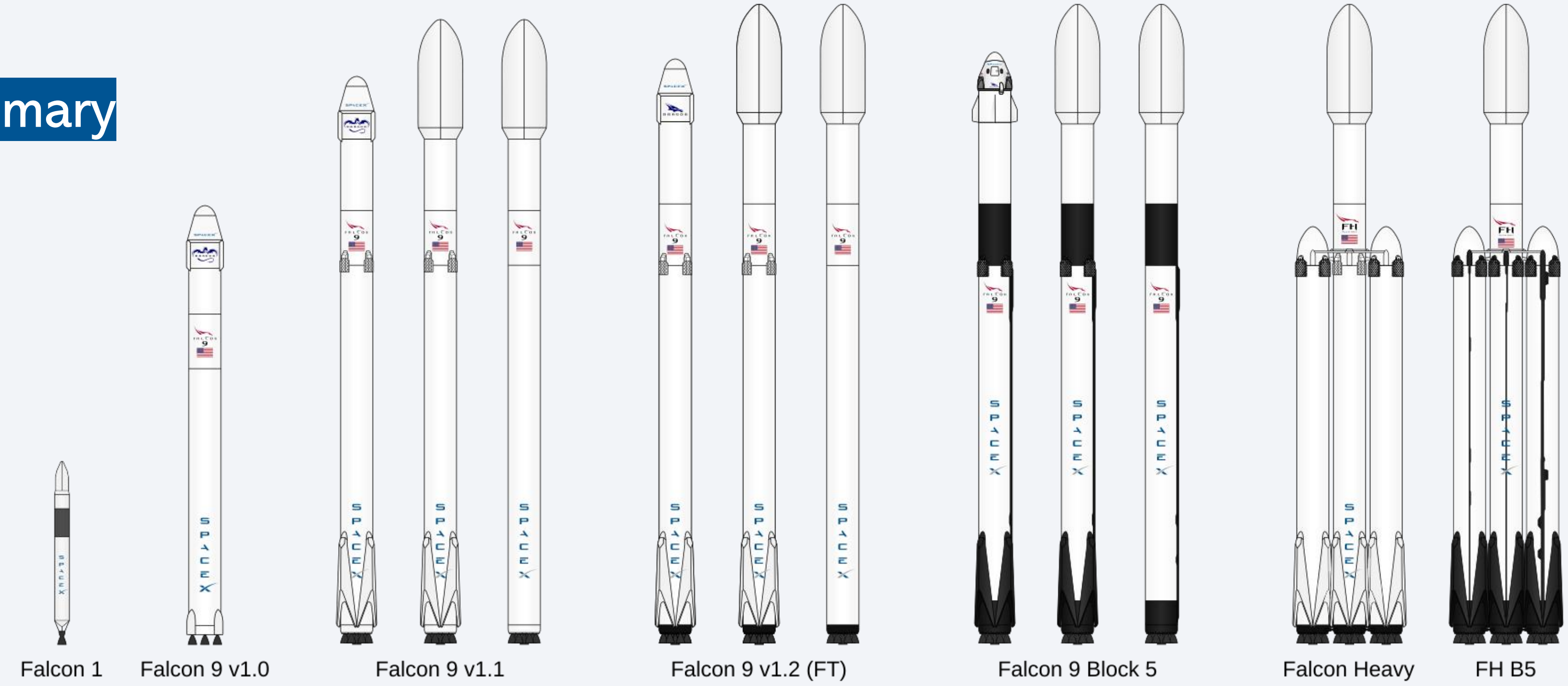
IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

Elena Vieth
02.11.2024

Outline

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



Lucabon (based on work of Markus Säynevirta and Craigboy and Ressi) - Eigenes Werk, CC BY-SA 4.0, https://commons.wikimedia.org/w/index.php?curid=75671798

Executive Summary – Data Collection & Processing

https://starship-spacex.fandom.com/wiki/Falcon_9?file=CRS-12_Mission_%2836571921725%29.jpg

This report presents an analysis of SpaceX rocket launches, utilizing data from the SpaceX API and Wikipedia's List of Falcon 9 and Falcon Heavy launches. Through data wrangling, exploratory data analysis (EDA), and predictive modeling, we provide insights into the factors influencing launch and landing success rates and the effectiveness of SpaceX's reusable rocket technology.

Data Collection and Processing. Data was sourced from the SpaceX API and Wikipedia for a comprehensive history of launches. After identifying and replacing missing values, additional summary and categorical variables were created to enrich the dataset. These adjustments enabled a clearer understanding of launch patterns, including payload mass, launch site, and booster version details.

Using SQL and various visualization techniques, we identified key trends in launch success rates over time. Interactive visualizations with Folium and Plotly Dash further enhanced this analysis, offering a dynamic view of launch site distribution and success trends. Findings revealed that success rates have improved over time, with significant milestones in 2017 and 2019 when landing success peaked.

Executive Summary – Key Insights & Findings

Key Insights and Findings. SpaceX has achieved a 76.9% launch success rate, with higher success rates for launches from Kennedy Space Center's LC-39A and Vandenberg Space Force Base's SLC-4E. Landing success rates have increased over time, indicating SpaceX's advancements in booster recovery technology. Launch sites are positioned on both the east and west coasts of the U.S. to support different orbital needs - Cape Canaveral and Kennedy Space Center for equatorial orbits and Vandenberg for polar orbits. Falcon 9 Block 5 boosters consistently deliver heavy payloads (15,600 kg), demonstrating high reliability for routine, high-capacity missions. VLEO missions (Very Low Earth Orbit) display the highest landing success rate, reflecting SpaceX's optimized reusability for specific orbital missions.

Classification models, including Logistic Regression, Support Vector Machine (SVM), and K-Nearest Neighbor (KNN), achieved over 94% accuracy in predicting landing outcomes, though each model showed a tendency for false positives. Future model improvements should adjust for false positive costs, as incorrect success predictions are more costly operationally.

Executive Summary – Future Steps

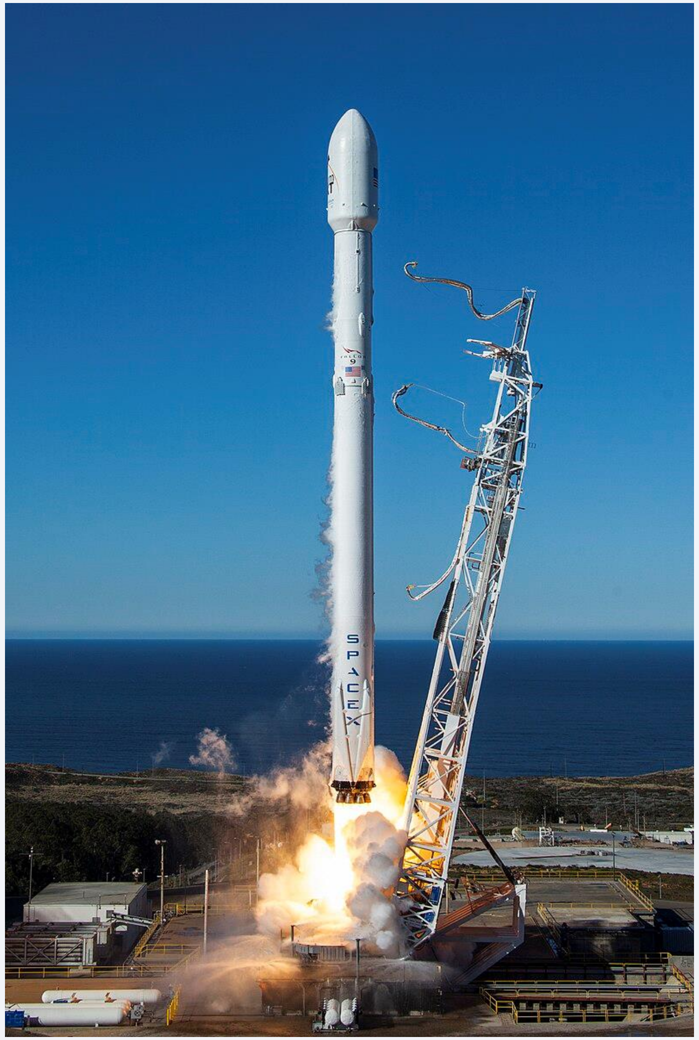
Recommendations and Future Steps. Future classification models should prioritize weighting schemes to reduce false positives, given the operational importance of correctly identifying unsuccessful landings. As SpaceX continues to launch and refine its technology, ongoing data collection and analysis will help track trends in reliability, launch efficiency, and payload performance.

This analysis provides a foundation for understanding SpaceX's mission success factors, optimizing launch site choices, and improving predictive models for booster recovery, contributing to the long-term goal of more sustainable and cost-effective space travel.

Introduction – Background & Context

Project Background and Context

SpaceX has transformed the aerospace industry by pioneering reusable rocket technology, drastically reducing the costs of space travel and making frequent launches possible. Since the debut of its Falcon 9 rocket, SpaceX has achieved significant milestones in both orbital payload delivery and booster recovery. By strategically positioning launch sites across the U.S. and refining reusable rockets, SpaceX has not only improved the economic viability of space missions but also set a new standard for sustainable space exploration. This project explores the launch and landing data of SpaceX rockets to better understand the factors influencing mission outcomes. Using datasets from the SpaceX API and detailed mission records from Wikipedia, we performed a comprehensive analysis that includes data wrangling, exploratory data analysis (EDA), and predictive modeling to identify patterns, trends, and key insights regarding SpaceX's operational success.



SpaceX - Iridium-1 Launch, CC0, https://commons.wikimedia.org/w/index.php?curid=55030122

Introduction – Main Objectives

The main objective of this project is to answer several critical questions surrounding SpaceX's launch operations and booster recovery, including:

1. **How have launch and landing success rates evolved over time?** Understanding the historical trends in mission success provides insight into the effectiveness of SpaceX's improvements in technology and operational procedures.
2. **What impact do launch sites and booster types have on mission outcomes?** By analyzing the role of launch sites and specific booster versions, we aim to identify factors that contribute to higher success rates and reusability.
3. **How does payload mass and orbital destination affect landing success?** Examining the relationship between payload characteristics and landing outcomes helps in assessing the reusability potential for missions of varying demands.
4. **Can we build reliable predictive models to forecast landing outcomes?** Using classification models, we aim to predict landing success and explore areas for improvement in model performance.

