IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

Thibault De La Selle
September 20th, 2024

Outline

EXECUTIVE SUMMARY

INTRODUCTION

METHODOLOGY

RESULTS

CONCLUSION

APPENDIX

Executive Summary

This capstone project aims to determine the success probability of the SpaceX Falcon 9 first stage given previously collected data.

Data manipulation & methodologies used:

- Use of SpaceX API & Web Scraping - BeautifulSoup
- Data wrangling (classification, data transformation) - Pandas, NumPy
- Python exploratory data analysis using SQL
- Visualizations using Plotly, MatPlotLib, Seaborn, Folium for interactive dashboard, maps
- Machine Learning for predictive analysis of rocket landing success - SKLearn

Results:

- SpaceX launch success climbed from 37% in 2014 to above 80% since 2019
- Higher payload mass (8.000kg and above) has a near flawless launch success
- ML models all predicted an 83.334% success rate (Falcon 9 booster landing)
- Further data would be necessary to increase precision of prediction

GitHub Project files: https://github.com/TheCanadianShield/Applied-Data-Science-Capstone

Introduction

Project Background & Context

- Private firms like SpaceX have decreased launch costs from $165 millions to $62 millions. These savings come partly from reusable rockets and boosters
- Determining booster landing rates helps predict launch costs, to be used by competitors in understanding the current competitive landscape

For valuable predictions, we must build upon the following:

- Discover the factors influencing a successful Falcon 9 booster landing
- Understand the conditions to optimize the probability of success
- Train a machine learning model to predict future SpaceX Falcon 9 booster recovery based on existing data

Section 1

Methodology

Methodology

Executive Summary

- Data collection methodology:
  - SpaceX REST API, web scraping of the SpaceX Wikipedia page
- Perform data wrangling
  - Isolated for Falcon 9 specific data
  - Use of One Hot Encoding to transform categorical variables as numerical values (for ML)
- Perform exploratory data analysis (EDA) using visualization and SQL
- Perform interactive visual analytics using Folium and Plotly Dash
- Perform predictive analysis using classification models
  - Building, tuning ML classification models with GridSearchCV

Data Collection

SpaceX API

API
- GET request to API for launch data

Extract
- Extract data from JSON file
- Convert to Pandas Data Frame

Clean
- Filter, clean and convert for a sanitized data set

Export
- Export data to a CSV file for next phase

Wikipedia Web Scraping

HTTP
- GET request to web URL for launch table data

Extract
- Parse data with BeautifulSoup

Clean
- Filter, clean and convert for a sanitized Pandas Data Frame

Export
- Export data to a CSV file for next phase

Data Collection – SpaceX API

API

Extract

Clean

Export

1. We're calling the API and getting all launch data to a normalized dataframe

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

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

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

2. Declaring variables for the dataframe

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

```
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}
```

3. Sorting and cleaning the data

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

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

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

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

```
data_falcon9 = df[df['BoosterVersion']!='Falcon 1']
```

```
# Calculate the mean value of PayloadMass column
payloadmassavg = data_falcon9['PayloadMass'].mean()
# Replace the np.nan values with its mean value
data_falcon9['PayloadMass'].replace(np.nan, payloadmassavg, inplace=True)
```

4. Extracting the data

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


Using the open-source SpaceX API available at https://github.com/r-spacex/SpaceX-API/
Project available at https://github.com/TheCanadianShield/Applied-Data-Science-Capstone/blob/main/Week%201%20-%20Lab%20-%20Data%20Collection%20API%20jupyter-labs-spacex-data-collection-api.ipynb

Data Collection - Scraping

1. HTTP request

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

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

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

2. Extract with the BeautifulSoup object

```
# Use BeautifulSoup() to create a BeautifulSoup object from a response text content
soup = BeautifulSoup(data, 'html5lib')
```

```
# Use soup.title attribute
print(soup.title)
```

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

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

```
# Let's print the third table and check its content
first_launch_table = html_tables[2]
print(first_launch_table)
```

3. Cleaning the data

```
for row in first_launch_table.find_all('th'):
    name = extract_column_from_header(row)
    if (name != None and len(name) > 0):
        column_names.append(name)
```

```
extracted_row = 0
#Extract each table
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()
        else:
            flag=False
        #get table element
        row=rows.find_all('td')
        #if it is number save cells in a dictonary
        if flag:
```

4. Export the data

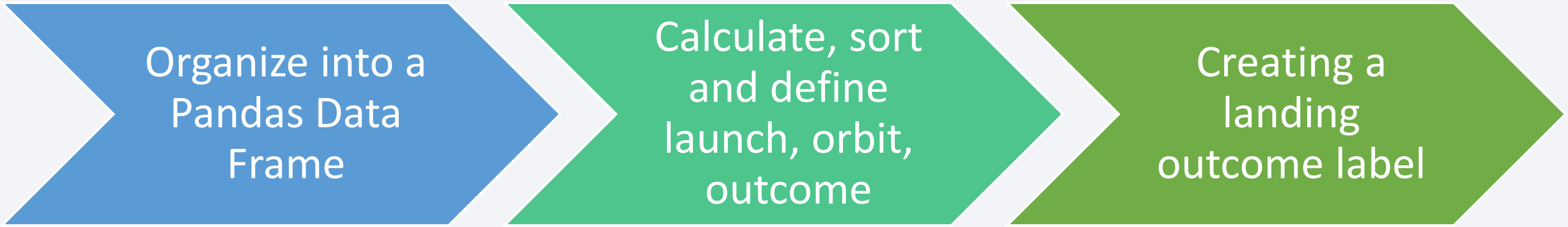
```
df.to_csv('spacex_web_scraped.csv', index=False)
```


Using the Wikipedia page available at
https://en.wikipedia.org/w/index.php?title=List_of_Falcon_9_and_Falcon_Heavy_launches&oldid=1027686922
Project available at https://github.com/TheCanadianShield/Applied-Data-Science-Capstone/blob/main/Week%201%20-%20Lab%20-%20Web%20Scraping%20jupyter-labs-webscraping.ipynb

Data Wrangling

Goal:

- Use Exploratory Data Analysis (EDA) for pattern matching, to be used for the machine learning model
- Creation of a label to define the outcome of each Falcon 9 flight success, known as ‘Class’ with a binary value, 1 being successful, 0 is not.
  - This also removes ambiguity in merging locations (Ocean, RTLS ground, ASDS drone ship) launches and sorts such data



Data Wrangling

1. Organize into a Pandas Data Frame
2. Calculate, sort and define launch, orbit, outcome
3. Creating a landing outcome label

1.

```
df=pd.read_csv("https://cf-courses-data.s3.us.cloud-object-storage.appdomain.cloud/IBM-DS0321EN-SkillsNetwork/datase
df.head(10)
```

	FlightNumber	Date	BoosterVersion	PayloadMass	Orbit	LaunchSite	Outcome	Flights	GridFins	Reused	Legs
0	1	2010-06-04	Falcon 9	6104.959412	LEO	CCAFS SLC 40	None None	1	False	False	False
1	2	2012-05-22	Falcon 9	525.000000	LEO	CCAFS SLC 40	None None	1	False	False	False

2.

```
# Apply value_counts() on column LaunchSite

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

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

```
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
```

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

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

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

```
{'False ASDS', 'False Ocean', 'False RTLS', 'None ASDS', 'None None'}
```


Project available at https://github.com/TheCanadianShield/Applied-Data-Science-Capstone/blob/main/Week%201%20-%20Lab%20-%20Data%20Wrangling%20labs-jupyter-spacex-Data%20wrangling.ipynb

Data Wrangling

1. Organize into a Pandas Data Frame → 2. Calculate, sort and define launch, orbit, outcome → 3. Creating a landing outcome label

3.

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

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

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

	Class
0	0
1	0
2	0
3	0
4	0
5	0
6	1
7	1

We can use the following line of code to determine the success rate:

