IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

Ching Lam Chan
2/8/2024

Outline

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

Executive Summary

- Summary of methodologies
  - Data collection
  - Data wrangling
  - Exploratory data analysis with data visualisation
  - Exploratory data analysis with data SQL
  - Interactive visual analytics and dashboard with Folium
  - Predictive analysis
- Summary of all results
  - Exploratory data analysis results
  - Interactive analytics demo in screenshots
  - Predictive analysis results

Introduction

- Project background and context
  - SpaceX's Falcon 9 rocket has transformed the space launch industry through its innovative reusable first-stage booster, significantly lowering launch costs to $62 million compared to over $165 million from other providers. The ability to reuse the first stage is a key factor in these savings, as evidenced by the successful landing and recovery of the first stage on numerous occasions. Predicting the success of these landings is crucial for accurately estimating launch costs, which is valuable information for any company looking to compete with SpaceX. By analyzing the factors that influence landing success, we can enhance our understanding of launch economics and provide insights for potential competitors in the space launch market.
- Problems you want to find answers
  - What are the critical factors influencing the successful landing of the Falcon 9 first stage?
  - How do specific rocket variables and their relationships impact the probability of a successful landing?
  - What optimal conditions should SpaceX aim for to maximize the success rate of first-stage landings?

Section 1

Methodology

Methodology

Executive Summary

- Data collection methodology:
    - Data was gathered through the SpaceX API and by web scraping information from Wikipedia
- Perform data wrangling
    - Categorical variables were transformed using one-hot encoding
- Perform exploratory data analysis (EDA) using visualization and SQL
    - Including scatterplot and bargraphs
- 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 acquisition involved two main methods:
  - SpaceX API:
    - We sent GET requests to the SpaceX API
    - JSON responses were decoded using the .json() method
    - We then converted the JSON data into a pandas DataFrame using json_normalize()
    - The resulting DataFrame underwent cleaning, including checks for missing values and appropriate data imputation where needed
  - Web Scraping:
    - We utilized BeautifulSoup to scrape Falcon 9 launch records from Wikipedia
    - The process involved extracting HTML tables containing launch information
    - These tables were then parsed and transformed into pandas DataFrames
    - The ultimate goal of this data collection process was to create comprehensive and clean datasets for subsequent analysis of SpaceX launches, particularly focusing on Falcon 9 missions.

Data Collection – SpaceX API

- Utilized GET requests to the SpaceX API to gather data
- Converted response to a .JSON file
- Cleaned data
- Filtered dataframe
- Exported to .CSV file

- please see: https://github.com/karenccl/SpaceX-Falcon-9-first-stage-Landing-Prediction/blob/main/jupyter-labs-webscraping.ipynb

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

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

In [13]:
```
# Use json_normalize meethod to convert the json result into a dataframe
data = pd.json_normalize(response.json())
```

In [16]:
```
#Global variables
BoosterVersion = []
PayloadMass = []
Orbit = []
LaunchSite = []
Outcome = []
Flights = []
GridFins = []
Reused = []
Legs = []
LandingPad = []
Block = []
ReusedCount = []
Serial = []
Longitude = []
Latitude = []
```

In [18]:
```
# Call getBoosterVersion
getBoosterVersion(data)
```

the list has now been update

In [19]:
```
BoosterVersion[0:5]
```

```
data_falcon9.to_csv('dataset_part_1.csv', index=False)
```

Data Collection - Scraping

- Retrieved HTML Content
- Parsed HTML with BeautifulSoup
- Located Relevant Tables
- Extracted Column Headers
- Initialized a Data Dictionary
- Populated the Dictionary
- Converted to DataFrame
- Exported to CSV

- Please see: https://github.com/karenccl/SpaceX-Falcon-9-first-stage-Landing-Prediction/blob/main/jupyter-labs-webscraping.ipynb

In [5]:
```
# use requests.get() method with the provided static_url
# assign the response to a object
response = requests.get(static_url).text
```

Create a `BeautifulSoup` object from the HTML `response`

In [8]:
```
# Use BeautifulSoup() to create a BeautifulSoup object from a response text content
soup = BeautifulSoup(response, 'html.parser')
```

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

In [9]:
```
# Use soup.title attribute
print(soup.title)
```

```
<title>List of Falcon 9 and Falcon Heavy launches - Wikipedia</title>
```

In [21]:
```
column_names = []

# Apply find_all() function with `th` element on first_launch_table
# Iterate each th element and apply the provided extract_column_from_header() to get a column name
# Append the Non-empty column name (`if name is not None and len(name) > 0`) into a list called column_names
temp = soup.find_all('th')
for x in range(len(temp)):
    try:
     name = extract_column_from_header(temp[x])
     if (name is not None and len(name) > 0):
        column_names.append(name)
    except:
     pass
```

In [23]:
```
launch_dict= dict.fromkeys(column_names)

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

# Let's initial the launch_dict with each value to be an empty list
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']=[]
```

Data Wrangling

- Determine the Launch Count per Site
  - Calculate the total number of launches conducted at each launch site
- Assess Orbit Frequency
  - Calculate the frequency and count of each orbit type
- Analyze Mission Outcomes by Orbit Type
  - Calculate the number and frequency of mission outcomes for each orbit type
- Generate Landing Outcome Labels
  - Create a new label for landing outcomes based on the data in the Outcome column
- Please see: https://github.com/karenccl/SpaceX-Falcon-9-first-stage-Landing-Prediction/blob/main/labs-jupyter-spacex-Data%20wrangling.ipynb

In [5]:
```
# Apply value_counts() on column LaunchSite
launches = df['LaunchSite'].value_counts()
launches
```

In [6]:
```
# Apply value_counts on Orbit column
occurence = df['Orbit'].value_counts()
occurence
```

In [7]:
```
# landing_outcomes = values on Outcome column
landing_outcomes = df['Outcome'].value_counts()
landing_outcomes
```

In [8]:
```
for i,outcome in enumerate(landing_outcomes.keys()):
    print(i,outcome)
```

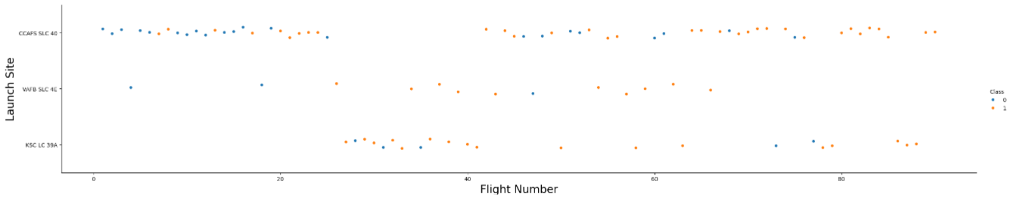
In [9]:
```
bad_outcomes=set(landing_outcomes.keys()[[1,3,5,6,7]])
bad_outcomes
```