SpaceX Photos - CRS-8 first stage landing, CC0, https://commons.wikimedia.org/w/index.php?curid=48152574

Section 1

Methodology

Methodology

- Data collection methodology:
  - Using SpaceX API (Rocket Launch Data)
  - Wikipedia (List of Falcon 9 and Falcon Heavy launches)
- Perform data wrangling
  - Identification and replacement of missing values
  - Creation of additional variables (summary; categorical)
- 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 – SpaceX API

1. The SpaceX API is accessed using a GET request to retrieve data about rocket launches.

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

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

2. Converting Response to a JSON file

```
data = pd.json_normalize(response.json())
```

3. Using custom functions to clean data

```
# Call getBoosterVersion
getBoosterVersion(data)
```

```
# Call getCoreData
getCoreData(data)
```

```
# Call getPayloadData
getPayloadData(data)
```

```
# Call getLaunchSite
getLaunchSite(data)
```

4. Combine columns in dictionary to create dataframe

```
launch_dict = {'FlightNumber': list(data['flight_number']),
'Date': list(data['date']),
'BoosterVersion':BoosterVersion,
'PayloadMass':PayloadMass,
```

```
data = pd.DataFrame.from_dict(launch_dict)
```

5. Filtering dataframe for Falcon 9 launches

```
data_falcon9 = data.drop(data[data['BoosterVersion']=='Falcon 1'].index)
```

6. Replacing missing values for Payload Mass with the Mean

```
PM_mean = data['PayloadMass'].mean()
# Replace the np.nan values with its mean value
data_falcon9['PayloadMass'].fillna(value=PM_mean, inplace=True)

data_falcon9.isnull().sum()
```

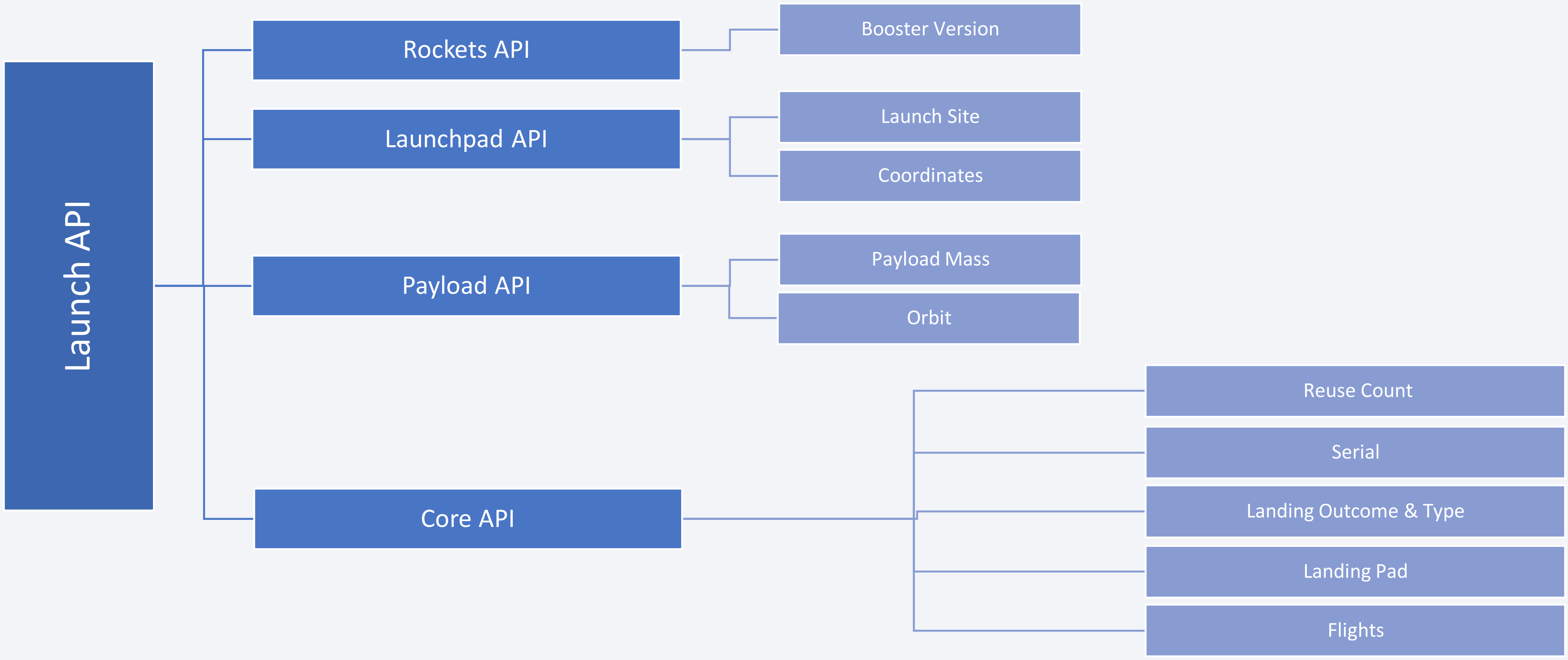
5. Export to CSV

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


Code on GitHub: https://github.com/vietnoir/IBM_DS_CapStone/blob/main/jupyter-labs-spacex-data-collection-api.ipynb

Data Collection – SpaceX API

Data Sources from SpaceX API



Code on GitHub: https://github.com/vietnoir/IBM_DS_CapStone/blob/main/jupyter-labs-spacex-data-collection-api.ipynb

Data Collection - Scraping

The Wikipedia Page '[List of Falcon 9 and Falcon Heavy Launches]' was scraped to retrieve additional data regarding dates and times of launches, booster version, landing status and mass (kg)

Past launches

2010 to 2013

[hide] Flight No.	Date and time (UTC)	Version, Booster [b]	Launch site	Payload[c]	Payload mass	Orbit	Customer	Launch outcome	Booster landing
1	4 June 2010, 18:45	F9 v1.0[7] B0003.1[8]	CCAFS, SLC-40	Dragon Spacecraft Qualification Unit		LEO	SpaceX	Success	Failure[9][10] (parachute)
	First flight of Falcon 9 v1.0.[11] Used a boilerplate version of Dragon capsule which was not designed to separate from the second stage.(more details below) Attempted to recover the first stage by parachuting it into the ocean, but it burned up on reentry, before the parachutes even deployed.[12]								
2	8 December 2010, 15:43[13]	F9 v1.0[7] B0004.1[8]	CCAFS, SLC-40	Dragon demo flight C1 (Dragon C101)		LEO (ISS)	NASA (COTS) NRO	Success[9]	Failure[9][14] (parachute)
	Maiden flight of Dragon capsule, consisting of over 3 hours of testing thruster maneuvering and reentry.[15] Attempted to recover the first stage by parachuting it into the ocean, but it disintegrated upon reentry, before the parachutes were deployed.[12] (more details below) It also included two CubeSats,[16] and a wheel of Brouère cheese.								

1. GET Request

```python
# use requests.get() method with the provided static_url
response = requests.get(static_url)
# assign the response to a object
text = response.text
```

```python
# Use BeautifulSoup() to create a BeautifulSoup object from a response text content
soup = BeautifulSoup(text, "html.parser")
```

2. Identify Table

```python
# Use the find_all function in the BeautifulSoup object,
# Assign the result to a list called `html_tables`
html_tables = soup.find_all('table')
```

```python
first_launch_table = html_tables[2]
print(first_launch_table)
```

3. Extract Column Names, Create Dictionary and Dataframe

```python
for th in names:
    name = extract_column_from_header(th)
    if name is not None and len(name) > 0:
        column_names.append(name)
```

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

```python
for table_number,table in enumerate(soup.find_all('table',"wikitable plainrowheaders collapsible")):
   # get table row
    for rows in table.find_all("tr"):
        #check to see if first table heading is as number corresponding to launch a number
        if rows.th:
            if rows.th.string:
                flight_number=rows.th.string.strip()
                flag=flight_number.isdigit()
```

```python
df.to_csv('spacex_web_scraped.csv', index=False)
```

Code on GitHub: https://github.com/vietnoir/IBM_DS_CapStone/blob/main/jupyter-labs-webscraping.ipynb

Data Wrangling

Identification of missing values

```
df.isnull().sum()/len(df)*100
```

```
FlightNumber       0.000000
Date               0.000000
BoosterVersion     0.000000
PayloadMass        0.000000
Orbit              0.000000
LaunchSite         0.000000
Outcome            0.000000
Flights            0.000000
GridFins           0.000000
Reused             0.000000
Legs               0.000000
LandingPad        28.888889
Block              0.000000
ReusedCount        0.000000
Serial             0.000000
Longitude          0.000000
Latitude           0.000000
dtype: float64
```

Variable Types

```
df.dtypes
```

```
FlightNumber        int64
Date               object
BoosterVersion     object
PayloadMass       float64
Orbit              object
LaunchSite         object
Outcome            object
Flights             int64
GridFins             bool
Reused               bool
Legs                 bool
LandingPad         object
Block             float64
ReusedCount         int64
Serial             object
Longitude         float64
Latitude          float64
dtype: object
```

Launch Sites & Occurence

```
# Apply value_counts() on column LaunchSite
df['LaunchSite'].value_counts()
```

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

Orbits & Occurence

```
# Apply value_counts on Orbit column
df['Orbit'].value_counts()
```

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

Landing Classes

```
for i,outcome in enumerate(landing_outcomes.keys()):
    print(i,outcome)
```

```
0 True ASDS
1 None None
2 True RTLS
3 False ASDS
4 True Ocean
5 False Ocean
6 None ASDS
7 False RTLS
```

Creating landing outcome label

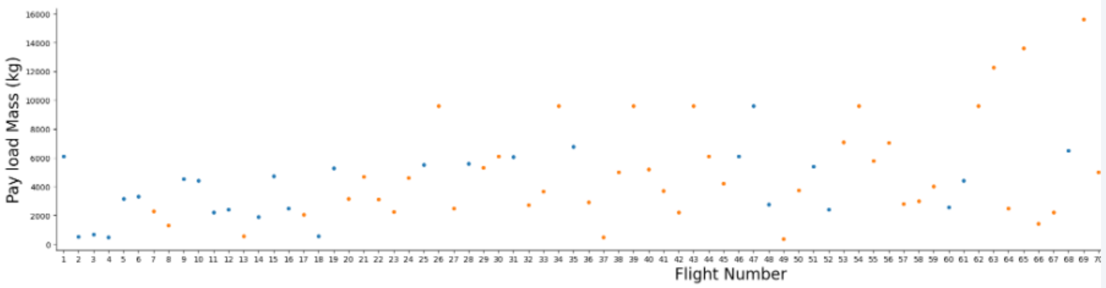
```
landing_class = []
# landing_class = 0 if bad_outcome
if df['Outcome'] in bad_outcomes:
    landing_class = 0
else:
    landing_class = 1
# landing_class = 1 otherwise
```


Code on GitHub: https://github.com/vietnoir/IBM_DS_CapStone/blob/main/labs-jupyter-spacex-Data%20wrangling.ipynb

EDA with Data Visualization

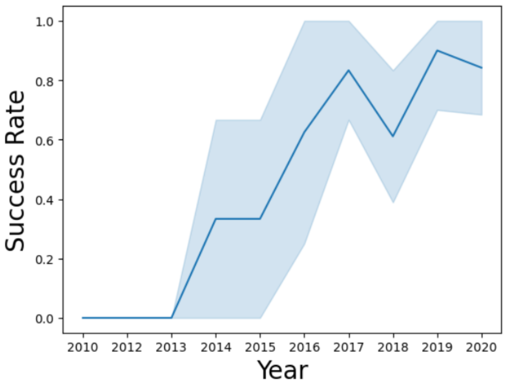
Charts were used to inspect relationships between variables (see slide 11ff.)
- Scatterplots for two continuous variables
- Barplots and Lineplots for one categorical and one continuous

```
sns.catplot(y="PayloadMass", x="FlightNumber", hue="Class", data=df, aspect = 5)
plt.xlabel("Flight Number",fontsize=20)
plt.ylabel("Pay load Mass (kg)",fontsize=20)
plt.show()
```