```python
df["Class"].mean()
```

```
0.6666666666666666
```

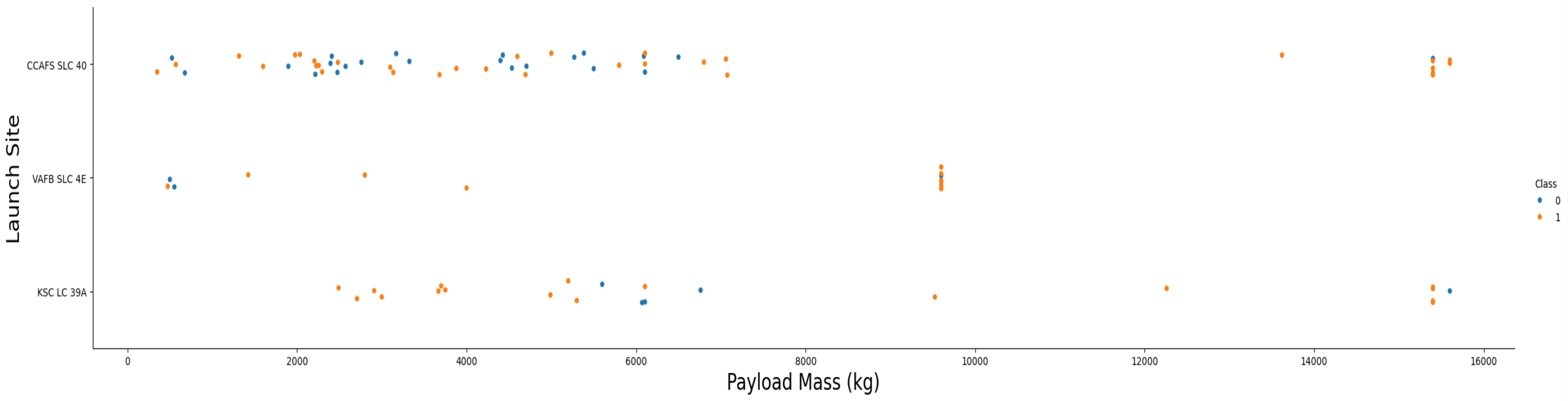
```python
df.to_csv("dataset_part_2.csv", index=False)
```


Project available at https://github.com/TheCanadianShield/Applied-Data-Science-Capstone/blob/main/Week%201%20-%20Lab%20-%20Data%20Wrangling%20labs-jupyter-spacex-Data%20wrangling.ipynb

EDA with Data Visualization

Data Visualization allows us to build knowledge and identify patterns, through tools like MatPlotLib and SeaBorn

This scatter plot highlights the relationship between Payload Mass and Launch Site. We can deduct that certain launch sites have payload restrictions.



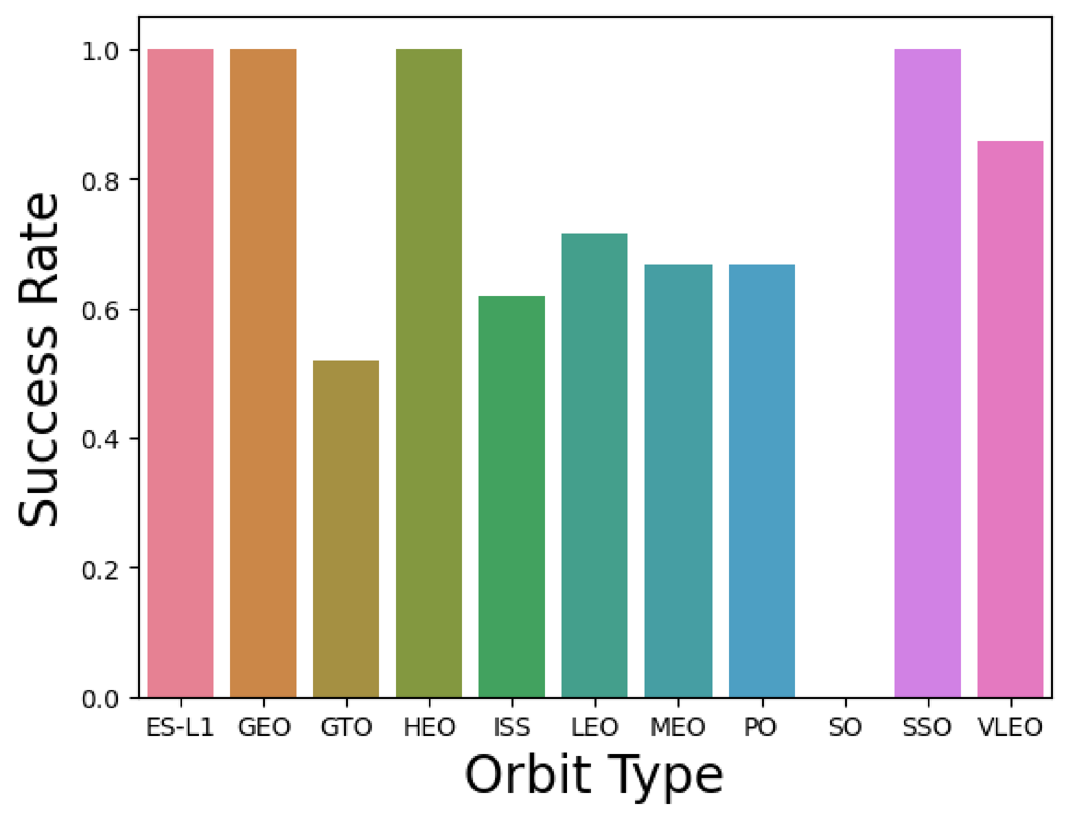
Plots help show relationships between 2 variables, such as Flight Number, Launch Site, Payload, or Orbit Type

```
sns.catplot(y="LaunchSite", x="PayloadMass", hue="Class", data=df, aspect = 5)
plt.xlabel("Payload Mass (kg)",fontsize=20)
plt.ylabel("Launch Site",fontsize=20)
plt.show()
```

Project available at https://github.com/TheCanadianShield/Applied-Data-Science-Capstone/blob/main/Week%202%20-%20Lab%20-%20EDA%20with%20Visualization%20Lab%20edadataviz.ipynb

EDA with Data Visualization

This bar chart highlights the relationship between the values of two variables. In this case, we can identify the risk associated with launching in each orbit type based on its success rate.

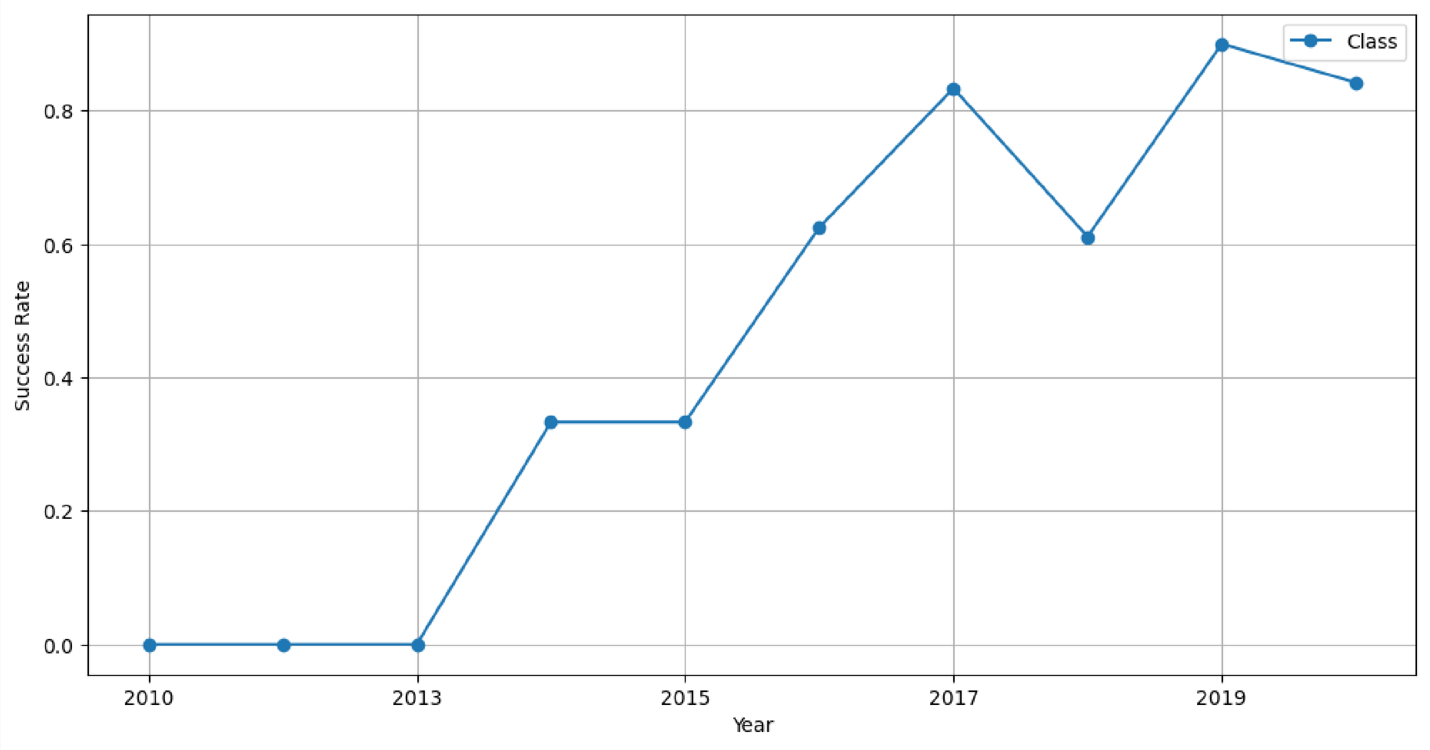



```python
subdf = df[['Orbit','Class']].groupby(['Orbit'],as_index=False).mean()
sns.barplot(x="Orbit", y="Class", hue="Orbit", data=subdf)
plt.xlabel("Orbit Type",fontsize=20)
plt.ylabel("Success Rate",fontsize=20)
plt.show()
```


Project available at https://github.com/TheCanadianShield/Applied-Data-Science-Capstone/blob/main/Week%202%20-%20Lab%20-%20EDA%20with%20Visualization%20Lab%20edadataviz.ipynb

EDA with Data Visualization

This line chart highlights the changes of a relationship of the value of a variable over time. Here, the success rate is rated a time scale of 10 years, indicating a sharp increase in successful launches since 2015.