EDA with Data Visualization

- Data visualization:
- Flight Number vs. Payload Mass
- Flight Number and Launch Site Relationship
- Payload and Launch Site Relationship
- Success Rate by Orbit Type
- Flight Number and Orbit Type Correlation
- Payload Mass and Orbit Type Correlation
- Annual Launch Success Trend

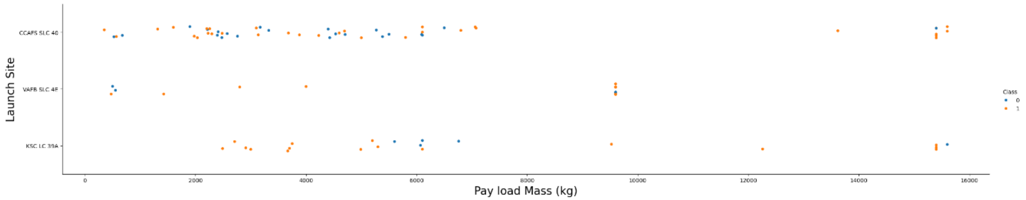
In [14]:

```
# 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 va
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()
```



In [9]:

```
# Plot a scatter point chart with x axis to be Pay Load Mass (kg) and y axis to be the launch site, and hue to be the cla
sns.catplot(y="LaunchSite", x="PayloadMass", hue="Class", data=df, aspect = 5)
plt.xlabel("Pay load Mass (kg)",fontsize=20)
plt.ylabel("Launch Site",fontsize=20)
plt.show()
```



- Please see: https://github.com/karenccl/SpaceX-Falcon-9-first-stage-Landing-Prediction/blob/main/edadataviz.ipynb

EDA with SQL

- Retrieve and display the names of all unique launch sites involved in the space missions.
- Display 5 records where the launch site names begin with the string 'CCA'.
- Calculate and display the total payload mass carried by boosters launched by NASA (CRS).
- Compute and display the average payload mass carried by the booster version F9 v1.1.
- List the date when the first successful landing outcome on a ground pad was achieved.
- List the names of boosters that have successfully landed on a drone ship and carried a payload mass between 4000 and 6000 kg.
- Display the total count of successful and failed mission outcomes.
- List the names of booster versions that have carried the maximum payload mass. Use a subquery for this task.
- Display records showing the month names, failed landing outcomes on drone ships, booster versions, and launch sites for the months in the year 2015.
- Rank the count of successful landing outcomes between the dates 04-06-2010 and 20-03-2017 in descending order.

- Please see: https://github.com/karenccl/SpaceX-Falcon-9-first-stage-Landing-Prediction/blob/main/jupyter-labs-eda-sql-coursera_sqllite%20(1).ipynb

Build an Interactive Map with Folium

- Map All Launch Sites
    - Plot the locations of all SpaceX launch sites on a geographical map.
    - Use appropriate markers or icons to represent each launch site.
- Visualize Launch Outcomes by Site
    - On the same map, differentiate between successful and failed launches for each site.
    - Use color-coding or different markers to represent success and failure.
    - For example, green markers for successful launches and red for failed ones.
- Calculate Proximity Distances
    - For each launch site, calculate distances to nearby points of interest or critical locations.
- Create a Distance Analysis
    - Compile the calculated distances into a table or dataset.
    - Consider creating a visualization (e.g., bar chart or heat map) to represent these distances graphically.
- Analyze Geographical Factors
- Use the map and distance data to analyze how geographical factors might correlate with launch success rates.

- Please see: https://github.com/karenccl/SpaceX-Falcon-9-first-stage-Landing-Prediction/blob/main/lab_jupyter_launch_site_location.ipynb

Build a Dashboard with Plotly Dash

- Continuous Availability
  - Hosted on Python Anywhere for 24/7 accessibility
  - Built using Flask and Dash web frameworks
- Visualizations
  - Pie Chart: Total Launches by Site
  - Scatter Graph: Outcome vs. Payload Mass
- Interactive Elements
  - Users can manipulate data views and explore different aspects of launches
- Please see: https://github.com/karenccl/SpaceX-Falcon-9-first-stage-Landing-Prediction/blob/main/space_dash.py

Predictive Analysis (Classification)

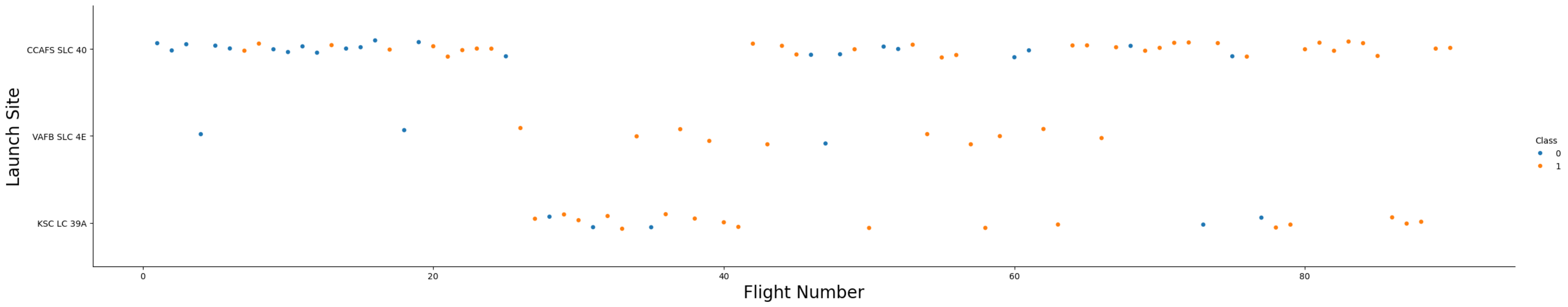
- Data Loading
  - Import the dataset into NumPy and Pandas for efficient manipulation
- Data Transformation
  - Preprocess and clean the data as necessary
  - Perform feature engineering
- Data Splitting
  - Divide the dataset into training and test sets
  - Verify the number of samples in the test set
- Algorithm Selection
  - Evaluate and choose appropriate machine learning algorithms for the task
- Hyperparameter Tuning Setup
  - Define parameter grids for each selected algorithm
  - Set up GridSearchCV objects with chosen parameters and algorithms
- Model Training
  - Fit the training data to the GridSearchCV objects
  - This process will train models with various hyperparameter combinations
- Model Evaluation
  - Assess model performance using the test set
  - Compare results across different algorithms and parameter settings
- Final Model Selection
  - Choose the best performing model based on evaluation metrics

