all launch sites are near the coast, most are in east coasts

# Folium Map – Labeled launch outcomes

- color-labeled launch outcomes (Green icon: successful, Red icon: fail)
  - West Coast: (VAFBSLC-4E)

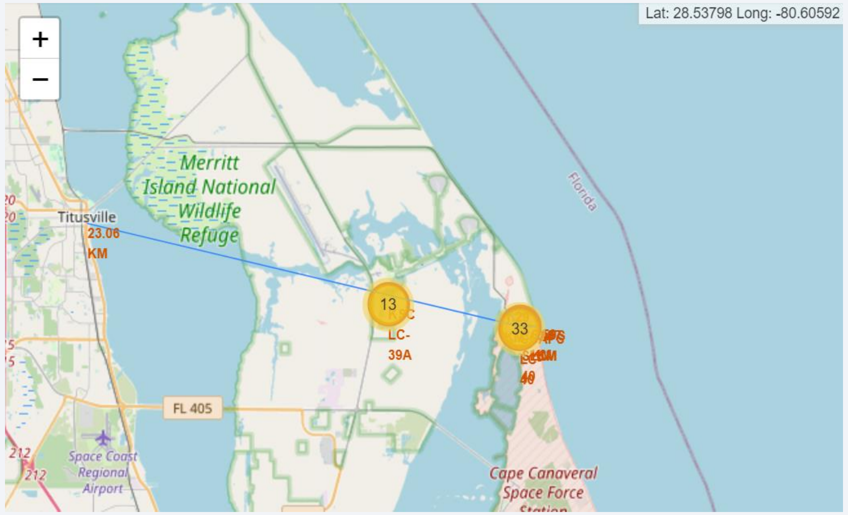


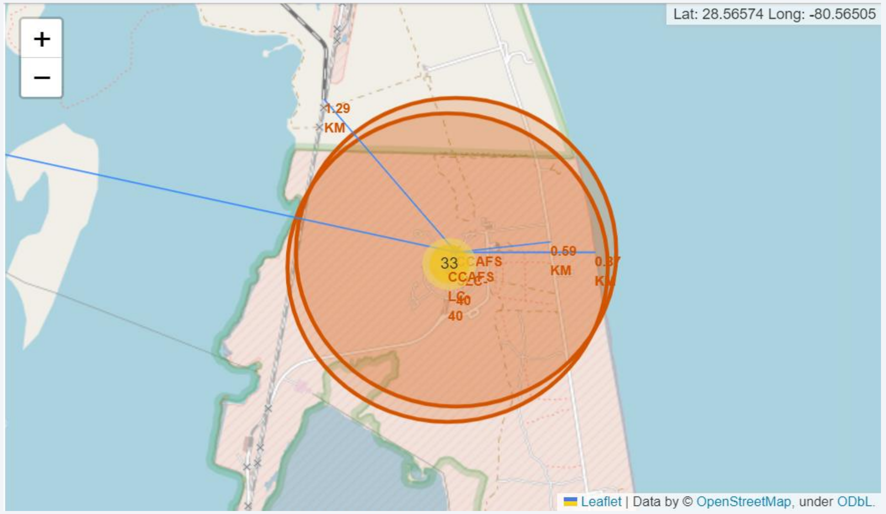
  - East Coast: (KSCLC-39A, CCAFS-SLC-40)



- East coast launches more successful than west coast. In east coast, KSCLC-39A is likely to be more successful than CCAFS-SLC-40