```
df1 = df[['Date','Class']].groupby(['Date']).mean()
df1.plot(kind='line',figsize=(12, 6), marker='o', linestyle='-')
plt.xlabel('Year')
plt.ylabel('Success Rate')
plt.grid(True)
plt.show()
```


Project available at https://github.com/TheCanadianShield/Applied-Data-Science-Capstone/blob/main/Week%202%20-%20Lab%20-%20EDA%20with%20Visualization%20Lab%20edadataviz.ipynb

EDA with SQL

SQL queries are an efficient gateway to SpaceX dataset insights. Here are the queries used against an SQLite database:

1. Display the names of the unique launch sites in the space mission
2. Display 5 records where launch sites begin with the string 'CCA'
3. Display the total payload mass carried by boosters launched by NASA (CRS)
4. Display average payload mass carried by booster version F9 v1.1
5. List the date when the first succesful landing outcome in ground pad was achieved
6. List the names of the boosters which have success in drone ship and have payload mass greater than 4000 but less than 6000
7. List the total number of successful and failure mission outcomes
8. List the names of the booster_versions which have carried the maximum payload mass. Use a subquery
9. List the records which will display the month names, failure landing_outcomes in drone ship ,booster versions, launch_site for the months in year 2015
10. Rank the count of successful landing_outcomes between the date 04-06-2010 and 20-03-2017 in descending order

Project available at https://github.com/TheCanadianShield/Applied-Data-Science-Capstone/blob/main/Week%202%20-%20Lab%20-%20SQL%20Notebook%20Peer%20Assignment%20eda-sql-coursera_sqllite.ipynb

Build an Interactive Map with Folium

- Folium is a python library supporting interactive maps for geospatial data visualization and analysis
- This tool was used to highlight the SpaceX launch pads in red circles, added lines and points to certain points of interests measuring their distance and impact to the launch pad.
- Marker clusters are a group of launches done at a landing pad, tracking the success of each flight done through colors, green indicating success, red indicating a failure.
- This interactive map helped answer questions on the proximity of railways, highways, coastlines and cities from SpaceX launchpads

Project available at https://github.com/TheCanadianShield/Applied-Data-Science-Capstone/blob/main/Week%203%20-%20Lab%20-%20Interactive%20Visual%20Analytics%20with%20Folium%20lab%20lab_jupyter_launch_site_location.ipynb

Build a Dashboard with Plotly Dash

- Plotly Dash is an interactive dashboard tool used for real-time data visualization.
- Built a dashboard including the following:
  - Dropdown menu for launch site selection impacting each item below
  - Pie chart for total successful launches
  - Slider for payload range selection
  - Scatter plot comparing payload and launch success
- This dashboard showcases the total success launches from each site and correlation of payload mass with mission outcome at each launch site

Project available at https://github.com/TheCanadianShield/Applied-Data-Science-Capstone/blob/main/Week%203%20-%20Lab%20-%20Interactive%20spacex_dash_app.py

Predictive Analysis (Classification)

1. Load the data into NumPy from the Data Frame
2. Preprocessing of the data with the SkLearn StandardScaler to standardize data
3. Split the data into train and test sets, where test size was 20% of total data size
4. Use of 4 models to train on the data:
    1. Logistic Regression
    2. Support Vector Machine
    3. Decision Tree
    4. K-Nearest Neighbors
5. Models were run and compared for accuracy

Project available at https://github.com/TheCanadianShield/Applied-Data-Science-Capstone/blob/main/Week%204%20-%20Lab%20-%20SpaceX_Machine%20Learning%20Prediction_Part_5.ipynb

Results

Results will be shared in 3 sections:

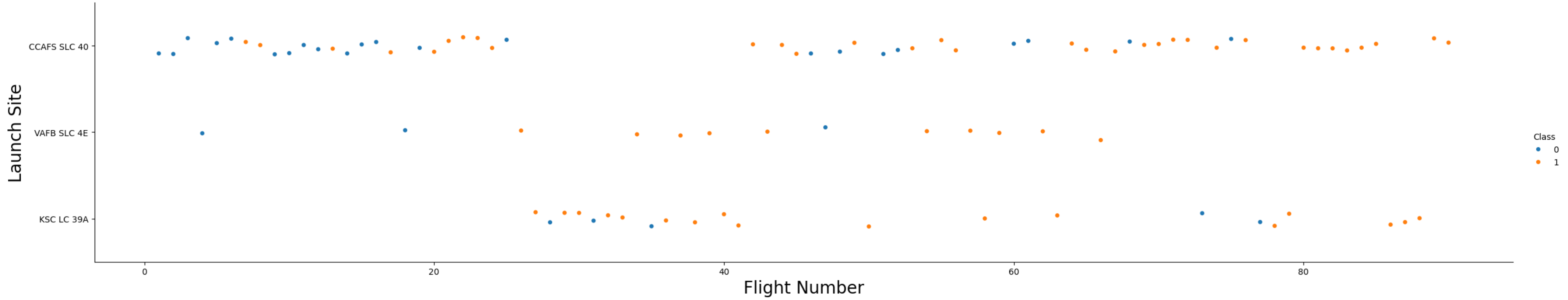
- Exploratory Data Analysis (EDA) results
- Interactive analytics demo in screenshots
- Predictive analysis results

Section 2

Insights drawn from EDA

Flight Number vs. Launch Site

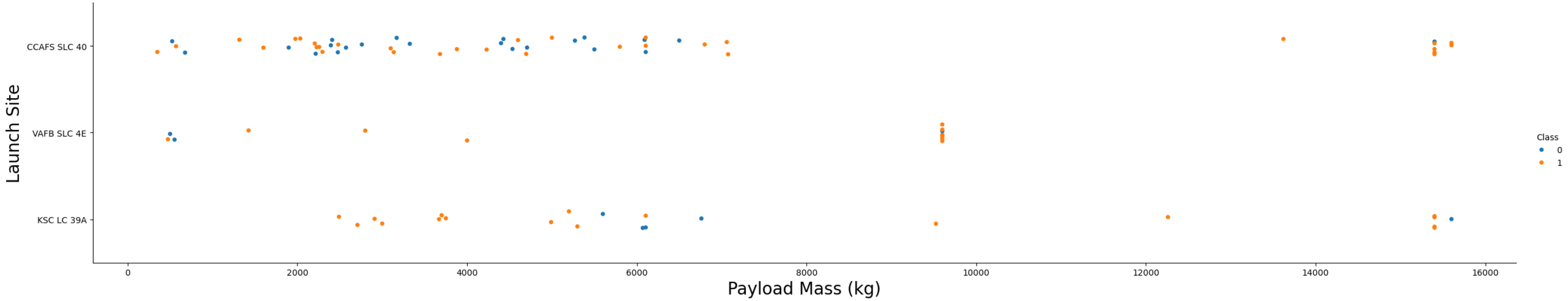
- The class represents a successful flight (1) or a failed flight (0)
- As the flight number increases, so does the success rate



Project available at https://github.com/TheCanadianShield/Applied-Data-Science-Capstone/blob/main/Week%202%20-%20Lab%20-%20EDA%20with%20Visualization%20Lab%20edadataviz.ipynb

Payload vs. Launch Site

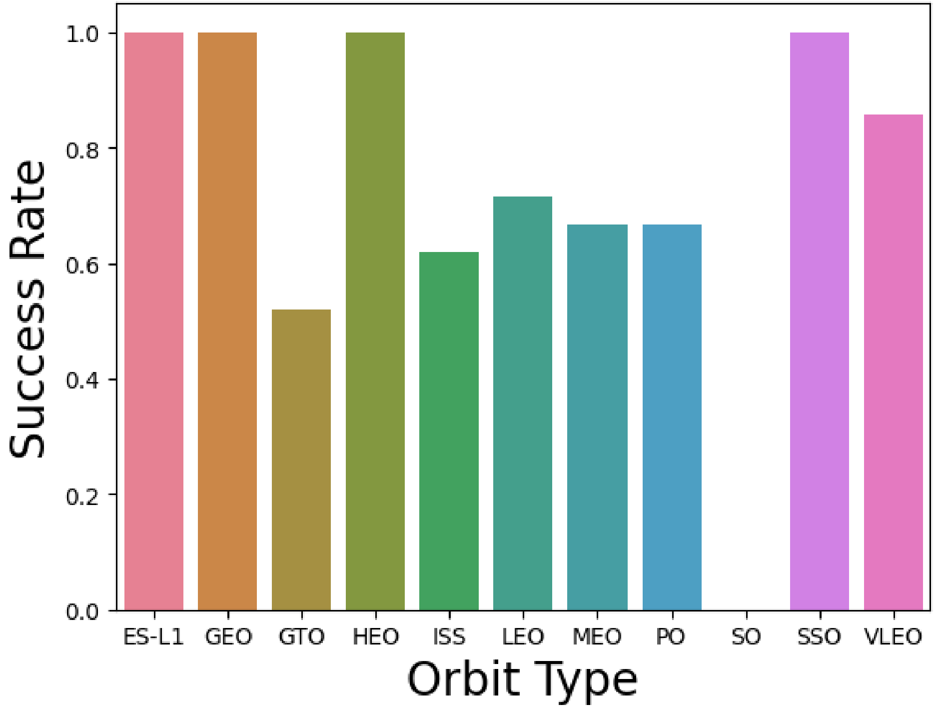
- VAFB SLC 4E has no launch above 10.000 kg of payload
- VAFB SLC 4E has a high rate of success, with very few failures (class = 0)
- Both CCAFS SLC 40 and VAFB SLC 4E have high success rates at higher payloads



Project available at https://github.com/TheCanadianShield/Applied-Data-Science-Capstone/blob/main/Week%202%20-%20Lab%20-%20EDA%20with%20Visualization%20Lab%20edadataviz.ipynb

Success Rate vs. Orbit Type

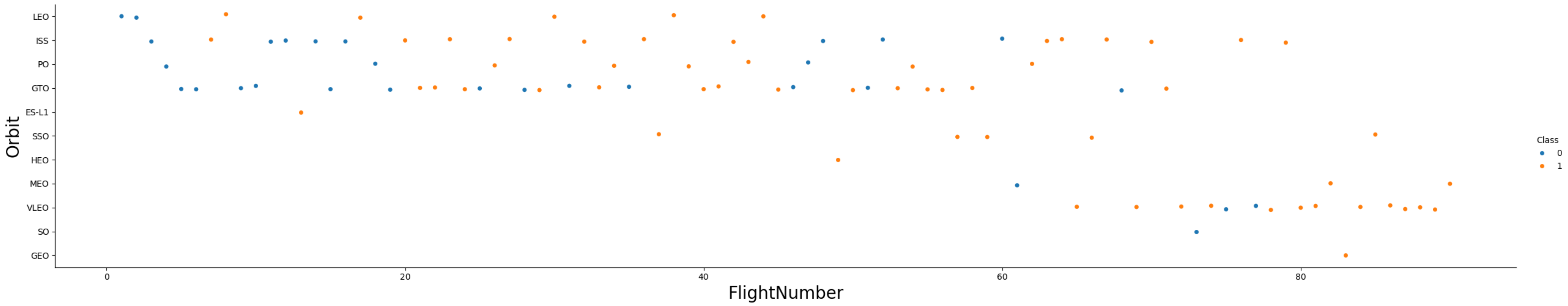
- ES-L1, GEO, HEP and SSO orbits have the best success rates
- GTO and ISS have the lowest success rates, with SO having no data



Project available at https://github.com/TheCanadianShield/Applied-Data-Science-Capstone/blob/main/Week%202%20-%20Lab%20-%20EDA%20with%20Visualization%20Lab%20edadataviz.ipynb

Flight Number vs. Orbit Type

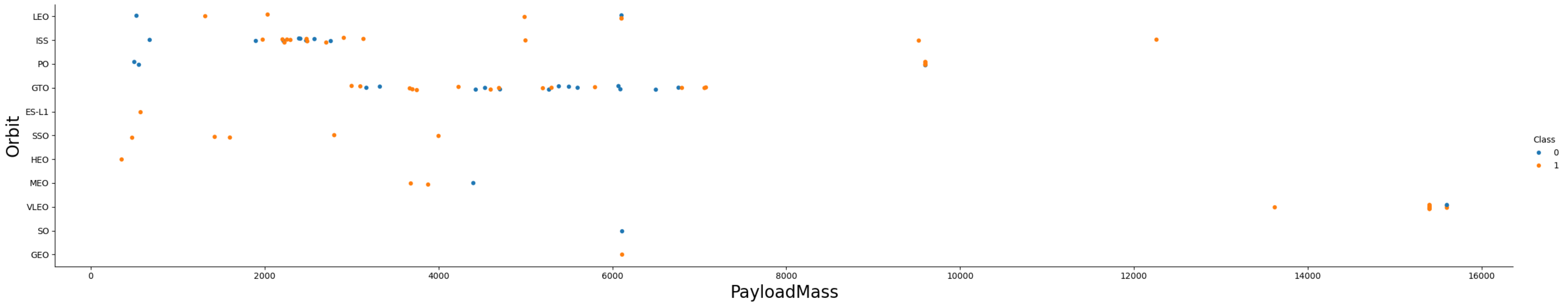
- SSO orbit contains only successful launches
- General trend where LEO, ISS, GTO and VLEO have better success at higher flight numbers



Project available at https://github.com/TheCanadianShield/Applied-Data-Science-Capstone/blob/main/Week%202%20-%20Lab%20-%20EDA%20with%20Visualization%20Lab%20edadataviz.ipynb

Payload vs. Orbit Type

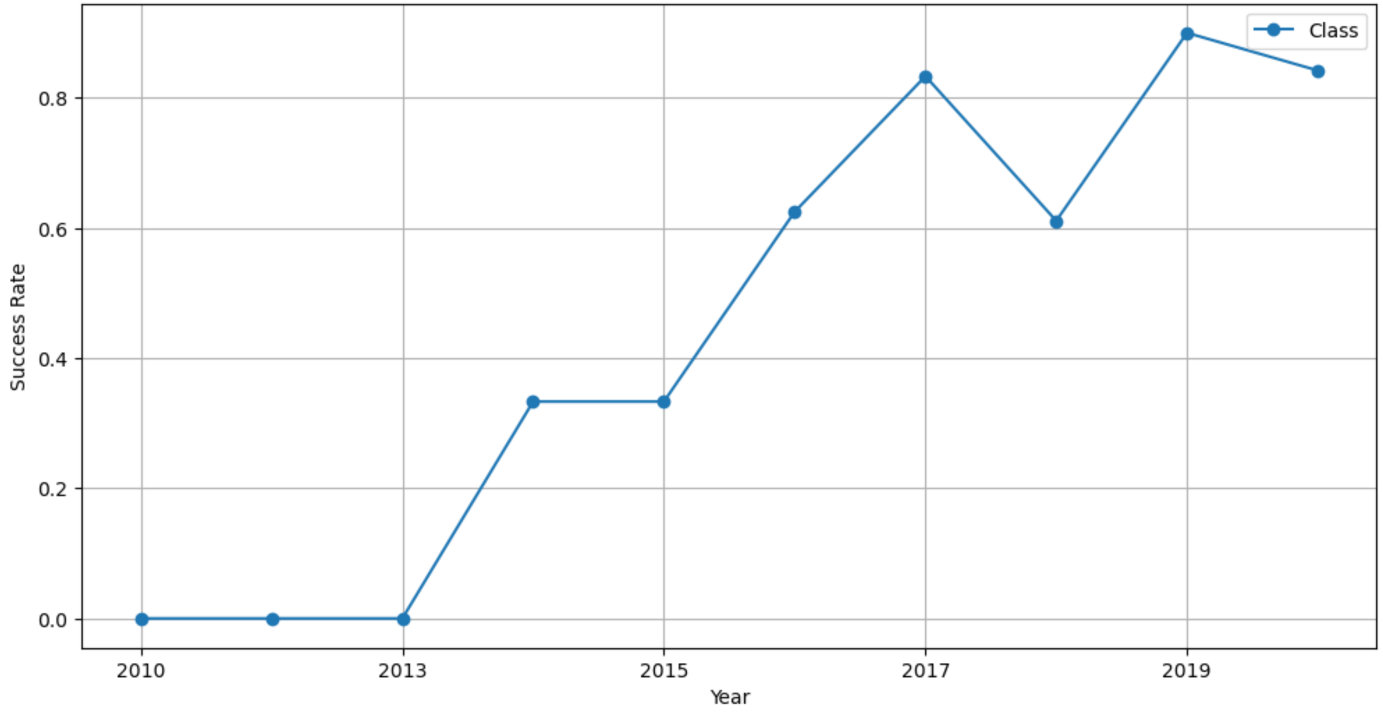
- General trend of higher payload mass having successful flights in all orbits
- Many orbits lack enough data for analysis



Project available at https://github.com/TheCanadianShield/Applied-Data-Science-Capstone/blob/main/Week%202%20-%20Lab%20-%20EDA%20with%20Visualization%20Lab%20edadataviz.ipynb

Launch Success Yearly Trend

- 2015-2017 saw the biggest improvements with a stagnation starting in 2019 in the mid 80 to 90% success rate.



Project available at https://github.com/TheCanadianShield/Applied-Data-Science-Capstone/blob/main/Week%202%20-%20Lab%20-%20EDA%20with%20Visualization%20Lab%20edadataviz.ipynb

All Launch Site Names

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