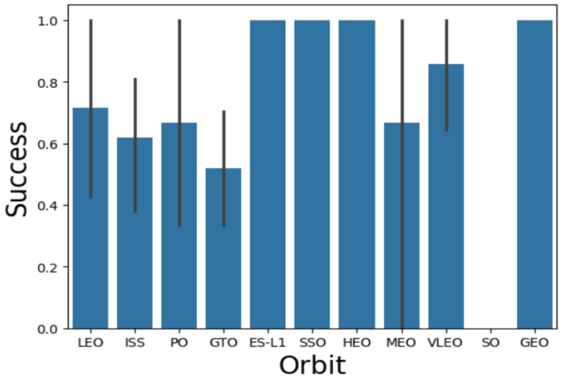
Example of a scatterplot displaying the relationship between Payload Mass (Y) and Flight Number (X)

```
# Plot a line chart with x axis to be the extracted year and y axis to be the success rate
sns.lineplot(y="Class", x="Date", data=df)
plt.xlabel("Year",fontsize=20)
plt.ylabel("Success Rate",fontsize=20)
plt.show()
```



Example of a lineplot displaying the relationship between Success Rate (Y) and Year (X)

```
# HINT use groupby method on Orbit column and get the mean of Class column
sns.barplot(x = 'Orbit', y = 'Class', data = df)
plt.xlabel("Orbit",fontsize=20)
plt.ylabel("Success",fontsize=20)
plt.show()
```



Example of a lineplot displaying the relationship between Success Rate (Y) and Orbit (X)

Code on GitHub: https://github.com/vietnoir/IBM_DS_CapStone/blob/main/edadataviz.ipynb

EDA with SQL

```
%sql SELECT DISTINCT Launch_Site FROM SPACEXTABLE
```

```
%%sql
SELECT Launch_Site
FROM SPACEXTABLE
WHERE Launch_Site LIKE 'CCA%'
LIMIT 5
```

```
%%sql
SELECT SUM(PAYLOAD_MASS__KG_), Customer FROM SPACEXTABLE
WHERE Customer = 'NASA (CRS)'
```

```
%%sql
SELECT AVG(PAYLOAD_MASS__KG_), Booster_Version FROM SPACEXTABLE
WHERE Booster_Version = 'F9 v1.1'
```

```
%%sql
SELECT MIN(Date), Landing_Outcome FROM SPACEXTABLE
WHERE Landing_Outcome = 'Success (ground pad)'
```

```
%%sql
SELECT DISTINCT Booster_Version, Landing_Outcome, PAYLOAD_MASS__KG_ FROM SPACEXTABLE
WHERE Landing_Outcome = 'Success (drone ship)' AND PAYLOAD_MASS__KG_ > 4000 AND PAYLOAD_MASS__KG_ <6000
```

```
%%sql
SELECT DISTINCT(Booster_Version), PAYLOAD_MASS__KG_ FROM SPACEXTABLE
WHERE PAYLOAD_MASS__KG_ = (SELECT MAX(PAYLOAD_MASS__KG_) FROM SPACEXTABLE)
```

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

```
%%sql
SELECT Mission_Outcome, COUNT(Mission_Outcome) AS outcome_count
FROM SPACEXTABLE
GROUP BY Mission_Outcome
```

```
%%sql
SELECT COUNT (Landing_Outcome) FROM SPACEXTABLE
WHERE Date <= '2010-06-04' AND Date >= '2017-03-20'
```

SQL was utilized to obtain descriptive statistics, proving insights into milestones and achievements of Falcon 9 launches

Code on GitHub: https://github.com/vietnoir/IBM_DS_CapStone/blob/main/jupyter-labs-eda-sql-coursera_sqllite.ipynb

Build an Interactive Map with Folium

1. **Map was initiated with start center for orientation**

→ site_map = folium.Map(location=nasa_coordinate, zoom_start=10)

2. **A circle and marker was added for each launch site in the data frame**

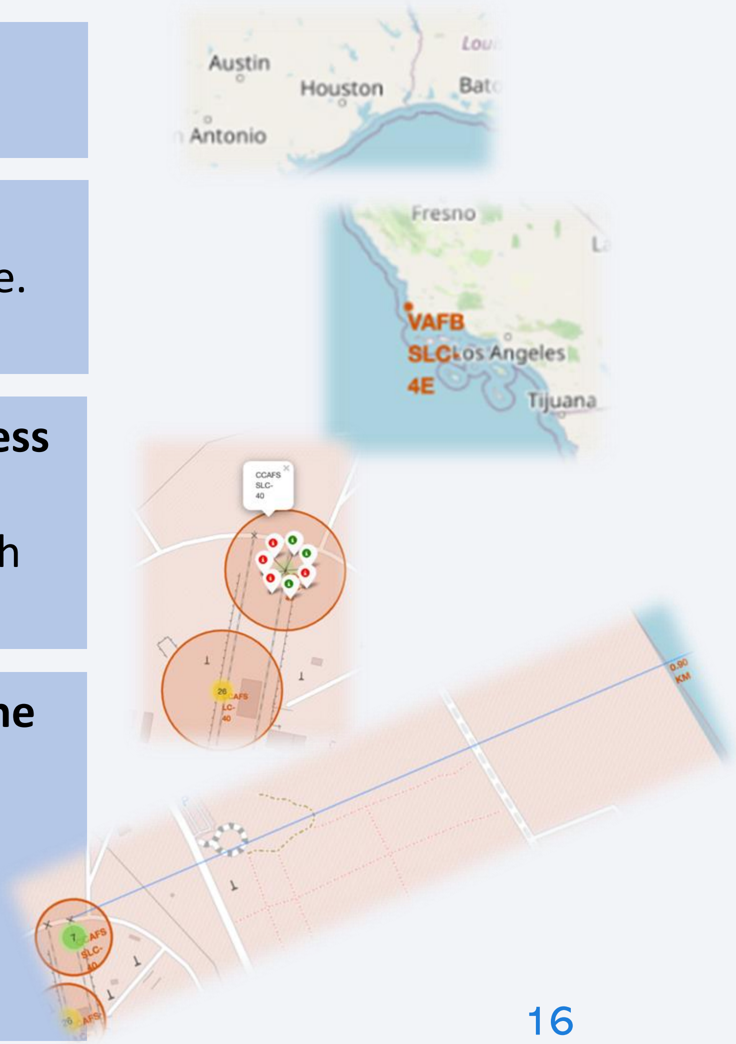
→ folium.Circle() to add a highlighted circle area with a text label on a specific coordinate.
→ folium.Marker() to add an icon showing a locations name

3. **The launch outcomes for each site were added to assess which sites have high success rates**

→ MarkerCluster() was utilized to obtain color coded clusters based on categorical launch outcomes (signifying success or failure)

3. **Distances between launch sites and its proximities were calculated to understand the placement of launch sites**

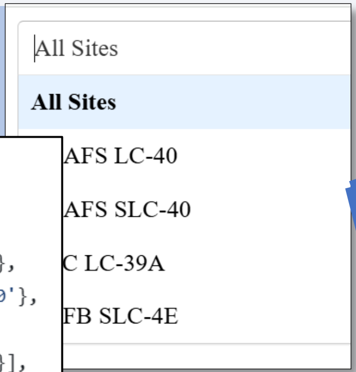
→ MousePosition() was added to retrieve coordinates for a mouse on the map
→ folium.Marker() was utilized for calculated distances between proximities and launch sites
→ folium.PolyLine() was used to draw a line between a selected proximity and site



Code on GitHub: https://github.com/vietnoir/IBM_DS_CapStone/blob/main/jupyter-labs-eda-sql-coursera_sqllite.ipynb

Build a Dashboard with Plotly Dash

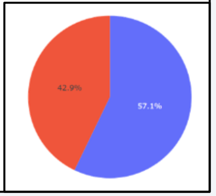
Drop Down Menu for the selection of Launch Site



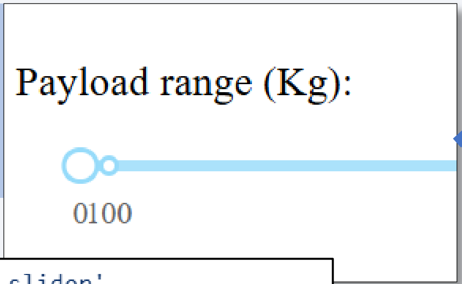

```
dcc.Dropdown(id='site-dropdown',
options=[
    {'label': 'All Sites', 'value': 'ALL'},
    {'label': 'CCAFS LC-40', 'value': 'CCAFS LC-40'},
    {'label': 'CCAFS SLC-40', 'value': 'CCAFS SLC-40'},
    {'label': 'KSC LC-39A', 'value': 'KSC LC-39A'},
    {'label': 'VAFB SLC-4E', 'value': 'VAFB SLC-4E'}],
value = 'ALL',
placeholder = 'Select a Launch Site here',
searchable=True
),
```

1. callback function to display a pie chart of success versus failure rate of the launch site(s)

```
@app.callback(Output(component_id='success-pie-chart', component_property='figure'),
              Input(component_id='site-dropdown', component_property='value'))
def get_pie_chart(entered_site):
    # Filter data based on the selected site
    if entered_site == 'ALL':
        # Calculate the total number of successful and failed launches across all sites
        success_counts = spacex_df['class'].value_counts().reset_index()
        success_counts.columns = ['class', 'count']
        fig = px.pie(success_counts,
                     values='count',
                     names='class',
                     title='Total Successful Launches for All Sites')
    else:
```



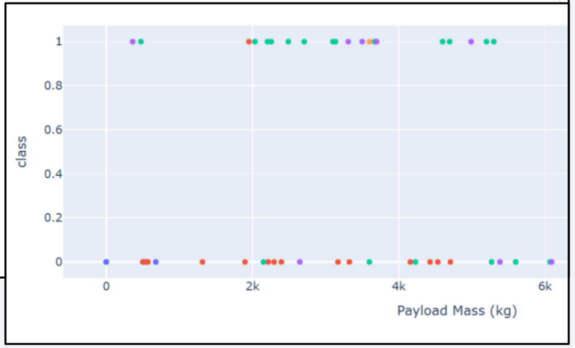
Range Slider for the selection of Payload range




```
dcc.RangeSlider(id='payload-slider',
                min=0, max=10000, step=1000,
                marks={0: '0',
                100: '100'},
                value=[min_payload, max_payload]),
```

