IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

Alexandro Cerezo Nagase
18/09/2024

Outline

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

Executive Summary

This study aims to analyze SpaceX Falcon 9 data from multiple sources and leverage machine learning models to forecast the success of first-stage landings, thereby aiding other space agencies in making informed decisions about competing with SpaceX.

- Summary of methodologies

Data collection and analysis for this research involved:

- **Data acquisition:** Utilizing APIs and web scraping techniques to gather relevant information.
- **Data preparation:** Cleaning and transforming the data through data wrangling processes.
- **Exploratory data analysis:** Employing SQL queries and data visualizations to gain insights into the data.
- **Geographic analysis:** Creating an interactive map with Folium to examine the proximity of launch sites.
- **Interactive dashboards:** Building a dashboard using Plotly Dash to explore launch records dynamically.
- **Predictive modeling:** Developing a machine learning model to forecast the successful landing of the Falcon 9's first stage.

- Summary of all results

- This report will present findings in the form of data analysis results, data visualizations, interactive dashboards, and predictive model analysis.

Introduction

- Project background and context

The recent advancements in private space travel have made the space industry increasingly accessible to the general public. While the cost of launch remains a significant hurdle for new entrants, SpaceX has gained a competitive edge through its reusable first-stage technology. Each SpaceX launch costs approximately $62 million, and the ability to reuse the first stage for future missions provides a substantial cost advantage over competitors, who typically spend around $165 million per launch.

- Problems you want to find answers

To accurately forecast SpaceX Falcon 9 first-stage landing outcomes, we'll examine key factors such as:

- **Payload mass:** Heavier payloads increase landing complexity.
- **Launch site:** Site-specific conditions (e.g., wind, temperature) influence landings.
- **Flight history:** Repeated flights can introduce wear and tear.
- **Orbit type:** Different orbits require unique landing trajectories.
- **Analyzing trends over time:** We'll assess if success rates have improved.
- **Predictive modeling:** Using algorithms like logistic regression, SVM, random forest, and GBM, we'll create models to predict landing success based on these factors.
- **Correlations:** We'll explore relationships between launch sites and success rates.
- **Goal:** Develop a robust model to optimize SpaceX operations and predict future landing outcomes.

Section 1

Methodology

Methodology

Executive Summary

- Data collection methodology:

  Data was collected from SpaceX API.

  From Wikipedia using Web Scrapping

- Perform data wrangling

  Labeling Mission Outcomes

  To train supervised models, we need to convert mission outcomes into binary labels:

  0: Unsuccessful landing

  1: Successful landing

  Data Filtering and Cleaning

  Filtering: Remove any irrelevant or redundant data points.

  Missing values: Handle missing values using techniques like imputation (e.g., mean, median, mode) or dropping rows/columns with excessive missing data.

- Perform exploratory data analysis (EDA) using visualization and SQL
- Perform interactive visual analytics using Folium and Plotly Dash
- Perform predictive analysis using classification models

Train models capable of predicting the success of Falcon 9 first stage landings, creating "class" column with binary labels (0: failure, 1: success). We experiment with various classification algorithms (logistic regression, decision trees, KNN and SVM), adjusting their hyperparameters to optimize their performance. Evaluate the models using metrics such as accuracy, recall, and F1-score on the testing set to select the most suitable model.

Data Collection

- Describe how data sets were collected

For this analysis, two data sources were combined: SpaceX's REST API and web scraping of a specific table on SpaceX's Wikipedia page. This dual strategy ensured complete data coverage for a more in-depth analysis.

Data Columns

From SpaceX's REST API: FlightNumber, Date, BoosterVersion, PayloadMass, Orbit, LaunchSite, Outcome, Flights, GridFins, Reused, Legs, LandingPad, Block, ReusedCount, Serial, Longitude, Latitude.

From Wikipedia: Flight No., Launch site, Payload, PayloadMass, Orbit, Customer, Launch outcome, Version Booster, Booster landing, Date, Time.

By combining these data sources, a robust dataset was created for the análisis.

Data Collection – SpaceX API

1.

```
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.

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

3.

```
getBoosterVersion(data)
getLaunchSite(data)
getPayloadData(data)
getCoreData(data)
```

4.

```
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}
```

5.

```
# 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)
```

[Linked to Github]()

Data Collection - Scraping

1.

```python
static_url = "https://en.wikipedia.org/w/index.php?title=List_of_Falcon_9_and_Falcon_Heavy_launches&oldid=1027686922"
```

```python
html_data = requests.get(static_url).text
```

2.

```python
soup = BeautifulSoup(html_data,"html.parser")
```

3.

```python
html_tables = soup.find_all ('table')
```

```python
column_names = []

# Apply find_all() function with `th` element on firs
# Iterate each th element and apply the provided extr
# Append the Non-empty column name (`if name is not N
colnames = soup.find_all('th')
for x in range (len(colnames)):
    name2 = extract_column_from_header(colnames[x])
    if (name2 is not None and len(name2) > 3):
        column_names.append(name2)
```

4.

```python
launch_dict= dict.fromkeys(column_names)

# Remove an irrelvant column
del launch_dict['Date and time ( )']

# Let's initial the launch_dict with each va
launch_dict['Flight No.'] = []
launch_dict['Launch site'] = []
launch_dict['Payload'] = []
launch_dict['Payload mass'] = []
launch_dict['Orbit'] = []
launch_dict['Customer'] = []
launch_dict['Launch outcome'] = []
# Added some new columns
launch_dict['Version Booster']=[]
launch_dict['Booster landing']=[]
launch_dict['Date']=[]
launch_dict['Time']=[]
```

5.

```python
def date_time(table_cells):
```

```python
def booster_version(table_cells):
```

```python
def landing_status(table_cells):
```

```python
def get_mass(table_cells):
```

6.