- Please see: https://github.com/karenccl/SpaceX-Falcon-9-first-stage-Landing-Prediction/blob/main/SpaceX_Machine%20Learning%20Prediction_Part_5.ipynb

Section 2

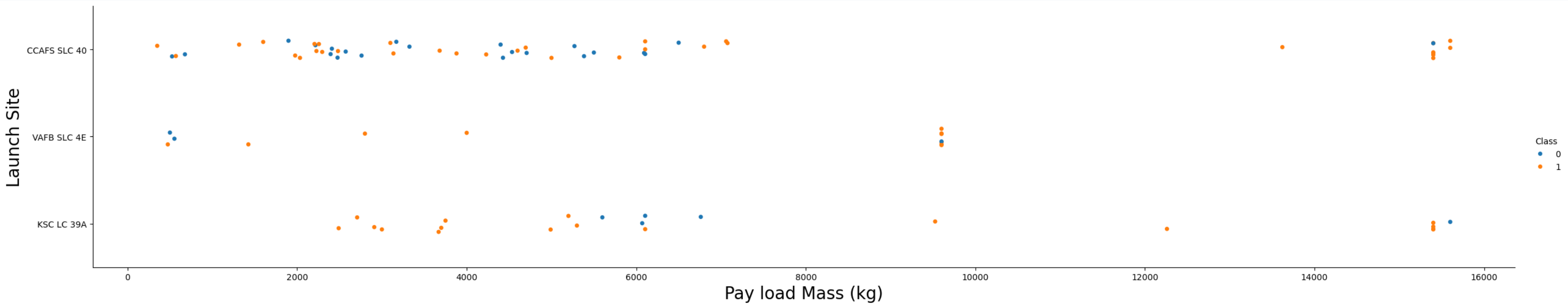
Insights drawn from EDA

Flight Number vs. Launch Site



- A higher number of flights at a launch site is associated with an increased success rate at that site

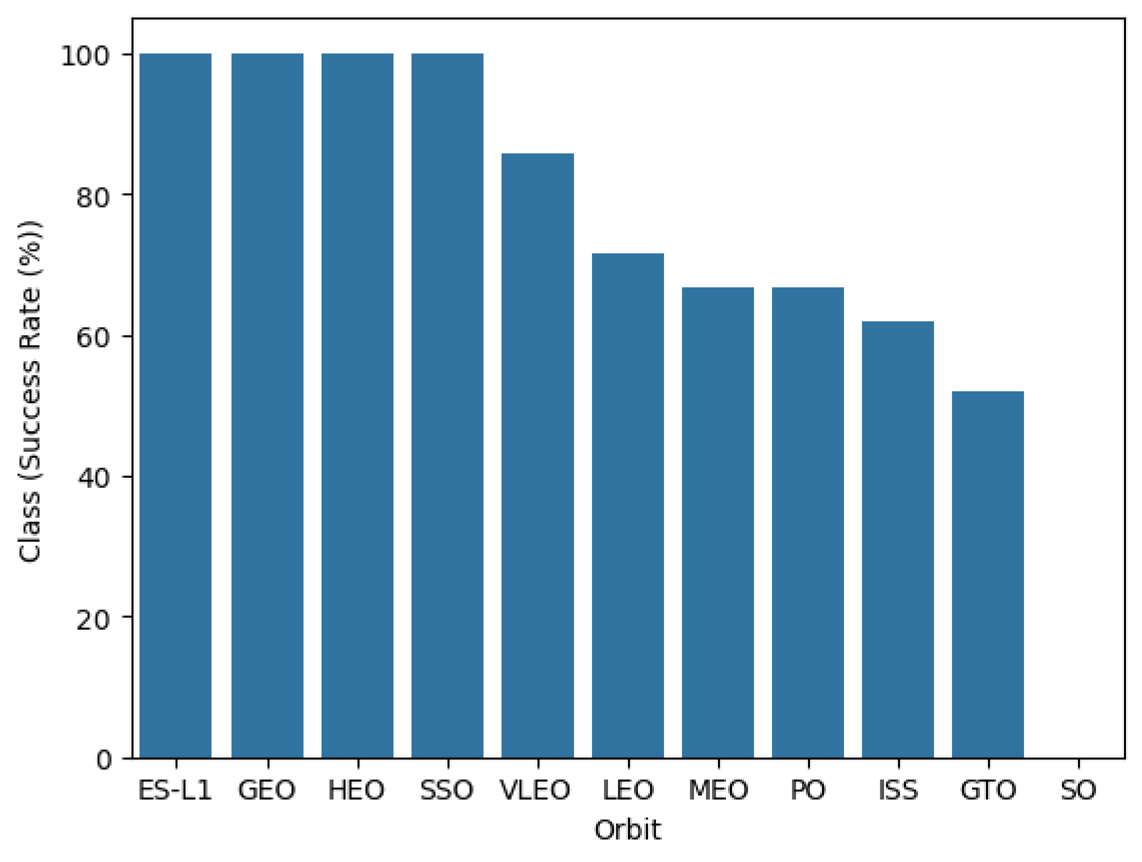
Payload vs. Launch Site



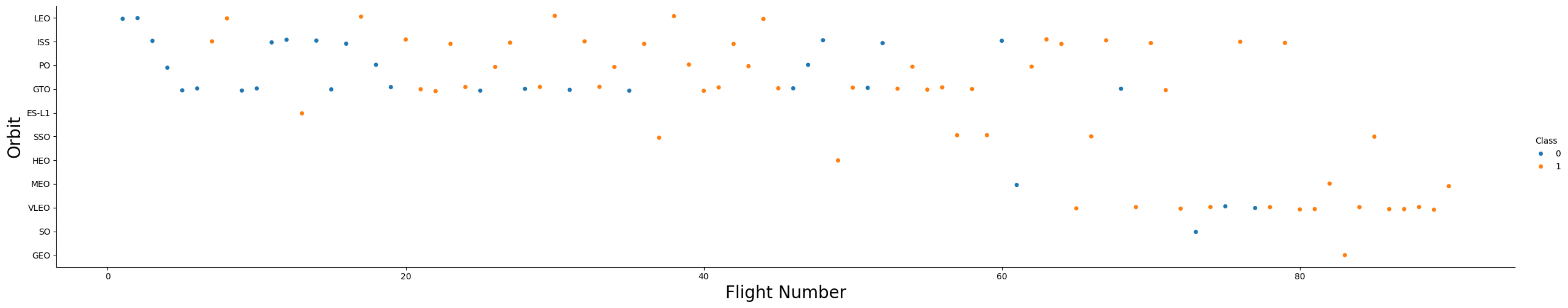
- The greater the payload mass for launch site CCAFS SLC 40, the higher the success rate for the rocket

