IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

<Name>
<Date>

Outline

Executive Summary

Introduction

Methodology

Results

Conclusion

Appendix

Executive Summary

- **In the upcoming capstone endeavor, our focus revolves around prognosticating the likelihood of a successful landing for the SpaceX Falcon 9 first stage through the employment of diverse machine learning classification methodologies.**
- **The primary phases encompassed within this undertaking encompass:**
- **Acquiring, refining, and structuring data**
- **Engaging in exploratory data analysis**
- **Crafting interactive data visualizations**
- **Conducting machine learning-based prognostication**
- **Our visual representations demonstrate correlations between certain attributes of rocket launches and their resultant outcomes, whether they be successes or failures.**
- **It has been deduced that the decision tree algorithm emerges as a promising contender for predicting the successful landing of the Falcon 9 first stage.**

Introduction

- Project Background and Context:
- The landscape of space exploration has witnessed significant transformations in recent years, marked by notable advancements in private space travel. Consequently, the space industry is progressively transitioning into a more mainstream and accessible realm for the general population.
- Despite these strides, the cost of launch operations continues to pose a formidable barrier for potential entrants into the space race.
- SpaceX, with its pioneering first stage reuse capabilities, enjoys a pivotal advantage over its competitors. While each SpaceX launch is estimated to cost approximately 62 million dollars, the company's ability to recycle the first stage for future launches confers a distinct edge. In contrast, other industry players expend upwards of 165 million dollars for each launch, lacking the cost-saving benefits of stage reuse.
- Problems to Address:

1. Determine the likelihood of successful first stage landings for SpaceX Falcon 9 rockets.
2. Assess the impact of various parameters and variables on the outcomes of first stage landings. These variables may include launch site, payload mass, booster version, among others.
3. Investigate potential correlations between launch sites and the success rates of Falcon 9 rocket landings.

Section 1

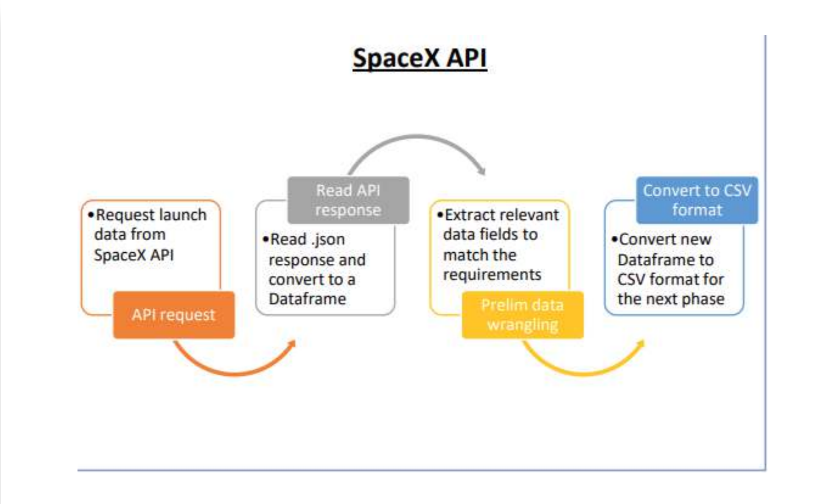
Methodology

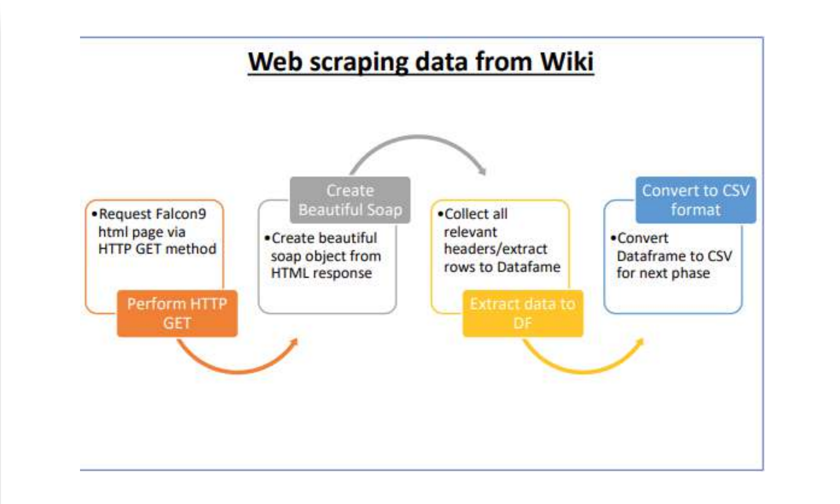
Methodology

- 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 collection involves gathering data from various sources, which can be structured, unstructured, or semi-structured. In our project, we collected data using the SpaceX API and by scraping relevant launch data from Wikipedia pages.





Data Collection – SpaceX API

- GitHub: https://github.com/sombir1/Final-Project/blob/main/jupyter-labs-spacex-data-collection-api%20(1).ipynb

1. Create API GET request, normalize data and read in to a Dataframe:

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

```
response = requests.get(spacex_url)
```

```
# Use json_normalize meethod to convert the json
data = pd.json_normalize(response.json())
```

2. Declare global variable lists that will store data returned by helper functions with additional API calls to get relevant data

```
#Global variables
BoosterVersion = []
PayloadMass = []
Orbit = []
LaunchSite = []
Outcome = []
Flights = []
GridFins = []
Reused = []
Legs = []
LandingPad = []
Block = []
ReusedCount = []
Serial = []
Longitude = []
Latitude = []
```

3. Call helper functions to get relevant data where columns have IDs (e.g., rocket column is an identification number)
   - getBoosterVersion(data)
   - getLaunchSite(data)
   - getPayloadData(data)
   - getCoreData(data)

4. Construct dataset from received data & combine columns into a dictionary:

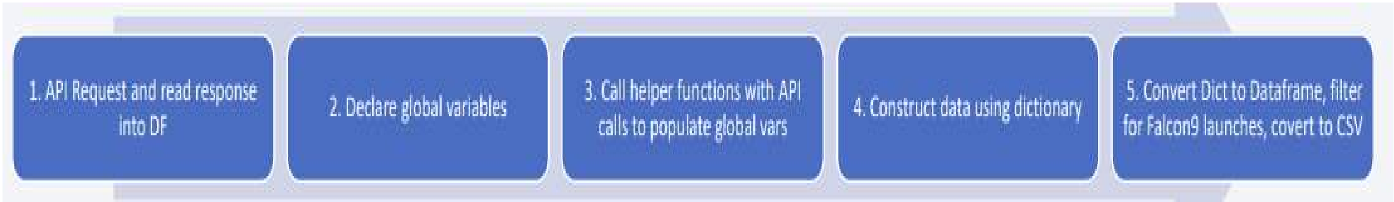
```
launch_dict = {'FlightNumber': list(data['flight_number']),
'Date': list(data['date']),
'BoosterVersion':BoosterVersion,
'PayloadMass':PayloadMass,
'Orbit':Orbit,
'LaunchSite':LaunchSite,
'Outcome':Outcome,
'Flights':Flights,
'GridFins':GridFins,
'Reused':Reused,
'Legs':Legs,
'LandingPad':LandingPad,
'Block':Block,
'ReusedCount':ReusedCount,
'Serial':Serial,
'Longitude': Longitude,
'Latitude': Latitude}
```

