Space X Data Analysis

Eddie Hammond

May 6, 2024

Outline

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

Executive Summary

- **SpaceX Falcon 9 First Stage Landing Success Prediction**
- **Objective:** Leverage data from diverse sources and machine learning techniques to predict the successful landing of Falcon 9's first stage, aiding space agencies in making informed bidding decisions.
- **Research Methodology:**

1. **Data Collection:**
    - Utilize APIs and web scraping techniques.
2. **Data Preparation:**
    - Perform data wrangling to transform raw data into a structured format.
3. **Exploratory Data Analysis:**
    - Analyze data using SQL queries and visualizations.
4. **Visualization Tools:**
    - Develop an interactive map with Folium for assessing launch site locations.
    - Create a dynamic dashboard with Plotly Dash to explore launch records.
5. **Predictive Modeling:**
    - Construct a model to forecast the likelihood of successful landings for Falcon 9’s first stage.

- **Results Presentation:**
- Detailed data analysis findings.
- Engaging data visuals and interactive dashboards.
- Evaluation of the predictive model's performance.

Introduction

- **Project Background:**
- **Private Space Travel Growth:** With private space ventures achieving recent successes, the industry is increasingly becoming mainstream, lowering barriers and expanding access to the general population.
- **Economic Challenges:** Despite advancements, high launch costs remain a significant barrier for new entrants in the space race.
- **SpaceX's Competitive Edge:**
- **Cost Efficiency and Reusability:** SpaceX's ability to reuse the first stage of Falcon 9 offers a substantial cost advantage. Each launch costs approximately $62 million, significantly lower than competitors' costs of over $165 million per launch, due to non-reusable components.
- **Research Questions:**

1. **Predictive Analysis:** Will the first stage of SpaceX Falcon 9 land successfully?
2. **Variable Impact:** What effect do different parameters (such as launch site, payload mass, and booster version) have on landing outcomes?
3. **Correlational Studies:** How do launch sites correlate with success rates?

Methodology

- **Data Collection Techniques:**
- **SpaceX API:** Utilize the official SpaceX API for real-time data.
- **Web Scraping:** Extract Falcon 9 and Falcon Heavy launch records from Wikipedia.
- **Data Preparation:**
- **Data Wrangling:** Clean and structure raw data for analysis.
- **Label Assignment:** Convert mission outcomes into training labels (0 for unsuccessful, 1 for successful).
- **Exploratory Data Analysis (EDA):**
  - Utilize visualization techniques and SQL to explore dataset characteristics.
- **Interactive Visual Analytics:**
  - Employ Folium for geospatial analysis of launch sites.
  - Use Plotly Dash for dynamic, interactive dashboard creation.
- **Predictive Modeling:**
- **Feature Engineering:** Create a ‘class’ column; standardize and transform data.
- **Model Selection:** Split data into training and testing sets; evaluate various classification algorithms including Logistic Regression, SVM, Decision Tree, and KNN to determine the best performer based on test data.

Data Collection-API

1. API, normalize and read data

```
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 will store data returned by helper functions

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

3. Create a dataset from received data and create a new dataset.

```
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. Filter data frame to keep only Falcon 9 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

1. Create API GET for Falcon 9 launch HTML site.

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

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

2. Create Beautiful Soup.

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

3. Extract column names from HTML header.

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

```
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. Create empty dictionary with keys from column names.

```
launch_dict= dict.fromkeys(column_names)

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

# Let's initial the launch_dict with each vc
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. Use the launch records to create data for launch_dict.

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

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

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

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

6. Create data frame

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

Data Wrangling

- **EDA Objectives:**
  - Identify patterns within the dataset to inform the training of supervised machine learning models.
- **Label Creation for Training:**
- **Data Set Labels:** Converted various mission outcomes into binary training labels:
  - **1:** Booster successfully landed.
  - **0:** Booster landing was unsuccessful.
- **Landing Scenarios for Labeling:**
- **True Ocean:** Successful ocean landing in a designated area.
- **False Ocean:** Unsuccessful ocean landing in a designated area.
- **RTLS (Return to Launch Site):** Successful landing on a ground pad.
- **False RTLS:** Unsuccessful landing on a ground pad.
- **True ASDS (Autonomous Spaceport Drone Ship):** Successful landing on a drone ship.
- **False ASDS:** Unsuccessful landing on a drone ship.

Data Wrangling – cont'd

1. Identify Patterns

i. Calculate the number of launches on each site

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

```
CCAFS SLC 40    55
KSC LC 39A      22
VAFB SLC 4E     13
```

ii. Calculate the number and occurrence of each orbit

```
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
```

