IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

<Vincent L>
<12/18/2023>

Outline

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

Executive Summary

- Summary of methodologies
  - Data collection API + webscraping
  - Data Wrangling
  - Exploratory Data Analysis with SQL
  - Exploratory Data Analysis and Data Visualization
  - Interactive Visual Analytics with Folium
  - Machine Learning Predictive Modeling
- Summary of all results
  - Exploratory Data Analysis results
  - Dashboard screenshots
  - Predictive Model Results

Introduction

- Project background and context
  - SpaceX advertises Falcon 9 rocket launches on its website with a cost of 62 million dollars; other providers cost upwards of 165 million dollars each, much of the savings is because SpaceX can reuse the first stage. Therefore, if we can determine if the first stage will land, we can determine the cost of a launch.
- Problems you want to find answers
  - We want to determine the price of each launch
  - Determine whether SpaceX will reuse the first stage of the launch
  - What factors determine if a rocket will land successfully in the launch and affect the success rate

Section 1

Methodology

Methodology

Executive Summary

- Data collection methodology:
  - SpaceX API and webscraping
- Perform data wrangling
  - Used SQL and Python to create dataset,
- 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

- Describe how data sets were collected.
  - Data was collected using SpaceX API and webscraping of Wikipedia using BeautifulSoup
  - Converted to editable pandas dataframe using .json_normalize()
  - Removed missing / unneeded values
  - Extracted launch records and converted to pandas dataframe for continued analysis and wrangling

Data Collection – SpaceX API

- SpaceX API get request to collect data, and used json_normalize() to clean, wrangle and format pandas dataframe

- Notebook: https://github.com/vincentmliang/IBM-DS-AppliedCapstone/blob/main/1_jupyter-labs-spacex-data-collection-api.ipynb

To make the requested JSON results more consistent, we will use the following static response object for this project:

```
static_json_url='https://cf-courses-data.s3.us.cloud-object-storage.appdomain.cloud/IBM-DS0321EN-SkillsNetw
```

We should see that the request was successfull with the 200 status response code

```
response.status_code
```

```
200
```

Now we decode the response content as a Json using `.json()` and turn it into a Pandas dataframe using `.json_normalize()`

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

Using the dataframe `data` print the first 5 rows

```
# Get the head of the dataframe
data.head(5)
```

Data Collection - Scraping

- Requested Falcon9 Launch Wiki using BeautifulSoup package and created editable pandas dataframe.

- Notebook: https://github.com/vincentmliang/IBM-DS-AppliedCapstone/blob/main/2_jupyter-labs-webscraping.ipynb

First, let's perform an HTTP GET method to request the Falcon9 Launch HTML page, as an HTTP response.

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

```
200
```

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

```
# Use BeautifulSoup() to create a BeautifulSoup object from a response text content

soup = BeautifulSoup(html_data.text, 'html.parser')
```

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

```
# Use soup.title attribute
soup.title
```

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

TASK 2: Extract all column/variable names from the HTML table header

Next, we want to collect all relevant column names from the HTML table header

Let's try to find all tables on the wiki page first. If you need to refresh your memory about `BeautifulSoup`, please check the external reference link towards the end of this lab

```
# 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')
```

Starting from the third table is our target table contains the actual launch records.

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

Data Wrangling

- Calculated exploratory data analysis in Python by calculating the number of launches on each site.
- Calculated the number and occurrence of each orbit
- Calculated the number and occurrences of mission outcomes of the orbits
- Created a landing outcome label using an if-else statement, labeling it landing_class
- Exported new dataframe to CSV file
- Notebook: https://github.com/vincentmliang/IBM-DS-AppliedCapstone/blob/main/3_labs-jupyter-spacex-Data%20wrangling(1).ipynb

EDA with Data Visualization

- Visualized relationship between flight number and launch site, payload and launch site, success rate of each orbit type, flight number and orbit type and the launch success in a yearly trend.
- Notebook: https://github.com/vincentmliang/IBM-DS-AppliedCapstone/blob/main/5_jupyter-labs-eda-dataviz.ipynb.jupyterlite.ipynb

EDA with SQL

- Used SQL queries to do exploratory data analysis and find insights on the following:
  - Names of unique launch sites
  - Total payload mass in kg carried by boosters launched by NASA
  - Average payload mass carried by booster version F9 V1.1
  - Total successful and failure mission outcomes
  - Failure landing outcomes in drone_ship, booster versions, and launch_site for months in year 2015.
- Notebook: https://github.com/vincentmliang/IBM-DS-AppliedCapstone/blob/main/4_jupyter-labs-eda-sql-coursera_sqllite.ipynb

Build an Interactive Map with Folium

- Using Folium, we created an interactive map marking launch sites, success or failure of launches
- Using color coordinated marker clusters, you can see launch sites that have higher success rates
- Using this map, you can determine where launch sites are easily and how geographical location can effect the outcome of a launch
- Notebook: https://github.com/vincentmliang/IBM-DS-AppliedCapstone/blob/main/6_lab_jupyter_launch_site_location.jupyterlite.ipynb