2. callback function to display correlation between selected payload range **and** selected launch site(s)

```
@app.callback(Output(component_id='success-payload-scatter-chart', component_property='figure'),
              [Input(component_id='site-dropdown', component_property='value'), Input(component_id="payload-slider", component_property="value")])
def update_scatter_plot(entered_site, payload_range):
    # Filter data based on payload range
    filtered_df = spacex_df[(spacex_df['Payload Mass (kg)'] >= payload_range[0]) &
                            (spacex_df['Payload Mass (kg)'] <= payload_range[1])]

    # Check if a specific launch site was selected
    if entered_site == 'ALL':
        # Render scatter plot for all sites
        fig = px.scatter(filtered_df, x='Payload Mass (kg)', y='class',
                         color='Booster Version Category',
                         title='Correlation between Payload and Success for All Sites')
```



Code on GitHub: https://github.com/vietnoir/IBM_DS_CapStone/blob/main/spacex_dash_app.py

Predictive Analysis (Classification)

1. Independent variables were standardized using the StandardScaler()

```
transform = preprocessing.StandardScaler()

X = transform.fit_transform(X)
```

2. Data was split into training and testing sets

```
X_train, X_test, Y_train, Y_test = train_test_split(X, Y, test_size = 0.2, random_state = 2)
```

3. GridSearch was utilized to find the optimal hyperparameters to minimize errors using cross-validation, e.g., for logistic regression

```
parameters ={'C':[0.01,0.1,1],
             'penalty':['l2'],
             'solver':['lbfgs']}
```

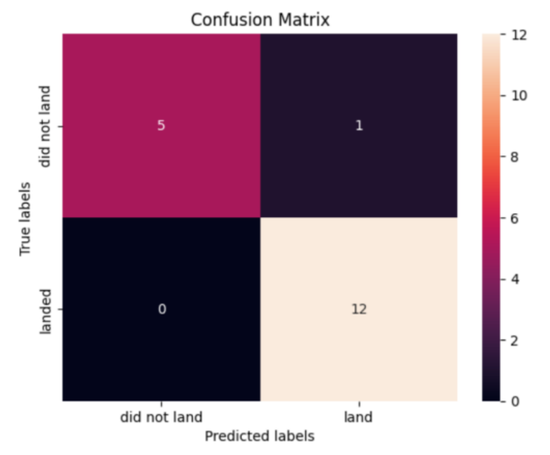
```
parameters ={"C":[0.01,0.1,1],'penalty':['l2'], 'solver':['lbfgs']}# l1 lasso l2 ridge
lr=LogisticRegression()

# Step 3: Create the GridSearchCV object with cv=10
logreg_cv = GridSearchCV(lr, parameters, cv=10)

# Step 4: Fit the GridSearchCV object to the data (assume X, y are defined)
logreg_cv.fit(X, Y)
```

4. Score method was utilized to obtain test accuracy, e.g. logreg:

```
logreg_cv.score(X_test, Y_test)
```

5. Confusion Matrices were used to analyze true and false positives in the predicted labels

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



Models Tested:

- Logistic Regression
- Support Vector Machine
- Decision Tree
- K-nearest Neighbor

Code on GitHub: https://github.com/vietnoir/IBM_DS_CapStone/blob/main/SpaceX_Machine%20Learning%20Prediction_Part_5.ipynb

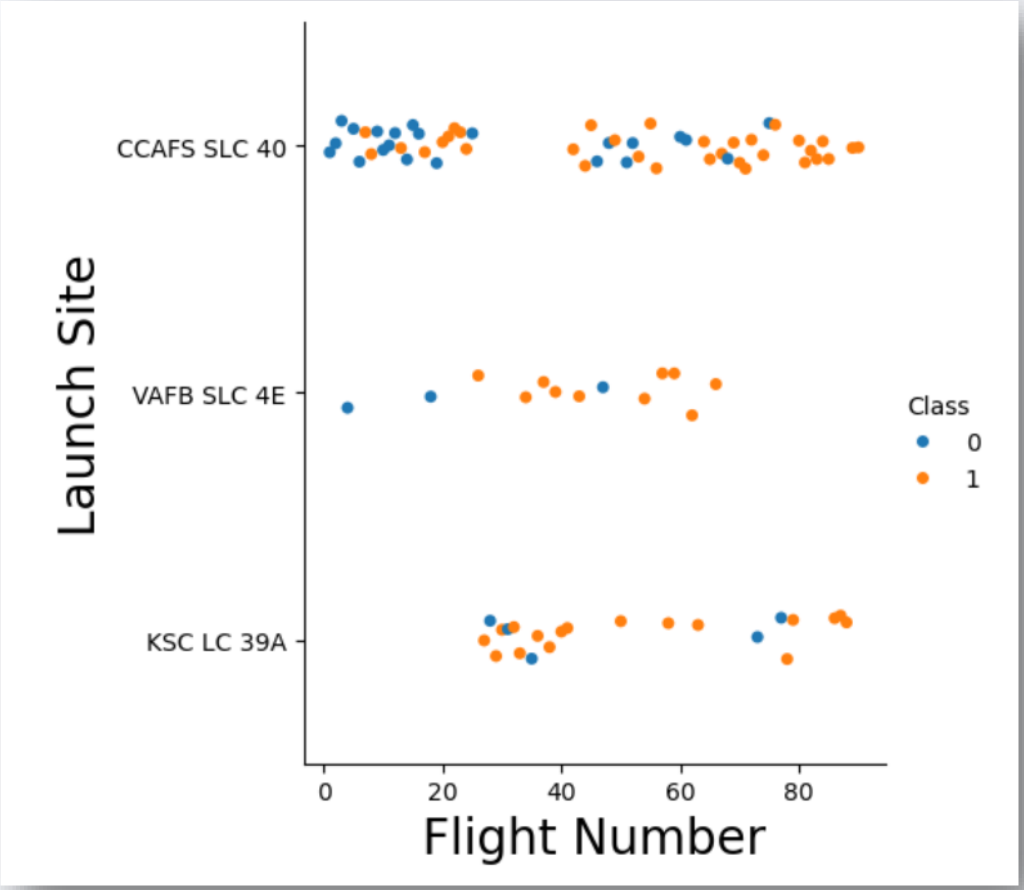
Section 2

Insights drawn from EDA

Flight Number vs. Launch Site

Successfull landings become more likely the more flights were conducted.

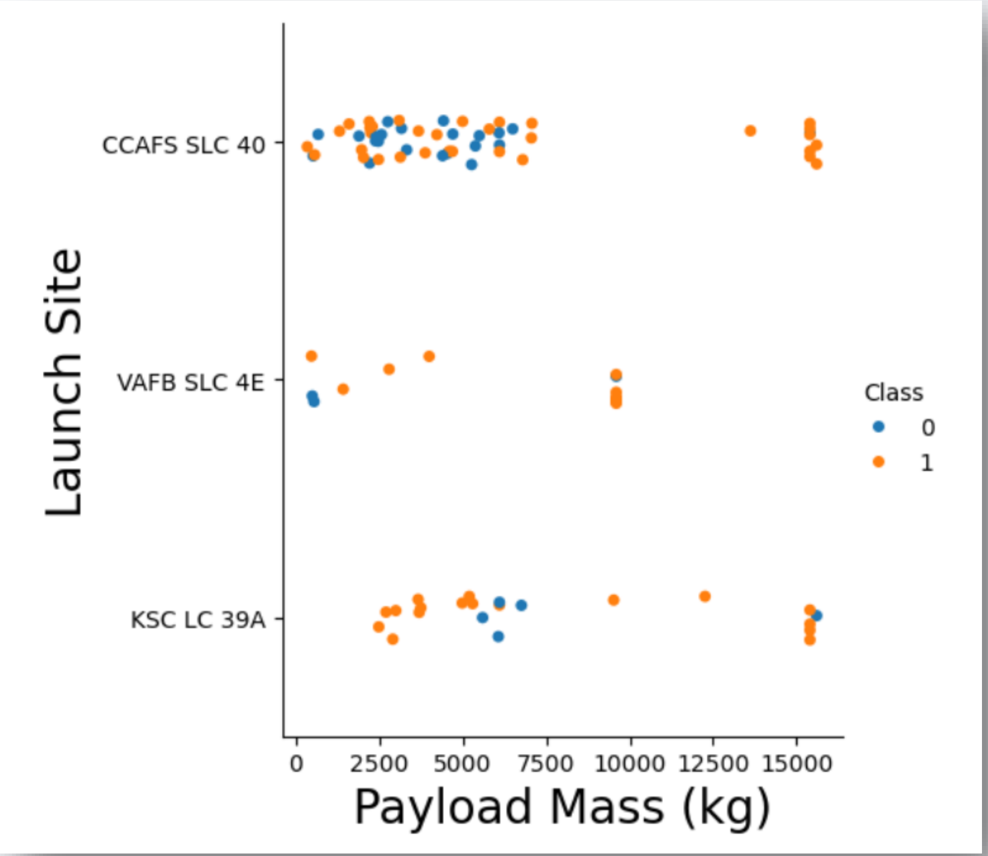
- Flights undertaken vary by launch site, with CCAFS SLC 40 being the most used site
- Site KSC LC 39A appears to be a later addition to the sites



Payload vs. Launch Site

The relationship between payload mass and success rate is less straightforward. However, heavier flights tend to have a higher landing success rate.

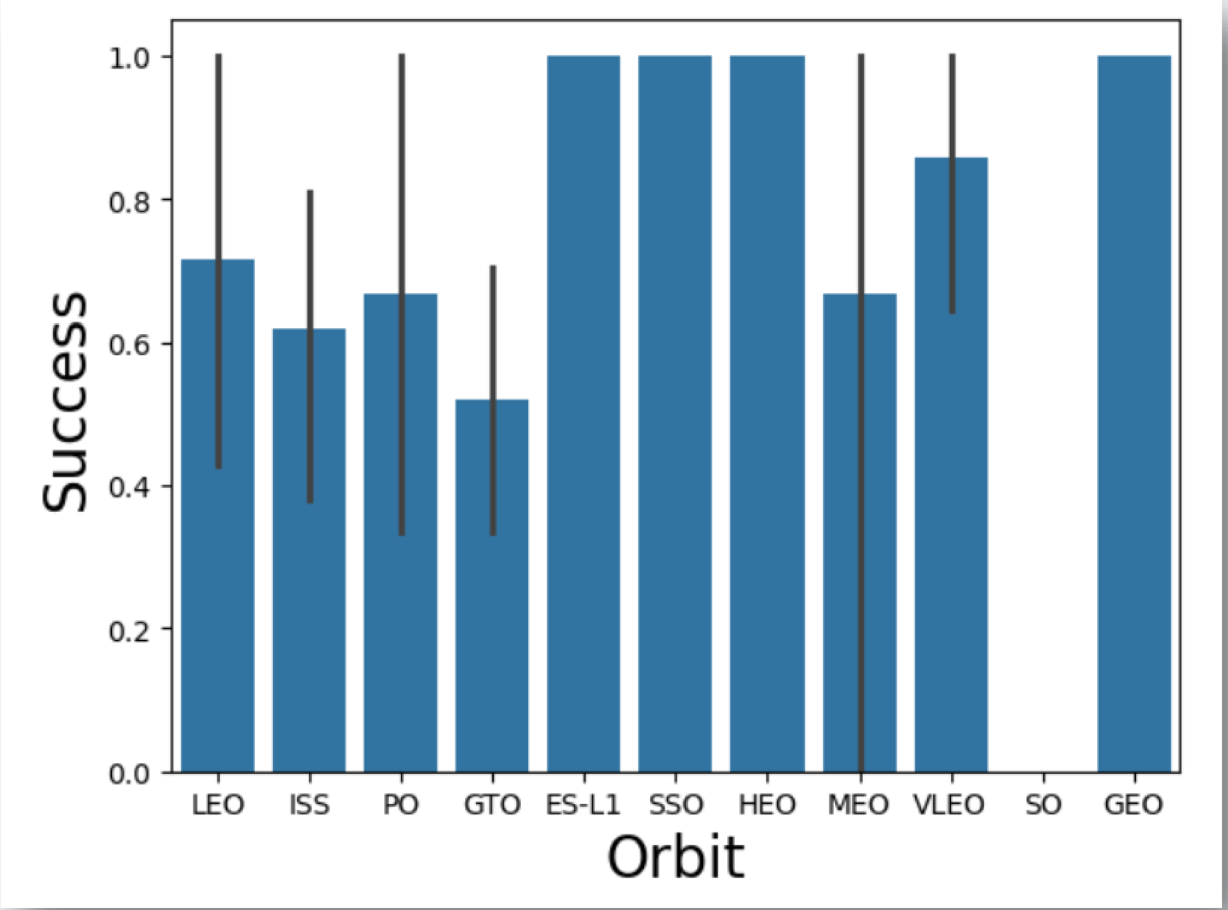
- Payloads above 7500kg occur less often.
- Site VAFB SLC 4E is not used for payloads greater than 10000kg.