4. Create Dataframe from dictionary and filter to keep only the Falcon9 launches:

```
# Create a data from launch_dict
df_launch = pd.DataFrame(launch_dict)
```

```
# Hint data['BoosterVersion']!='Falcon 1'
data_falcon9 = df_launch[df_launch['BoosterVersion']!= 'Falcon 1']
```

```
data_falcon9.to_csv('dataset_part\_1.csv', index=False)
```



Data Collection - Scraping

GitHub: https://github.com/sombir1/Final-Project/blob/main/jupyter-labs-webscraping.ipynb

1. Fetching HTML content from Wikipedia
2. Employing BeautifulSoup with the html5lib Parser
3. Locating the HTML table containing launch information
4. Converting the obtained dictionary into a DataFrame
5. Iterating through the table cells to extract data and populate the dictionary
6. Generating a dictionary with the extracted data

Data Wrangling

- • Conducted an Exploratory Data Analysis (EDA) to identify patterns within the dataset and establish labels for training supervised models. • The dataset encompassed diverse mission outcomes, which were transformed into Training Labels. A value of 1 indicated a successful booster landing, while 0 indicated an unsuccessful landing. The following landing scenarios were considered to formulate the labels:
- True Ocean: Denoted a mission outcome successfully landed in a specific region of the ocean.
- False Ocean: Indicated an unsuccessful landing in a specific region of the ocean.
- RTLS (Return to Launch Site): Signified a successful landing on a ground pad.
- False RTLS: Represented an unsuccessful landing attempt on a ground pad.
- True ASDS (Autonomous Spaceport Drone Ship): Reflected a successful landing on a drone ship.
- False ASDS: Corresponded to an unsuccessful landing attempt on a drone ship.
- GitHub: https://github.com/sombir1/Final-Project/blob/main/labs-jupyter-spacex-Data%20wrangling%20(1).ipynb

EDA with Data Visualization

- As part of the Exploratory Data Analysis (EDA), the following charts were generated to delve deeper into the dataset:
- Scatter Plot:
  - A scatter plot displays the relationship or correlation between two variables, facilitating the observation of patterns.
  - The following charts were plotted to visualize:
    - Relationship between Flight Number and Launch Site
    - Relationship between Payload and Launch Site
    - Relationship between Flight Number and Orbit Type
    - Relationship between Payload and Orbit Type
- Bar Chart:
  - Bar charts are effective for comparing the values of a variable at a specific point in time. They provide a clear visual representation of which groups are most common and how other groups compare.
  - The following bar chart was generated to visualize:
    - Relationship between the success rate of each orbit type
- Line Chart:
  - Line charts are commonly used to track changes over a period of time, enabling the observation of trends.
  - The following line chart was plotted to observe:
    - Average launch success yearly trend
- GitHub: https://github.com/sombir1/Final-Project/blob/main/edadataviz.ipynb

EDA with SQL

- **Unique Launch Sites:**
  - The data analysis unveils the distinct launch sites contributing to the space mission.
- **Total Payload Mass by NASA (CRS):**
  - The investigation yields the cumulative payload mass carried by boosters launched under NASA's CRS program.
- **Average Payload Mass of Booster Version F9 v1.1:**
  - Insights reveal the mean payload mass transported by boosters of version F9 v1.1.
- GitHub: https://github.com/sombir1/Final-Project/blob/main/jupyter-labs-eda-sql-coursera_sqllite.ipynb

Build an Interactive Map with Folium

- Utilizing Folium interactive maps aids in analyzing geospatial data, enabling more engaging visual analytics and deeper insights into factors such as location and proximity of launch sites, which influence launch success rates.
- Key elements were incorporated into the map object:
- All launch sites were marked on the map to provide a visual representation.
- 'folium.circle' and 'folium.marker' were utilized to highlight circle areas with text labels over each launch site.
- A 'MarkerCluster()' was added to display launch success (green) and failure (red) markers for each launch site, enhancing visual clarity.
- Distances between launch sites and proximities (e.g., coastline, railroad, highway, city) were calculated and displayed.
- 'MousePosition()' was employed to retrieve coordinates for a mouse position over a point on the map.
- 'folium.Marker()' was added to indicate distances (in KM) on the map for various proximities (e.g., coastline, railroad, highway, city).
- 'folium.Polyline()' was used to draw lines between points on the map and the launch site, enhancing spatial understanding.
- These interactive features facilitated answering pertinent questions:
- 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 the coastline? YES
- Do launch sites maintain a certain distance from cities? YES
- Overall, building the interactive map with Folium provided a comprehensive understanding of the spatial relationships between launch sites and various geographical features, aiding in the analysis of factors influencing launch success rates.
- Github:https://github.com/sombir1/Final-Project/blob/main/lab_jupyter_launch_site_location.ipynb

Build a Dashboard with Plotly Dash

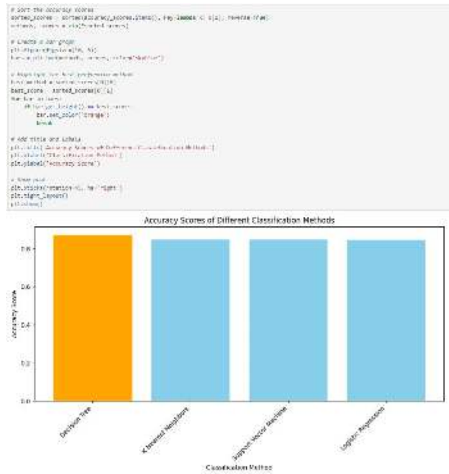
- • Developed a Plotly Dash web application for interactive visual analytics on SpaceX launch data, featuring a Launch Site Drop-down, Pie Chart, Payload range slider, and Scatter chart.
- Integrated a Launch Site Drop-down Input component, enabling filtering by all launch sites or a specific one.
- Implemented a Pie Chart displaying total successful launches when 'All Sites' is selected, and success/failure counts for a chosen site.
- Incorporated a Payload range slider for easy selection of different payload ranges to identify visual patterns.
- Included a Scatter chart to explore payload correlation with mission outcomes for selected site(s), with Booster version color-labeled on each point.
- • Dashboard insights addressed the following questions:
- Which site has the largest successful launches? KSC LC-39A with 10.
- Which site has the highest launch success rate? KSC LC-39A with 76.9% success.
- Which payload range(s) has the highest launch success rate? 2000 – 5000 kg.
- Which payload range(s) has the lowest launch success rate? 0-2000 and 5500 - 7000.
- Which F9 Booster version (v1.0, v1.1, FT, B4, B5, etc.) has the highest launch success rate? FT.
- Github: https://github.com/sombir1/Final-Project/blob/main/spacex_dash_app.py

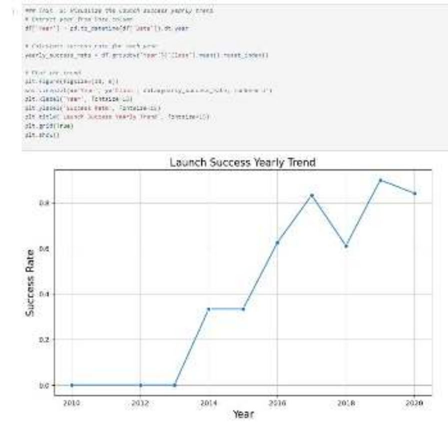
Predictive Analysis (Classification)