Build a Dashboard with Plotly Dash

- Interactive dashboard using Plotly Dash, plotting pie charts showing the total launches per site, along with scatterplot showing the relationships between outcome vs payload mass in kilograms.
- This plots show relationships that may affect the outcome of certain launches.
- Notebook: https://github.com/vincentmliang/IBM-DS-AppliedCapstone/blob/main/7_DashboardwPlotly

Predictive Analysis (Classification)

- Transformed and split train and test data using train_test_split
- Coded different amchine learning models and performed hyperparameter tuning using GridSearchCV
- Used accuracy to metric – found best accuracy to be Decision Tree classifier
- Notebook: https://github.com/vincentmliang/IBM-DS-AppliedCapstone/blob/main/SpaceX_Machine_Learning_Prediction_Part_5.jupyterlite(1).ipynb

Results

- Exploratory data analysis results
- Interactive analytics demo in screenshots
- Predictive analysis results

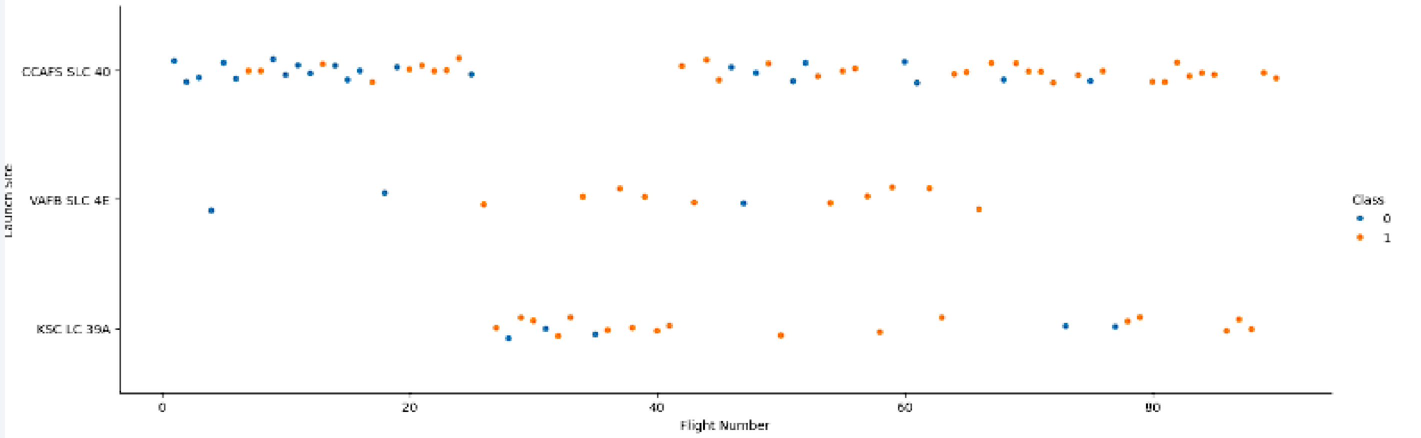
Section 2

Insights drawn from EDA

Flight Number vs. Launch Site

- Flight number vs. launch site. Shows that typically, the more the launch site has more flights, the more likely / successful their launches become.
- Over time, as flight numbers increase, successes (class 1) becomes more frequent

```
# Plot a scatter point chart with x axis to be Flight Number and y axis to be the launch site, and hue to b
sns.catplot(x = "FlightNumber", y = "LaunchSite", hue = "Class", data = df, aspect = 3)
plt.xlabel("Flight Number")
plt.ylabel("Launch Site")
plt.show()
```