iii. Calculate number/occurrence of mission outcomes per orbit type

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

2. Create landing outcome label

```
# 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

Exploratory Data Analysis (EDA) Using Data Visualization

- **Data Visualization Techniques:**
- **Scatter Plots:**
  - **Purpose:** Identify relationships or correlations between two variables, making patterns easily observable.
  - **Charts Created:**
    - 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 Charts:**
  - **Purpose:** Ideal for comparing values of a variable at specific points in time, allowing for easy comparison between different groups.
  - **Chart Created:**
    - Success rate across different orbit types.
- **Line Charts:**
  - **Purpose:** Track changes over time to illustrate trends.
  - **Chart Created:**
    - Average yearly trend in launch success.

Exploratory Data Analysis (EDA) Using SQL

SQL Queries Performed on IBM DB2 Cloud Instance:

- **Launch Site Analysis:**
  - Retrieve names of all unique launch sites used in space missions.
  - Display records for launch sites starting with 'CCA'.
- **Payload and Booster Insights:** 3. Sum of payload mass for missions launched by NASA (CRS).
  - Average payload mass carried by the booster version F9 v1.1.
  - Date of the first successful landing on a ground pad.
- **Booster Performance:** 6. Identify boosters successful in drone ship landings with payload masses between 4,000 and 6,000 kg.
  - Total number of successful vs. failed mission outcomes.
  - Booster versions carrying the maximum payload (utilizing a subquery).
- **Specific Outcome Details:** 9. Failed landing outcomes on drone ships in 2015, including booster versions and launch sites.
  - Rank landing outcomes (e.g., Failure (drone ship), Success (ground pad)) from 2010-06-04 to 2017-03-20 in descending order.

Interactive Map with Folium

Purpose of the Folium Interactive Map:
- Analyze geospatial data to understand how the location and proximity of launch sites influence launch success rates.

Map Features and Functionalities:
- **Launch Sites Visualization:**
  - Mark all launch sites to visually represent their locations on the map.
- **Enhanced Site Details:**
  - Use **folium.circle** and **folium.marker** to highlight areas and provide text labels over each launch site.
  - Implement **MarkerCluster()** to display launch outcomes with color-coded markers (green for success, red for failure).
- **Distance Measurements:**
  - Calculate and display distances from each launch site to nearby coastlines, railroads, highways, and cities.
  - Integrate **MousePosition()** to display coordinates when hovering over points.
  - Add **folium.Marker()** to show distances in kilometers at various points.
  - Use **folium.Polyline()** to draw lines connecting launch sites with their nearby features.

Insights Gained:
- Proximity of launch sites to critical infrastructure:
  - Close to railways? **Yes**
  - Close to highways? **Yes**
  - Near the coastline? **Yes**
  - Maintains a certain distance from cities? **Yes**

Plotly Dashboard

- **Dashboard Overview:**
- Developed a Plotly Dash web application for real-time interactive visual analytics on SpaceX launch data.
- **Dashboard Features:**

1. **Launch Site Drop-down:**
    1. Allows users to filter visualizations by all launch sites or a specific launch site.
2. **Pie Chart:**
    1. Displays total successful launches when 'All Sites' is selected.
    2. Shows counts of successful and failed launches when a specific site is selected.
3. **Payload Range Slider:**
    1. Facilitates the selection of different payload ranges to identify visual patterns.
4. **Scatter Chart:**
    1. Explores correlations between payload and mission outcomes for selected sites.
    2. Color-labeled booster versions on each scatter point indicate mission outcomes with different boosters.

- **Insights Gained from the Dashboard:**
- **Site with Most Successful Launches:**
    - Kennedy Space Center LC-39A, with 10 successful launches.
- **Highest Launch Success Rate by Site:**
    - Kennedy Space Center LC-39A, with a 76.9% success rate.
- **Payload Ranges and Success Rates:**
    - Highest launch success rate: 2000 – 5000 kg.
    - Lowest launch success rate: 0-2000 and 5500 - 7000 kg.
- **Booster Version with Highest Success Rate:**
    - Falcon 9 Full Thrust (FT).

Predictive Analysis (Classification)

1.Load data, Split into training and testing

```
data = pd.read_csv("https://cf-courses-data.s3.us.cloud-object
et_part_2.csv")
```

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

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

```
# 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)
```

2. Conduct Logistic Regression. Refine as needed.

i. Create Logistic Regression object and then create a GridSearchCV object

ii. Fit train data set in to the GridSearchCV object and train the Model

```
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)
```

iii. Find and display best hyperparameters and accuracy score

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

iv. Check the accuracy on the test data by creating a confusion matrix

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

v. Repeat above steps for Decision Tree, KNN, and SVM algorithms

3. Identify and use the best model