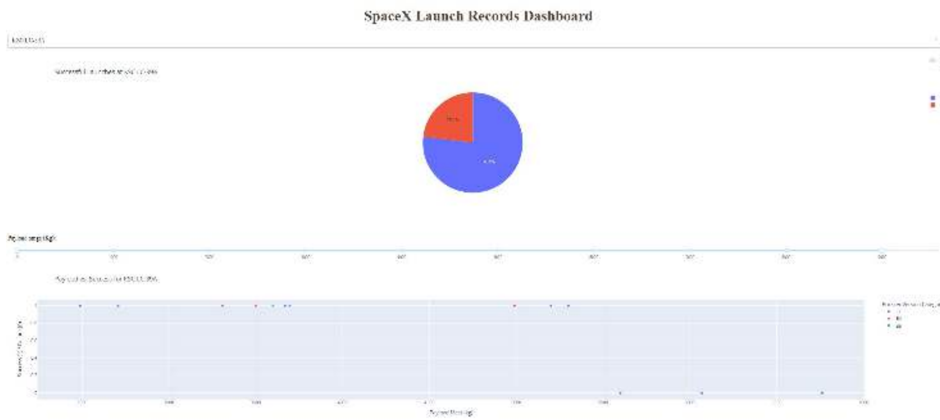
- When comparing the results of all four models side by side, it's evident that they all achieve the same accuracy score and confusion matrix when tested on the test set. Therefore, their GridSearchCV best scores are used to rank them. Based on the GridSearchCV best scores, the models are ranked as follows, with the best model listed first and the worst model listed last:
- Decision Tree (GridSearchCV best score: 0.8892857142857142)
- K Nearest Neighbors, KNN (GridSearchCV best score: 0.8482142857142858)
- Support Vector Machine, SVM (GridSearchCV best score: 0.8482142857142856)
- Logistic Regression (GridSearchCV best score: 0.8464285714285713)
- GitHub:https://github.com/sombir1/Final-Project/blob/main/SpaceX_Machine%20Learning%20Prediction_Part_5.ipynb

Results

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







Section 2

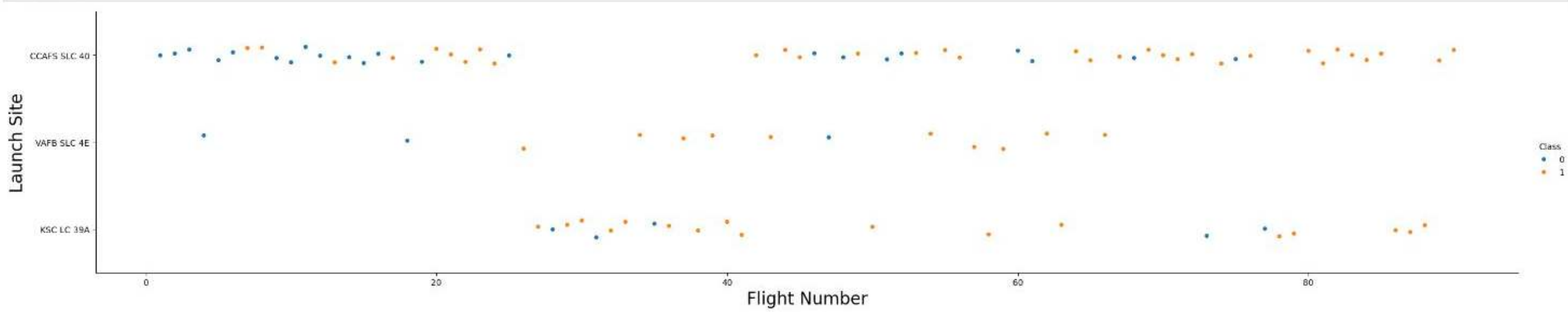
Insights drawn from EDA

Flight Number vs. Launch Site

- The success rates (Class=1) demonstrate an increase as the number of flights increases.
- Specifically, for launch site 'KSC LC 39A', it typically requires approximately 25 launches before achieving the first successful launch.

[7]:

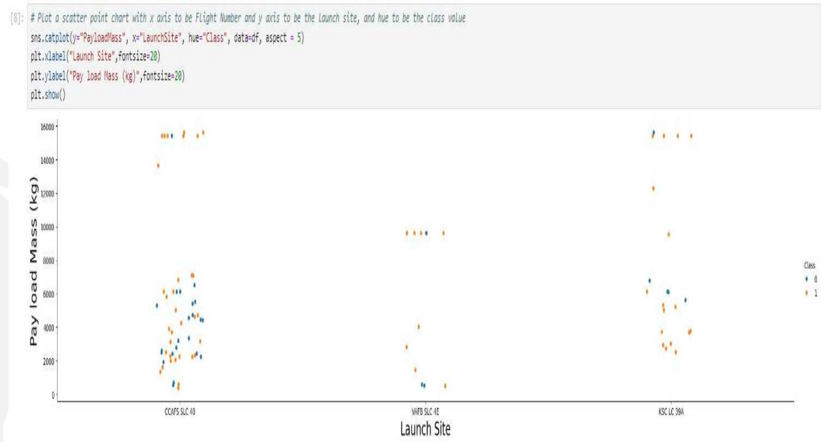
```
### TASK 1: Visualize the relationship between Flight Number and Launch Site
# Plot a scatter point chart with x axis to be Flight Number and y axis to be the launch site, and hue to be the class value
sns.catplot(y="LaunchSite", x="FlightNumber", hue="Class", data=df, aspect = 5)
plt.xlabel("Flight Number",fontsize=20)
plt.ylabel("Launch Site",fontsize=20)
plt.show()
```

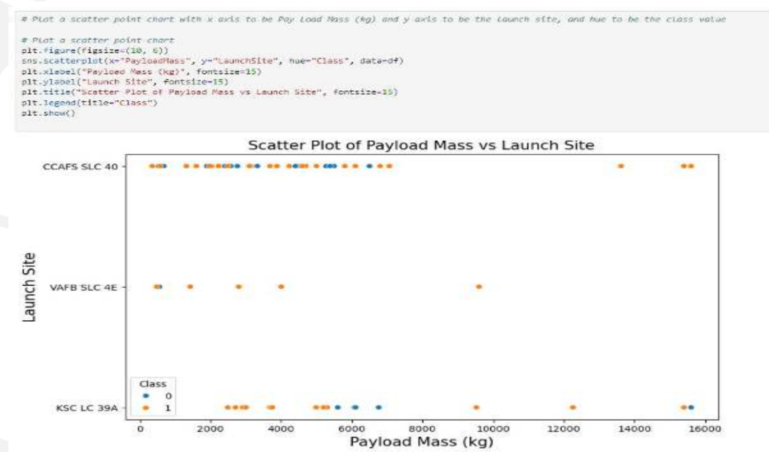


Use the function `catplot` to plot `FlightNumber` vs `LaunchSite` , set the parameter `x` parameter to `FlightNumber` ,set the `y` to `Launch Site` and set the parameter `hue` to `'class'`

Payload vs. Launch Site

- At launch site 'VAFB SLC 4E', no rockets are launched for payloads greater than 10,000 kg.
- The percentage of successful launches (Class=1) at launch site 'VAFB SLC 4E' tends to increase as the payload mass increases.
- There is no discernible correlation or pattern observed between launch site and payload mass.