```python
df=pd.DataFrame(launch_dict)
```

Linked to Github

Data Wrangling

To gain insights into the data and prepare it for supervised learning, we conducted an exploratory data analysis (EDA). During this process, we defined labels for training models based on mission outcomes.

Label Definition

1: Successful landing

0: Unsuccessful landing

Mission outcomes were classified as follows:

True Ocean: Successful landing in a specific ocean region

False Ocean: Unsuccessful landing in a specific ocean region

RTLS: Successful landing on a ground pad

False RTLS: Unsuccessful landing on a ground pad

True ASDS: Successful landing on a drone ship

False ASDS: Unsuccessful landing on a drone ship

These labels provided a clear target variable for our machine learning models.

Data Wrangling

1.

```
df=pd.read_csv("https://cf-courses-data.s3.us.cloud-object-storage.appd
art_1.csv")
```

2.

```
df['LaunchSite'].value_counts()
```

3.

```
df['Orbit'].value_counts()
```

```
GTO      27
ISS      21
VLEO     14
PO        9
LEO       7
SSO       5
MEO       3
GEO       1
HEO       1
SO        1
ES-L1     1
```

4.

```
landing_outcomes = df['Outcome'].value_counts()
```

5.

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

landing_class = []
for i in df['Outcome']:
    if i in bad_outcomes:
        landing_class.append(0)
    else:
        landing_class.append(1)
```

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

	Class
0	0
1	0
2	0
3	0
4	0

Linked to Github

EDA with Data Visualization

- Summarize what charts were plotted and why you used those charts

Scatter plots are used to visualize the relationship between two variables. They can reveal patterns or correlations that might be useful for machine learning models. The texts mention several scatter plots that were created to analyze relationships between flight number, launch site, payload, and orbit type.

Bar charts are effective for comparing different categories. They can show which groups are the highest or most common and how they compare to each other. The texts describe using bar charts to visualize the success rates of different orbit types.

Line charts are ideal for tracking changes over time. They can help identify trends and patterns in data. The texts explain using line charts to observe the yearly trend in average launch success.

Linked to Github

EDA with SQL

- Using bullet point format, summarize the SQL queries you performed

In essence, a series of SQL queries were executed on a SpaceX dataset stored in an IBM DB2 cloud instance to extract meaningful insights.

These queries allowed us to:

Identify: Unique launch sites, specific missions (such as NASA's CRS missions), and distinct rocket versions.

Compare: Different landing types (on land or on droneships) and the payload mass carried by each rocket.

Analyze trends: Such as the evolution of launch success rates over time.

Linked to Github

Build an Interactive Map with Folium

- Summarize what map objects such as markers, circles, lines, etc. you created and added to a folium map

An interactive map was created using Folium to analyze launch site data. This map allows for a deeper understanding of how location and proximity to certain features like railways, highways, and coastlines might affect launch success rates.

The map includes:

Markers for all launch sites, with details like text labels and circles highlighting their locations.

Colored markers (green for success, red for failure) grouped by launch site to show success rates visually.

Distance measurements between launch sites and nearby features like railways, highways, coastlines, and cities.

- Explain why you added those objects

By analyzing this interactive map, we can answer questions about the proximity of launch sites to various features and their potential impact on launch success.

Linked to Github

Build a Dashboard with Plotly Dash

- Summarize what plots/graphs and interactions you have added to a dashboard

The Dash web application provides an interactive platform for analyzing SpaceX launch data in real-time.

Key features and functionalities:

Launch Site Dropdown: Allows users to filter data by specific launch sites or view overall statistics.

Pie Chart: Visualizes the success and failure rates for selected launch sites.

Payload Range Slider: Enables users to explore data within different payload ranges.

Scatter Chart: Displays the correlation between payload mass and mission outcomes, color-coded by booster version.

Insights gained from the dashboard:

Most successful launch site: KSC LC-39A with 10 successful launches.

Highest launch success rate: KSC LC-39A with 76.9% success.

Optimal payload range: 2000-5000 kg for highest success rate.

Least successful payload range: 0-2000 kg and 5500-7000 kg.

Best-performing booster version: FT.

Linked to Github

- Explain why you added those plots and interactions

Overall, the dashboard offers a valuable tool for understanding SpaceX launch data and identifying key trends and patterns.

Predictive Analysis (Classification)

1.

```
data = pd.read_csv("https://cf-courses-data.s3.us.cloud-object
et part 2.csv")
```

```
Y = data['Class'].to_numpy()
```

2.

```
X= preprocessing.StandardScaler().fit(X).transform(X)
```

3.

```
# Split data for training and testing data sets
from sklearn.model_selection import train_test_split
X_train, X_test, Y_train, Y_test = train_test_split
( X, Y, test_size=0.2, random_state=2)
print ('Train set:', X_train.shape,  Y_train.shape)
print ('Test set:', X_test.shape,  Y_test.shape)
```

4.

```
parameters ={"C":[0.01,0.1,1],'penalty':['l2'], 'solver':['lbfgs']}
LR = LogisticRegression()
logreg_cv = GridSearchCV(LR, parameters,cv=10)
logreg_cv.fit(X_train, Y_train)
```

5.

```
print("tuned hpyerparameters :(best parameters) ",logreg_cv.best_params_)
print("accuracy :",logreg_cv.best_score_)
```

6.

```
yhat=logreg_cv.predict(X_test)
plot_confusion_matrix(Y_test,yhat)
```

7.

```
Model_Performance_df = pd.DataFrame({'Algo Type': ['Logistic Regression', 'SVM','Decision Tree','KNN'],
'Accuracy Score': [logreg_cv.best_score_, svm_cv.best_score_, tree_cv.best_score_, knn_cv.best_score_],
'Test Data Accuracy Score': [logreg_cv.score(X_test, Y_test), svm_cv.score(X_test, Y_test),
tree_cv.score(X_test, Y_test), knn_cv.score(X_test, Y_test)]})
```

```
i = Model_Performance_df['Accuracy Score'].idxmax()
print('The best performing alogrithm is '+ Model_Performance_df['Algo Type'][i]
+ ' with score ' + str(Model_Performance_df['Accuracy Score'][i]))
```