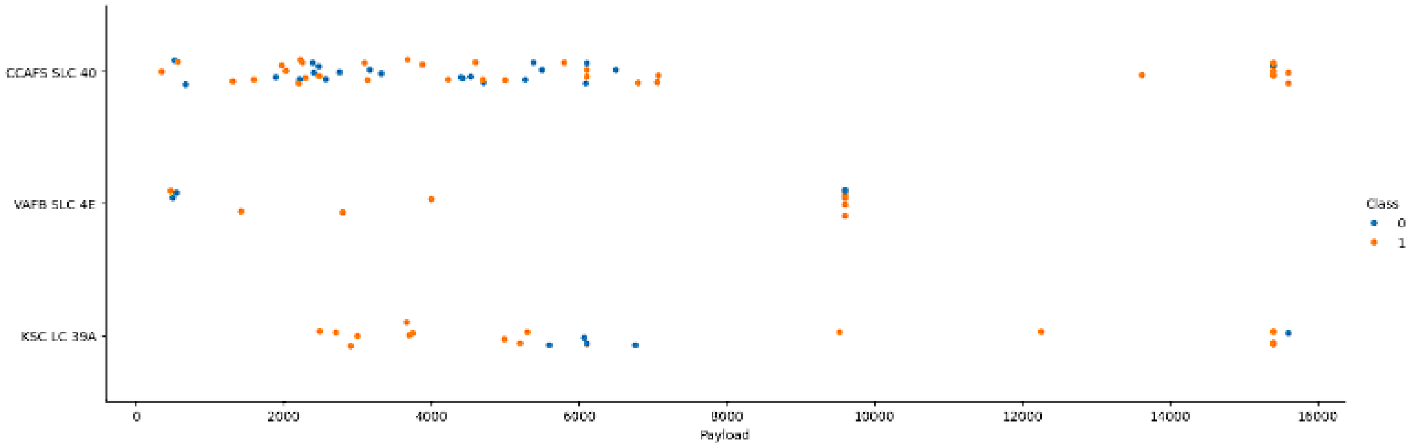


Payload vs. Launch Site

- Scatterplot of payload in kilgorams vs. launch site.
- Typically for launchsite CCAFS SLC 40, higher payloads at 16,000 kg had a high success rate, and mixed results from the 0-8000 kg range

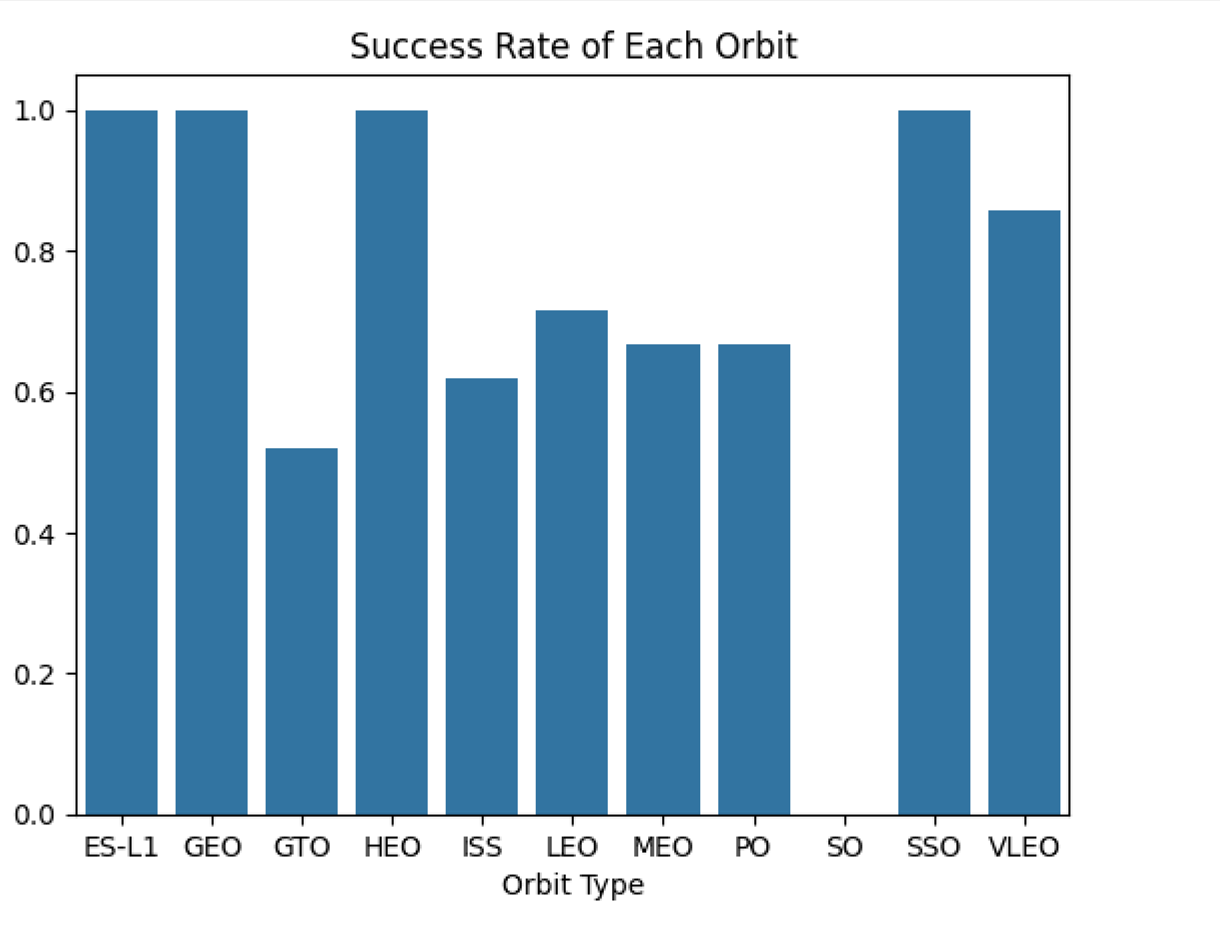
```
# Plot a scatter point chart with x axis to be Pay Load Mass (kg) and y axis to be the launch site, and hue

sns.catplot(x = "PayloadMass", y = "LaunchSite", hue = "Class", data = df, aspect = 3)
plt.xlabel("Payload")
plt.ylabel("Launch Site")
plt.show()
```