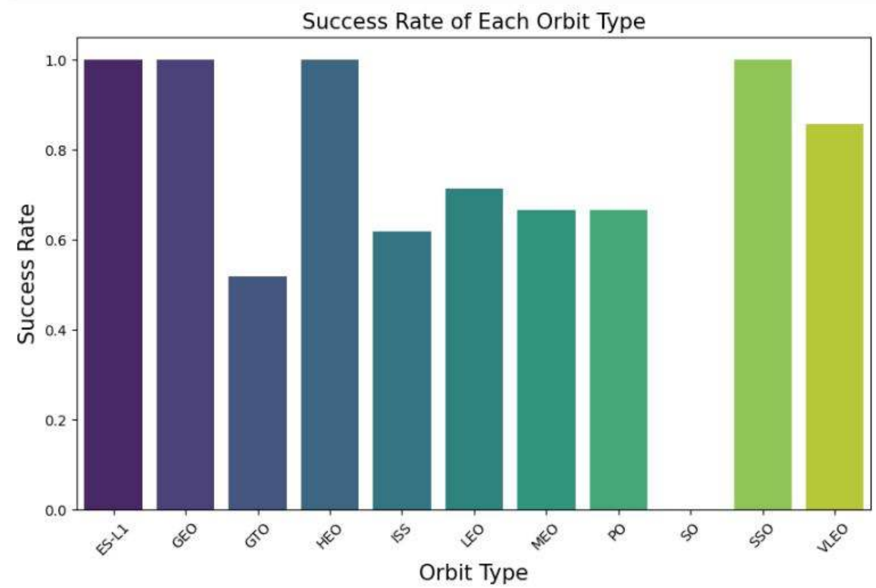




Success Rate vs. Orbit Type

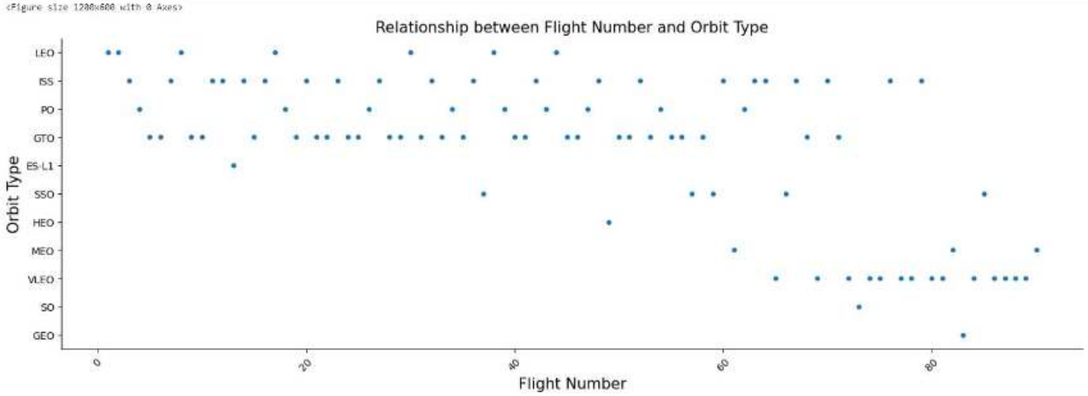
- Orbits ES-LI, GEO, HEO, and SSO exhibit the highest success rates.
- Conversely, the GTO orbit demonstrates the lowest success rate.

```
### TASK  3: Visualize the relationship between success rate of each orbit type
# Plot a bar plot
plt.figure(figsize=(10, 6))
sns.barplot(x="Orbit", y="Class", hue="Orbit", data=orbit_success_rate, palette="viridis", legend=False)
plt.xlabel("Orbit Type", fontsize=15)
plt.ylabel("Success Rate", fontsize=15)
plt.title("Success Rate of Each Orbit Type", fontsize=15)
plt.xticks(rotation=45)
plt.show()
```



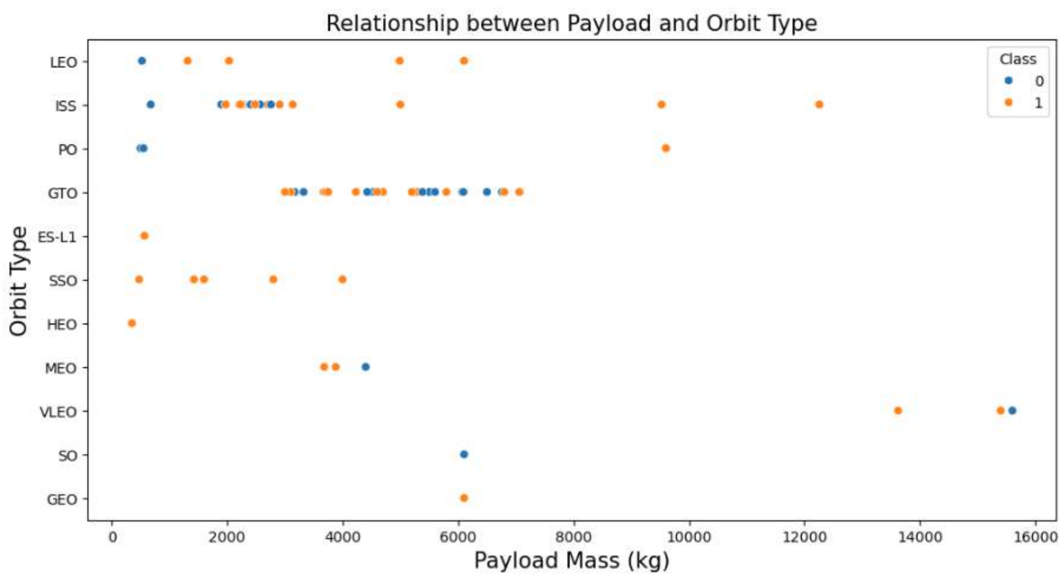
```python
### TASK  4: Visualize the relationship between FlightNumber and Orbit type
# Plot a categorical plot to visualize the relationship between FlightNumber and Orbit type
plt.figure(figsize=(12, 6))
sns.catplot(x="FlightNumber", y="Orbit", data=df, kind="swarm", aspect=3)
plt.xlabel("Flight Number", fontsize=15)
plt.ylabel("Orbit Type", fontsize=15)
plt.title("Relationship between Flight Number and Orbit Type", fontsize=15)
plt.xticks(rotation=45)
plt.show()
```

```
<Figure size 1200x600 with 0 Axes>
```



Flight Number vs. Orbit Type

- In orbit VLEO, the first successful landing (class=1) typically doesn't occur until after 60 flights.
- Across most orbits (LEO, ISS, PO, SSO, MEO, VLEO), successful landing rates generally increase with flight numbers.
- However, there appears to be no discernible relationship between flight number and orbit for GTO.