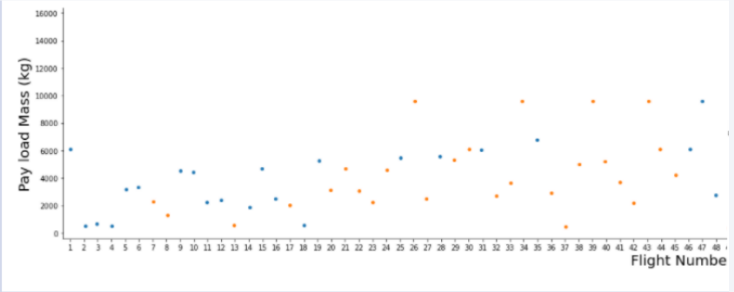
```
The best performing alogrithm is Decision Tree with score 0.875
```

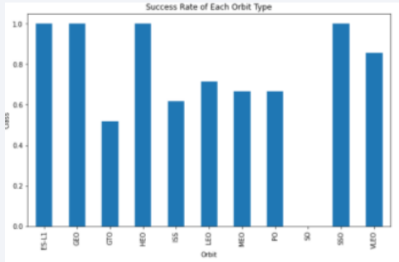
	Algo Type	Accuracy Score	Test Data Accuracy Score
2	Decision Tree	0.875000	0.833333
3	KNN	0.848214	0.833333
1	SVM	0.848214	0.833333
0	Logistic Regression	0.846429	0.833333

Linked to Github

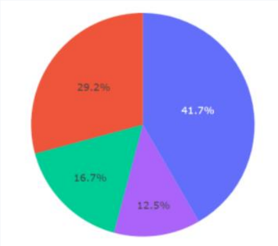
Results

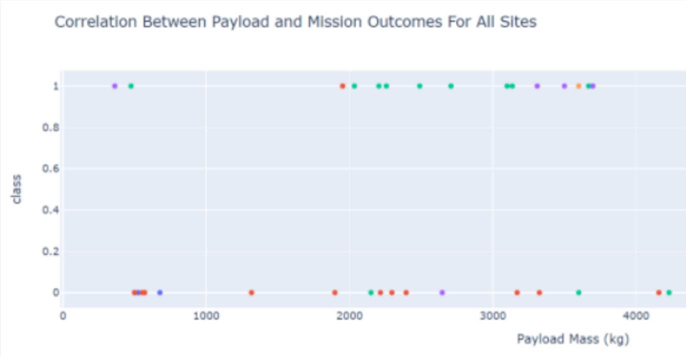
- Exploratory data analysis results





- Interactive analytics demo in screenshots





- Predictive analysis results

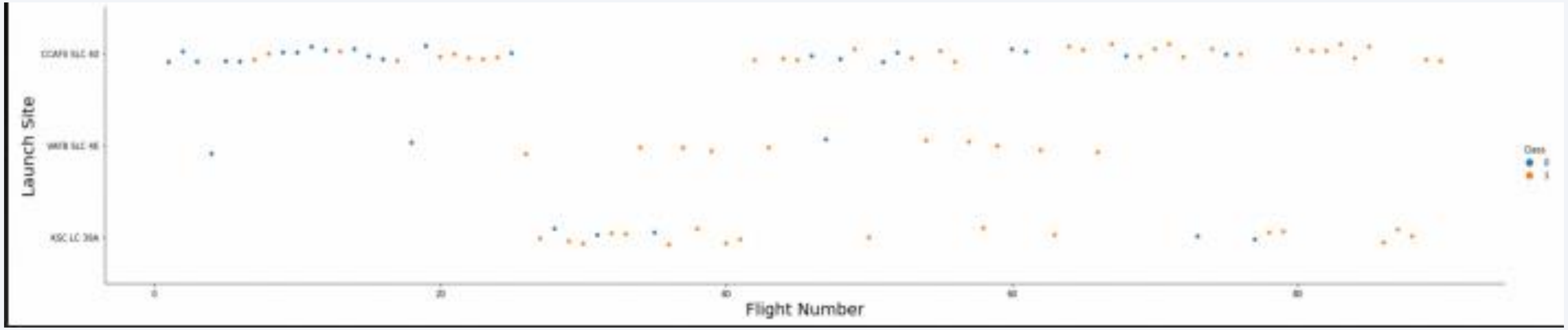
	Algo Type	Accuracy Score
2	Decision Tree	0.903571
3	KNN	0.848214
1	SVM	0.848214
0	Logistic Regression	0.846429

Section 2

Insights drawn from EDA

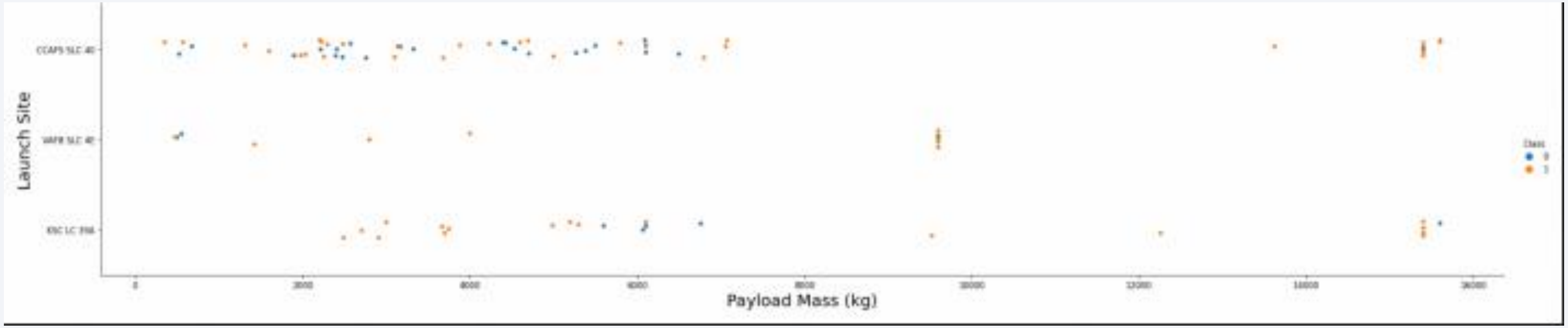
Flight Number vs. Launch Site

The early flights of space exploration were unsuccessful, while the more recent ones have been mostly successful.
The launch site CCAFS SLC 40 has accounted for approximately half of all launches, but VAFB SLC 4E and KSC LC 39A have higher success rates.
It appears that each new launch increases the overall success rate, and for KSC LC 39A, it typically takes about 25 launches before a successful one occurs.