```
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
```

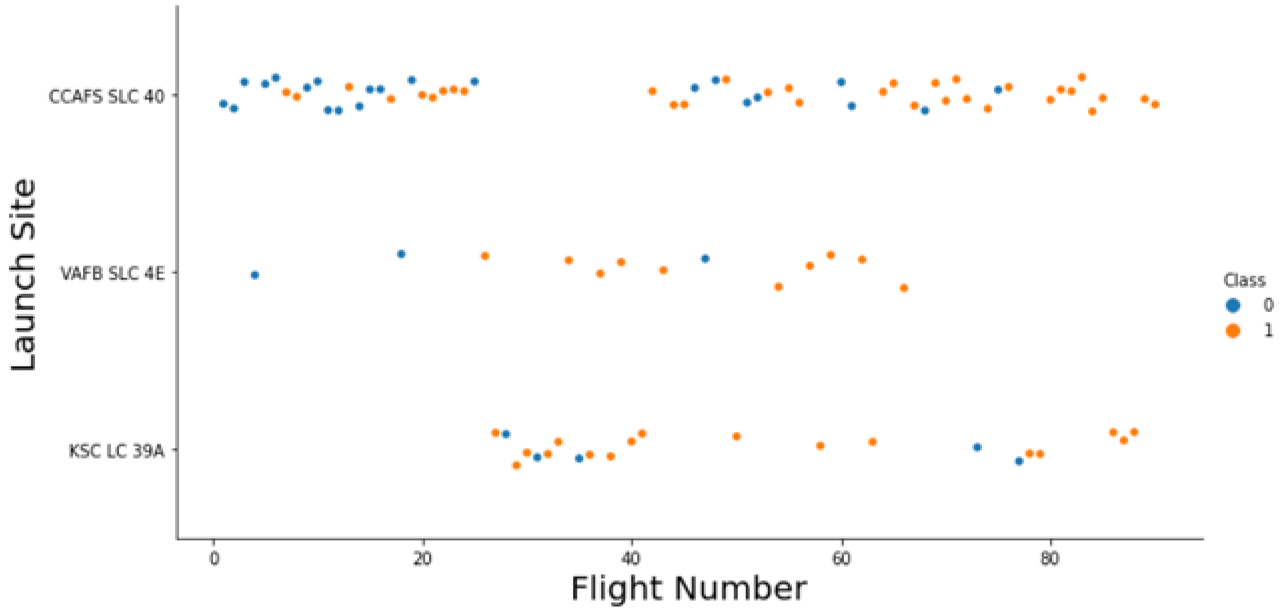
	Algo Type	Accuracy Score	Test Data Accuracy Score
2	Decision Tree	0.875000	0.833333
3	KNN	0.848214	0.833333
1	SVM	0.848214	0.833333
0	Logistic Regression	0.846429	0.833333

EDA Results

Flight Number vs. Launch Site

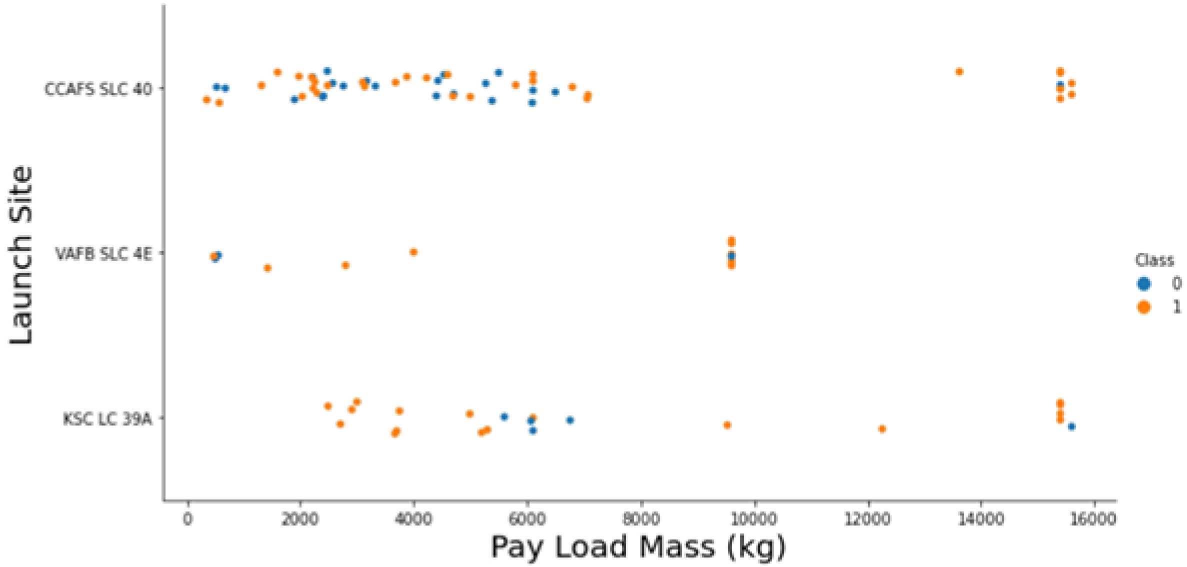
Based on the Scatter plot below:

- Success rates (Class=1) increases as the number of flights increase
- For launch site ‘KSC LC 39A’, it takes at least around 25 launches before a first successful launch



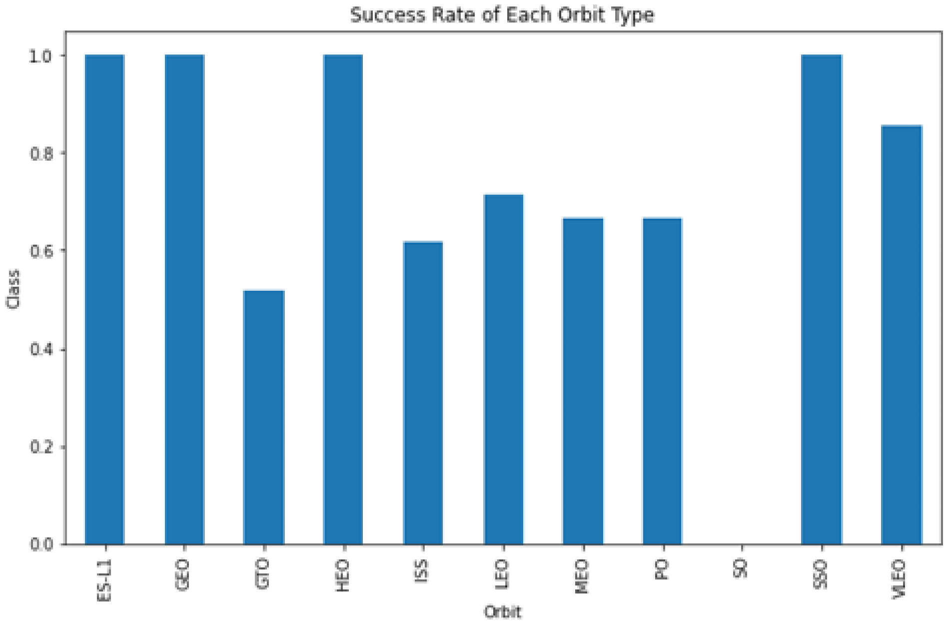
Payload vs. Launch Site

- Percentage of successful launch (Class=1) increases for launch site 'VAFB SLC 4E' as the payload mass increases mass
- The data shows no clear correlation between launch site and payload



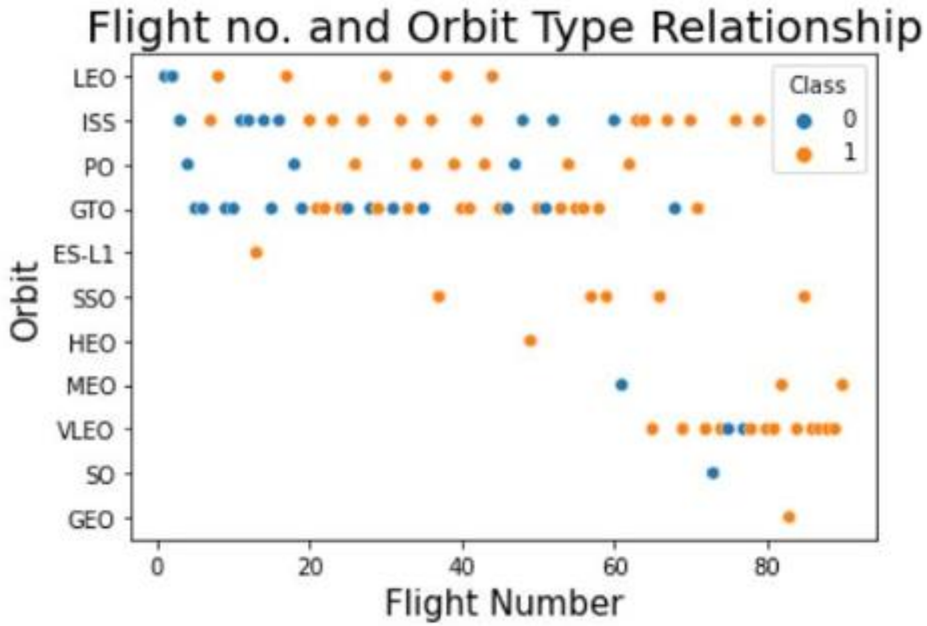
Success Rate vs. Orbit Type

- Orbits ES-LI, GEO, HEO, and SSO have the highest success rates
- GTO orbit has the lowest success rate



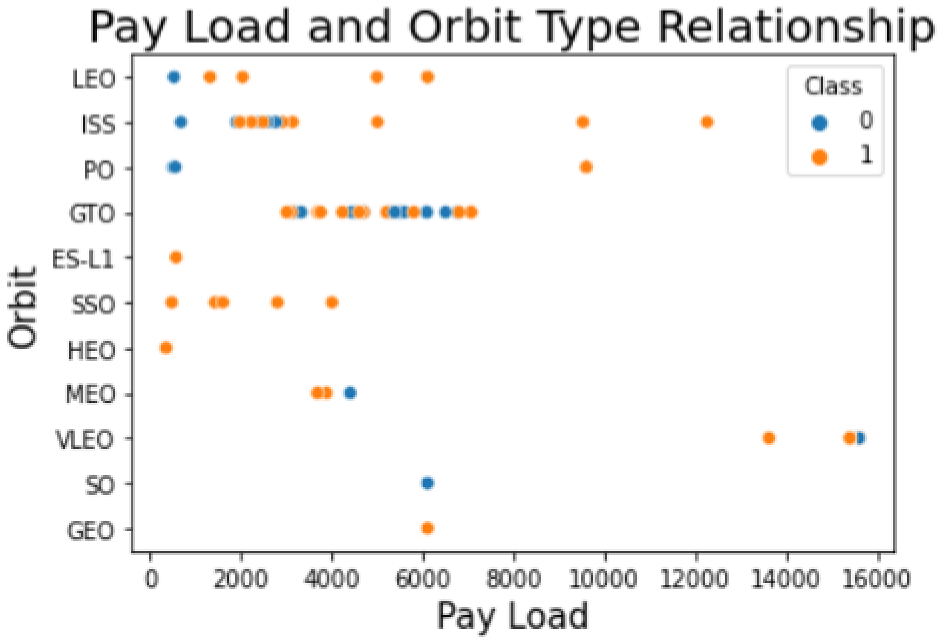
Flight Number vs. Orbit Type