```
%sql SELECT DISTINCT LAUNCH_SITE FROM SPACEXTABLE;
```

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

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

- Find the names of the unique launch sites
- The function retrieves all unique values for Launch_Site

Project available at https://github.com/TheCanadianShield/Applied-Data-Science-Capstone/blob/main/Week%202%20-%20Lab%20-%20SQL%20Notebook%20Peer%20Assignment%20eda-sql-coursera_sqllite.ipynb

Launch Site Names Begin with 'CCA'

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

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

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

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

- Find 5 records where launch sites begin with `CCA`
- This query returns the launch sites that include the characters CCA (as demonstrated by the wildcard %), limiting to the first 5

Project available at https://github.com/TheCanadianShield/Applied-Data-Science-Capstone/blob/main/Week%202%20-%20Lab%20-%20SQL%20Notebook%20Peer%20Assignment%20eda-sql-coursera_sqllite.ipynb

Total Payload Mass

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

```
%sql SELECT SUM(PAYLOAD_MASS__KG_) FROM SPACEXTABLE WHERE Customer LIKE 'NASA (CRS)%';
```

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

SUM(PAYLOAD_MASS__KG_)
48213

- Calculate the total payload carried by boosters from NASA

- We are adding up the values for the payload mass for each mission where the customer's name included NASA (CRS)

Project available at https://github.com/TheCanadianShield/Applied-Data-Science-Capstone/blob/main/Week%202%20-%20Lab%20-%20SQL%20Notebook%20Peer%20Assignment%20eda-sql-coursera_sqllite.ipynb

Average Payload Mass by F9 v1.1

```
%sql SELECT AVG(PAYLOAD_MASS__KG_) FROM SPACEXTABLE WHERE Booster_Version LIKE "F9 v1.1%";
```

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

AVG(PAYLOAD_MASS__KG_)
2534.6666666666665

- Calculate the average payload mass carried by booster version F9 v1.1

- This command selects all flights done by the F9 v1.1 and averages its payload mass

Project available at https://github.com/TheCanadianShield/Applied-Data-Science-Capstone/blob/main/Week%202%20-%20Lab%20-%20SQL%20Notebook%20Peer%20Assignment%20eda-sql-coursera_sqllite.ipynb

First Successful Ground Landing Date

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

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

MIN(DATE)
2015-12-22

- Find the dates of the first successful landing outcome on ground pad

- The command selects all landing outcomes that are successful and landed on the ground pad. It then selected the earliest date from that list.

Project available at https://github.com/TheCanadianShield/Applied-Data-Science-Capstone/blob/main/Week%202%20-%20Lab%20-%20SQL%20Notebook%20Peer%20Assignment%20eda-sql-coursera_sqllite.ipynb

Successful Drone Ship Landing with Payload between 4000 and 6000

```
%%sql SELECT BOOSTER_VERSION, PAYLOAD_MASS__KG_, LANDING_OUTCOME FROM SPACEXTABLE
WHERE Landing_Outcome = 'Success (drone ship)'
    AND PAYLOAD_MASS__KG_ BETWEEN 4000 AND 6000
```

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

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

- List the names of boosters which have successfully landed on drone ship and had payload mass greater than 4000 but less than 6000
- We find the booster names with 3 conditions: landing outcome is ‘Success (drone ship)’, payload mass is greater than 4000kg and less than 6000kg.

Project available at https://github.com/TheCanadianShield/Applied-Data-Science-Capstone/blob/main/Week%202%20-%20Lab%20-%20SQL%20Notebook%20Peer%20Assignment%20eda-sql-coursera_sqllite.ipynb

Total Number of Successful and Failure Mission Outcomes

```
%%sql
SELECT MISSION_OUTCOME, COUNT(MISSION_OUTCOME) AS TOTAL_NUMBER
FROM SPACEXTABLE
GROUP BY MISSION_OUTCOME;
```

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

Mission_Outcome	TOTAL_NUMBER
Failure (in flight)	1
Success	98
Success	1
Success (payload status unclear)	1

- Calculate the total number of successful and failure mission outcomes

- We use the Group By function, which arranges repeating variables into a single group, connected with a count function.

Project available at https://github.com/TheCanadianShield/Applied-Data-Science-Capstone/blob/main/Week%202%20-%20Lab%20-%20SQL%20Notebook%20Peer%20Assignment%20eda-sql-coursera_sqllite.ipynb

Boosters Carried Maximum Payload

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

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

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

- List the names of the booster which have carried the maximum payload mass
- We use a subquery to select and sort the maximum payload mass. The table shows the unique booster versions with the highest payload calculated from the subquery

Project available at https://github.com/TheCanadianShield/Applied-Data-Science-Capstone/blob/main/Week%202%20-%20Lab%20-%20SQL%20Notebook%20Peer%20Assignment%20eda-sql-coursera_sqllite.ipynb

2015 Launch Records