Payload vs. Launch Site

The success rate of a launch increases with the payload mass, and most launches carrying payloads over 7000 kg have been successful.
KSC LC 39A has a perfect success rate for payloads under 5500 kg, while VAFB SLC 4E has not launched any rockets with payloads exceeding 10,000 kg.
The success rate of VAFB SLC 4E increases with the payload mass, but there is no clear relationship between launch site and payload mass in general.

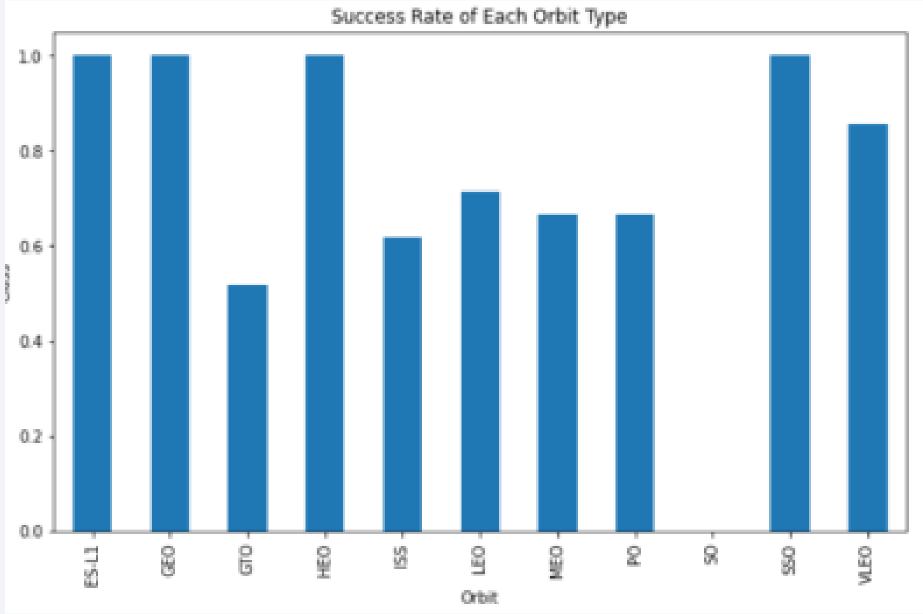


Success Rate vs. Orbit Type

Orbits like ES-L1, GEO, HEO, and SSO have a 100% success rate, while SO orbits have a 0% success rate.

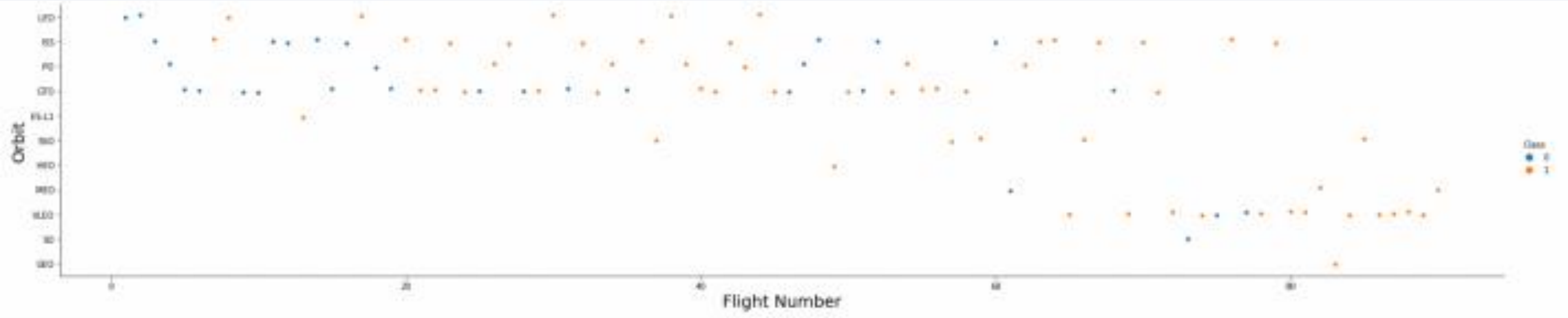
Orbits such as GTO, ISS, LEO, MEO, and PO have success rates between 50% and 85%.

ES-L1, GEO, HEO, and SSO are the most successful orbits, while GTO has the lowest success rate.



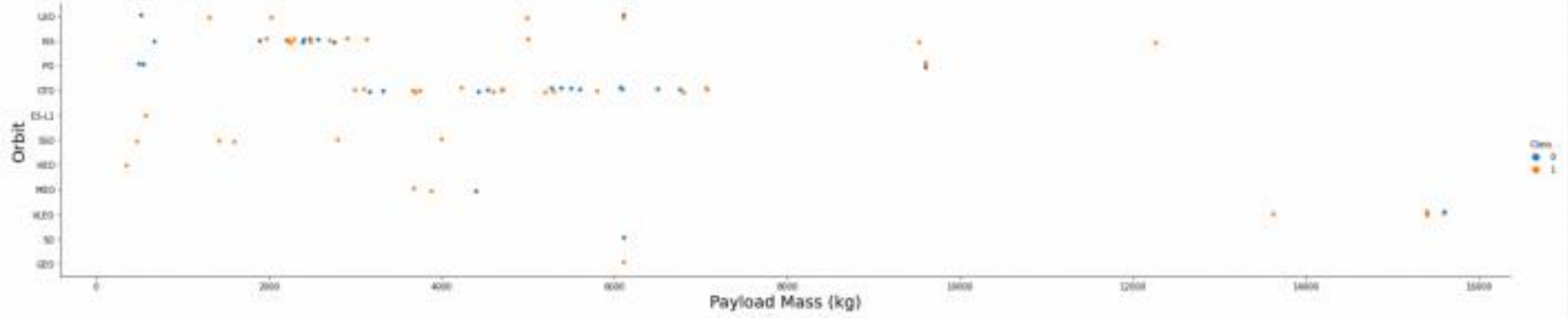
Flight Number vs. Orbit Type

The success of LEO orbits seems to be linked to the number of flights, while there appears to be no correlation between flight number and success in GTO orbits.
For most orbits, including LEO, ISS, PO, SSO, MEO, and VLEO, successful landing rates tend to increase with the number of flights.
However, there is no such relationship between flight number and orbit for GTO.