- **VLEO Orbit Insights:**
  - The first successful landing (class=1) in the VLEO orbit was not achieved until after 60+ flights.
- **General Trends Across Most Orbits:**
  - For orbits such as LEO, ISS, PO, SSO, MEO, and VLEO, there is a noticeable trend where successful landing rates tend to increase as flight numbers rise.
- **GTO Orbit Analysis:**
  - No discernible relationship exists between the flight number and orbit type for GTO.



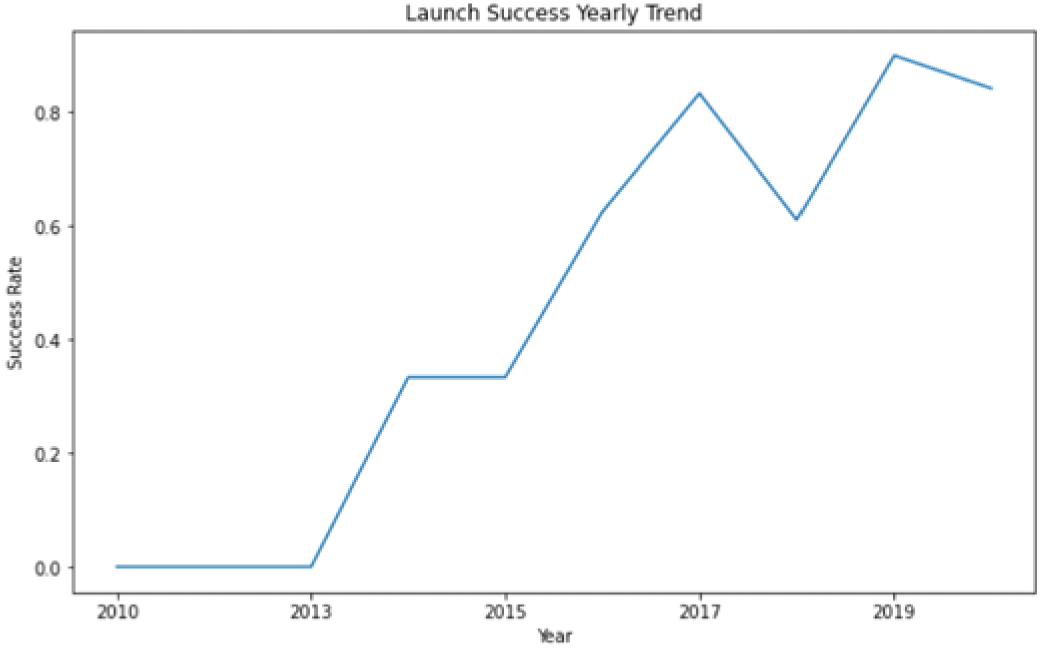
Payload vs. Orbit Type

- **Trends in Certain Orbits:**
  - In orbits such as LEO, ISS, PO, and SSO, successful landing rates (Class=1) tend to increase with the payload size.
- **GEO Orbit Observations:**
  - No consistent pattern emerges between payload size and landing success in the GEO orbit.



Launch Success Yearly Trend

- Success rate (Class=1) increased by about 80% between 2013 and 2020
- Success rates remained the same between 2010 and 2013 and between 2014 and 2015
- Success rates decreased between 2017 and 2018 and between 2019 and 2020



Launch Site Names

SQL Query