Success Rate vs. Orbit Type

- ES-L1, GEO, HEO, SSO had the most success rate

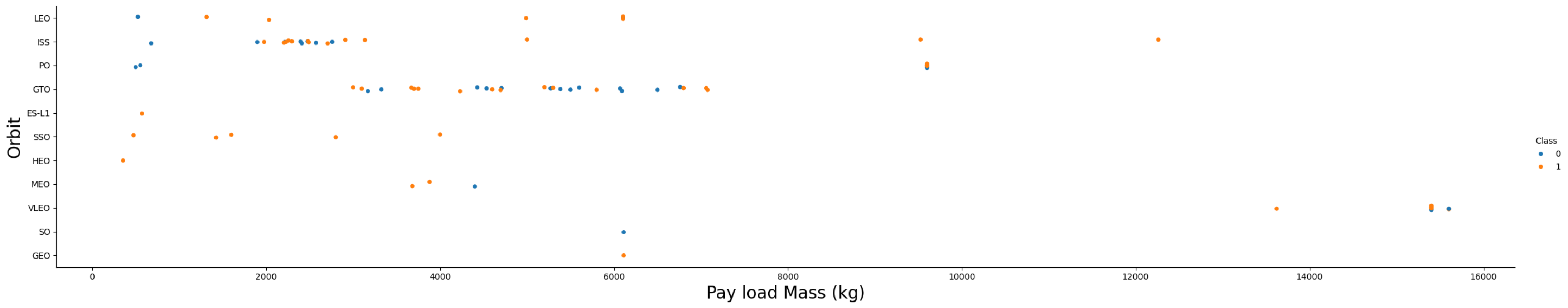


Flight Number vs. Orbit Type



- The LEO orbit's success is related to the number of flights whereas in the GTO orbit, there is no relationship between flight number and the orbit

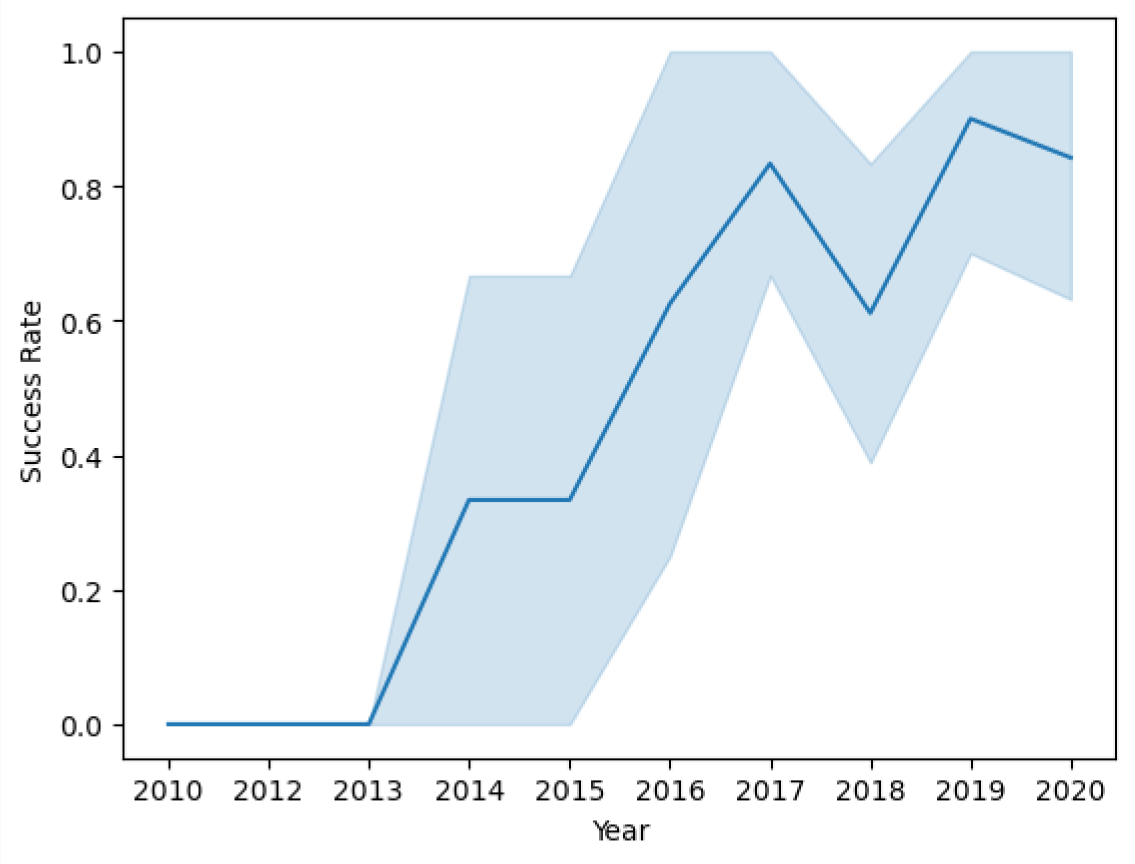
Payload vs. Orbit Type



- Heavy payloads are more likely to achieve successful landings PO, LEO, and ISS orbit.