Success Rate vs. Orbit Type

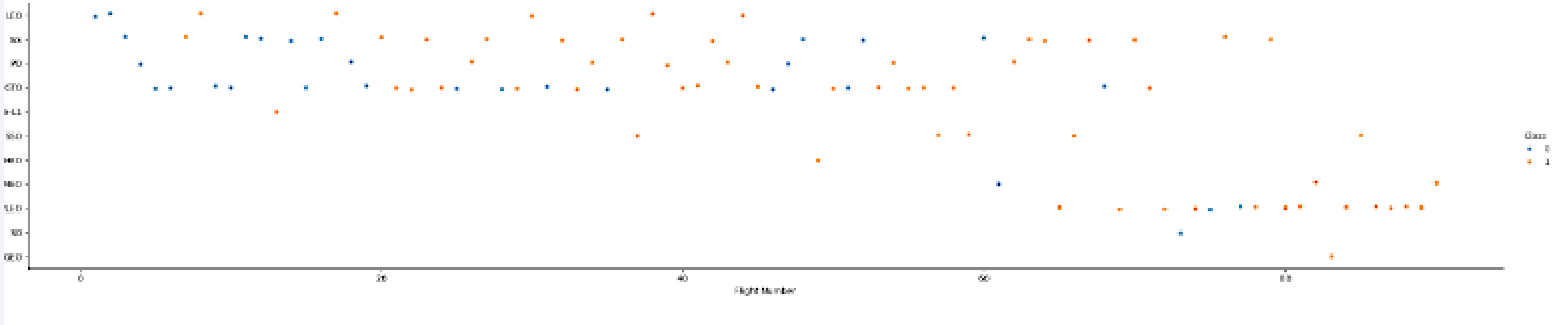
- Bar chart of success rates vs orbit type.
- Orbit type ES-L1, GEO, HEO, and SSO had the best success rates
- SO had no successful launches



Flight Number vs. Orbit Type

- Scatterplot shows flight number vs orbit type. Shows over time how each launch site goes through their successes and failures. For example, had failures in the early flight numbers, but as number of flights got higher, they became more successful

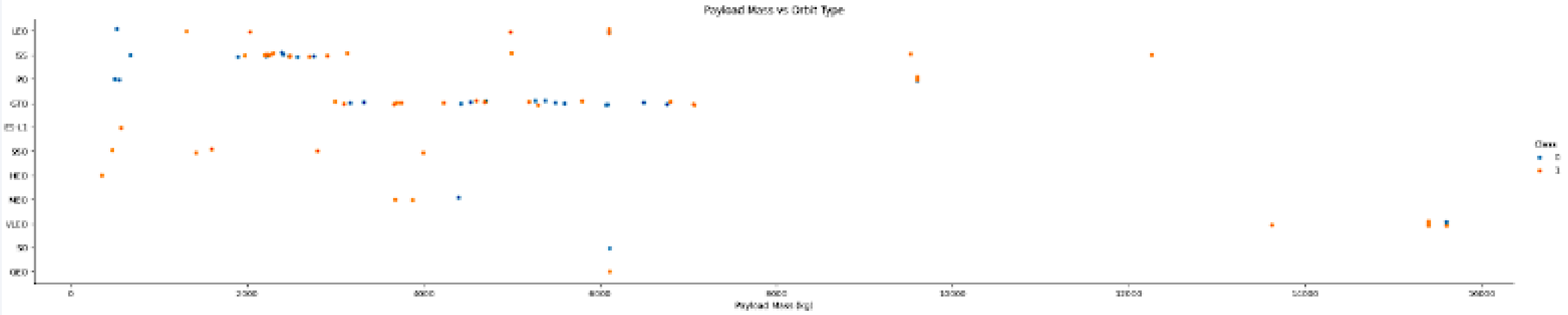
```
sns.catplot(x = "FlightNumber", y = "Orbit", hue = "Class", data = df, aspect = 5)
plt.xlabel("Flight Number")
plt.ylabel("Orbit")
plt.show()
```