```
select distinct Launch_Site from spacextbl
```

Results

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

Launch Site Names Begin with 'CCA'

Query

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

Results

DATE	time__utc_	booster_version	launch_site	payload	payload_mass__kg_	orbit	customer	mission_outcome	landing__outcome
2010-04-06	18:45:00	F9 v1.0 B0003	CCAFS LC-40	Dragon Spacecraft Qualification Unit	0	LEO	SpaceX	Success	Failure (parachute)
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 (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
2012-08-10	0:35:00	F9 v1.0 B0006	CCAFS LC-40	SpaceX CRS-1	500	LEO (ISS)	NASA (CRS)	Success	No attempt
2013-01-03	15:10:00	F9 v1.0 B0007	CCAFS LC-40	SpaceX CRS-2	677	LEO (ISS)	NASA (CRS)	Success	No attempt

Total Payload Mass

Query

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

Results

45596

Average Payload Mass by F9 v1.1

Query

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

Results

2928

First Successful Ground Landing Date

Query

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

Results

min_date
2015-12-22

Successful Drone Ship Landing with Payload between 4000 and 6000

Query

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

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

Query

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

Results

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

Boosters Carried Maximum Payload

Query

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

Results

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

2015 Launch Records

Query

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

Results

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

Query

```
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;
```

Results

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

SpaceX Falcon9 - Launch Sites

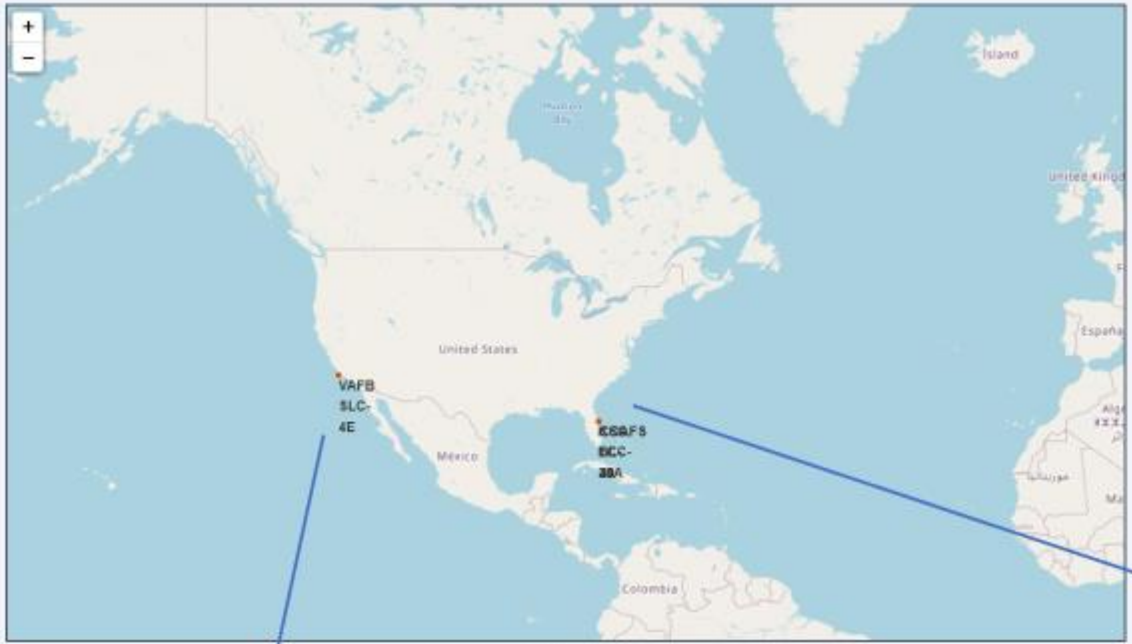


Fig 1 – Global Map

- **Figure 1: Global Overview**
  - Displays a map highlighting Falcon 9 launch sites located in the United States, specifically in California and Florida.
  - Each site is marked with a circle, label, and popup to showcase the site's name and location.
  - Notably, all launch sites are situated near the coast.
    - VAFB SLC-4E (California)
    - CCAFS LC-40 (Florida)
    - KSC LC-39A (Florida)
- **Figures 2 and 3: Detailed Site Views**
  - Provide closer views of the launch sites, focusing on:
    - CCAFS SLC-40 (Florida)

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

Fig 1 – US map with all Launch Sites

- **Figure 1: Overview of US Launch Sites**
  - Displays a map of the US with all Falcon 9 launch sites, where each site is annotated with the total number of successful and failed launches.
- **Figures 2, 3, 4, and 5: Detailed Site Analysis**
  - These figures zoom into each launch site, using green markers to denote successful launches and red markers for failures.
- **Key Observation:**
  - The KSC LC-39A Launch Site displays the highest number of successful launches, as evident from the detailed site maps.

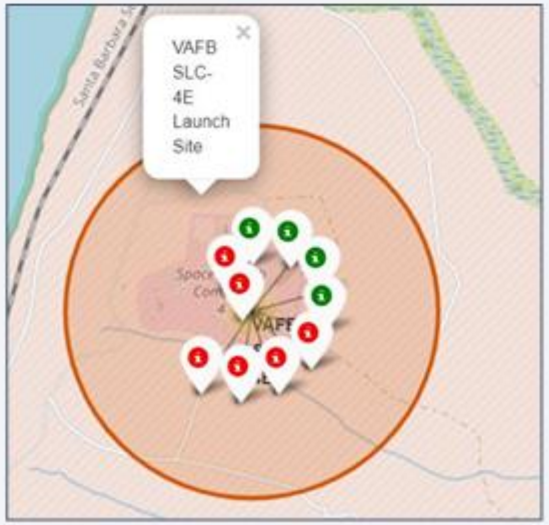


Fig 2 – VAFB Launch Site with success/failed markers

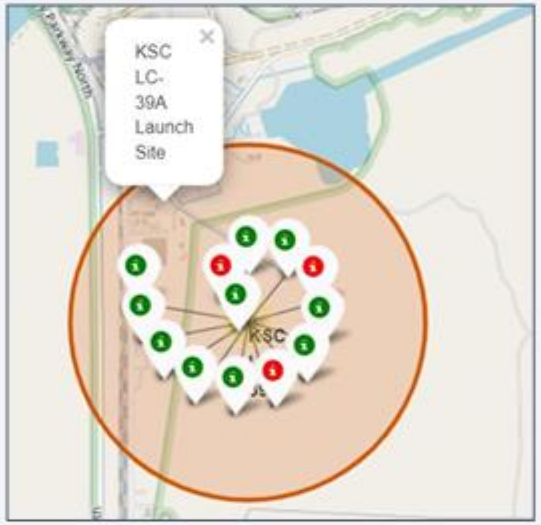


Fig 3 – KSC LC-39A success/failed markers

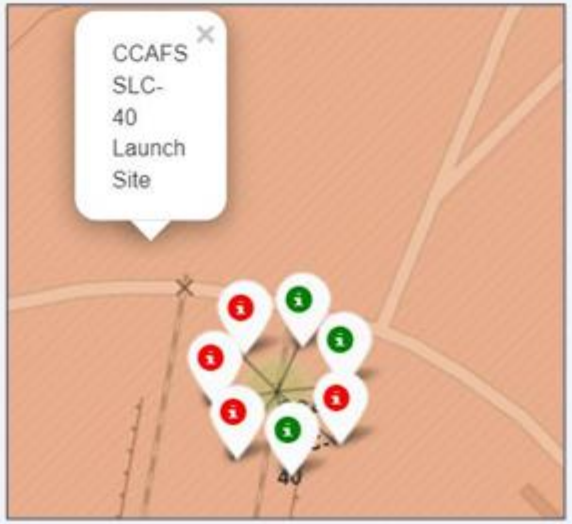


Fig 4 – CCAFS SLC-40 success/failed markers

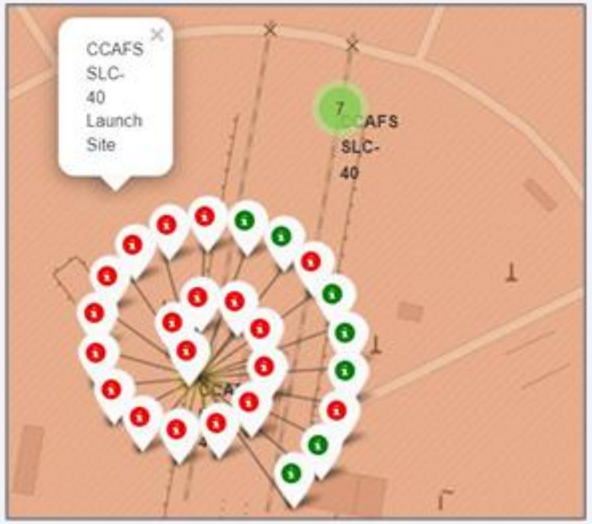