```
%%sql SELECT substr(Date, 6,2) as Month, substr(Date,0,5) as Year, Landing_Outcome, Booster_Version,  Launch_Site
FROM SPACEXTABLE
WHERE Date Like '%2015%' and Landing_Outcome LIKE '%Failure (drone ship)%'
```

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

Month	Year	Landing_Outcome	Booster_Version	Launch_Site
01	2015	Failure (drone ship)	F9 v1.1 B1012	CCAFS LC-40
04	2015	Failure (drone ship)	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

- The output generated is the landing outcome, booster version and launch site, which is filtered by year (2015) and outcome ('Failure (drone ship)) with wild marks to cover all bases

Project available at https://github.com/TheCanadianShield/Applied-Data-Science-Capstone/blob/main/Week%202%20-%20Lab%20-%20SQL%20Notebook%20Peer%20Assignment%20eda-sql-coursera_sqllite.ipynb

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

```
%%sql SELECT Landing_Outcome, count(*) as Count_Landing_Outcome
FROM SPACEXTABLE
WHERE (Date BETWEEN  '2010-06-04' and '2017-03-20')
GROUP BY Landing_Outcome ORDER BY Count_Landing_Outcome DESC;
```

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

Landing_Outcome	Count_Landing_Outcome
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

- Rank the count of landing outcomes (such as Failure (drone ship) or Success (ground pad)) between the date 2010-06-04 and 2017-03-20, in descending order

- Group function organizes results by landing outcome in descending order, from a list of outcomes sorted by date

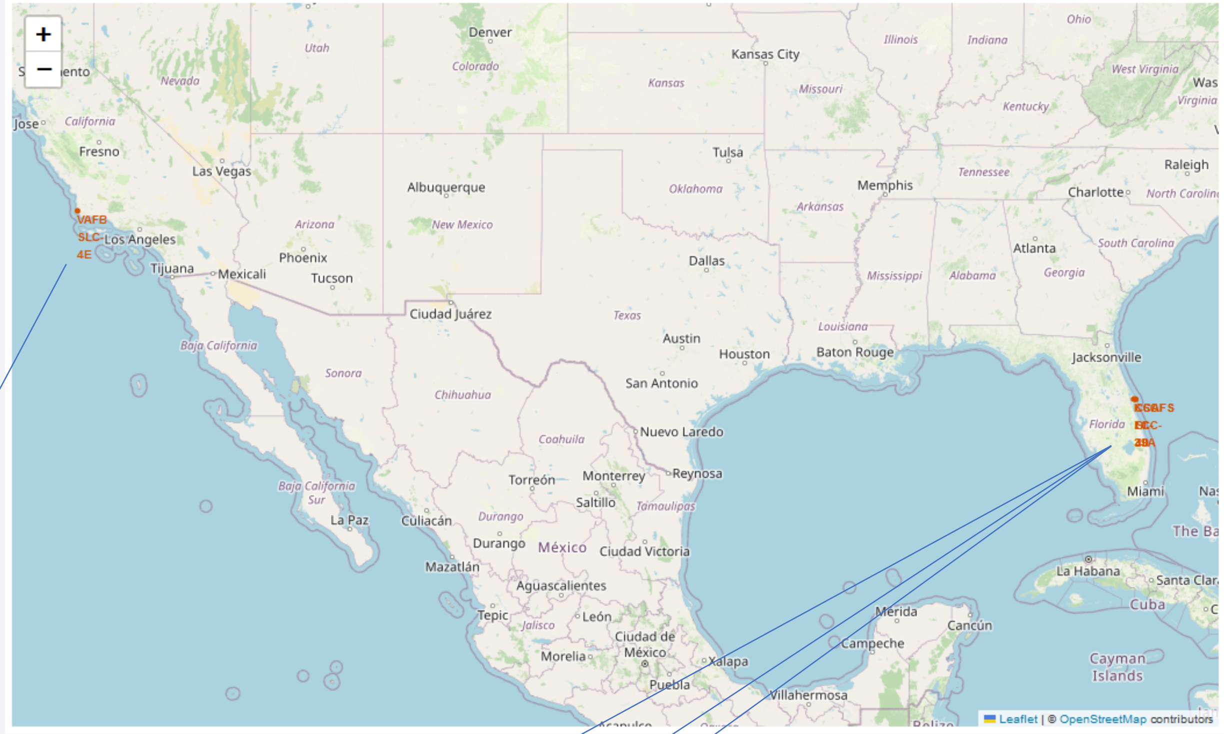
Project available at https://github.com/TheCanadianShield/Applied-Data-Science-Capstone/blob/main/Week%202%20-%20Lab%20-%20SQL%20Notebook%20Peer%20Assignment%20eda-sql-coursera_sqllite.ipynb

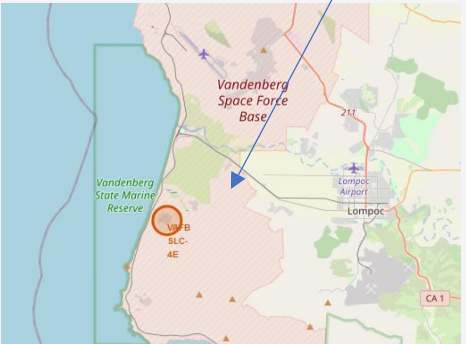
Section 3

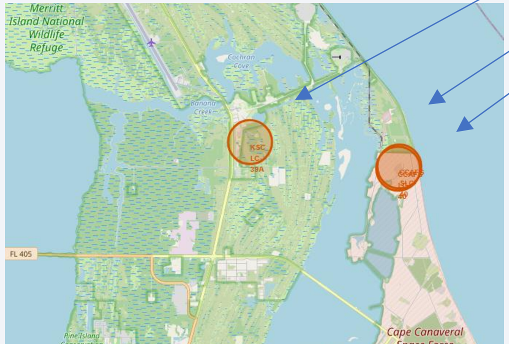
Launch Sites Proximities Analysis

SpaceX Falcon 9 Launch Sites

- The map highlights all launch sites in the United States for SpaceX using markers
- Florida hosts CCAFS SLC 40, CCAFS LC 40 and KSC LC 39A pads
- California hosts the VAFB SLC 4E pad







Project available at https://github.com/TheCanadianShield/Applied-Data-Science-Capstone/blob/main/Week%203%20-%20Lab%20-%20Interactive%20Visual%20Analytics%20with%20Folium%20lab%20lab_jupyter_launch_site_location.ipynb

Launch Site Success of Falcon 9 Mapped

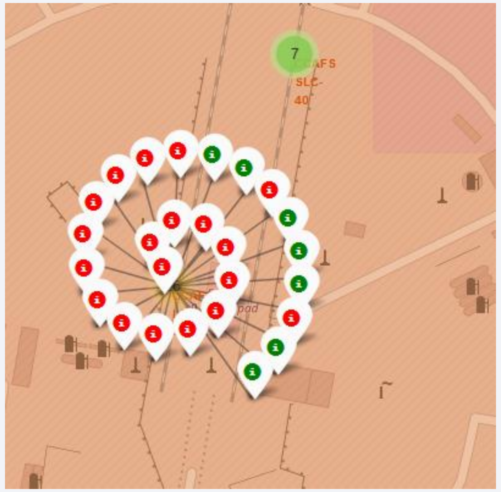


Figure 1: CCAFS LC-40

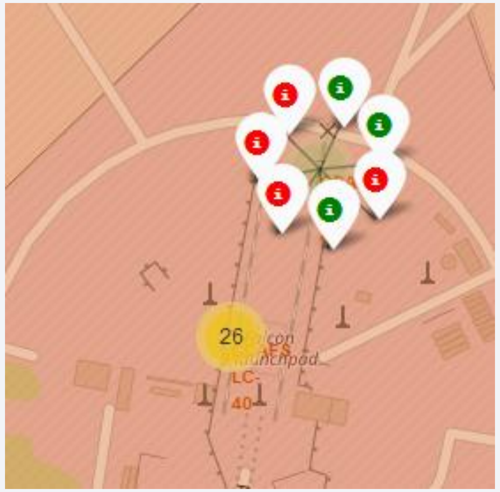