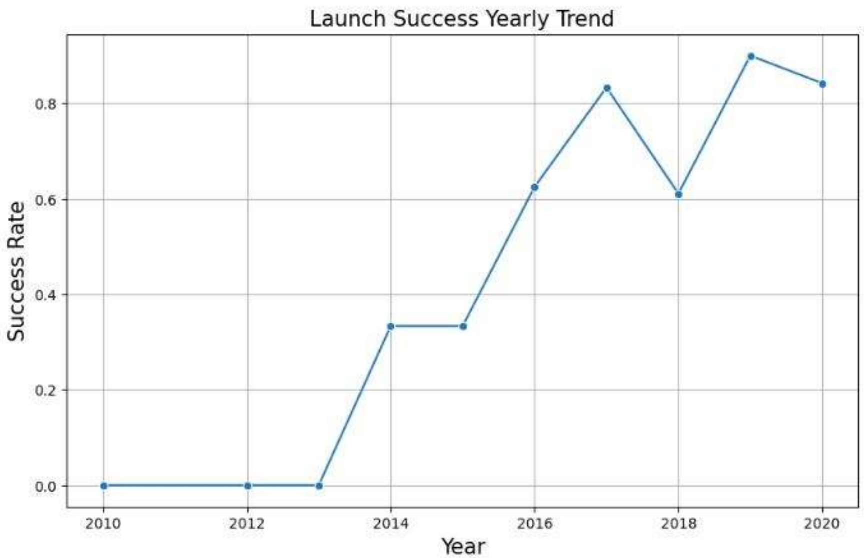
Payload vs. Orbit Type

- Successful landing rates (Class=1) show an apparent increase with payload for orbits LEO, ISS, PO, and SSO.
- Conversely, for the GEO orbit, there is no clear pattern observed between payload and orbit for either successful or unsuccessful landings.

```
### TASK  6: Visualize the launch success yearly trend
# Extract year from Date column
df['Year'] = pd.to_datetime(df['Date']).dt.year

# Calculate success rate for each year
yearly_success_rate = df.groupby('Year')['Class'].mean().reset_index()

# Plot the trend
plt.figure(figsize=(10, 6))
sns.lineplot(x='Year', y='Class', data=yearly_success_rate, marker='o')
plt.xlabel('Year', fontsize=15)
plt.ylabel('Success Rate', fontsize=15)
plt.title('Launch Success Yearly Trend', fontsize=15)
plt.grid(True)
plt.show()
```



Launch Success Yearly Trend

- The success rate (Class=1) witnessed a significant increase of approximately 80% between 2013 and 2020.
- Success rates remained stable between 2010 and 2013 and between 2014 and 2015.
- However, there was a decrease in success rates between 2017 and 2018, as well as between 2019 and 2020.

All Launch Site Names

Description:

- Utilizing the 'distinct' function retrieves only unique values from the queries column, specifically Launch_Site.
- Analysis reveals a total of 4 unique launch sites.

Task 1

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

```
%%sql
SELECT DISTINCT "launch_site" FROM SPACEXTABLE1;
```

```
 * 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'

- Description:
- Employing the keyword 'Like' with the format 'CCA%' retrieves records where the 'Launch_Site' column commences with "CCA".
- The addition of 'Limit 5' restricts the number of returned records to 5.

Task 2

Display 5 records where launch sites begin with the string 'CCA'

```
%sql SELECT * FROM SPACEXTABLE1 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_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

Task 3

Display the total payload mass carried by boosters launched by NASA (CRS)

```
%%sql
SELECT SUM("PAYLOAD_MASS__KG_") AS TotalPayloadMass
FROM SPACEXTABLE1
WHERE "Customer" LIKE 'NASA (CRS)';
```

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

TotalPayloadMass
45596

- Description:
- The function 'sum' computes the sum of the 'PAYLOAD_MASS_KG' column.
- Specifically, it returns the total payload mass for customers designated as 'NASA (CRS)'.

Average Payload Mass by F9 v1.1

- Description:
- Utilizing the 'avg' keyword calculates the average payload mass in the 'PAYLOAD_MASS_KG' column.
- This calculation specifically focuses on instances where the booster version is 'F9 v1.1'.

Task 4

Display average payload mass carried by booster version F9 v1.1

```
%%sql
SELECT AVG("PAYLOAD_MASS__KG_") AS AveragePayloadMass
FROM SPACEXTABLE1
WHERE "Booster_Version" = "F9 v1.1";
```

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

AveragePayloadMass
2928.4

First Successful Ground Landing Date

Task 5

List the date when the first succesful landing outcome in ground pad was acheived.

Hint:Use min function

```
%%sql
SELECT MIN("Date") AS FirstSuccessfulLandingDate
FROM SPACEXTABLE1
WHERE "Landing_Outcome" = 'Success (ground pad)';
```

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

FirstSuccessfulLandingDate
2015-12-22

- Description:
- The expression 'min(Date)' selects the earliest date from the 'Date' column where the first successful landing on a ground pad occurred.
- The 'Where' clause specifies the condition to retrieve the date for instances where the 'Landing_Outcome' value equals 'Success (ground pad)'.

Successful Drone Ship Landing with Payload between 4000 and 6000

- Description:
- This query identifies the booster versions where the payload mass is greater than 4000 but less than 6000, and the landing outcome is successful on a drone ship.
- The 'and' operator in the 'where' clause ensures that only booster versions meeting both conditions are returned.
-

Task 6

List the names of the boosters which have success in drone ship and have payload mass greater than 4000 but less than 6000

```
%%sql
SELECT "Booster_Version"
FROM SPACEXTABLE1
WHERE "Landing_Outcome" = 'Success (drone ship)'
AND "PAYLOAD_MASS__KG_" > 4000
AND "PAYLOAD_MASS__KG_" < 6000;
```

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

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

Total Number of Successful and Failure Mission Outcomes

- Description:
- The 'group by' keyword organizes identical data in a column into groups.
- Specifically, the number of mission outcomes is grouped by types of outcomes in the 'counts' column.

Task 7

List the total number of successful and failure mission outcomes

```
%%sql
SELECT "Mission_Outcome", COUNT(*) AS "COUNT"
FROM SPACEXTABLE1
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

- Description:
- The subquery retrieves the maximum payload mass using the 'max' keyword on the payload mass column.
- In the main query, the booster versions and their corresponding payload masses are returned where the payload mass equals the maximum value of 15600.

Task 8

List the names of the booster_versions which have carried the maximum payload mass. Use a subquery

```
%%sql
SELECT "Booster_Version"
FROM SPACEXTABLE1
WHERE "PAYLOAD_MASS__KG_" = (
    SELECT MAX("PAYLOAD_MASS__KG_")
    FROM SPACEXTABLE
);
```

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

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

2015 Launch Records

- Description:
- The query lists the landing outcome, booster version, and launch site where the landing outcome is a failure on a drone ship in the year 2015.
- The 'and' operator in the where clause ensures that both conditions are met for the returned results.
- The 'year' keyword extracts the year from the 'Date' column.
- The results identify the launch site as 'CCAFS LC-40' and the booster versions as F9 v1.1 B1012 and B1015, which experienced failed landing outcomes on a drone ship in the year 2015.

Task 9

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.

Note: SQLLite does not support monthnames. So you need to use substr(Date, 6,2) as month to get the months and substr(Date,0,5)='2015' for year.