Launch Success Yearly Trend

- The success rate has consistently increased from 2013 to 2020



All Launch Site Names

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

In [10]:
```
%sql select distinct(LAUNCH_SITE) from SPACEXTBL
```

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

Out[10]:

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

- Names of the unique launch sites:
    - CCAFS LC-40
    - VAFB SLC-4E
    - KSC LC-39A
    - CCAFS SLC-40

Launch Site Names Begin with 'CCA'

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

```
In [11]: %sql select * from SPACEXTBL where LAUNCH_SITE like 'CCA%' limit 5
```

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

Out[11]:

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

- 5 records where launch sites begin with `CCA`

Total Payload Mass

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

In [12]:
```
%sql select sum(PAYLOAD_MASS__KG_) from SPACEXTBL where CUSTOMER = 'NASA (CRS)'
```

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

Out[12]:

sum(PAYLOAD_MASS__KG_)
45596

- Total payload carried by boosters from NASA: 45596 kg

Average Payload Mass by F9 v1.1

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

In [13]:
```
%sql select avg(PAYLOAD_MASS__KG_) from SPACEXTBL where BOOSTER_VERSION = 'F9 v1.1'
```

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

Out[13]:

avg(PAYLOAD_MASS__KG_)
2928.4

- Average payload mass carried by booster version F9 v1.1: 2928.4 kg

First Successful Ground Landing Date

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

Hint:Use min function

In [14]:

```
%sql select min(DATE) from SPACEXTBL where Landing_Outcome = 'Success (ground pad)'
```

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

Out[14]:

min(DATE)
2015-12-22

- Date of the first successful landing outcome on ground pad: 2015/12/22

Successful Drone Ship Landing with Payload between 4000 and 6000

In [15]:
```
%sql select Booster_Version from SPACEXTBL WHERE Landing_Outcome = 'Success (drone ship)' and PAYLOAD_MASS__KG_ > 4000 and P
```

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

Out[15]:

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

- Names of boosters which have successfully landed on drone ship and had payload mass greater than 4000 but less than 6000:
  - F9 FT B1022
  - F9 FT B1026
  - F9 FT B1021.2
  - F9 FT B1031.2

Total Number of Successful and Failure Mission Outcomes

List the total number of successful and failure mission outcomes

In [16]:

```
%sql select count(Mission_Outcome) from SPACEXTBL WHERE Mission_Outcome = 'Success' or Mission_Outcome = 'Failure (in flight
```

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

Out[16]:

count(Mission_Outcome)
99

- Total number of successful and failure mission outcomes: 99

Boosters Carried Maximum Payload

```
In [17]: %sql select Booster_Version from SPACEXTBL where PAYLOAD_MASS__KG_ = (select max(PAYLOAD_MASS__KG_) from SPACEXTBL)

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

Out[17]:

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

- Names of the booster which have carried the maximum payload mass

2015 Launch Records

In [20]:

```
%sql SELECT SUBSTR(Date,6,2) AS Month, Booster_Version, Launch_site FROM SPACEXTBL WHERE Landing_Outcome LIKE 'Failure%drone
```

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

Out[20]:

Month	Booster_Version	Launch_Site
01	F9 v1.1 B1012	CCAFS LC-40
04	F9 v1.1 B1015	CCAFS LC-40

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

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

```
In [25]: %sql SELECT Landing_Outcome, COUNT(*) AS Numbers FROM SPACEXTBL WHERE Landing_Outcome LIKE 'Failure%' OR Landing_Outcome LIK

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

Out[25]:

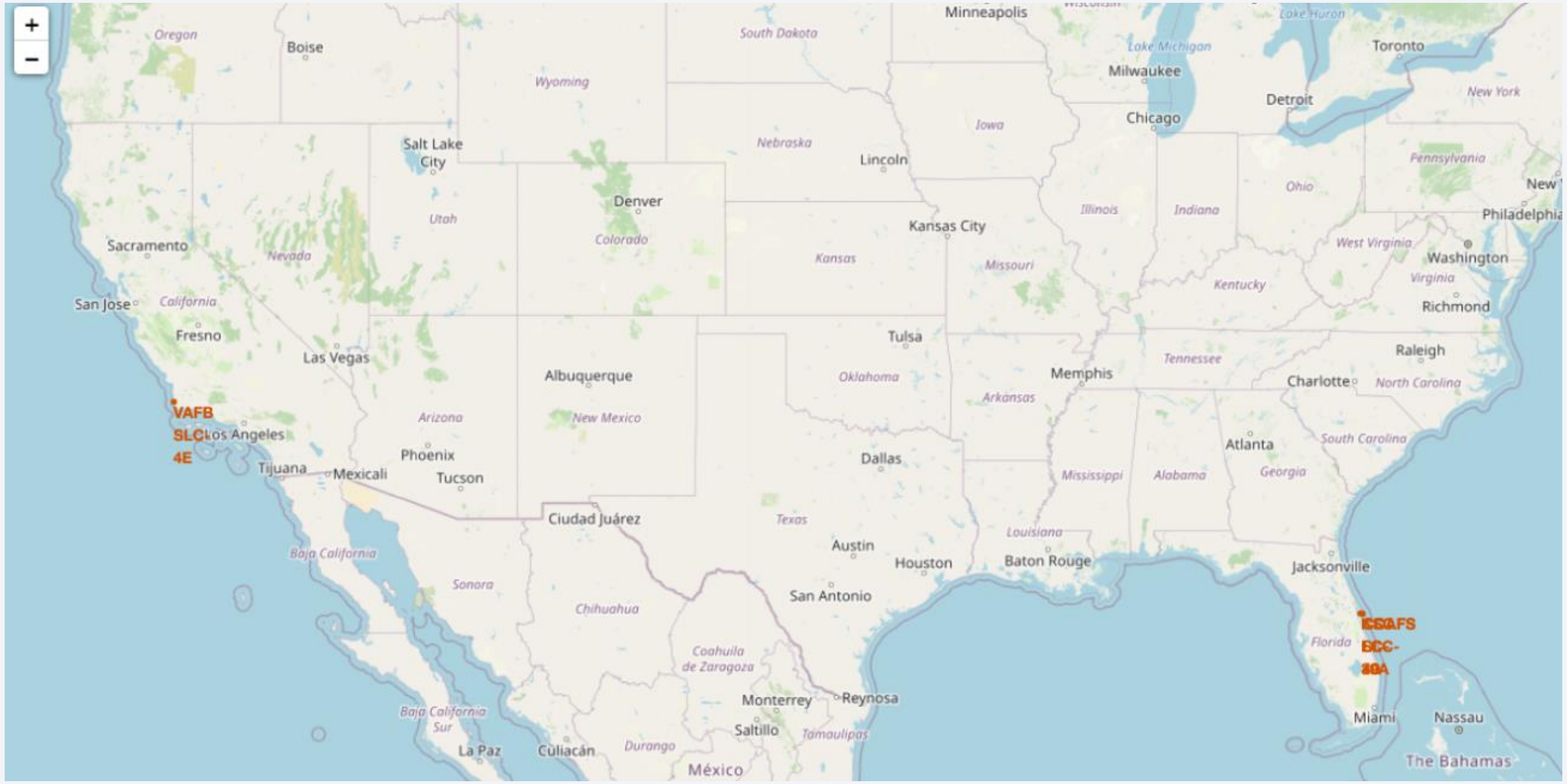
Landing_Outcome	Numbers
Failure (parachute)	10