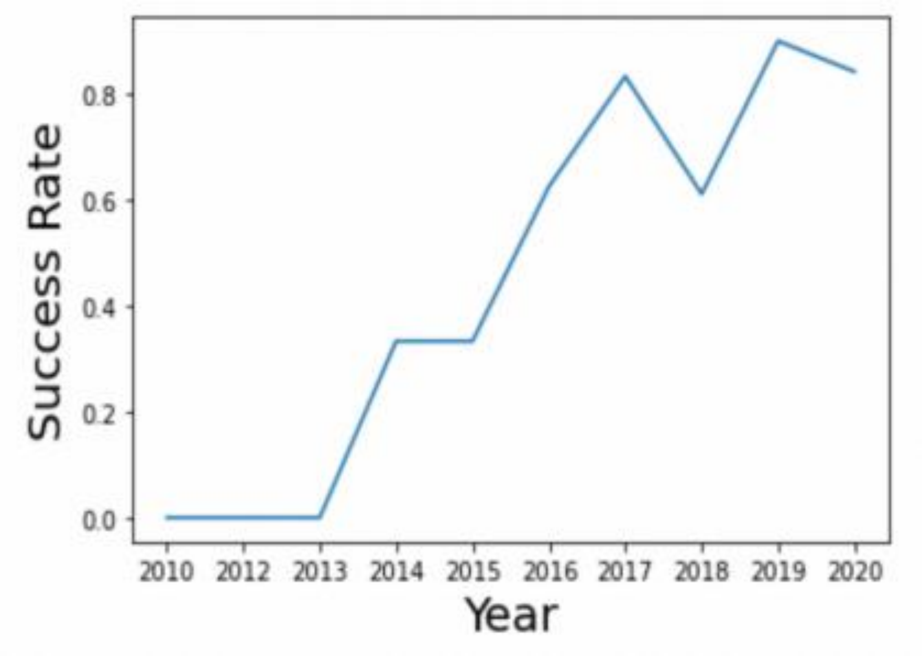
Payload vs. Orbit Type

Heavy payloads have a negative impact on GTO orbits but a positive impact on GTO and Polar LEO (ISS) orbits.
In orbits like LEO, ISS, PO, and SSO, successful landing rates tend to increase with payload mass.
However, there is no clear relationship between payload and orbit for successful or unsuccessful landings in GEO orbits.



Launch Success Yearly Trend

The success rate increased significantly between 2013 and 2020, with an increase of about 80%.
However, the success rate remained stable between 2010 and 2013 and between 2014 and 2015.
Additionally, the success rate decreased between 2017 and 2018 and between 2019 and 2020.



All Launch Site Names

```
select distinct Launch_Site from spacextbl
```

- There are four distinct launch sites used in the space missions.

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

Launch Site Names Begin with 'CCA'

```
select * from spacextbl where Launch_Site LIKE 'CCA%' limit 5;
```

- This query selects all columns (*) from the SPACEXTBL table where the Launch_site column starts with “CCA”.
- The LIKE ‘CCA%’ condition ensures that only records where the launch site begins with “CCA” area included.
- The limit 5 clause limits the results to the first 5 rows.

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	07:44:00	F9 v1.0 B0005	CCAFS LC-40	Dragon demo flight C2	525	LEO (ISS)	NASA (COTS)	Success	No attempt
2012-10-08	00:35:00	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

```
select sum(PAYLOAD_MASS__KG_) from spacextbl where Customer = 'NASA (CRS)'
```

- This query calculates the sum of the PAYLOAD_MASS_KG column for all rows where the Customer column equals "NASA (CRS)".
- The SUM() function aggregates the values, and the AS Total_Payload_Mass clause assigns an alias to the calculated value for better readability

45596

Average Payload Mass by F9 v1.1

```
select avg(PAYLOAD_MASS__KG_) from spacextbl where Booster_Version LIKE 'F9 v1.1'
```

- This query calculates the average of the PAYLOAD_MASS_KG column for all rows where the Booster_Version column equals “F9 v1.1”.
- The AVG() function aggregates the values, and the AS Average_Payload_Mass clause assigns an alias to the calculated value for better readability.

2534

First Successful Ground Landing Date

```
select min(Date) as min_date from spacextbl where Landing__Outcome = 'Success (ground pad)';
```

- This query selects the minimum (earliest) date from the Date column for all rows where the Landing_Outcome column equals "Success (ground pad)".
- The MIN() function finds the smallest value, and the AS First_Successful_Landing_Date clause assigns an alias to the calculated value for better readability.

2015-12-22

Successful Drone Ship Landing with Payload between 4000 and 6000

```
select Booster_Version from spacextbl where (PAYLOAD_MASS__KG_> 4000 and PAYLOAD_MASS__KG_ < 6000)
and (Landing__Outcome = 'Success (drone ship)');
```

- This query selects the Booster_Version column for all rows where the PAYLOAD_MASS_KG column is greater than 4000 and less than 6000, and where the Landing_Outcome column equals "Success (drone ship)".
- The AND operator ensures that both conditions must be true for a row to be included in the results.