Payload vs. Orbit Type

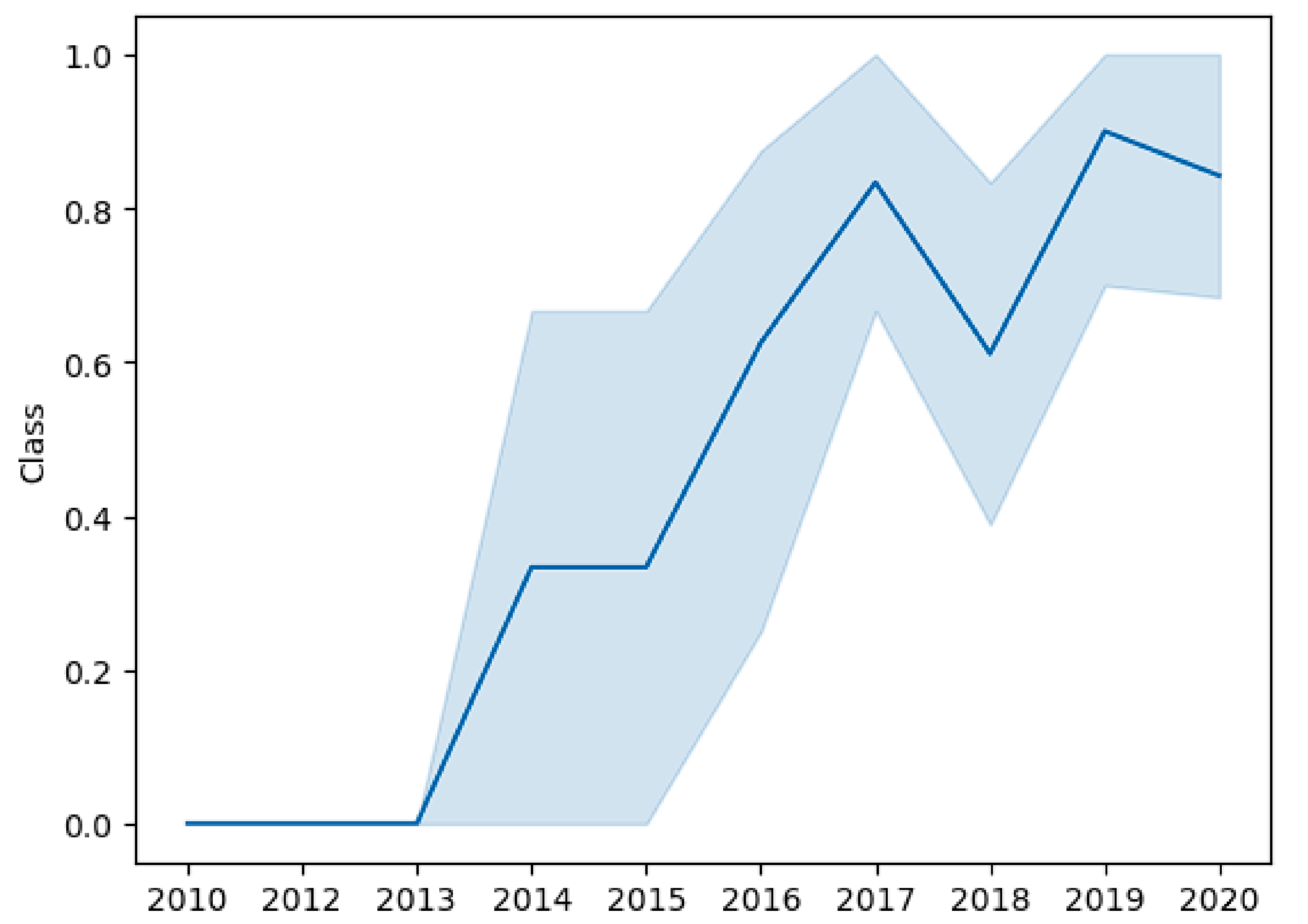
- Scatterplot of payload (kg) vs orbit type. Showcases that the heavier payloads have higher success rates, in particular for LEO, ISS and PO orbits.

```
# Plot a scatter point chart with x axis to be Payload and y axis to be the Orbit, and hue to be the class
sns.catplot(x = "PayloadMass", y = "Orbit", hue = "Class", data = df, aspect = 5)
plt.xlabel("Payload Mass (kg)")
plt.ylabel("Orbit")
plt.title("Payload Mass vs Orbit Type")
plt.show()
```



Launch Success Yearly Trend

- Line chart showing the launch success yearly trend
- Success rate has steadily increased over time, with a slight dip in 2017 – 2018



All Launch Site Names

- SQL query with all launch site names. Used distinct to show only unique names from the SpaceX data
- Launch site names are as follows:
  - CCAFS LC-40
  - VAFB SLC-4E
  - KSC LC-39A
  - CCAFS SLC-40

```
%sql SELECT Distinct LAUNCH_SITE FROM SPACEXTABLE
```

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

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

Launch Site Names Begin with 'CCA'

- Find 5 records where launch sites begin with `CCA`
- Used LIMIT 5 to only show the first five records where launch sites begin with 'CCA'