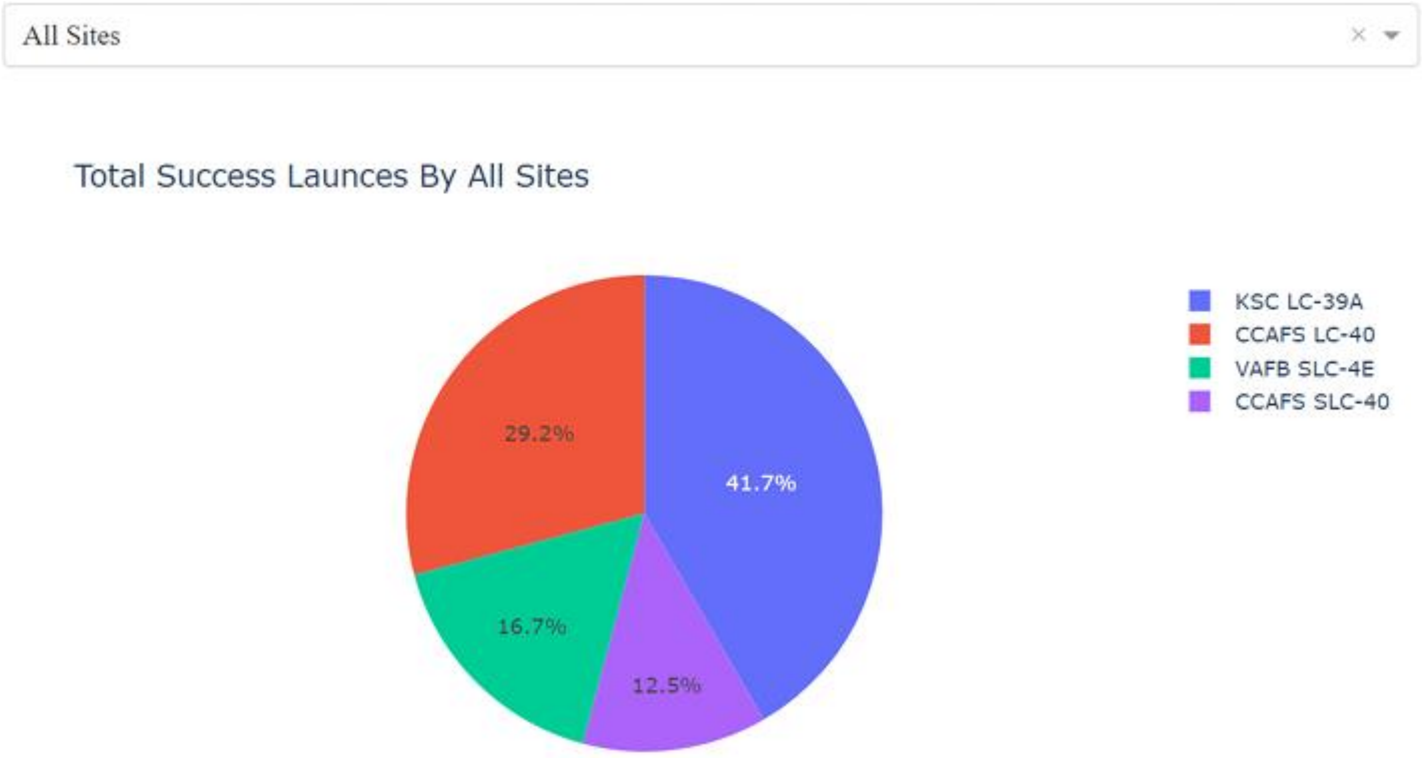


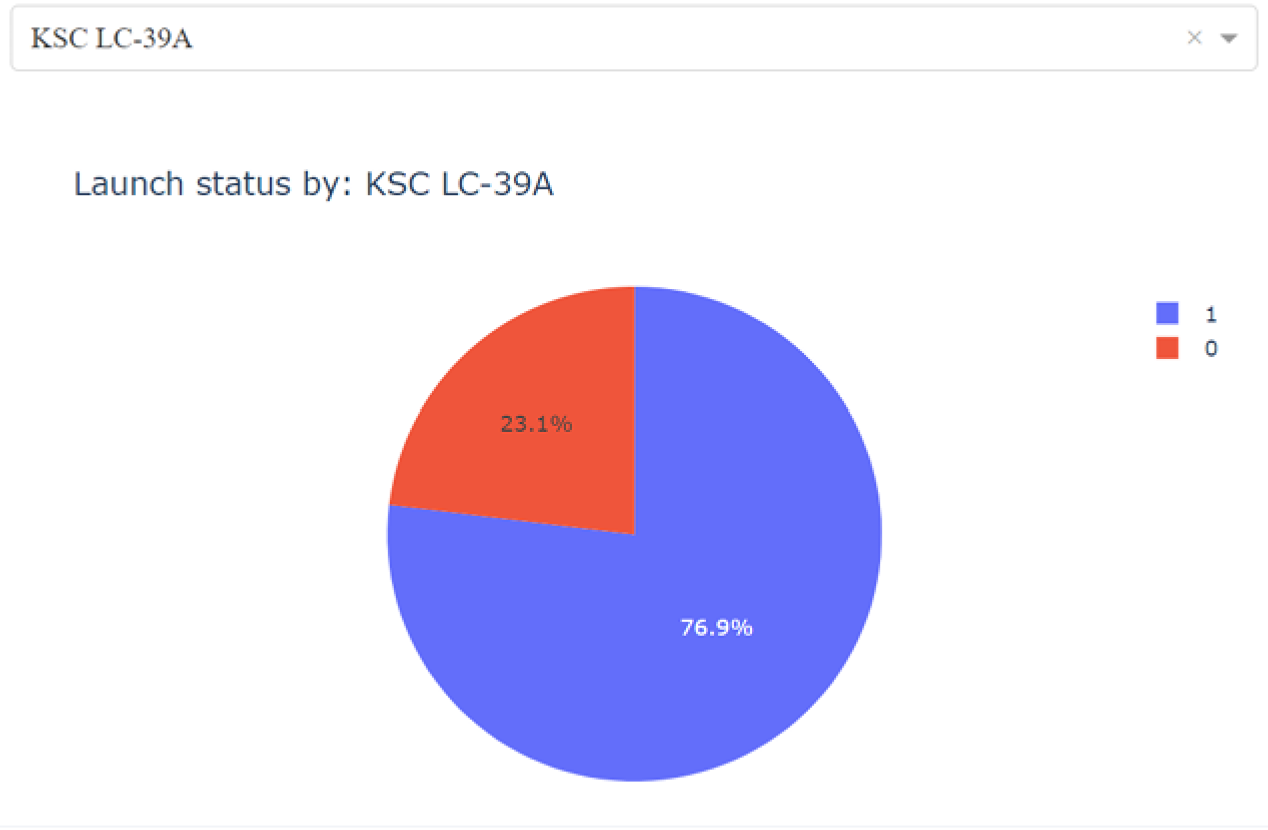
Fig 5 – CCAFS SLC-40 success/failed markers

Plotly Dashboard: Launch Success Counts For All Sites

- Launch Site ‘KSC LC-39A’ has the highest launch success rate
- Launch Site ‘CCAFS SLC 40’ has the lowest launch success rate



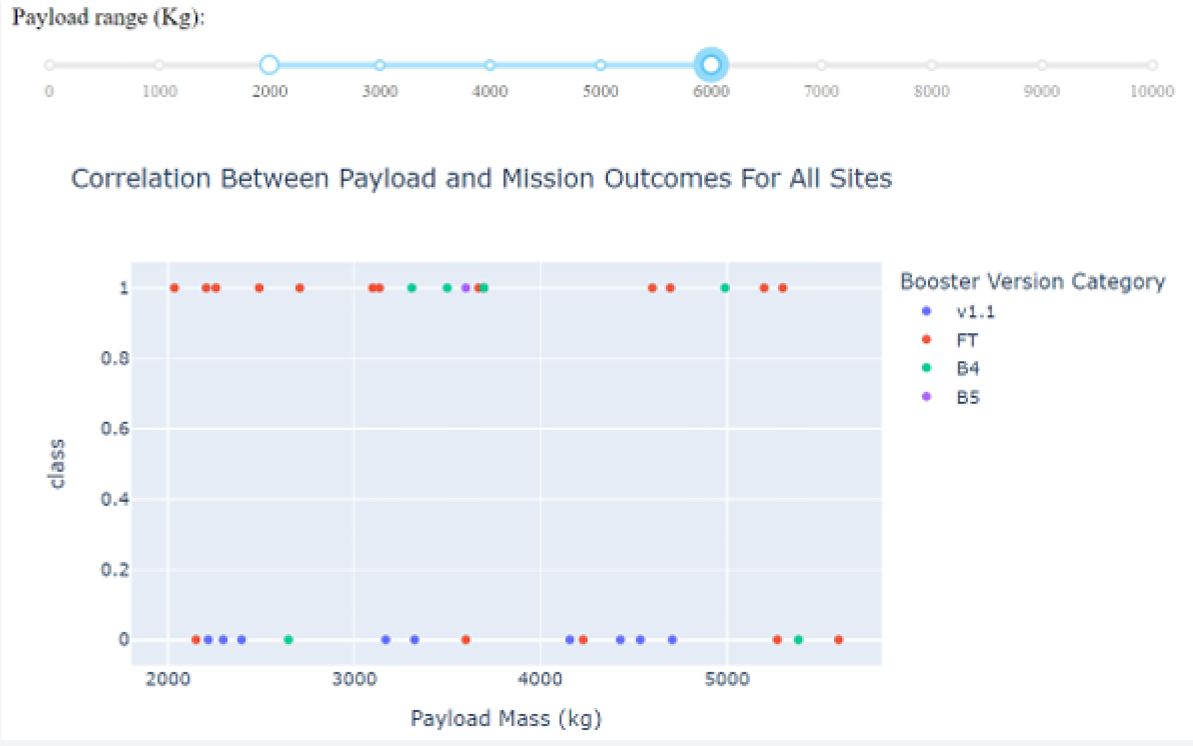
Plotly Dashboard: Launch Site with Highest Launch Success Ratio



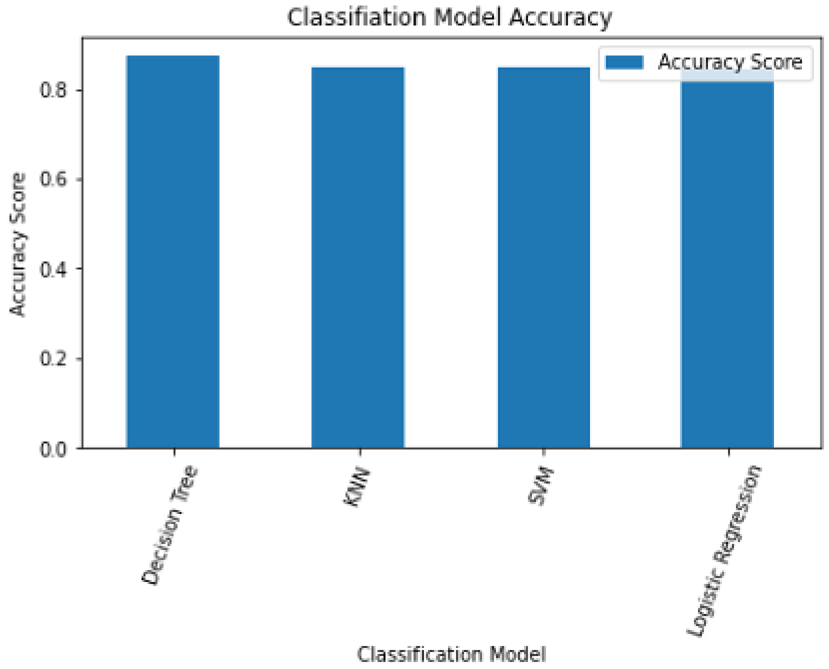
- KSC LC-39A Launch Site has the highest launch success rate and count
- Launch success rate is 76.9%
- Launch success failure rate is 23.1%

Plotly Dashboard: Payload vs. Launch Outcome Scatter Plot for All Sites

- Most successful launches are in the payload range from 2000 to about 5500
- Booster version category ‘FT’ has the most successful launches
- Only booster with a success launch when payload is greater than 6k is ‘B4’



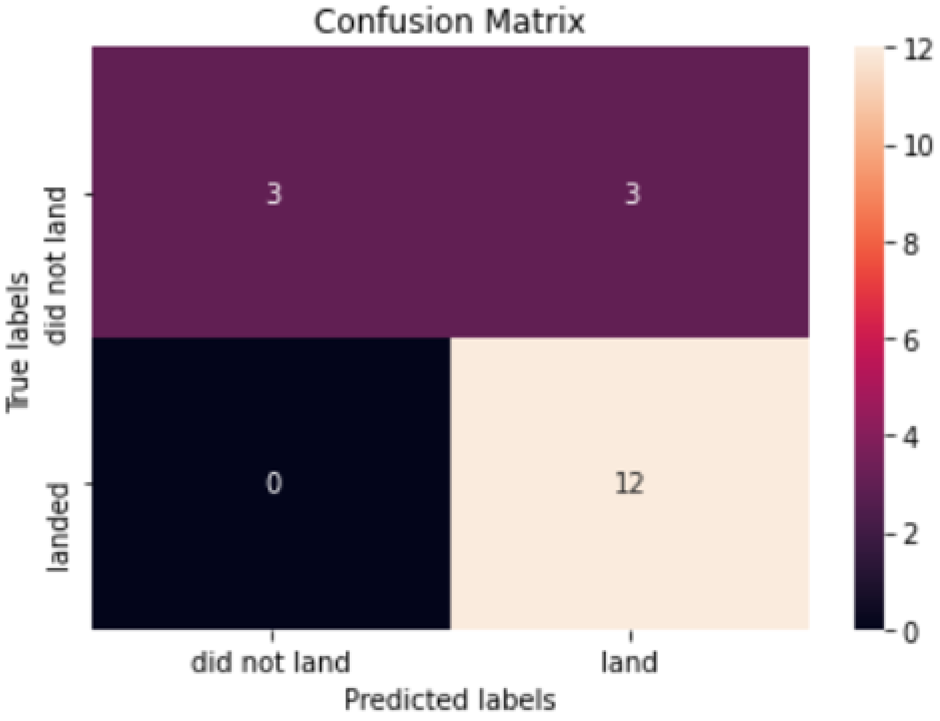
Predictive Analysis: Classification Accuracy



	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

- Top Performer:
- The Decision Tree algorithm achieved the highest classification score, as indicated by an accuracy of 0.8750, which is confirmed by the bar chart.
- Uniform Test Scores:
- All classification algorithms demonstrated the same accuracy score of 0.8333 on the test data.
- Considerations for Improvement:
- Given the closely matched accuracy scores and identical test results across models, expanding the dataset may be necessary to further refine and optimize the models.

Predictive Analysis: Confusion Matrix



- **Uniform Matrix Across Models:**
  - The confusion matrix was consistent for all models tested (Logistic Regression, SVM, Decision Tree, KNN).
- **Prediction Breakdown:**