```
%%sql
SELECT SUBSTR("Date", 6, 2) AS "Month",
       "Landing_Outcome",
       "Booster_Version",
       "Launch_Site"
FROM SPACEXTABLE
WHERE SUBSTR("Date", 0, 5) = '2015'
      AND "Landing_Outcome" = 'Failure (drone ship)';
```

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

Month	Landing_Outcome	Booster_Version	Launch_Site
01	Failure (drone ship)	F9 v1.1 B1012	CCAFS LC-40
04	Failure (drone ship)	F9 v1.1 B1015	CCAFS LC-40

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

- Description:
- The 'group by' keyword organizes data in the 'Landing_Outcome' column into groups.
- The 'between' and 'and' keywords filter data between the dates 2010-06-04 and 2017-03-20.
- The 'order by' keyword sorts the counts column in descending order.
- The query yields a ranked list of landing outcome counts per the specified date range.

Task 10

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 "Count"
FROM SPACEXTABLE1
WHERE "Date" BETWEEN '2010-06-04' AND '2017-03-20'
GROUP BY "Landing_Outcome"
ORDER BY "Count" DESC;
```

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

Landing_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

Section 3

Launch Sites Proximities Analysis

SpaceX Falcon9 - Launch Sites Map

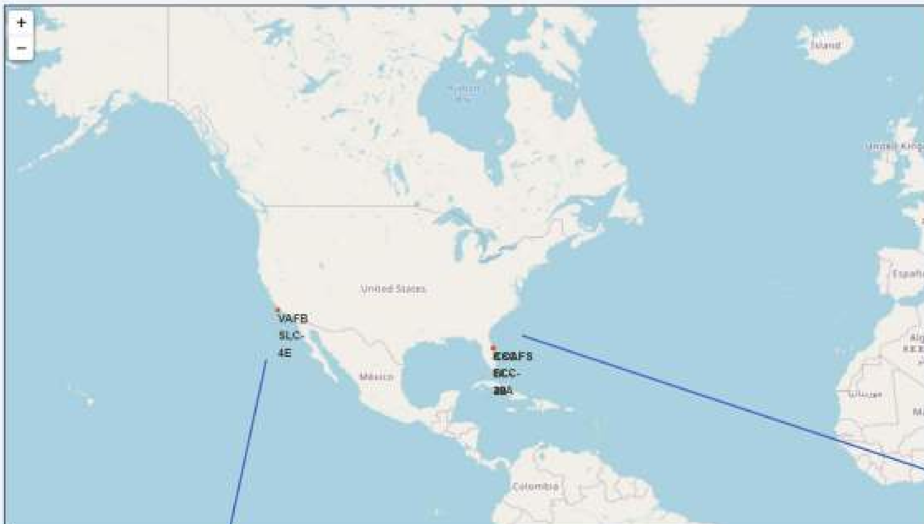


Fig 1 – Global Map

Fig 2 – Zoom 1

Figure 1 on left displays the Global map with Falcon 9 launch sites that are located in the United States (in California and Florida). Each launch site contains a circle, label, and a popup to highlight the location and the name of the launch site. It is also evident that all launch sites are near the coast.

Figure 2 and Figure 3 zoom in to the launch sites to display 4 launch sites:

- VAFB SLC-4E (CA)
- CCAFS LC-40 (FL)
- KSC LC-39A (FL)
- CCAFS SLC-40 (FL)

Fig 3 – Zoom 2

SpaceX Falon9 – Success/Failed Launch Map for all Launch Sites

- Figure 1 is the US map with all the Launch Sites. The numbers on each site depict the total number of successful and failed launches
- Figure 2, 3, 4, and 5 zoom in to each site and displays the success/fail markers with green as success and red as failed
- By looking at each site map, KSC LC-39A Launch Site has the greatest number of successful launches