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

Total Number of Successful and Failure Mission Outcomes

```
select Mission_Outcome, count(Mission_Outcome) as counts from spacextbl group by Mission_Outcome
```

- This query counts the occurrences of each unique value in the Mission_Outcome column.
- The COUNT(Mission_Outcome) function calculates the count, and the GROUP BY Mission_Outcome clause groups the results by the Mission_Outcome column.
- The AS Counts clause assigns an alias to the calculated count for better readability.

mission_outcome	counts
Failure (in flight)	1
Success	99
Success (payload status unclear)	1

Boosters Carried Maximum Payload

```
select Booster_Version, PAYLOAD_MASS__KG_ from spacextbl where PAYLOAD_MASS__KG_ = (select max(PAYLOAD_MASS__KG_) from spacextbl)
```

- This query first finds the maximum payload mass using a subquery: (SELECT MAX(PAYLOAD_MASS_KG) FROM SPACEXTBL).
- Then, the main query selects the Booster_Version and PAYLOAD_MASS_KG columns for all rows where the PAYLOAD_MASS_KG column is equal to the maximum value found in the subquery.
- This ensures that only the booster versions with the maximum payload mass are returned

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

```
select Landing__Outcome, Booster_Version, Launch_Site from spacextbl where Landing__Outcome = 'Failure (drone ship)' and year(Date) = '2015'
```

- This query selects the Landing_Outcome, Booster_Version, and Launch_Site columns for all rows where the Landing_Outcome column equals "Failure (drone ship)" and the year extracted from the Date column is 2015.
- The YEAR(Date) = 2015 condition ensures that only records from the year 2015 are included.

landing__outcome	booster_version	launch_site
Failure (drone ship)	F9 v1.1 B1012	CCAFS LC-40
Failure (drone ship)	F9 v1.1 B1015	CCAFS LC-40

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

```
select Landing__Outcome, count(*) as LandingCounts from spacextbl where Date between '2010-06-04' and '2017-03-20'
group by Landing__Outcome
order by count(*) desc;
```

This query:

1.**Filters** the data to include only records between the specified dates using the WHERE Date BETWEEN '2010-06-04' AND '2017-03-20' clause.

2.**Groups** the data by Landing_Outcome using the GROUP BY Landing_Outcome clause.

3.**Counts** the number of occurrences of each landing outcome using the COUNT(Landing_Outcome) AS Counts expression.

4.**Orders** the results in descending order based on the Counts column using the ORDER BY Counts DESC clause.

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

Section 3

Launch Sites Proximities Analysis

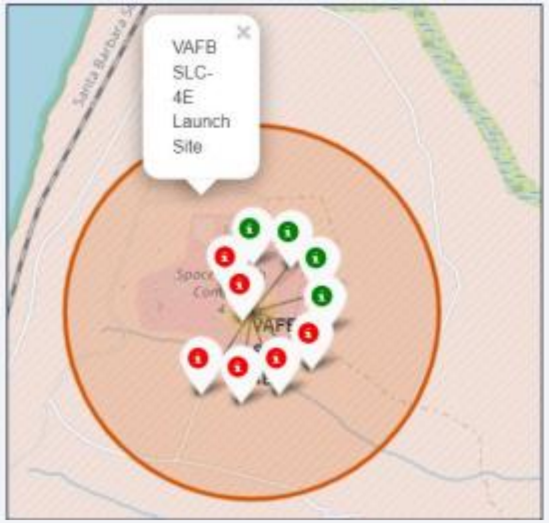
<Folium Map Screenshot 1>

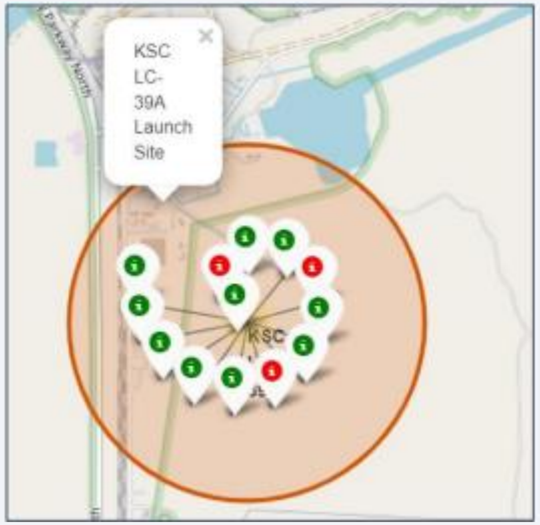
The strategic placement of rocket launch sites is driven by two primary factors:

1. **Equatorial Advantage:** Most launch sites are situated near the Equator because the Earth's rotation is fastest at this latitude. This inherent eastward velocity provides a significant boost to rockets, reducing the amount of fuel required to achieve orbital velocity. By launching eastward from the Equator, a rocket can effectively harness this rotational energy, making it more efficient and cost-effective to reach space.
2. **Coastal Safety:** To minimize the risks associated with rocket launches, including potential failures or debris falling back to Earth, launch sites are almost always located near a coastline. This ensures that any hazardous events occur over uninhabited areas of the ocean, safeguarding both people and property.

The map illustrating SpaceX's Falcon 9 launch sites in the United States highlights these principles. Both the California and Florida-based facilities are positioned along the coast, taking advantage of the Earth's rotation and minimizing potential risks. Specifically, the map shows the following launch sites: VAFB SLC-4E (CA), CCAFS LC-40 (FL), KSC LC-39A (FL), and CCAFS SLC-40 (FL).

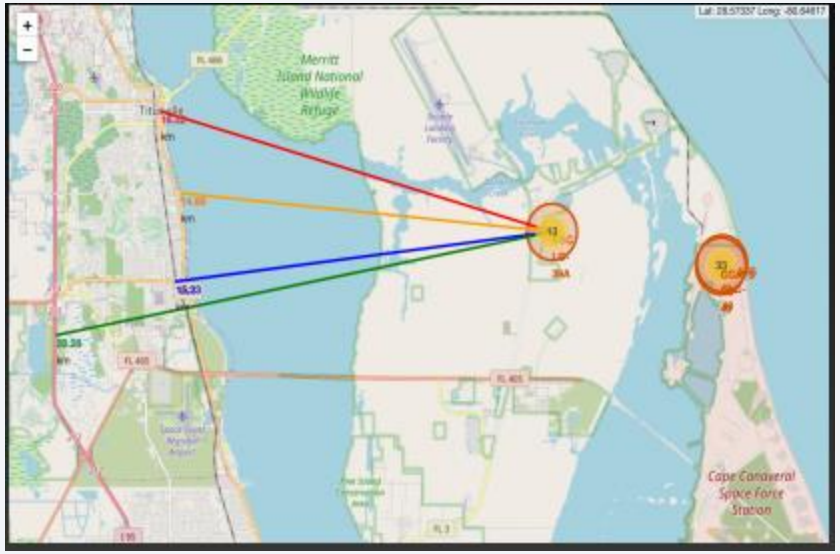
<Folium Map Screenshot 2>





•**Green markers** denote successful launches.
•**Red markers** indicate failed launches.
•**Launch Site KSC LC-39A** exhibits a notably high success rate."
This color-coding system offers a quick and easy way to compare the performance of various launch facilities.