- 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

Section 3

Launch Sites Proximities Analysis

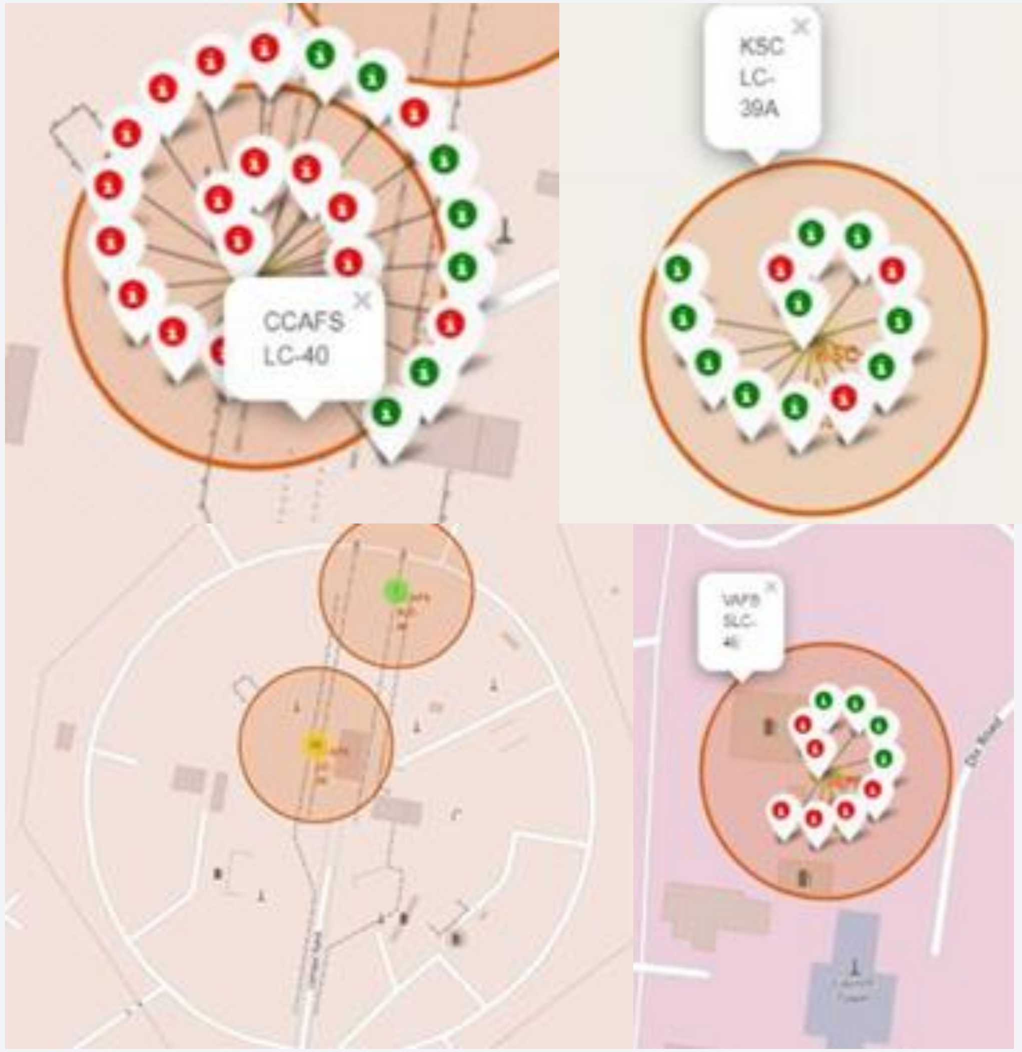
All Launch Sites Global Map Markers



- SpaceX launch sites are in the United States of America coasts
  - Florida and California