Success Rate vs. Orbit Type

Flights in Orbit VLEO have the highest landing success rate.

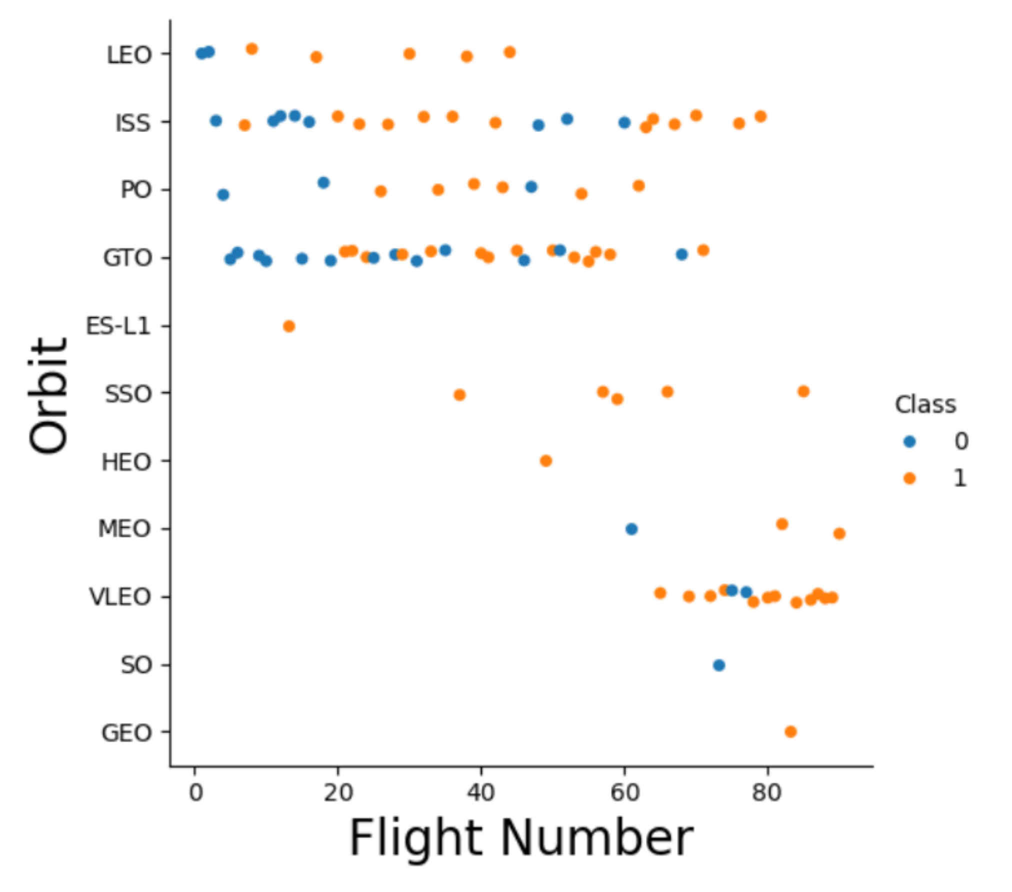
- The Orbit GTO has the lowest landing succes rate.
- Flights in Orbits ES-L1, SSO, HEO and GEO have been used each for one successful landing.



Flight Number vs. Orbit Type

While landing success generally increased with flight number, selected Orbits have changed and affected landing success as well.

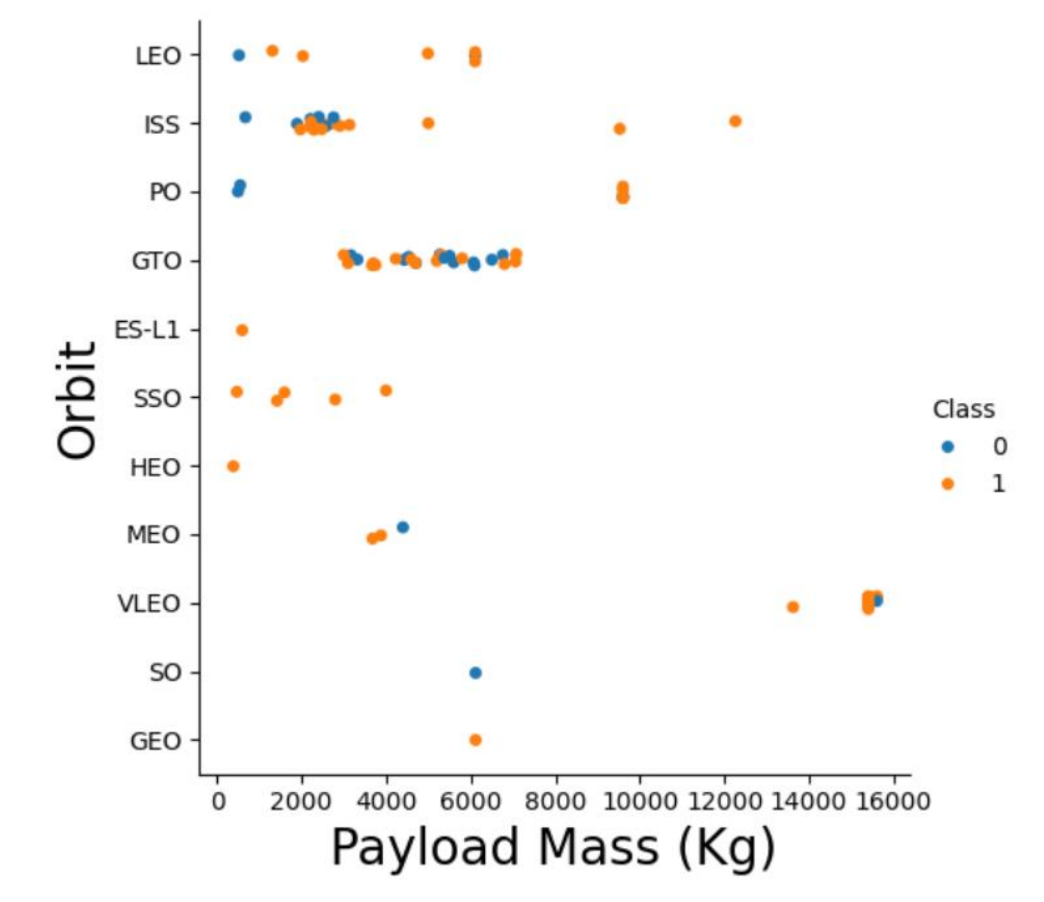
- The earlier flights were mostly restricted to the Orbits LEO, ISS, PO and GTO.
- The later flights were mostly undertaken in Orbits ISS, SSO, MEO and VLEO.



Payload vs. Orbit Type

For orbits LEO, ISS and PO, landing success increases with payload mass.

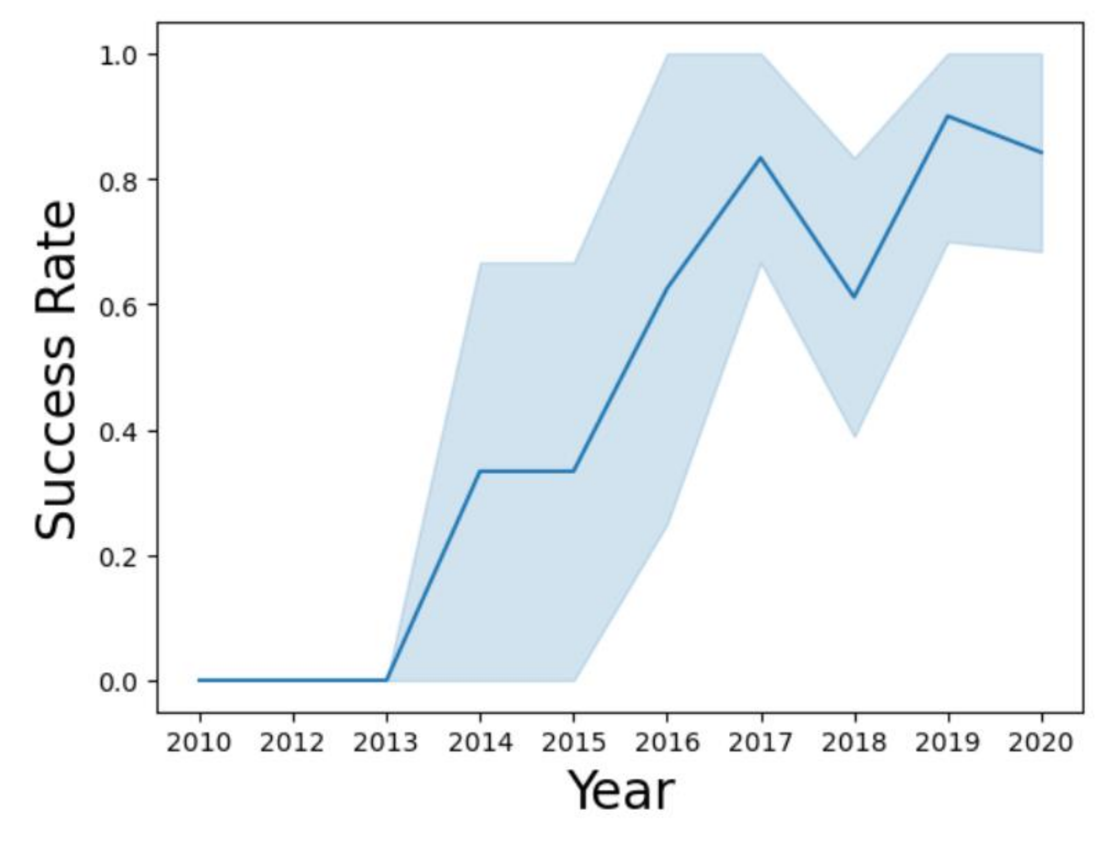
- Several orbits have not been utilized for payloads above 7500.
- The orbit VLEO is the only one successfully used for payloads around 16000



Launch Success Yearly Trend

The landing success rate has been increasing since 2013 until 2020.