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

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

Date	Time (UTC)	Booster_Version	Launch_Site	Payload	PAYLOAD_MASS__KG_	Orbit	Customer	Mission_Outcome
2010-06-04	18:45:00	F9 v1.0 B0003	CCAFS LC-40	Dragon Spacecraft Qualification Unit	0	LEO	SpaceX	Success
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
2012-05-22	7:44:00	F9 v1.0 B0005	CCAFS LC-40	Dragon demo flight C2	525	LEO (ISS)	NASA (COTS)	Success
2012-10-08	0:35:00	F9 v1.0 B0006	CCAFS LC-40	SpaceX CRS-1	500	LEO (ISS)	NASA (CRS)	Success
2013-03-01	15:10:00	F9 v1.0 B0007	CCAFS LC-40	SpaceX CRS-2	677	LEO (ISS)	NASA (CRS)	Success

Total Payload Mass

- Calculate the total payload carried by boosters from NASA
- Selected the total sum of payload mass from the SpaceX Table where the customer is equal to ‘NASA (CRS)’. Total sum of all the payloads carried by boosters launched by NASA was 45,596 kg.

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

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

SUM(PAYLOAD_MASS__KG_)
45596

Average Payload Mass by F9 v1.1

- Calculate the average payload mass carried by booster version F9 v1.1
- Selected the average payload mass of booster version F9 v1.1 only. Average payload mass was 2928.4

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

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

AVG(PAYLOAD_MASS__KG_)
2928.4

First Successful Ground Landing Date

- Find the dates of the first successful landing outcome on ground pad
- Selected the first date using MIN(date) to determine first successful outcome where the landing outcome was ‘Success (ground pad)’.
- Result was 12/22/2015

```
%sql SELECT min(DATE) FROM SPACEXTABLE WHERE LANDING_OUTCOME = 'Success (ground pad)'
```

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

min(DATE)
2015-12-22

Successful Drone Ship Landing with Payload between 4000 and 6000

- List the names of boosters which have successfully landed on drone ship and had payload mass greater than 4000 but less than 6000
- Selected any booster versions from the SpaceX table where the payload mass was BETWEEN 4000 and 6000 AND landing outcome contained ‘Success’

```
%sql SELECT BOOSTER_VERSION FROM SPACEXTABLE WHERE PAYLOAD_MASS__KG_ between 4000 and 6000 AND LANDING_OUTC
```

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

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

Total Number of Successful and Failure Mission Outcomes

- Calculate the total number of successful and failure mission outcomes
- Made a query to count all results in SpaceX table included something like ‘Success’ or ‘Failure’ in the mission outcomes column.
- This counted to 98 occurrences of successful or failed mission outcomes.

```
%sql SELECT COUNT(*) FROM SPACEXTABLE WHERE MISSION_OUTCOME like 'Success' OR MISSION_OUTCOME like 'Failur
```

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

COUNT(*)
98

Boosters Carried Maximum Payload

- List the names of the booster which have carried the maximum payload mass
- Selected any booster_version from the SpaceX table where payload mass was the maximum. This lists all names of boosters that have carried maximum payload mass.

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

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

Booster_Version
F9 B5 B1048.4
F9 B5 B1049.4
F9 B5 B1051.3
F9 B5 B1056.4
F9 B5 B1048.5
F9 B5 B1051.4
F9 B5 B1049.5
F9 B5 B1060.2
F9 B5 B1058.3
F9 B5 B1051.6
F9 B5 B1060.3
F9 B5 B1049.7

2015 Launch Records

- Lists failed landing outcomes (drone ship) and puts out corresponding booster version, launch site in the year 2015.
- Used BETWEEN to restrict results in 2015 year.
- %sql SELECT BOOSTER_VERSION, LAUNCH_SITE, LANDING_OUTCOME FROM SPACEXTABLE WHERE LANDING_OUTCOME LIKE 'Failure (drone ship)' AND "DATE" BETWEEN '2015-01-01' AND '2015-12-31'

Booster_Version	Launch_Site	Landing_Outcome
F9 v1.1 B1012	CCAFS LC-40	Failure (drone ship)
F9 v1.1 B1015	CCAFS LC-40	Failure (drone ship)

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

- Ranks the landing outcomes between 2010 through 2017.
- %sql SELECT LANDING_OUTCOME, 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
- Groups by the count of landing outcome in a descending order

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

Section 3

Launch Sites Proximities Analysis

<Folium Map Screenshot 1>

- Replace <Folium map screenshot 2> title with an appropriate title
- Explore the generated folium map and make a proper screenshot to include all launch sites' location markers on a global map
- Explain the important elements and findings on the screenshot

<Folium Map Screenshot 2>

- Replace <Folium map screenshot 2> title with an appropriate title
- Explore the folium map and make a proper screenshot to show the color-labeled launch outcomes on the map
- Explain the important elements and findings on the screenshot