Markers Showing Launch Sites With Color Labels

- Green Marker shows successful Launches
- Red Marker shows Failures



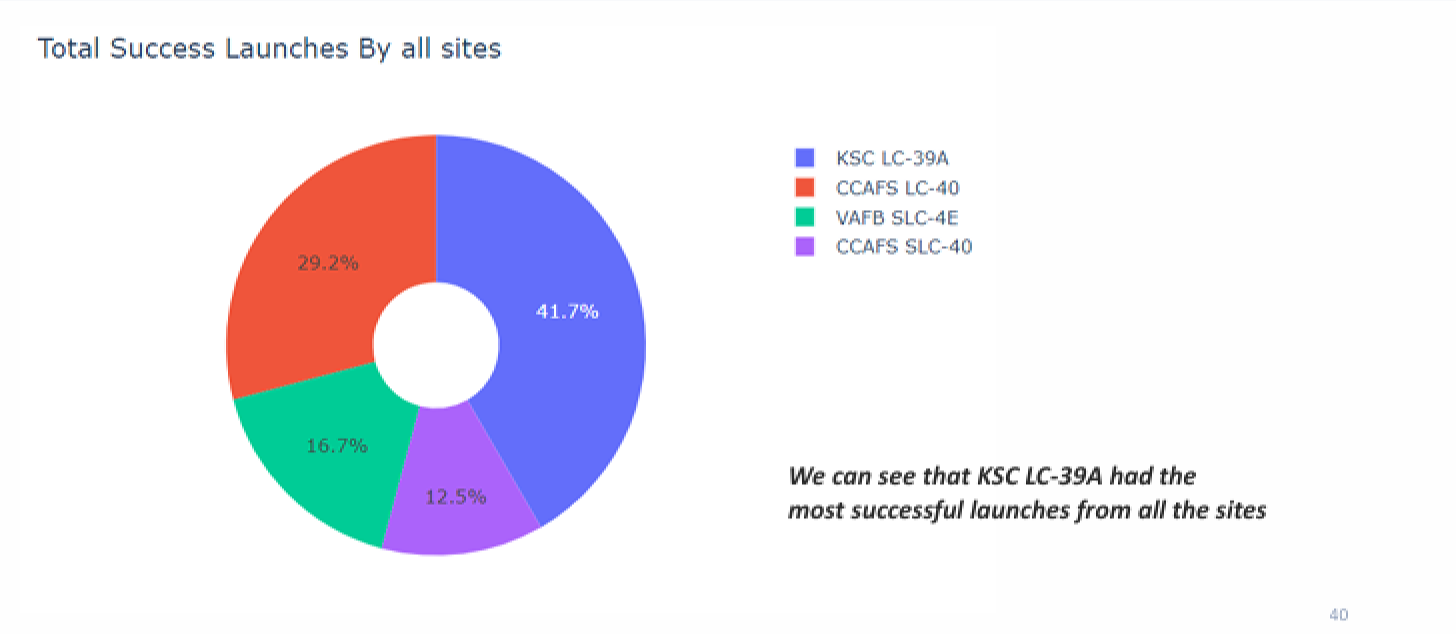
Launch Site Distance To Landmarks

- Railways and Highways: Not in close proximity
- Coastline: Close proximity observed
- Cities: Maintained at a distance

Section 4

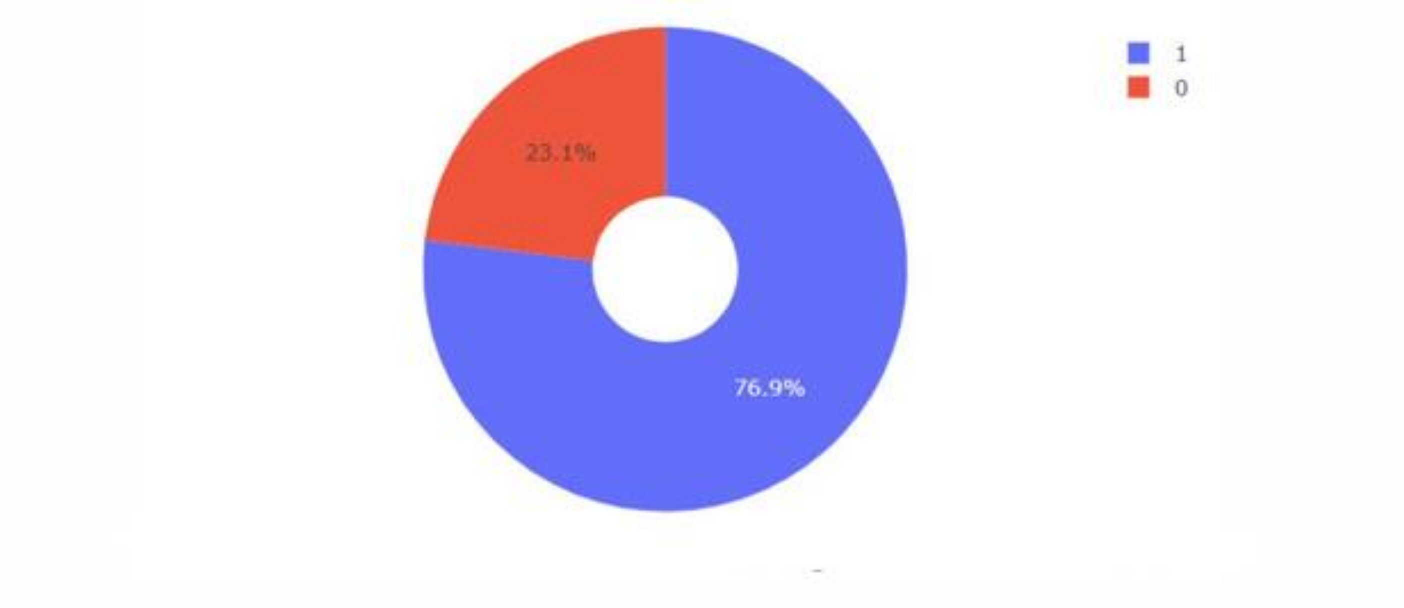
Build a Dashboard with Plotly Dash

Pie Chart Showing The Success Percentage Achieved By Each Launch Site



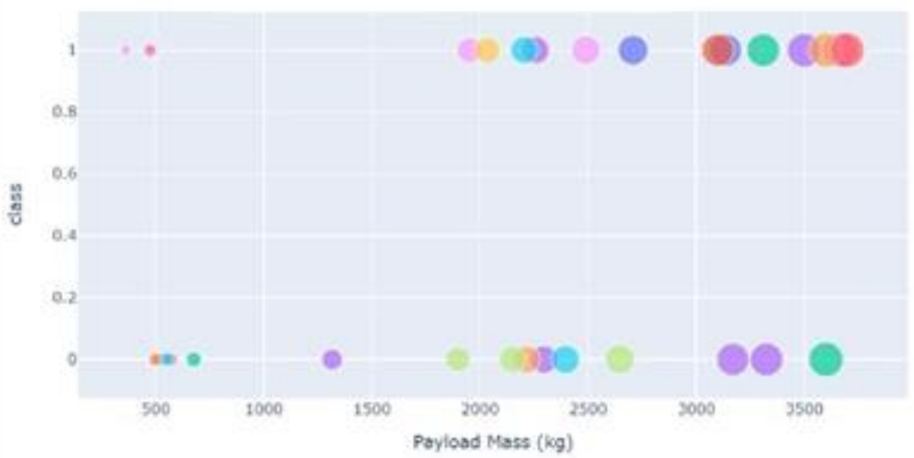
- KSC LC-39A had the highest total success launches

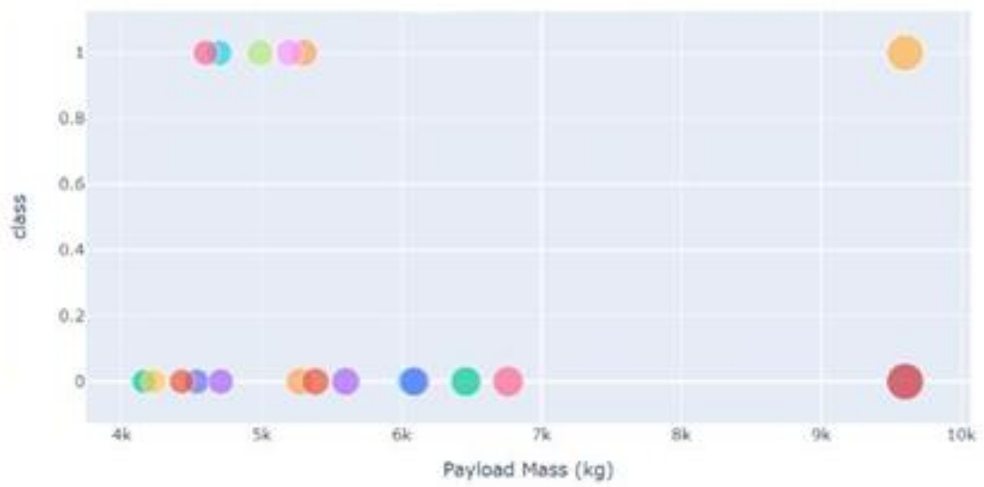
Pie Chart Showing The Launch Site With The Highest Launch Success Ratio