<Folium Map Screenshot 3>



Launch Site KSC LC-39A's Proximity Analysis:

•**Located relatively near:**

- Railway (15.23 km)
- Highway (20.28 km)
- Coastline (14.99 km)
- City of Titusville (16.32 km)

Safety Considerations:

•Failed rockets can cover significant distances (15-20 km) in a short time, posing potential risks to populated areas.

•Launch sites are generally situated away from cities to minimize these risks.

Strategic Location:

•Launch sites are often located near coastlines, railways, and highways to facilitate resource access and transportation.

Launch Site VAFB SLC-4E:

•City of Lompoc is located further away from this launch site compared to other proximities.

•The map shows distances to the coastline, railroad, and highway.

Overall, the strategic placement of launch sites aims to balance accessibility and safety

Section 4

Build a Dashboard with Plotly Dash

<Dashboard Screenshot 1>

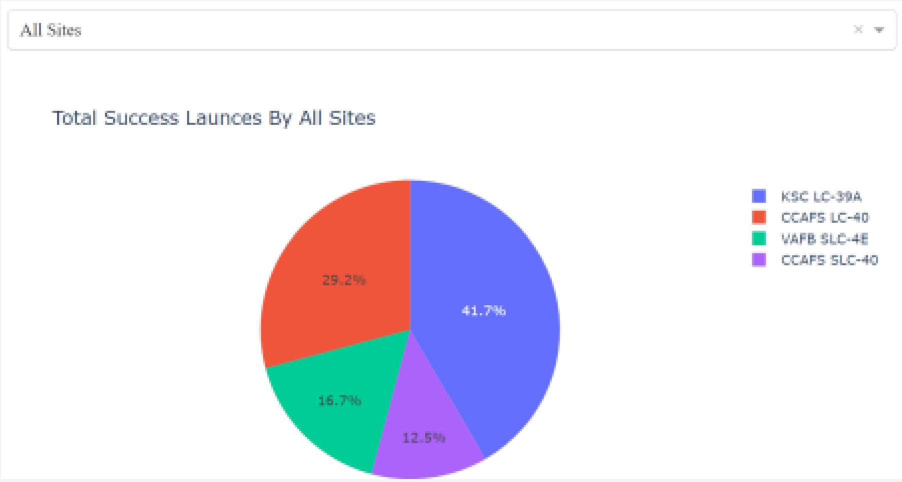
Launch Site Comparison:

•**KSC LC-39A** has the highest launch success rate.

•**CCAFS SLC-40** has the lowest launch success rate.

Data Visualization:

•The chart visually demonstrates that KSC LC-39A outperforms other launch sites in terms of successful launches.



<Dashboard Screenshot 2>

Launch Site KSC LC-39A:

•**Highest launch success rate:** 76.9%

•**Successful launches:** 10

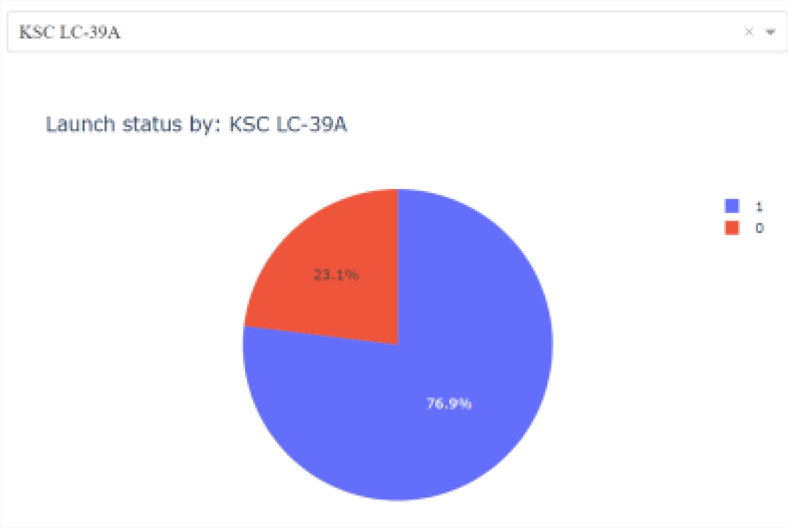
•**Failed launches:** 3

Performance Metrics:

•**Launch success rate:** 76.9%

•**Launch failure rate:** 23.1%

This data highlights the exceptional performance of KSC LC-39A in terms of launch reliability



<Dashboard Screenshot 3>

Payload and Success Rate:

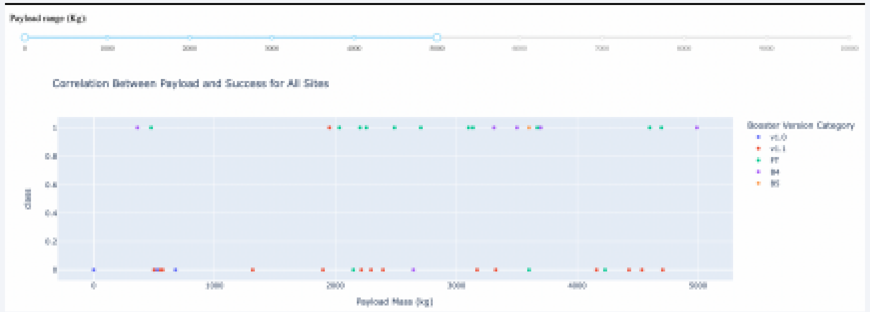
•**Optimal payload range:** 2000-5500 kg