Figure 2: CCAFS SLC-40

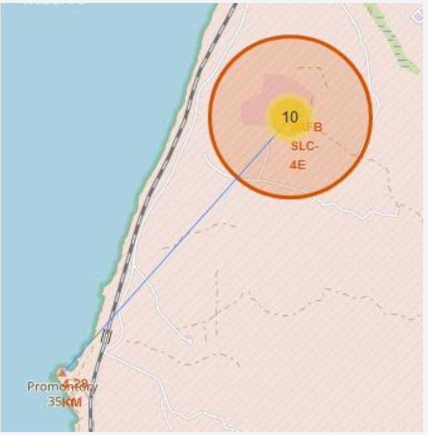
Figure 3: KSC LC-39A

Figure 4: VAFB SLC 4E

- The green markers represent a success, red represent a failure
- KSC LC-39A is the pad with the most successful launches
- CCAFS SL-40 has improved the most in successful launches

Project available at https://github.com/TheCanadianShield/Applied-Data-Science-Capstone/blob/main/Week%203%20-%20Lab%20-%20Interactive%20Visual%20Analytics%20with%20Folium%20lab%20lab_jupyter_launch_site_location.ipynb

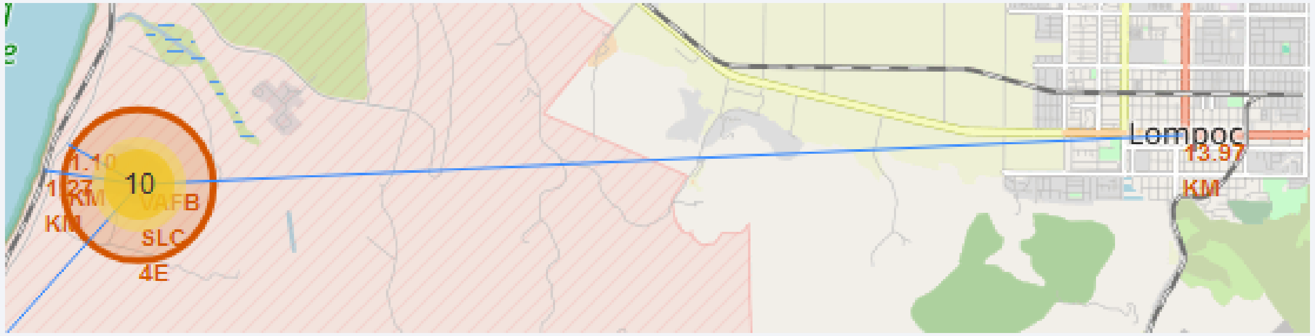
Proximities to SpaceX Launch Sites



VAFB SLC 4E

- 4.29km away from the Promontory
- 13.97km away from the city of Lompoc
- 1.10km from the nearest road
- 1.27 from the nearest railway

Launch pads must be a safe distance from cities for safety while being close to roads, railways for transport and coastlines for accessibility.

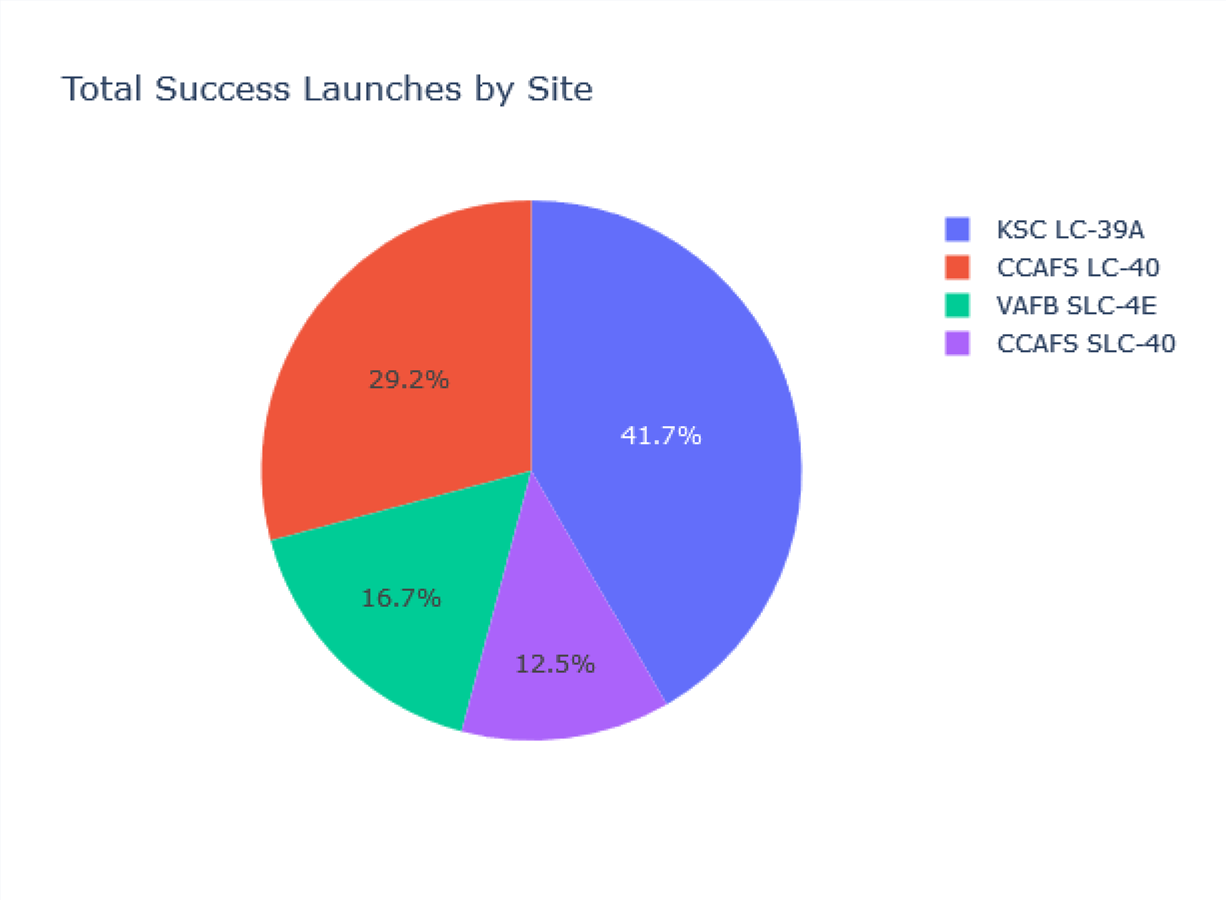


Project available at https://github.com/TheCanadianShield/Applied-Data-Science-Capstone/blob/main/Week%203%20-%20Lab%20-%20Interactive%20Visual%20Analytics%20with%20Folium%20lab%20lab_jupyter_launch_site_location.ipynb

Section 4

Build a Dashboard with Plotly Dash

Launch Success per Launch Pad



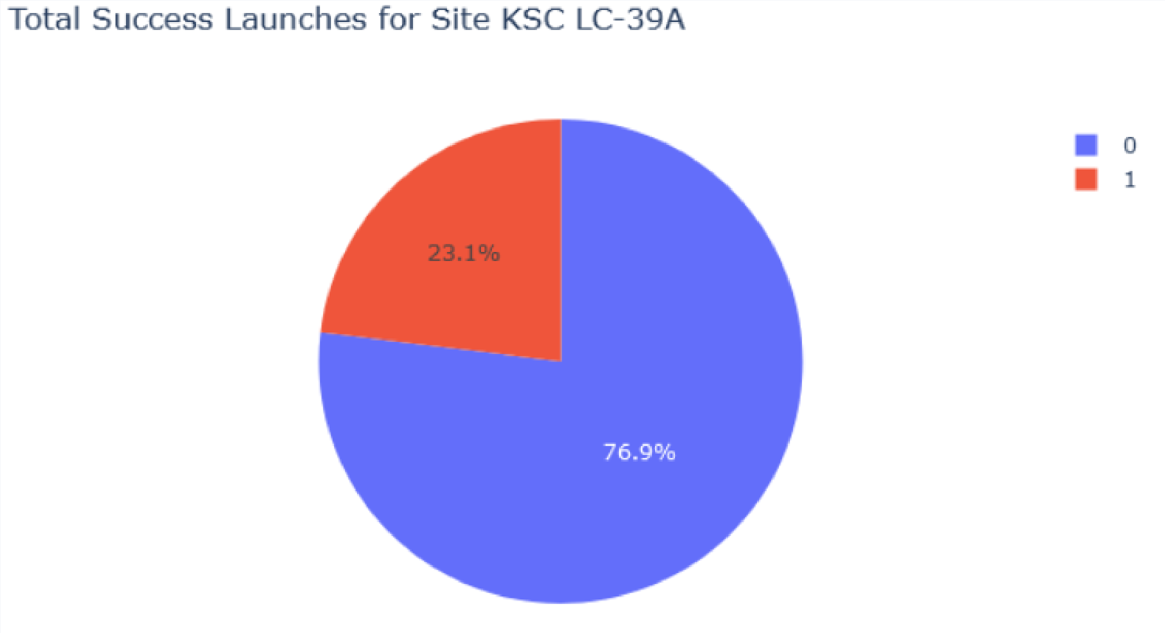
- KSC LC-39A holds the highest launch success rate
- CCAFS SLC-40 holds the lowest launch success rate

Project available at https://github.com/TheCanadianShield/Applied-Data-Science-Capstone/blob/main/Week%203%20-%20Lab%20-%20Interactive%20spacex_dash_app.py

Highest Success Ratio Launch Pad

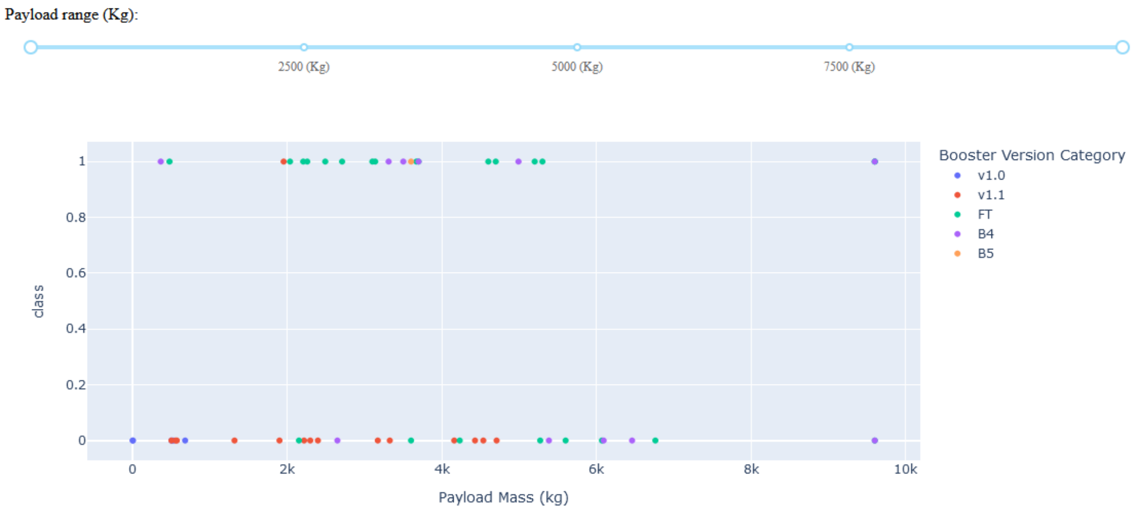
KSC LC-39A's success rate:

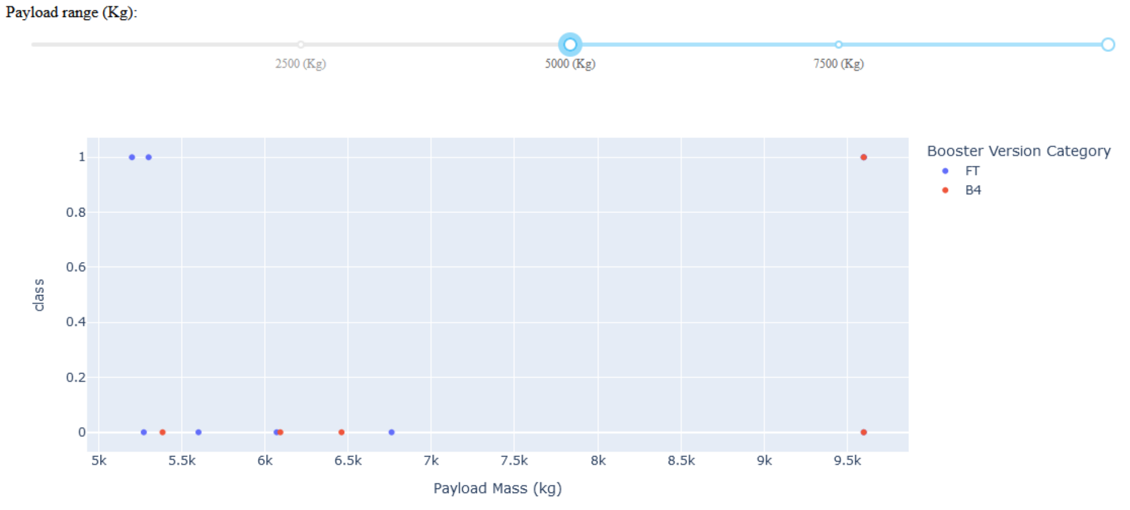
- 76.9% success
- 23.1% failure



Project available at https://github.com/TheCanadianShield/Applied-Data-Science-Capstone/blob/main/Week%203%20-%20Lab%20-%20Interactive%20spacex_dash_app.py

Comparison of Main Variables for Launch Outcome





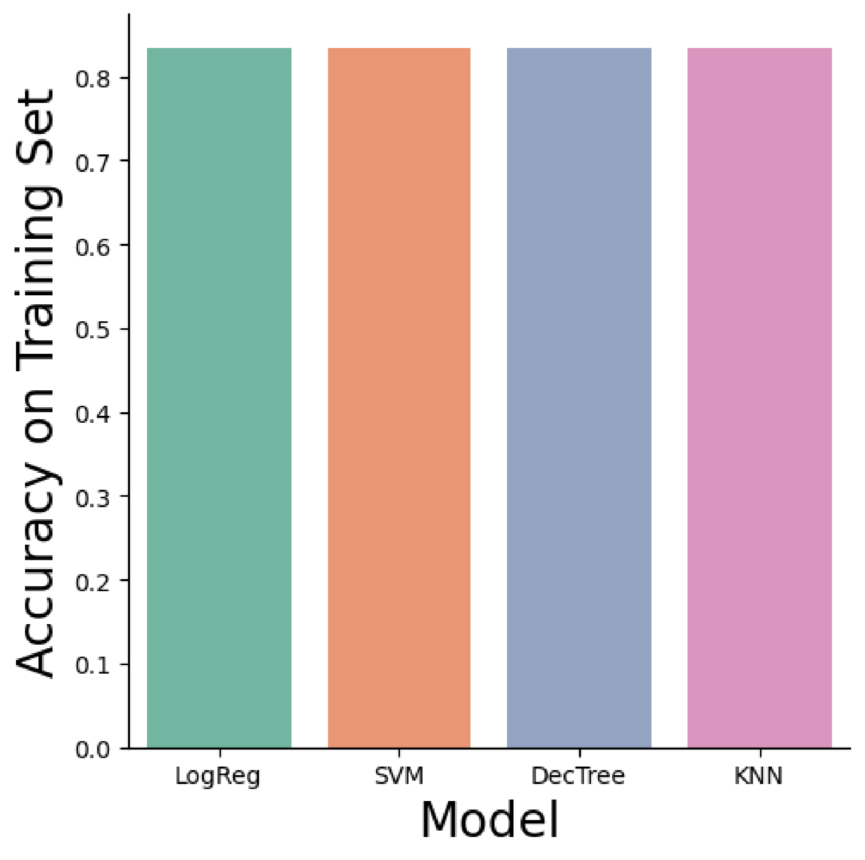
- The most successful payload launch is the 2.000kg to 5.000kg range.
- The FT booster variant is the most successful with booster v1.1 being the least successful
- Booster B4 launched the biggest payload successfully

Project available at https://github.com/TheCanadianShield/Applied-Data-Science-Capstone/blob/main/Week%203%20-%20Lab%20-%20Interactive%20spacex_dash_app.py

Section 5

Predictive Analysis (Classification)

Classification Accuracy

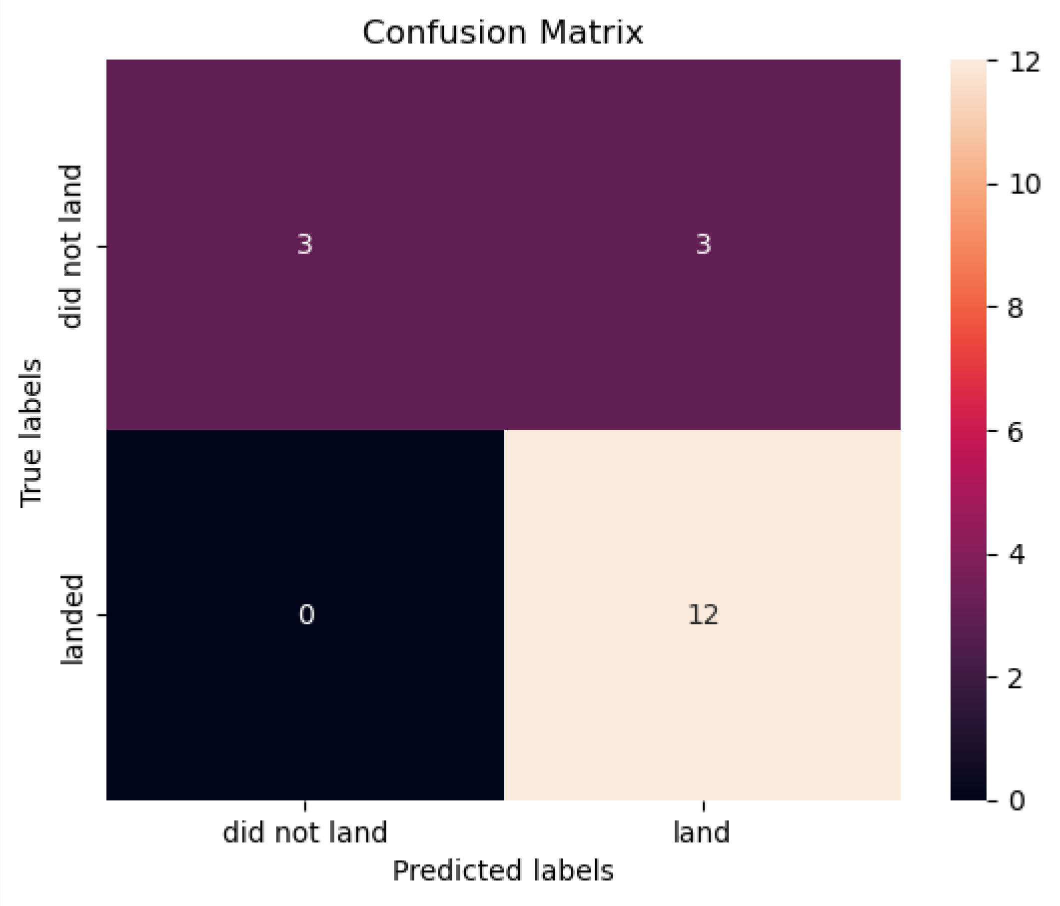