- KSC LC-39A had a 76.9% success rate and a 23.1% failure rate

Scatter Plot Of Payload Vs Launch Outcome For All Sites, With Different Payload Selected In The Range Slider





- The success rates for lighter payloads are higher than those for heavier payloads

Section 5

Predictive Analysis (Classification)

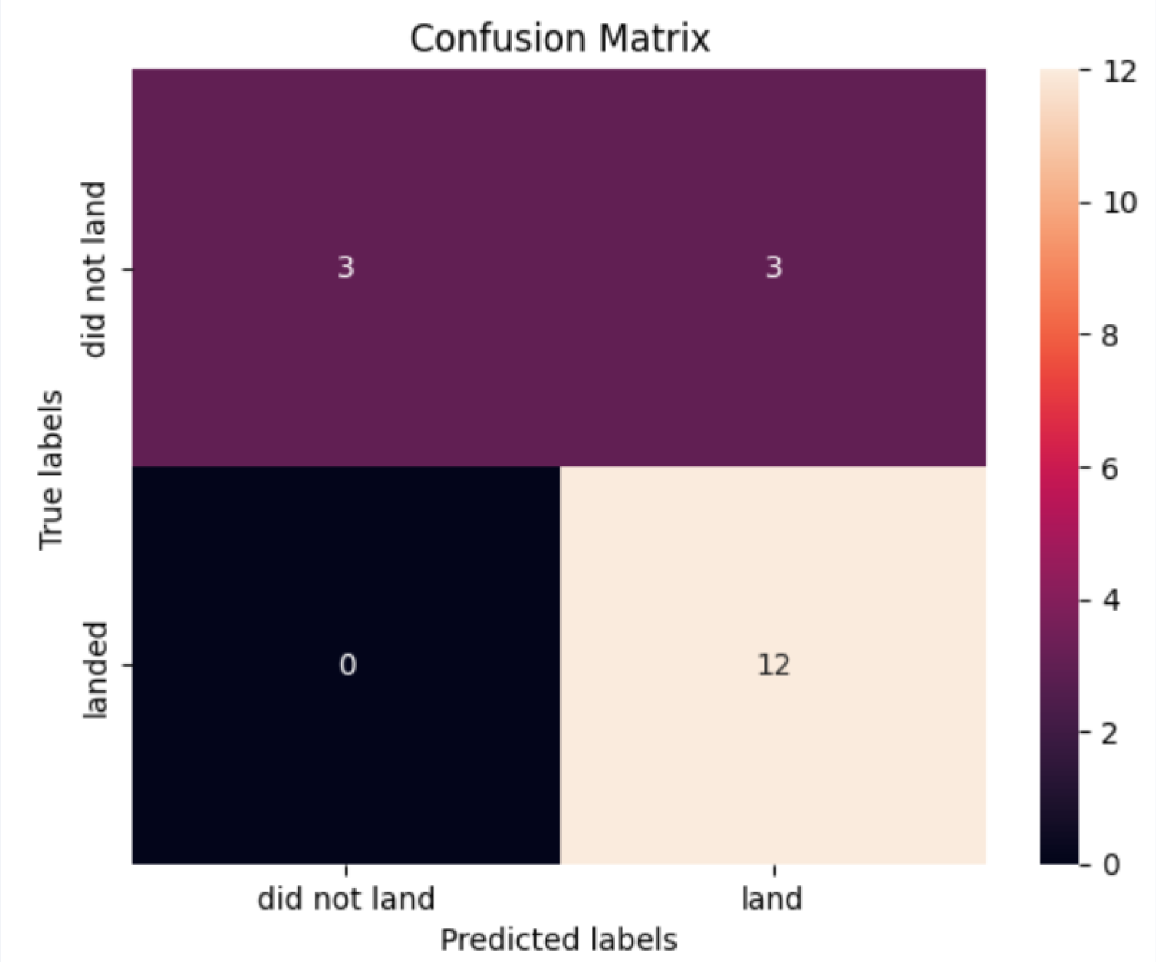
Classification Accuracy

Find the method performs best:

In [39]:
```
Report = pd.DataFrame({'Method' : ['Test Data Accuracy']})

knn_accuracy=knn_cv.score(X_test, Y_test)
Decision_tree_accuracy=tree_cv.score(X_test, Y_test)
SVM_accuracy=svm_cv.score(X_test, Y_test)
Logistic_Regression=logreg_cv.score(X_test, Y_test)

Report['Logistic_Reg'] = [Logistic_Regression]
Report['SVM'] = [SVM_accuracy]
Report['Decision Tree'] = [Decision_tree_accuracy]
Report['KNN'] = [knn_accuracy]

Report.transpose()
```

- 0.8333333333333334
- The Tree Classifier Algorithm is the best for machine learning for this dataset

Confusion Matrix



Conclusions

- SpaceX Launch Data Analysis Insights
  - Model Performance
    - The Tree Classifier Algorithm proved most effective for this dataset
  - Achieved a high accuracy of 84% using optimized parameters
- Payload Impact
  - Lighter payloads generally showed better performance than heavier ones
  - Increased payload mass correlates with lower first stage return success
- Launch Success Trends
  - SpaceX's launch success rate shows a positive correlation with time
  - Continuous improvement observed, indicating ongoing refinement of launch processes
- Launch Site Performance
  - KSC LC-39A emerged as the most successful launch site
- Orbit Type Success Rates
  - Highest success rates: GEO, HEO, SSO, and ES-L1 orbits
  - Lowest success rate: SO (Sub-Orbital) launches
- Site-Success Correlation
  - Clear relationship between launch sites and success rates
  - Suggests some sites may offer more favorable launch conditions
- Temporal Trends
  - Marked increase in success rates from 2013 to 2020
  - Indicates significant technological and operational improvements over time

Thank you!