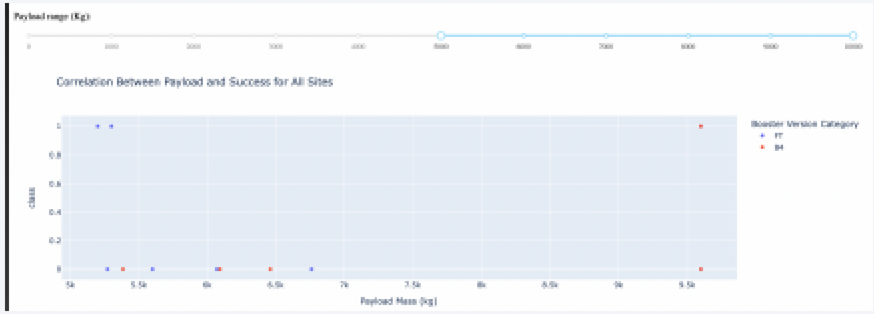
•**Booster version:** 'FT' has the most successful launches

Payload Capacity:

•Only 'B4' booster has successfully launched payloads exceeding 6,000 kg.

These findings suggest that there is an optimal payload range for achieving higher launch success rates and that specific booster versions are better suited for handling heavier payloads.



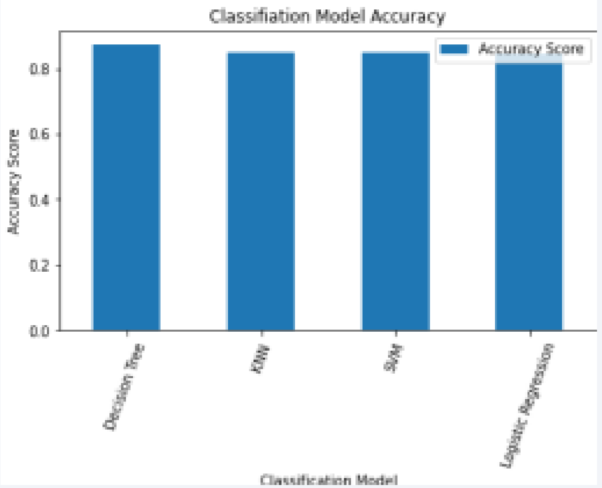


Section 5

Predictive Analysis (Classification)

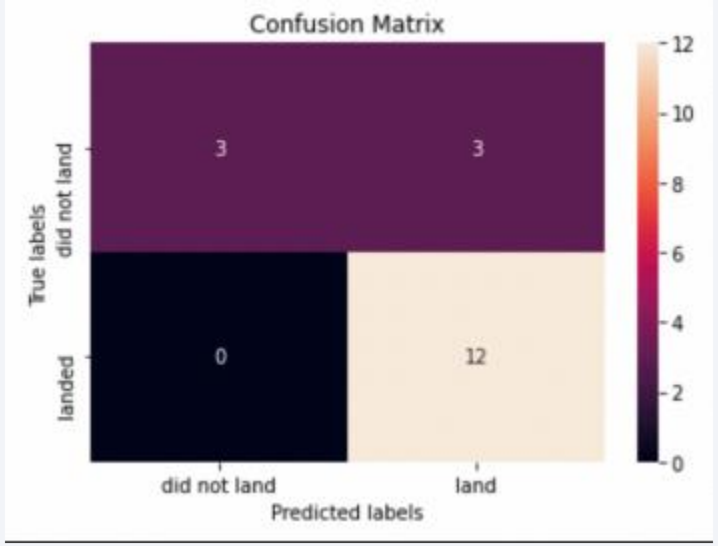
Classification Accuracy

Model Evaluation and Performance:
•**Initial test set analysis:** Unable to determine the best-performing method due to small sample size (18 samples).
•**Full dataset evaluation:** Confirmed Decision Tree Model as the superior model based on higher scores and accuracy.
Key Findings:
•**Decision Tree Model:** Highest classification score (.8750) on the full dataset.
•**Test data accuracy:** Consistent (.8333) across all classification algorithms.
•**Need for larger dataset:** Close accuracy scores and limited test data suggest the need for a more extensive dataset to refine model tuning.



	LogReg	SVM	Tree	KNN
Jaccard_Score	0.833333	0.845070	0.882353	0.819444
F1_Score	0.909091	0.916031	0.937500	0.900763
Accuracy	0.866667	0.877778	0.911111	0.855556

Confusion Matrix



Confusion Matrix Analysis:

•**Logistic Regression:** Capable of distinguishing between classes, but prone to false positives.

•**Consistent Confusion Matrix:** All models (LR, SVM, Decision Tree, KNN) exhibit the same confusion matrix.

Prediction Accuracy:

•**Total Predictions:** 18

•**True Positives:** 12

•**True Negatives:** 3

•**False Positives:** 3

Performance Metrics:

•**Overall Accuracy:** 83%

•**Misclassification Rate:** 16.5%

These results indicate that while the classifier is generally accurate, it could benefit from improvements in reducing false positive predictions.

Conclusions

Key Findings and Recommendations:

- **Decision Tree Model:** Outperforms other models for this dataset.
- **Payload Mass:** Lower payloads correlate with better results.
- **Launch Site Location:** Equatorial proximity and coastal location are common.
- **Launch Success Rate:** Improving over time.
- **KSC LC-39A:** Highest success rate among launch sites.
- **Orbits:** ES-L1, GEO, HEO, and SSO have 100% success rates.

Additional Insights:

- **Flight Frequency:** Increased flights correlate with higher first-stage landing success.
- **Payload Mass and Success:** Positive correlation, but not definitive.
- **Launch Success Improvement:** Significant increase (80%) from 2013 to 2020.
- **Model Evaluation:**
- **Decision Tree Model:** Best performance overall.
- **Test Data Accuracy:** Consistent across models.
- **Data Requirements:** More data may be necessary for further model tuning.

Overall, the analysis suggests that the Decision Tree model is well-suited for this dataset, and continued data collection and model refinement could lead to further improvements in launch prediction and optimization.

Appendix

- Include any relevant assets like Python code snippets, SQL queries, charts, Notebook outputs, or data sets that you may have created during this project

Thank you!