```
['logistic regression', 'support vector machine', 'decision tree classifier', 'k nearest neighbors']
[0.8333333333333334, 0.8333333333333334, 0.8333333333333334, 0.8333333333333334]
```

- Each model shows the same level of accuracy (83.334%)
- Results will have a small level of variance based on the random seed given to some of these models, such as the randomness of test vs train data split

Project available at https://github.com/TheCanadianShield/Applied-Data-Science-Capstone/blob/main/Week%204%20-%20Lab%20-%20SpaceX_Machine%20Learning%20Prediction_Part_5_with_visualization.ipynb

Confusion Matrix for K-Nearest Neighbors



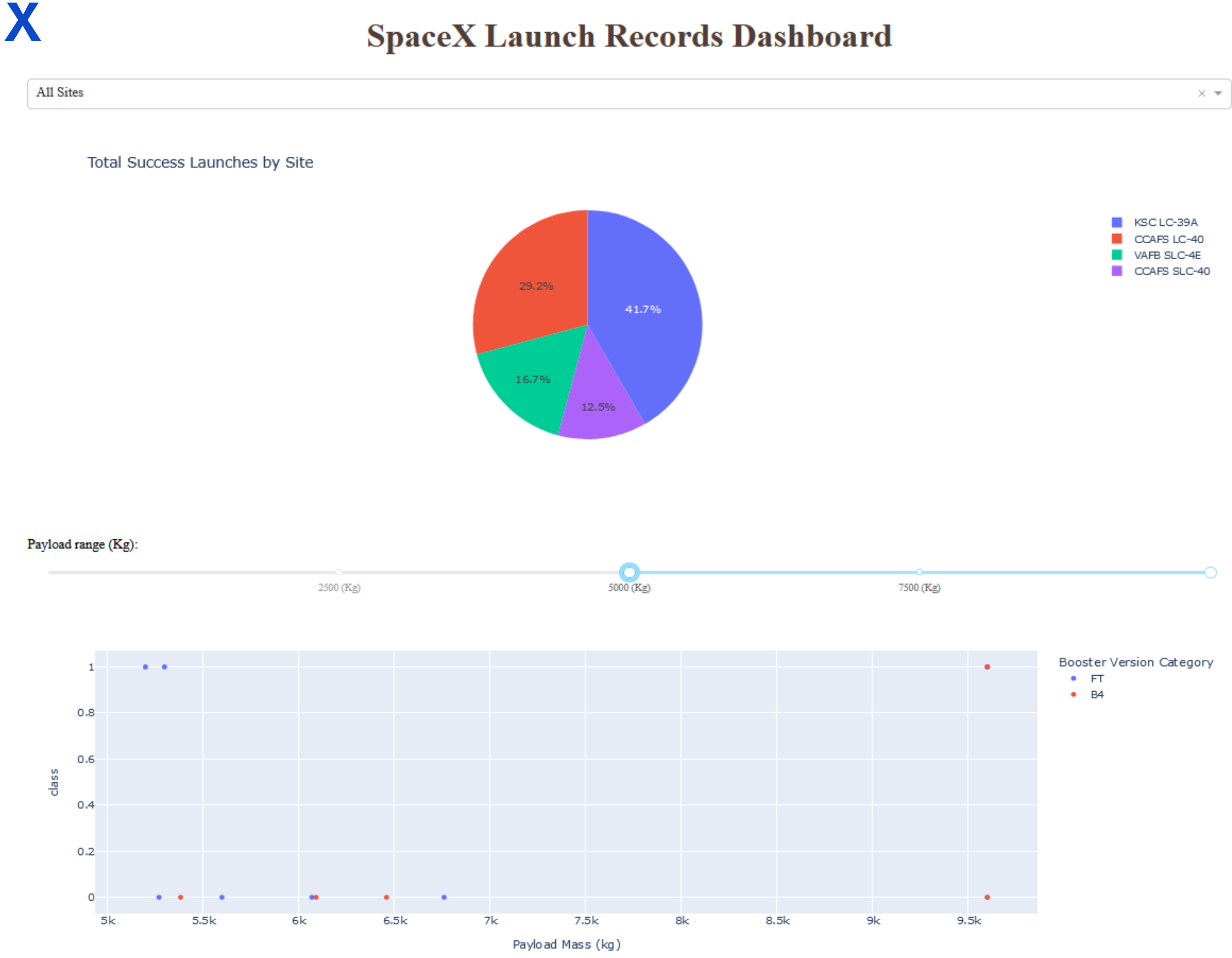
- This matrix highlights the distribution of results from the K-Nearest Neighbors with both the test and train data.
- The labels with 'did not land' have a 50% accuracy, as false positives

Project available at https://github.com/TheCanadianShield/Applied-Data-Science-Capstone/blob/main/Week%204%20-%20Lab%20-%20SpaceX_Machine%20Learning%20Prediction_Part_5_with_visualization.ipynb

Conclusions

- SpaceX launch success rate continues to climb, now above 80%
- Higher payload mass launches have higher success rates
- SSO, HEO, GEO and ES_L1 orbits obtain better rates of success than other orbits
- Launch pad KSC LC 39A is the pad with the most successful launches
- Booster version FT is the top booster for reliability
- Each machine learning model confirmed an accuracy of prediction at 83.34%
- A combination of the above would make for a very safe launch and high likelihood of success. Likewise, the use of other launch pads, boosters, orbits and payload mass would result in lower chances of success.

Appendix



The interactive dashboard built using Dash in python

Project available at https://github.com/TheCanadianShield/Applied-Data-Science-Capstone/blob/main/Week%203%20-%20Lab%20-%20Interactive%20spacex_dash_app.py

Appendix