# Folium Map – Distances



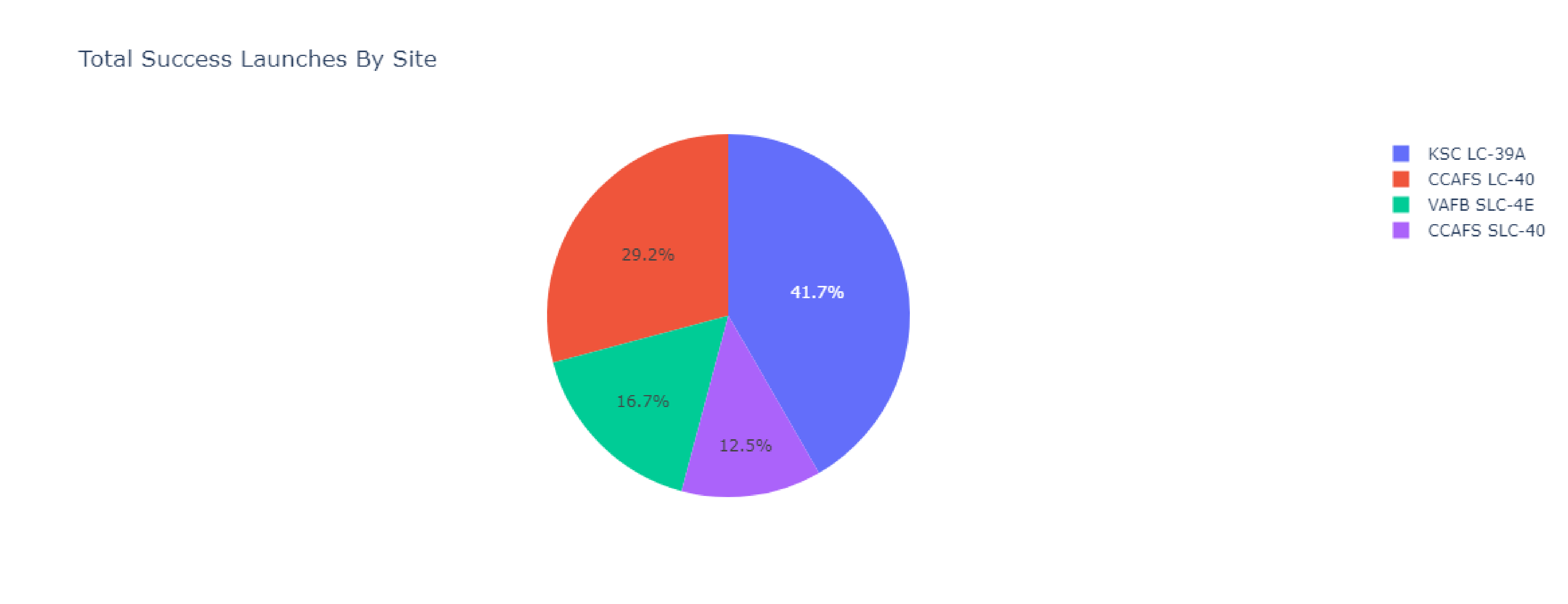




- Are launch sites in close proximity to railways? Yes
- Are launch sites in close proximity to highways? Yes
- Are launch sites in close proximity to coastline? Yes
- Do launch sites keep certain distance away from cities? No