<Folium Map Screenshot 3>

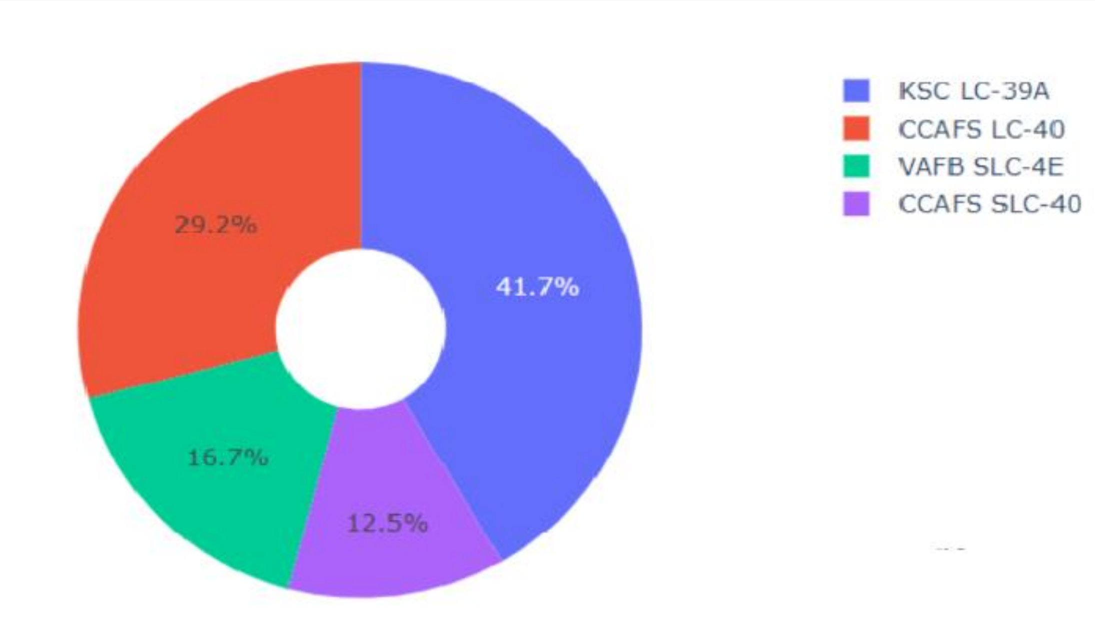
- Replace <Folium map screenshot 3> title with an appropriate title
- Explore the generated folium map and show the screenshot of a selected launch site to its proximities such as railway, highway, coastline, with distance calculated and displayed
- Explain the important elements and findings on the screenshot

Section 4

Build a Dashboard with Plotly Dash

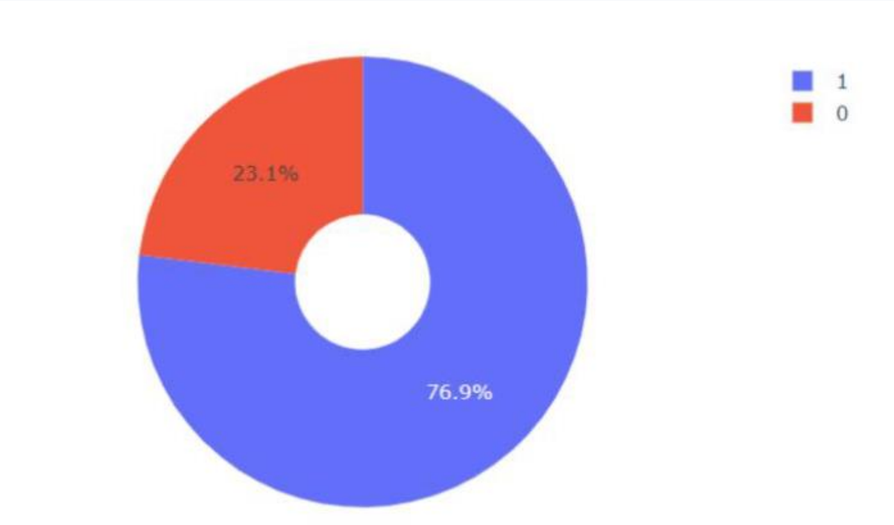
PlotlyDash Results – Sucesses per Launch Site

- This dashboard result shows the percentage of successful launches per launch site. Some insights gathered is that KSC LC-39A accounts for about 41.7% of the total successful launches