  - Total predictions made: 18
  - True Positives (Correct 'Yes'): 12 scenarios were accurately predicted as successful landings.
  - True Negatives (Correct 'No'): 3 scenarios were accurately predicted as unsuccessful landings.
  - False Positives (Incorrect 'Yes'): 3 scenarios were incorrectly predicted as successful landings.
- **Accuracy and Misclassification:**
  - Overall accuracy: The classifier correctly predicted outcomes 83% of the time.
  - Error rate: The misclassification rate stood at approximately 16.5%.

Conclusion

Insights from Flight and Payload Data Analysis

- **Flight Frequency and Success:**
  - The likelihood of successful first stage landings increases with the number of flights.
- **Payload and Success Rates:**
  - While success rates tend to rise with increased payload, there is no definitive correlation between payload mass and success rates.
- **Improvement Over Time:**
  - Launch success rates have improved by approximately 80% from 2013 to 2020.
- **Launch Site Performance:**
  - 'KSC LC-39A' records the highest launch success rate, while 'CCAFS SLC 40' shows the lowest.
- **Orbital Success Rates:**
  - Orbits ES-L1, GEO, HEO, and SSO exhibit the highest launch success rates, with GTO orbit showing the lowest.
- **Strategic Location of Launch Sites:**
  - Launch sites are strategically positioned away from cities and closer to coastlines, railroads, and highways.
- **Model Performance:**
  - The Decision Tree model outperforms others with an accuracy of about 87.5%. All models tested showed an accuracy of around 83% on test data, suggesting that additional data might be required to further refine the models and potentially identify a better fit.