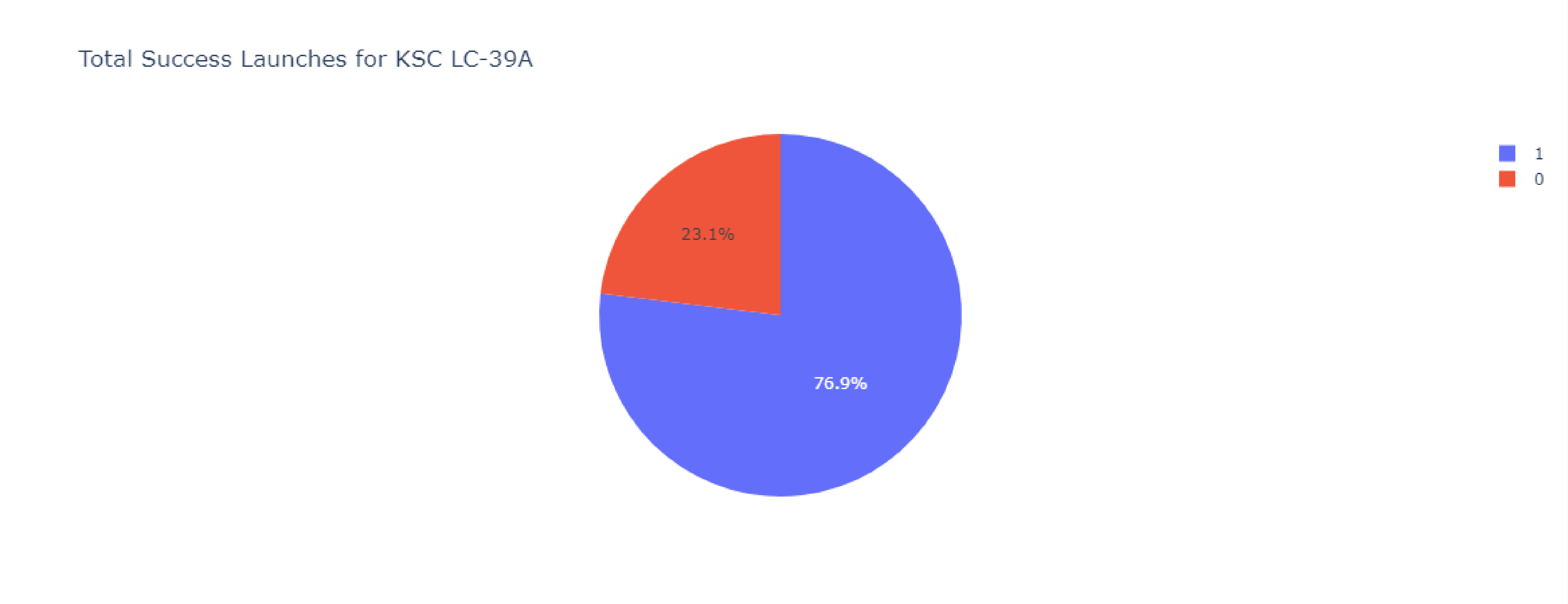
Section 4

# Build a Dashboard with Plotly Dash

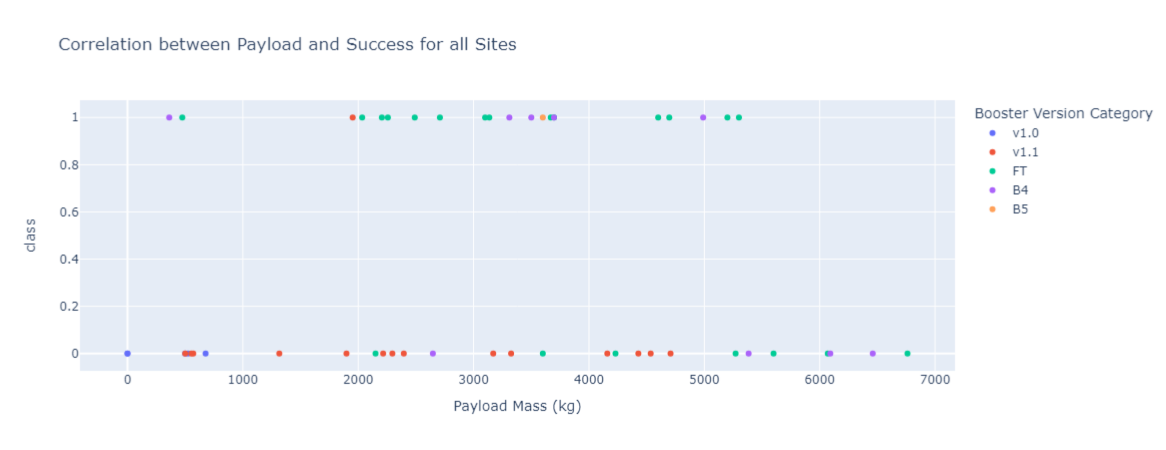
# Dashboard – Pie chart of launch success count for all sites