- Years 2017 and 2019 have seen the most successful landings, with success rates peaking above 80%.
- In 2018, success rate dropped to 60% but recovered in the following year.



All Launch Site Names

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

Four launch sites are used for SpaceX Falcon 9 flights.

- **CCAFS LC-40: Cape Canaveral Air Force Station, Launch Complex 40**
  - LC-40 has been a primary site for SpaceX launches
- **CCAFS SLC-40: Cape Canaveral Space Launch Complex 40**
  - Similar to CCAFS LC-40 but represents a different designation within the launch site
- **VAFB SLC-4E: Vandenberg Air Force Base, Space Launch Complex 4 East**
  - SLC-4E is especially used for Falcon 9 launches requiring a southward trajectory
- **KSC LC-39A: Kennedy Space Center, Launch Complex 39A**
  - Historic launch site used for Apollo missions and Space Shuttle launches.
  - Modified for Falcon 9 and Falcon Heavy launches, including crewed missions to the International Space Station (ISS)

Launch Site Names Begin with 'CCA'

Launch_Site
CCAFS LC-40
CCAFS LC-40
CCAFS LC-40
CCAFS LC-40
CCAFS LC-40

CCAFS LC-40

(Cape Canaveral Air Force Station, Launch Complex 40)

- Located at the Cape Canaveral Space Force Station in Florida
- Commonly used by SpaceX for launching Falcon 9 rockets
- Commercial missions and NASA missions.
- **LC-40 has been a primary site or SpaceX launches.**

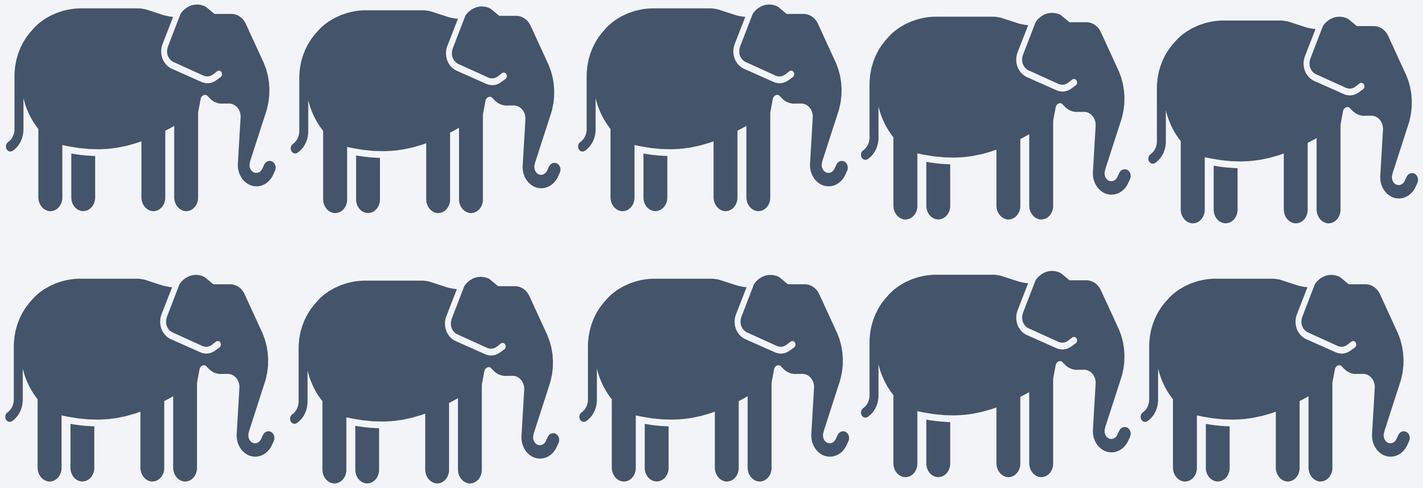
http://www.spacex.com/media-gallery/detail/1662/991 CC0 1.0

Total Payload Mass

SUM(PAYLOAD_MASS__KG_)	Customer
45596	NASA (CRS)

The total payload carried by boosters from NASA amounts to 45,596 kg.

This is approximately as much as the weight of ten African elephants.



Average Payload Mass by F9 v1.1

AVG(PAYLOAD_MASS__KG_)	Booster_Version
2928.4	F9 v1.1

Avialuh (based on work of Lucabon) – CC A-S 4.0

The average payload of approximately 2,928 kg suggests that the F9 v1.1 booster was capable of carrying medium-weight payloads.

These are suitable for various types of satellite launches and cargo missions to the International Space Station (ISS).

First Successful Ground Landing Date

MIN(Date)	Landing_Outcome
2015-12-22	Success (ground pad)

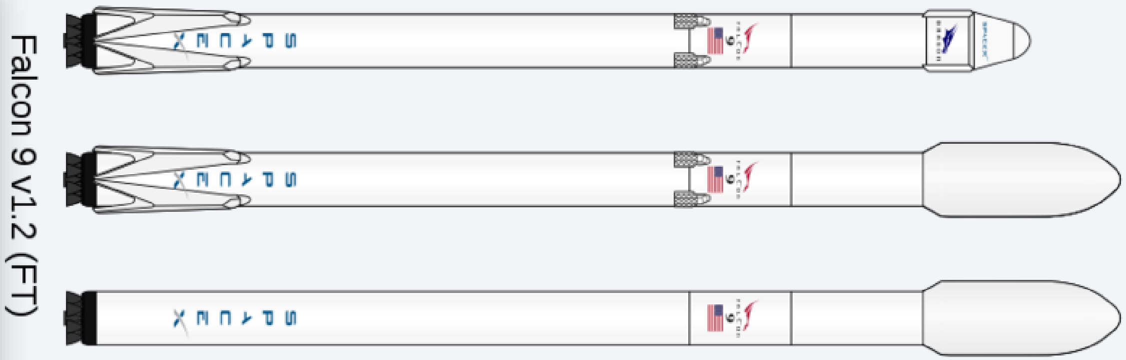
The December 22, 2015, landing was a historic event for SpaceX and the space industry, demonstrating that rockets could be reliably recovered after launch.

Falcon 9 Flight 20 successfully delivered 11 satellites to orbit before returning to Cape Canaveral, Florida.

https://www.flickr.com/photos/spacex/23802549782/ CC0 1.0

Successful Drone Ship Landing with Payload between 4000 and 6000

Booster_Version	Landing_Outcome	PAYLOAD_MASS__KG_
F9 FT B1022	Success (drone ship)	4696
F9 FT B1026	Success (drone ship)	4600
F9 FT B1021.2	Success (drone ship)	5300
F9 FT B1031.2	Success (drone ship)	5200



Avialuh (based on work of Lucabon) – CC A-S 4.0

The Falcon 9 Full Thrust (F9 FT) boosters were successful in delivering heavy payloads while also achieving drone ship landings.

This capability underscores SpaceX's ability to launch substantial payloads while still recovering and reusing rockets.

Total Number of Successful and Failure Mission Outcomes

SpaceX displays a very high mission success rate.

A mission is considered successful if the rocket delivers its payload to the intended orbit or completes its primary objective, regardless of whether the booster is recovered or not.

Mission_Outcome	total_count
Failure (in flight)	1
Success	98

Boosters Carried Maximum Payload

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

Avialuh (based on work of Lucabon) – CC A-S 4.0

Falcon 9 Block 5 boosters are suitable for carrying a consistent payload of 15,600 kg.

The reuse of boosters with multiple flights demonstrates SpaceX's success in achieving reusability.

2015 Launch Records

substr(Date, 6,2)	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

https://www.wired.com/2015/04/analysis-falcon-9-crash-landing/

Two Falcon 9 version 1.1 missions from 2015, both launched from CCAFS LC-40 and both attempting drone ship landings ended in failure.

The use of drone ships suggests these missions required trajectories that made ground landings impractical. The booster version (F9 v1.1) may also have lacked some of the landing improvements introduced in later Falcon 9 versions, which could have contributed to landing challenges.

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

SpaceX has successfully landed boosters on both drone ships and ground pads.

Entries for controlled and uncontrolled ocean landings indicate that some missions did not attempt a full recovery.

Landing_Outcome	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

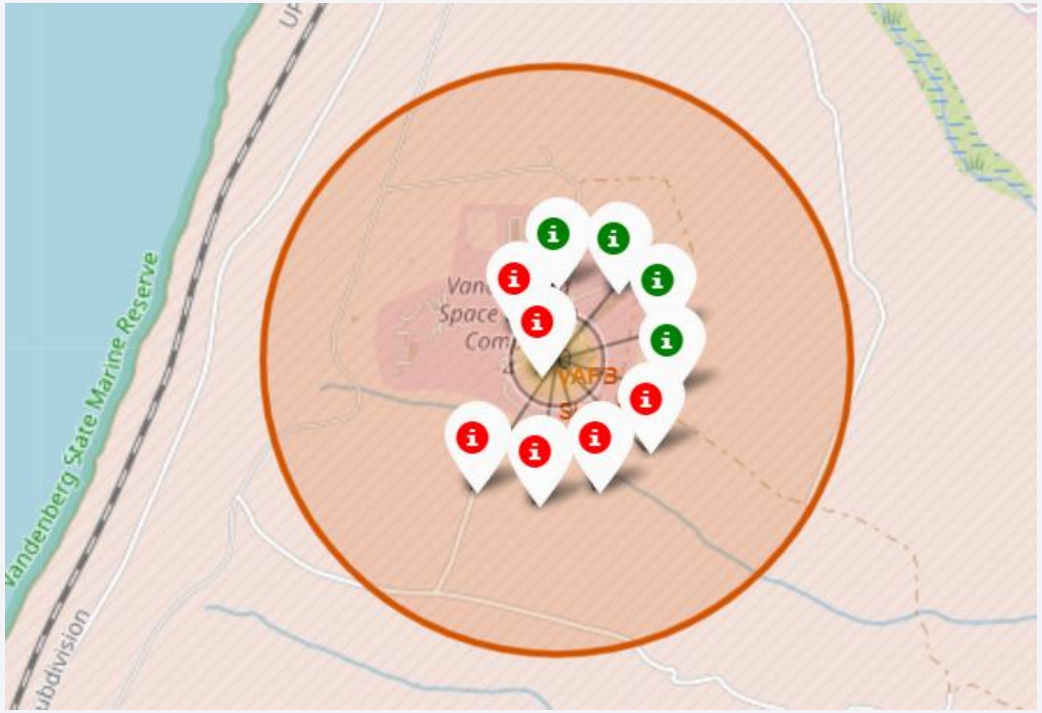
SpaceX Launch Sites

Launch Sites are positioned on both east and west cost to optimize the launch trajectory for different orbital requirements.