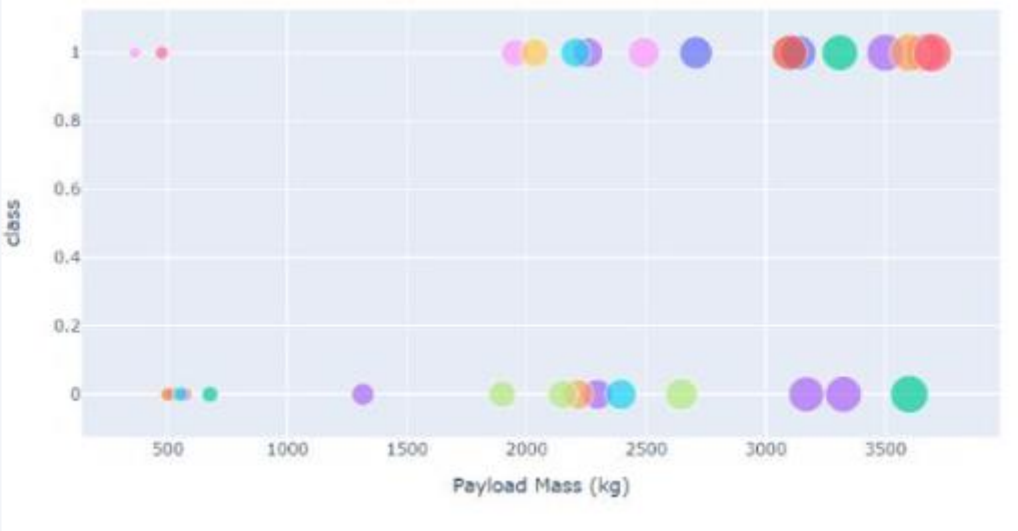
PlotlyDash Result – Success Rate KSC LC-39A

- From the last slide, we gathered that launch site KSC LC-39A accounts for almost half of the successful launches. When we breakdown the success rate of that launch site, we can see they successfully launch about 77% of the time.



<Dashboard Screenshot 3>

- Payload vs Success rate for all sites shown below. This dashboard screenshot shows the success class (0 or 1) based on the payload mass (kg).
- This screenshot shows the distribution of successful launches. Here we can see that there are many successful launches around the lower end of the scale of payload mass.



Section 5

Predictive Analysis (Classification)

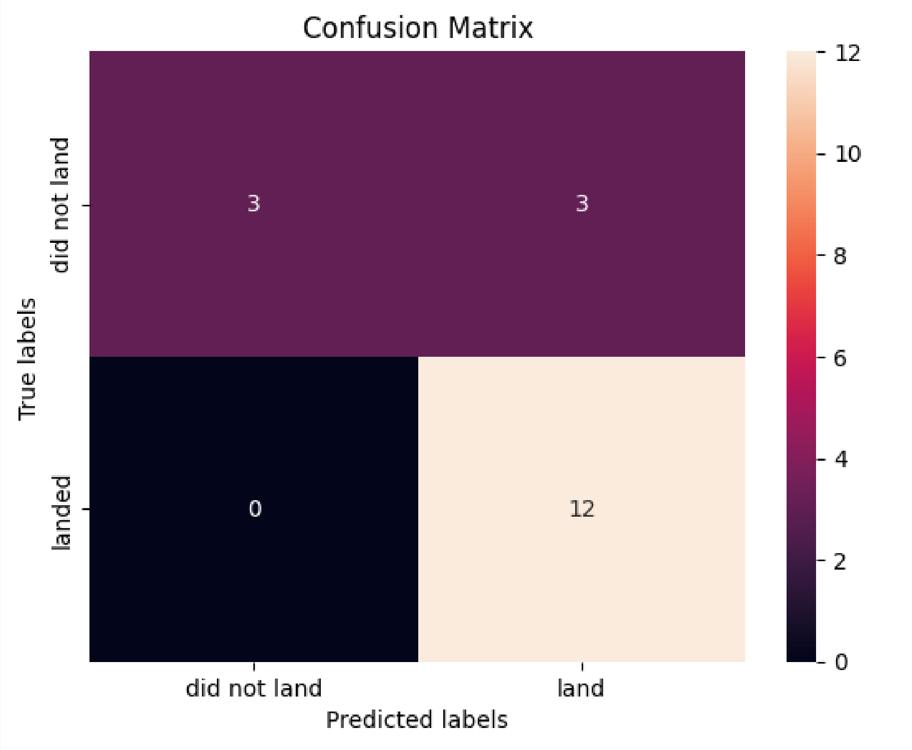
Classification Accuracy

- We found that our decision tree classifier had the highest accuracy metric with 0.87 accuracy.

```
tuned hpyerparameters :(best parameters)  {'criterion': 'gini', 'max_depth': 4, 'max_features': 'sqrt', 'min_samples_
leaf': 4, 'min_samples_split': 2, 'splitter': 'random'}
accuracy : 0.8625
```

Confusion Matrix

- Decision Tree classifier confusion matrix. Shows 12 True positives, 3 true negatives, and 3 false positives which unsuccessfully marked as "landed" when in reality it did not.
- Overall decently good accuracy



Conclusions

- KSC LC-39A had the highest number of successful launches, with a 77% success rate
- Launch sites become more successful when having larger flight numbers/amount of launches
- Launch success rate steadily increased across 2013-2020
- Decision Tree Classifier was the most accurate machine learning algorithm for determining whether a launch would be successful

Thank you!