# Dashboard – Pie chart for the launch site with highest launch success ratio



# Dashboard – Payload vs. Launch Outcome scatter plot for all sites

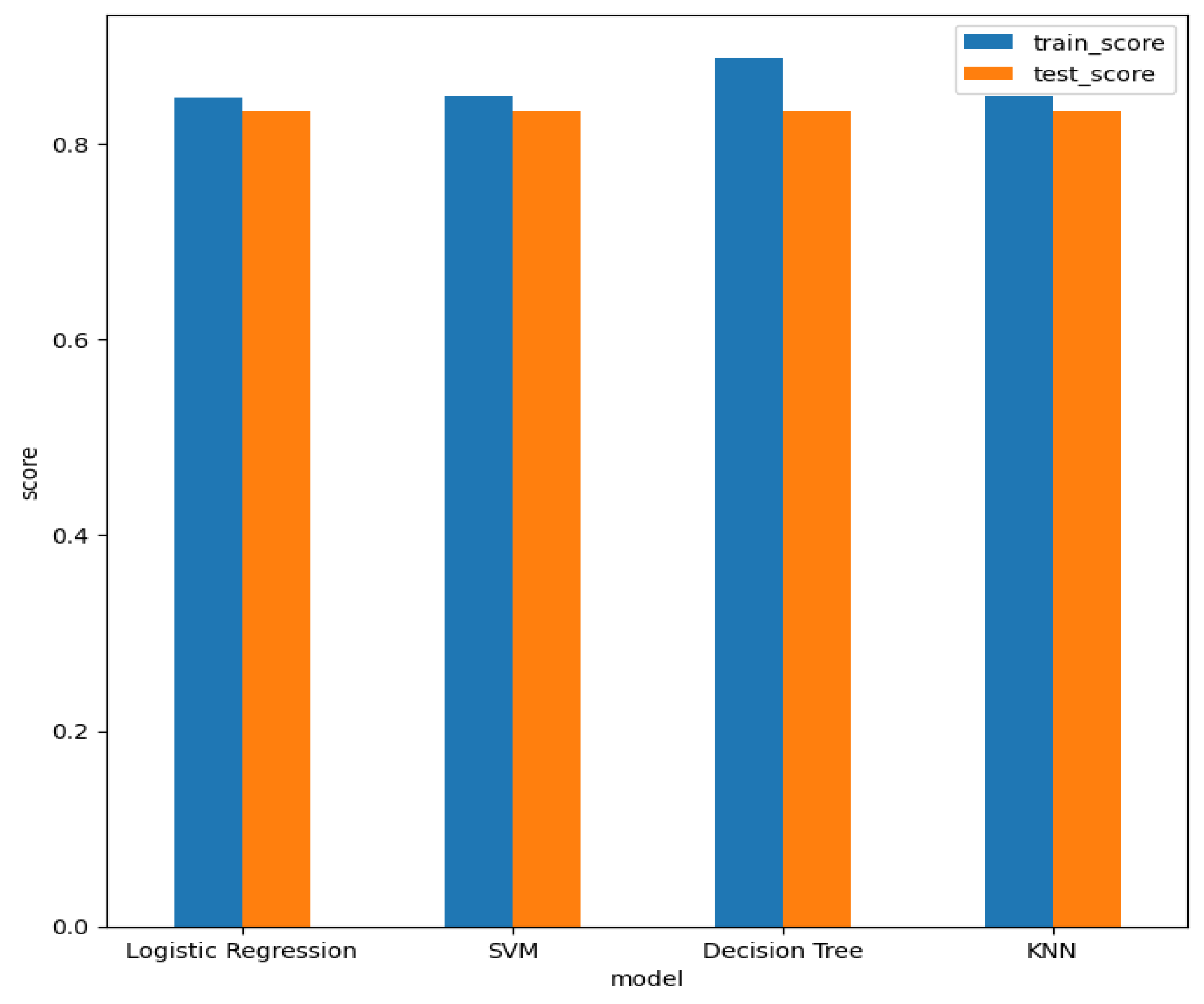





Section 5

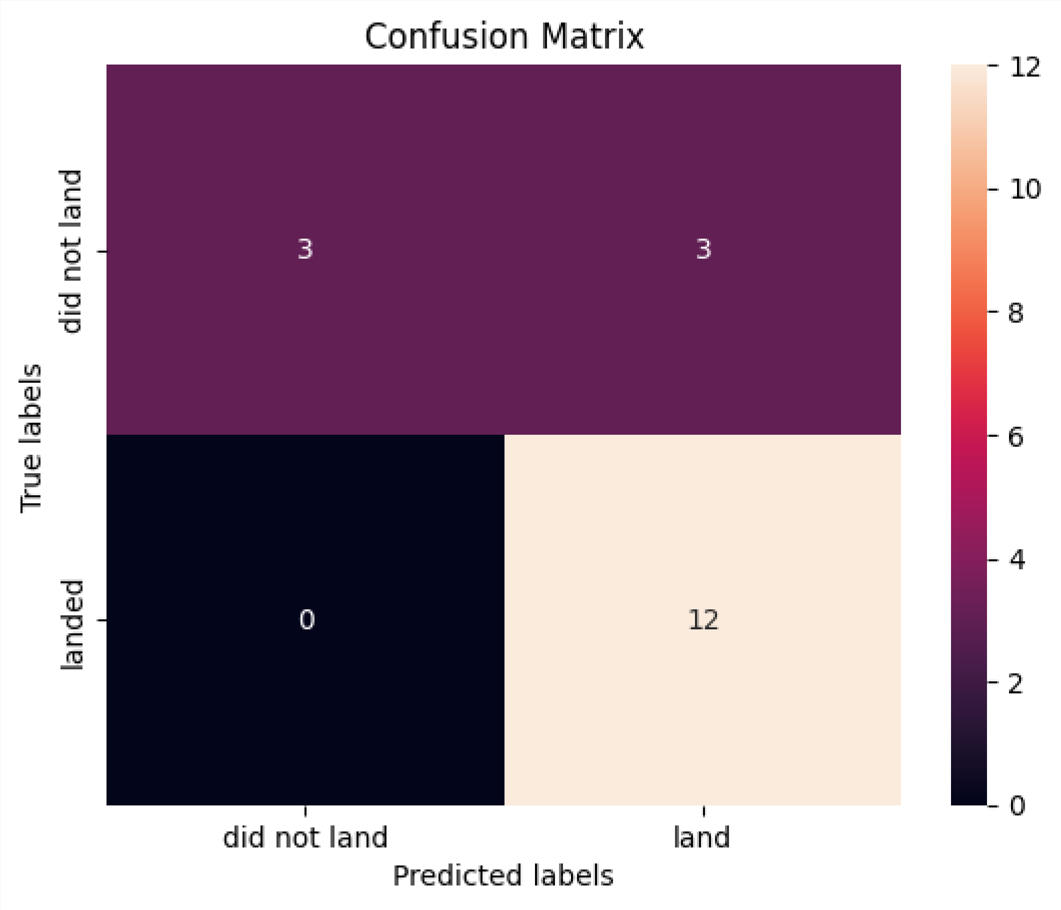
# Predictive Analysis (Classification)

# Classification Accuracy




From the bar chart, Decision Tree has the highest classification accuracy

# Confusion Matrix



# Conclusions

- There are several factors influences the result, such as the launch site, the orbit and especially the number of previous launches.
- Orbits with the best success rates: GEO, HEO, SSO, ES-L1.
- Generally heavy weighted payloads perform better than low weighted payloads.
- We choose the Decision Tree Algorithm as the best model for its better training accuracy. However, the test accuracy between all the models used is identical.

# Thank you!