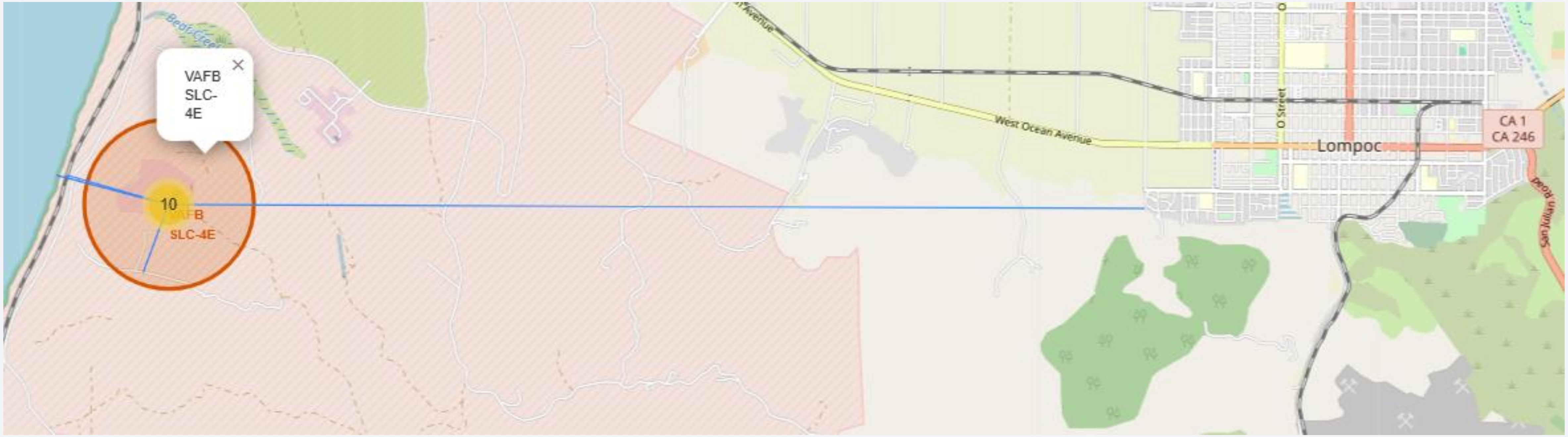
Launch Outcomes on Site VAFB SLC-4E

Landing outcomes at VAFB SLC-4E have included a mix of successful and failed drone ship landings.

At VAFB SLC-4E launches typically head southward over the Pacific Ocean, with boosters usually recovered on drone ships. Ground landings are generally not attempted here due to trajectory and logistical constraints.



Location of Site VAFB SLC-4E and its Proximities



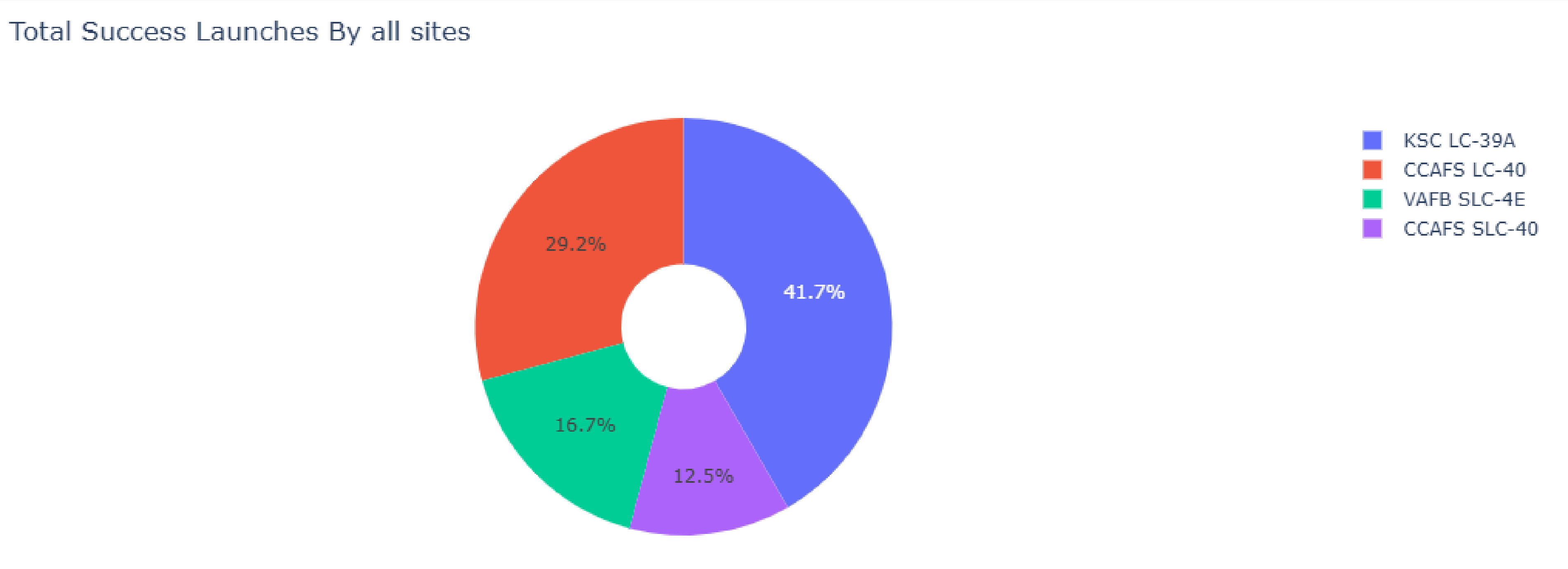
VAFB SLC-4E is positioned along the Pacific Ocean, allowing safe southward launch trajectories for missions to polar and sun-synchronous orbits without passing over populated areas.

The sites coastal and relatively remote location minimizes risks to populated areas, making it ideal for space launches. The nearest town, Lompoc (CA), is about 10 miles away.

Section 4

Build a Dashboard with Plotly Dash

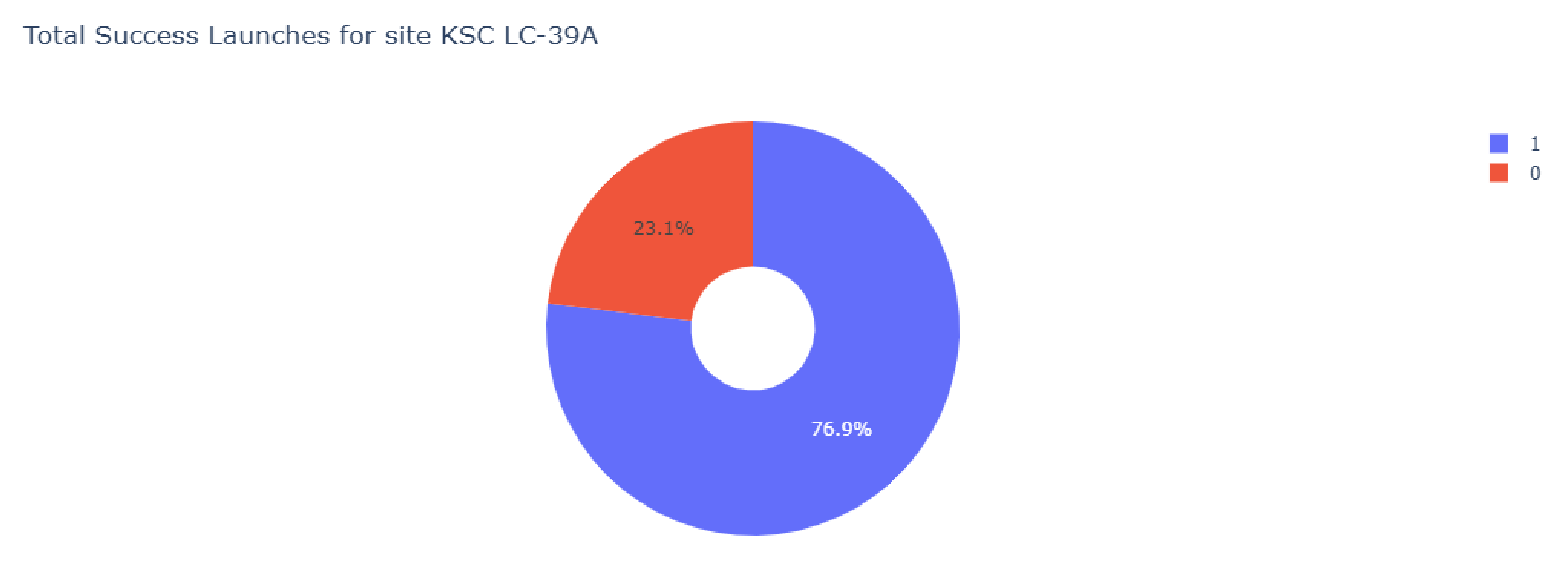
Total Successful Launches for All Sites



The pie chart shows the total amount of successful launches by all sites.

We can see that site KSC LC-39A has had the most successful launches.