Fig 1 – US map with all Launch Sites

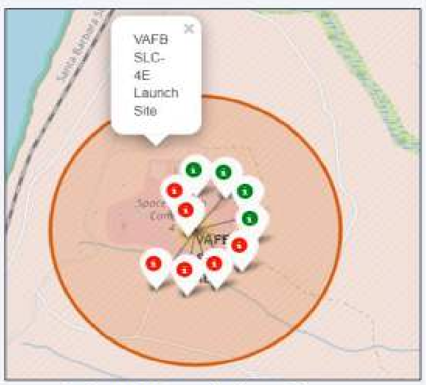


Fig 2 – VAFB Launch Site with success/failed markers

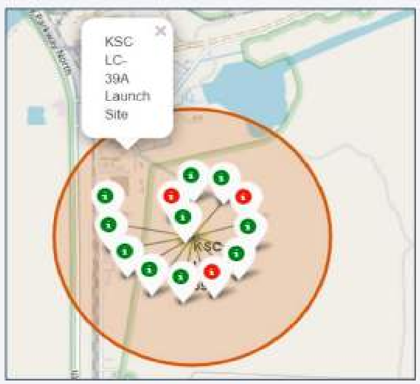


Fig 3 – KSC LC-39A success/failed markers

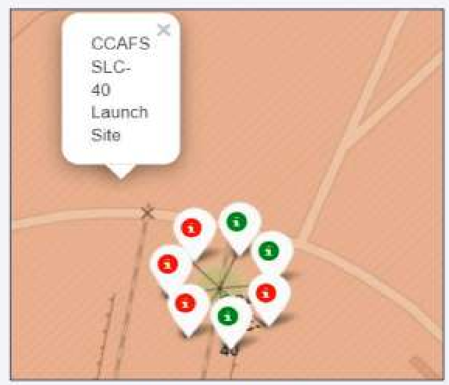


Fig 4 – CCAFS SLC-40 success/failed markers

Fig 5 – CCAFS SLC-40 success/failed markers

SpaceX Falcon9 – Launch Site to proximity Distance Map

Fig 1 – Proximity site map for VAFB SLC-4E

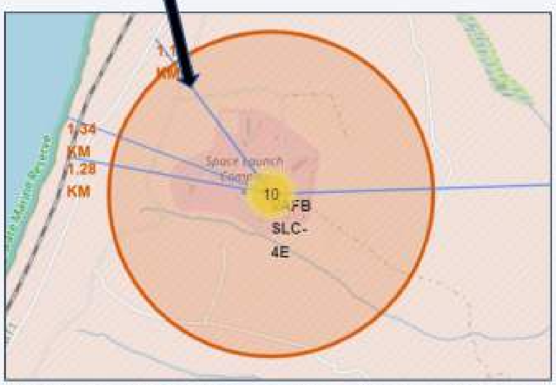
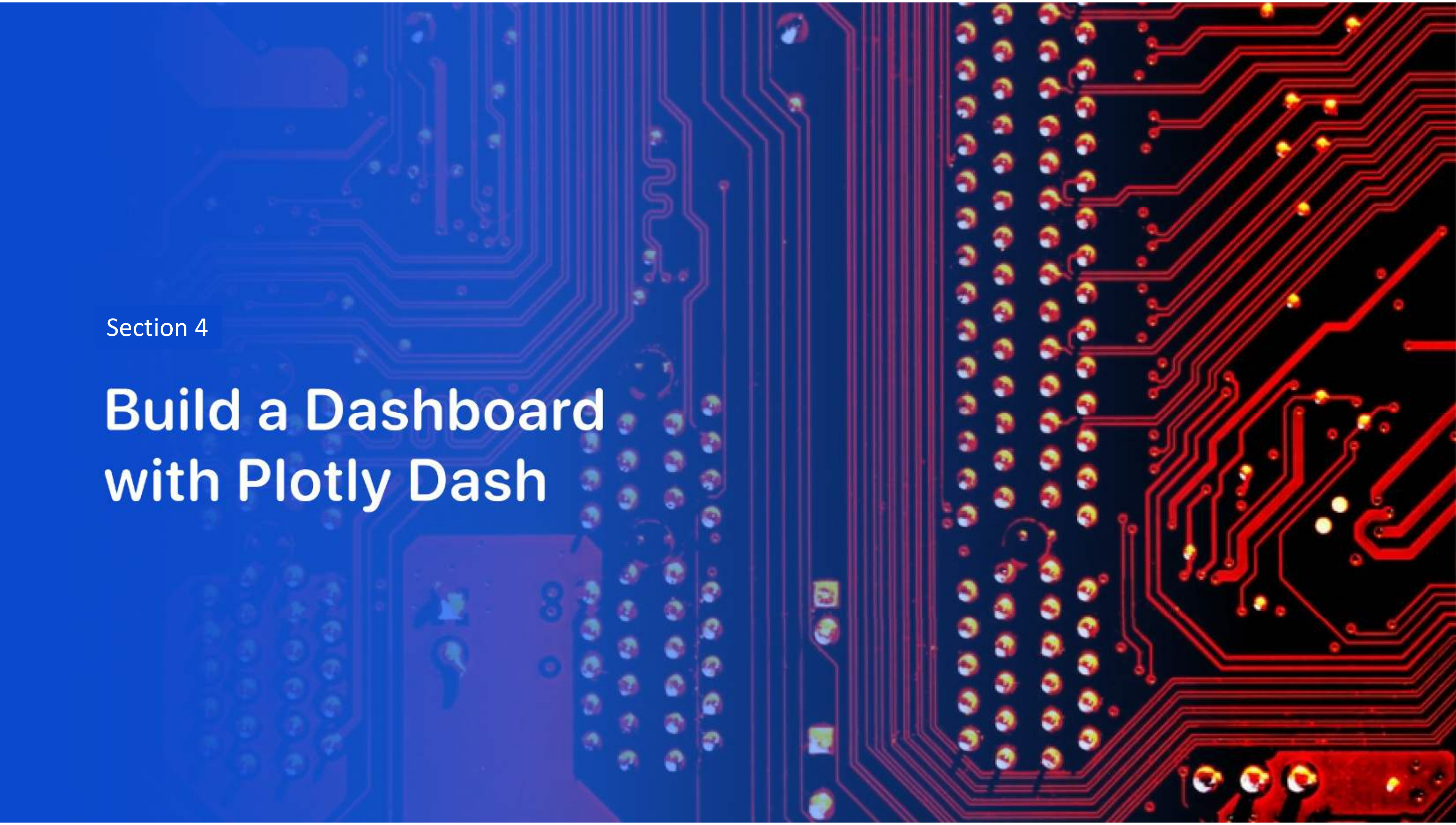


Fig 2 – Zoom in for sites – coastline, railroad, and highway

Figure 1 displays all the proximity sites marked on the map for Launch Site VAFB SLC-4E. City Lompoc is located further away from Launch Site compared to other proximities such as coastline, railroad, highway, etc. The map also displays a marker with city distance from the Launch Site (14.09 km)

Figure 2 provides a zoom in view into other proximities such as coastline, railroad, and highway with respective distances from the Launch Site

In general, cities are located away from the Launch Sites to minimize impacts of any accidental impacts to the general public and infrastructure. Launch Sites are strategically located near the coastline, railroad, and highways to provide easy access to resources.



Section 4

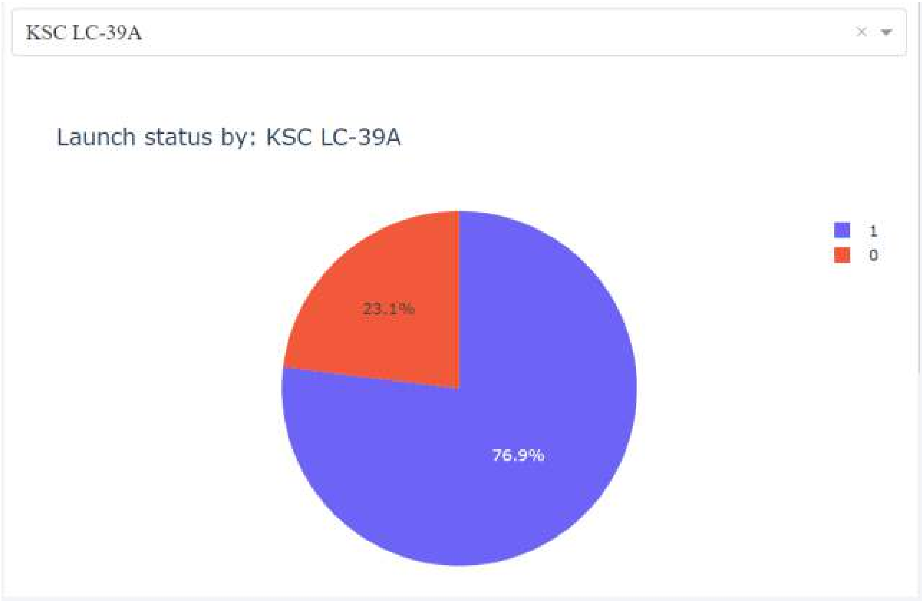
Build a Dashboard with Plotly Dash

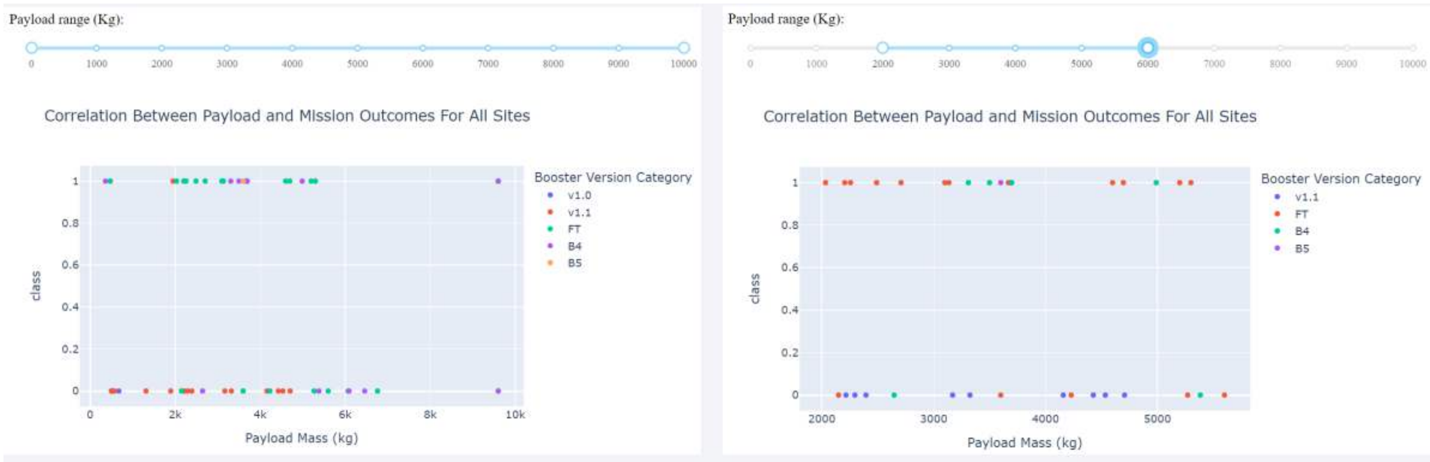
Launch Success Counts For All Sites

- Launch Site 'KSC LC-39A' boasts the highest launch success rate.
- Conversely, Launch Site 'CCAFS SLC-40' exhibits the lowest launch success rate.