```
# Get the head of the dataframe
data.head()
```

	static_fire_date_utc	static_fire_date_unix	net	window	rocket	success	failures	details	crew	ships	capsules	payloads	launchpad	flight_number	name	date_utc	date_unix	
0	2006-03-17T00:00:00.000Z	1.142554e+09	False	0.0	5e9d0d95eda69955f709d1eb	False	[{'time': 33, 'altitude': None, 'reason': 'merlin engine failure'}]	Engine failure at 33 seconds and loss of vehicle	[]	[]	[]	[5eb0e4b5b6c3bb0006eeb1e1]	5e9e4502f5090995de566f86	1	FalconSat	2006-03-24T22:30:00.000Z	1143239400	2(
1	None	NaN	False	0.0	5e9d0d95eda69955f709d1eb	False	[{'time': 301, 'altitude': 289, 'reason': 'harmonic oscillation leading to premature engine shutdown'}]	Successful first stage burn and transition to second stage, maximum altitude 289 km, Premature engine shutdown at T+7 min 30 s, Failed to reach orbit, Failed to recover first stage	[]	[]	[]	[5eb0e4b6b6c3bb0006eeb1e2]	5e9e4502f5090995de566f86	2	DemoSat	2007-03-21T01:10:00.000Z	1174439400	2(
							[{'time':											

Original data parsed from SpaceX API JSON into a Pandas dataframe

Project available at https://github.com/TheCanadianShield/Applied-Data-Science-Capstone/blob/main/Week%201%20-%20Lab%20-%20Data%20Collection%20API%20jupyter-labs-spacex-data-collection-api.ipynb

Appendix

```
from js import fetch
import io

URL1 = "https://cf-courses-data.s3.us.cloud-object-storage.appdomain.cloud/IBM-DS0321EN-SkillsNetwork/datasets/dataset_part_2.csv"
resp1 = await fetch(URL1)
text1 = io.BytesIO((await resp1.arrayBuffer()).to_py())
data = pd.read_csv(text1)
```

```
data.head()
```

	FlightNumber	Date	BoosterVersion	PayloadMass	Orbit	LaunchSite	Outcome	Flights	GridFins	Reused	Legs	LandingPad	Block	ReusedCount	Serial	Longitude	Latitude	Class
0	1	2010-06-04	Falcon 9	6104.959412	LEO	CCAFS SLC 40	None None	1	False	False	False	NaN	1.0	0	B0003	-80.577366	28.561857	0
1	2	2012-05-22	Falcon 9	525.000000	LEO	CCAFS SLC 40	None None	1	False	False	False	NaN	1.0	0	B0005	-80.577366	28.561857	0
2	3	2013-03-01	Falcon 9	677.000000	ISS	CCAFS SLC 40	None None	1	False	False	False	NaN	1.0	0	B0007	-80.577366	28.561857	0
3	4	2013-09-29	Falcon 9	500.000000	PO	VAFB SLC 4E	False Ocean	1	False	False	False	NaN	1.0	0	B1003	-120.610829	34.632093	0
4	5	2013-12-03	Falcon 9	3170.000000	GTO	CCAFS SLC 40	None None	1	False	False	False	NaN	1.0	0	B1004	-80.577366	28.561857	0

A cleaned dataset ready for the machine learning portion of the project

Project available at https://github.com/TheCanadianShield/Applied-Data-Science-Capstone/blob/main/Week%204%20-%20Lab%20-%20SpaceX_Machine%20Learning%20Prediction_Part_5_with_visualization.ipynb

Appendix

TASK 4

Create a logistic regression object then create a GridSearchCV object `logreg_cv` with cv = 10. Fit the object to find the best parameters from the dictionary `parameters`.

```
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()
logreg_cv = GridSearchCV(lr, parameters, cv = 10 )
logreg_cv = logreg_cv.fit(X_train, Y_train)
```

We output the `GridSearchCV` object for logistic regression. We display the best parameters using the data attribute `best_params_` and the accuracy on the validation data using the data attribute `best_score_`.

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

```
tuned hpyerparameters :(best parameters)  {'C': 0.01, 'penalty': 'l2', 'solver': 'lbfgs'}
accuracy : 0.8464285714285713
```

Building a logistic regression model. Other models used include:

- Support Vector Machine
- Decision Tree Classifier
- K-Nearest Neighbors

Project available at https://github.com/TheCanadianShield/Applied-Data-Science-Capstone/blob/main/Week%204%20-%20Lab%20-%20SpaceX_Machine%20Learning%20Prediction_Part_5_with_visualization.ipynb

Appendix

TASK 5

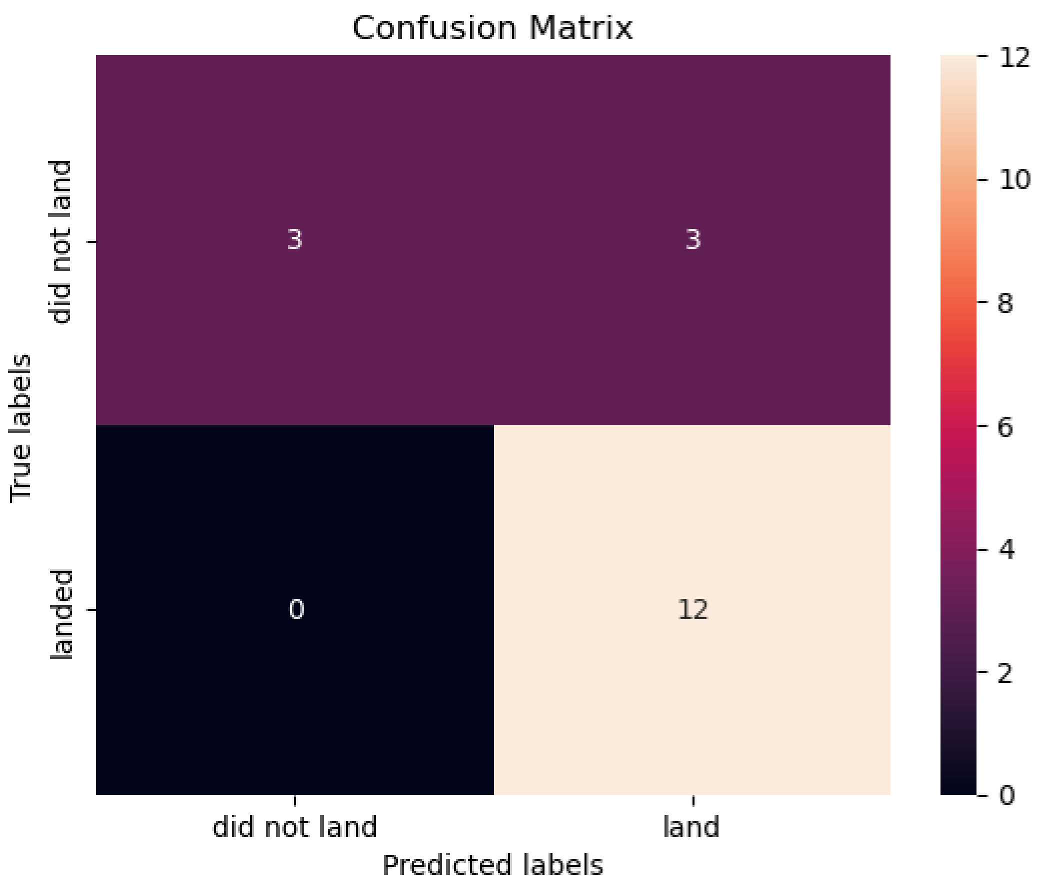
Calculate the accuracy on the test data using the method `score` :

```
accuracy=[]
methods=[]
accuracy.append(logreg_cv.score(X_test,Y_test))
methods.append('logistic regression')
logreg_cv.score(X_test,Y_test)
```

```
0.8333333333333334
```

Lets look at the confusion matrix:

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



Testing the logistic regression model with a confusion matrix

Project available at https://github.com/TheCanadianShield/Applied-Data-Science-Capstone/blob/main/Week%204%20-%20Lab%20-%20SpaceX_Machine%20Learning%20Prediction_Part_5_with_visualization.ipynb

Thank you!