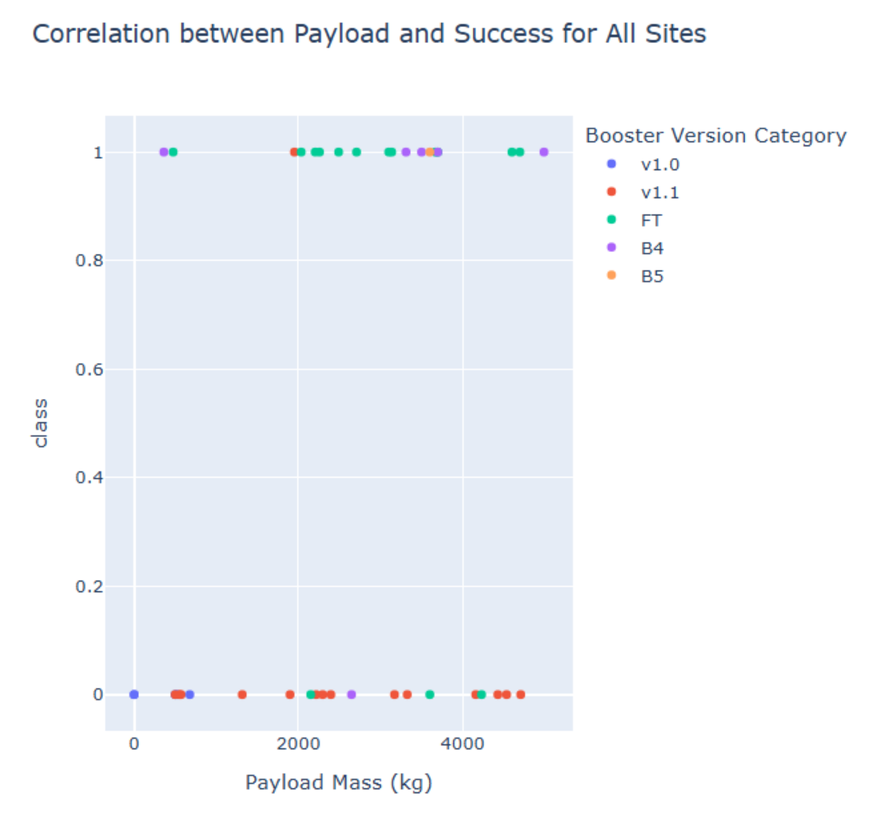
Successful Launches for Site KSC LC-39A

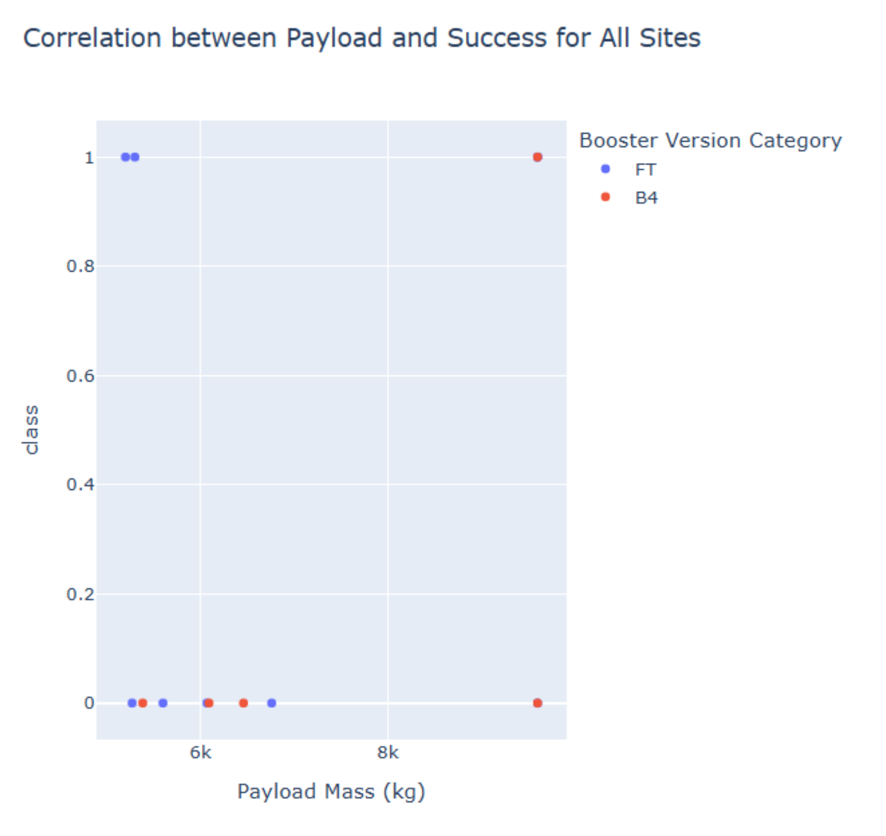


The pie chart shows the total amount of successful launches for site KSC LC-39A.

We can see that that a successful launch for site KSC LC-39A is about 3.3 times more likely than a failure.

Launch Success and Payload Range





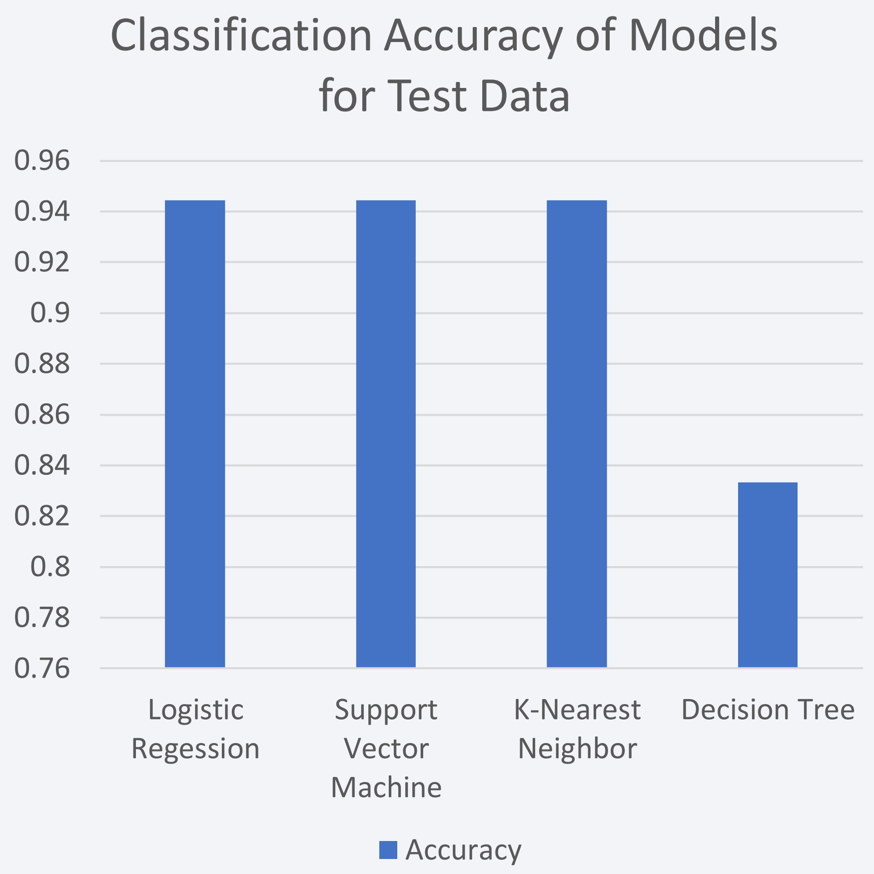
Successful launches are more likely for lower compared to higher payloads.

Section 5

Predictive Analysis (Classification)

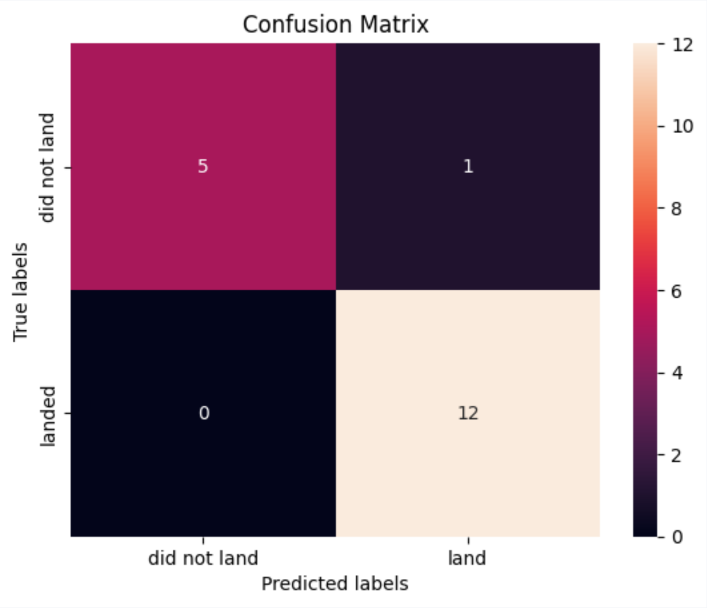
Classification Accuracy

The Logistic Regression, Support Vector Machine and K-Nearest Neighbor perform equally well in predicting landing outcomes.

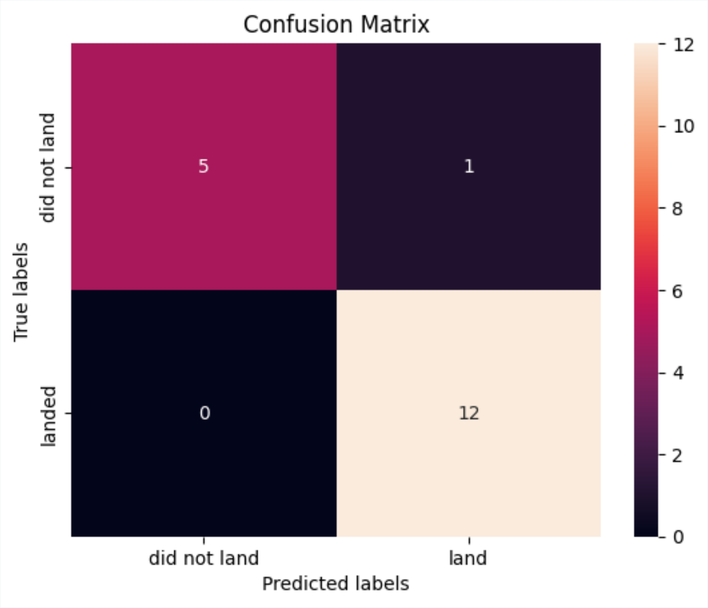


Confusion Matrix

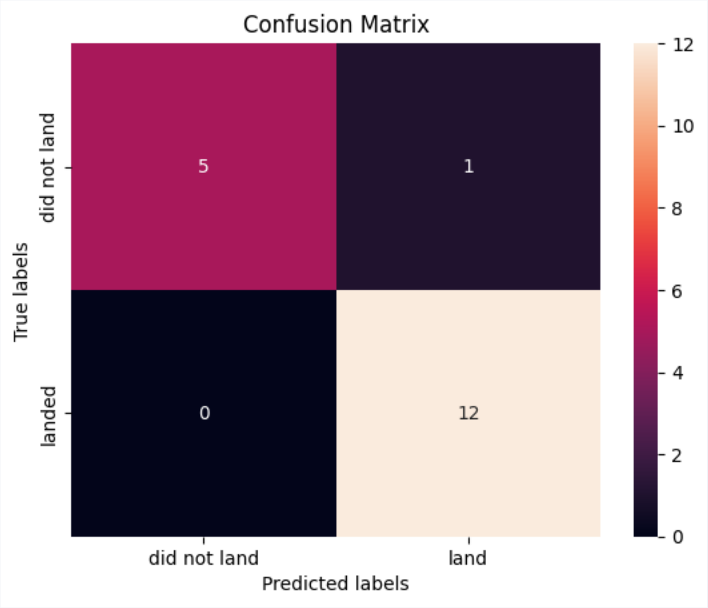
Logistic Regression



Support Vector Machine



K-Nearest Neighbor



All the models can successfully distinguish between the different landing outcomes. All models display issues with identifying false positives.

Conclusions

- SpaceX's landing success rate has steadily improved with each passing year, especially peaking in 2017 and 2019 with over 80% success.
- Launch sites on both the east and west coasts maximize trajectory options, tailored for different orbital needs.
- Flights in Orbit VLEO have the highest landing success rate.
- The KSC LC-39A site makes a successful launch 3.3 times more likely than a failure.
- Logistic Regression, SVM, and K-Nearest Neighbor models all accurately predict landing outcomes, though false positives remain a challenge.

Future Steps: AI models should be adjusted to prioritize accuracy in identifying false positives, as they are more costly compared to false negatives.

Appendix

Main Sources

- https://www.spacex.com
- https://www.nasa.gov
- https://www.faa.gov
- https://en.wikipedia.org/wiki/List_of_Falcon_9_and_Falcon_Heavy_launches

Thank you!