Launch Site with Highest Launch Success Ratio

- The KSC LC-39A Launch Site achieves both the highest launch success rate and count.
- The launch success rate stands at 76.9%.
- Meanwhile, the launch failure rate is 23.1%.





- The majority of successful launches fall within the payload range from 2000 to approximately 5500.
- Among booster version categories, 'FT' records the highest number of successful launches.
- Interestingly, the only booster with a successful launch when the payload exceeds 6000 kg is 'B4'.

Payload vs. Launch Outcome Scatter Plot for All Sites

Section 5

Predictive Analysis (Classification)

Classification Accuracy

- The Decision Tree algorithm achieves the highest classification score of 0.8750, as indicated by the Accuracy scores and confirmed by the bar chart.
- The Accuracy Score remains consistent across all classification algorithms on the test data, with a value of 0.8333.
- Since the Accuracy scores for classification algorithms are very close and the test scores are identical, it suggests the necessity of a broader dataset for further model tuning.

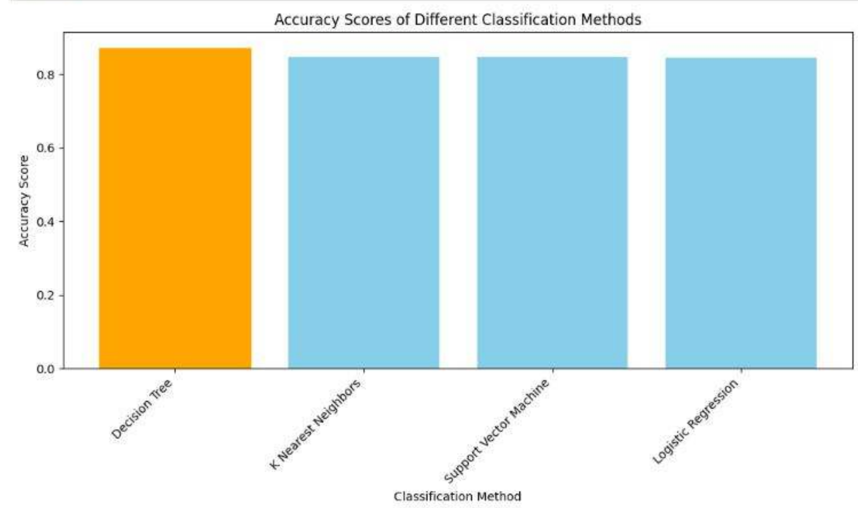
```python
# Sort the accuracy scores
sorted_scores = sorted(accuracy_scores.items(), key=lambda x: x[1], reverse=True)
methods, scores = zip(*sorted_scores)

# Create a bar graph
plt.figure(figsize=(10, 6))
bars = plt.bar(methods, scores, color='skyblue')

# Highlight the best-performing method
best_method = sorted_scores[0][0]
best_score = sorted_scores[0][1]
for bar in bars:
    if bar.get_height() == best_score:
        bar.set_color('orange')
        break

# Add title and labels
plt.title('Accuracy Scores of Different Classification Methods')
plt.xlabel('Classification Method')
plt.ylabel('Accuracy Score')

# Show plot
plt.xticks(rotation=45, ha='right')
plt.tight_layout()
plt.show()
```

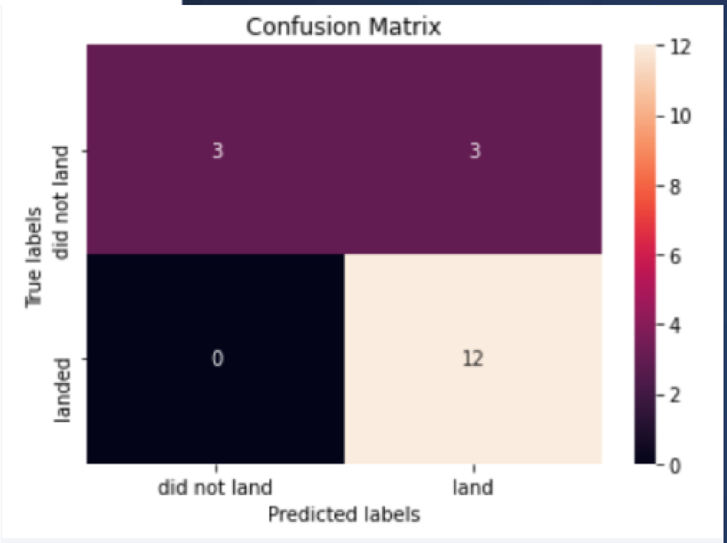


Confusion Matrix

-

The confusion matrix remains consistent across all models (LR, SVM, Decision Tree, KNN).

- According to the confusion matrix, the classifier made a total of 18 predictions.
- Out of these, 12 scenarios were predicted as 'Yes' for landing, and they did land successfully (True positive).
- Additionally, 3 scenarios were correctly predicted as 'No' for landing, and they did not land (True negative).
- However, there were also 3 scenarios where the classifier predicted 'Yes' for landing, but they did not land successfully (False positive).
- Overall, the classifier's accuracy is approximately 83% ((TP + TN) / Total), with a misclassification or error rate of about 16.5% ((FP + FN) / Total).



Conclusions

- With an increase in the number of flights, the first stage is more likely to land successfully.
- While success rates tend to rise with increasing payload, there's no clear correlation between payload mass and success rates.
- The launch success rate surged by approximately 80% from 2013 to 2020.
- KSC LC-39A boasts the highest launch success rate, whereas CCAFS SLC-40 has the lowest.
- Orbits ES-L1, GEO, HEO, and SSO exhibit the highest launch success rates, while GTO has the lowest.
- Launch sites are strategically located away from cities and closer to coastline, railroads, and highways.
- The Decision Tree emerges as the best-performing Machine Learning Classification Model with an accuracy of about 87.5%. However, all models achieved an accuracy score of about 83% when scored on the test data. Further data may be required for fine-tuning the models and potentially finding a better fit.

Appendix

GITHUB: REPOSITORY URL: HTTPS://GITHUB.COM/SOMBIR1/FINAL-PROJECT

SPECIAL THANKS TO COURSERA!!


Thank you!