IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

Kevin Reynolds
30 Jan 2023

Outline

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

Executive Summary

- Summary of methodologies
  - Data Collection
    - Collect and format data from the SpaceX API
    - Collect Falcon 9 historical launch data from Wikipedia using web scraping
  - Exploratory Data Analysis
    - Identify patterns and determine labels for supervised training models
    - Understand the SpaceX dataset and load dataset into DB2 database table
    - EDA and feature engineering
    - Data visualization with Folium and Plotly
  - Predictive Analysis
    - Created 5 ML models to predict if the first stage will land
- Summary of all results
  - Collected and cleaned data
  - Explored data
  - Model prediction results

Introduction

- Project background and context
  - In this capstone, we will predict if the Falcon 9 first stage will land successfully. SpaceX advertises Falcon 9 rocket launches on its website with a cost of 62 million dollars; other providers cost upward 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. This information can be used if an alternate company wants to bid against SpaceX for a rocket launch.
- Problems you want to find answers
  - Can we use SpaceX and other public data sources to predict if the first stage of SpaceX rocket will land successfully?

Section 1

Methodology

Methodology

Executive Summary

- Data collection methodology:
  - The data was collected from two sources; the SpaceX API (APIC calls) and Wikipedia (web scraping)
- Perform data wrangling
  - Data was analyzed for null values and data completeness was analyzed by evaluating number of launch sites, types of orbits, and mission outcomes. Categorical values were converted to numerical representations
- 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

- Data Collection Methods
  - The data was collected from SpaceX and Wikipedia
  - SpaceX Data
    - The SpaceX data was collected using SpaceX API calls
    - The data was decoded using the json_normalize method and converted to a Pandas dataframe
  - Wikipedia
    - The launch data collected from Wikipedia was collected using the Beautiful Soup web scraping library
    - The launch records were extracted from HMTL tables
  - Both data sets were evaluated for null values and completeness
  - Both data sets were saved to CSV files

Data Collection – SpaceX API

- The data collection required the following 3 steps:
  - Download the data using the requests.get() method
  - Define the JSON object
  - Load the JSON object onto a Pandas Dataframe
- GitHub URL of the completed SpaceX API calls notebook:

[IBM-Data-Science-Capstone/jupyter-labs-spacex-data-collection-api (1).ipynb at main · Data-Made-Simple/IBM-Data-Science-Capstone (github.com)]

Now let's start requesting rocket launch data from SpaceX API with the following URL:

```
In [6]:  1  spacex_url="https://api.spacexdata.com/v4/launches/past"

In [7]:  1  response = requests.get(spacex_url)
```

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

```
In [9]:  1  static_json_url='https://cf-courses-data.s3.us.cloud-object-storage.appdomain.cloud/IBM-DS0321EN-SkillsNetwork/datasets/API_
```

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

```
In [10]: 1  response.status_code
Out[10]: 200
```

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

```
In [11]: 1  # Use json_normalize meethod to convert the json result into a dataframe
         2  data = pd.json_normalize(response.json())
```

Data Collection - Scraping

- The data collection required three steps to complete:
  - Request the URL data with the requests.get() method.
  - Create a Beautiful Soup object to parse the data
  - Store the HTML tables in a variable

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

In [6]:
```
# use requests.get() method with the provided static_url
# assign the response to a object
data = requests.get(static_url)
```

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

In [7]:
```
# Use BeautifulSoup() to create a BeautifulSoup object from a response text content
soup = BeautifulSoup(data.content, "html.parser")
```

In [9]:
```
# 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("tr")
```

- GitHub URL of the completed web scraping notebook:

[IBM-Data-Science-Capstone/jupyter-labs-webscraping (1).ipynb at main · Data-Made-Simple/IBM-Data-Science-Capstone (github.com)]

Data Wrangling

- The data was processed per steps listed below:
  - Null values were identified and counted
  - Calculated launches per site
  - Displayed count of orbit types
  - Calculated landing outcomes
  - Converted landing class from categorical to numerical
  - Saved the data as a CSV file
- GitHub URL

[IBM-Data-Science-Capstone/labs-jupyter-spacex-Data wrangling (1).ipynb at main · Data-Made-Simple/IBM-Data-Science-Capstone (github.com)]

EDA with Data Visualization

- Plots Charted and Rationale
  - Cat plot of launch site and flight number. Provides a visual of activity per launch site
  - Scatter plot of payload mass and launch site. Displays any potential relationship between launch and payload mass capabilities.
  - Bar plot of success rate and orbit. Displays any potential relationship between mission outcomes and orbit type
  - Scatter plot of flight number and orbit type. Shows distribution of orbit types over time.
  - Scatter plot of payload mass and orbit. Shows any potential relationship between payload mass and orbit type.
  - Line plot of mission outcome and launch year. Displays an increasing mission outcome reliability over time.
- GitHub URL

IBM-Data-Science-Capstone/jupyter-labs-eda-dataviz (1).ipynb at main · Data-Made-Simple/IBM-Data-Science-Capstone (github.com)

EDA with SQL

- SQL Query Summary
  - Select distinct launch sites
  - Select launches from launch sites beginning with ‘CCA’
  - Select the sum of payload mass for the NASA (CRS) customer
  - Select the average payload mass for the F9 v1.1 booster version
  - Select the date when the first successful landing outcome occurred on a ground pad
  - Select the booster names which have success in drone ship and a payload mass between 4000 and 6000
  - Select the total number of successful and failure mission outcomes per mission outcome
  - Select the names of the booster versions which have carried the maximum payload mass
  - Select the failed landing outcomes in drone ship, their booster version, and launch site names for 2015
  - Select the count of landing outcomes between 2010-06-04 and 2017-03-20 and sort in a descending order
- GitHub URL

IBM-Data-Science-Capstone/jupyter-labs-eda-sql-coursera (1).ipynb at main · Data-Made-Simple/IBM-Data-Science-Capstone (github.com)

Build an Interactive Map with Folium

- Map Object SummarySummarize what map objects such as markers, circles, lines, etc. you created and added to a folium map
  - Markers: NASA JSC, all launch sites, and all launch records
  - Circles: NASA JSC, all launch sites, and nearest coastline
  - Lines: Line between coastline and launch site and another line between the launch site and Cape Canaveral, FL
- These object were created to provide a visual representation of launch locations, launches per location, and the relative proximity of those launch locations to the ocean and nearby cities.
- GitHub URL

IBM-Data-Science-Capstone/lab_jupyter_launch_site_location (2).ipynb at main · Data-Made-Simple/IBM-Data-Science-Capstone (github.com)

Build a Dashboard with Plotly Dash

- The Plotly graph contains 3 key items:
  - Drop down menu of all launch sites
  - Pie chart showing successful launches per site
  - Scatter plot payload mass at each site with a custom slider bar
- These graphs allow the user to compare and contract successful launches per site and analyze the payload mass from each site.
- GitHub URL

IBM-Data-Science-Capstone/spacex_dash_app.py at main · Data-Made-Simple/IBM-Data-Science-Capstone (github.com)

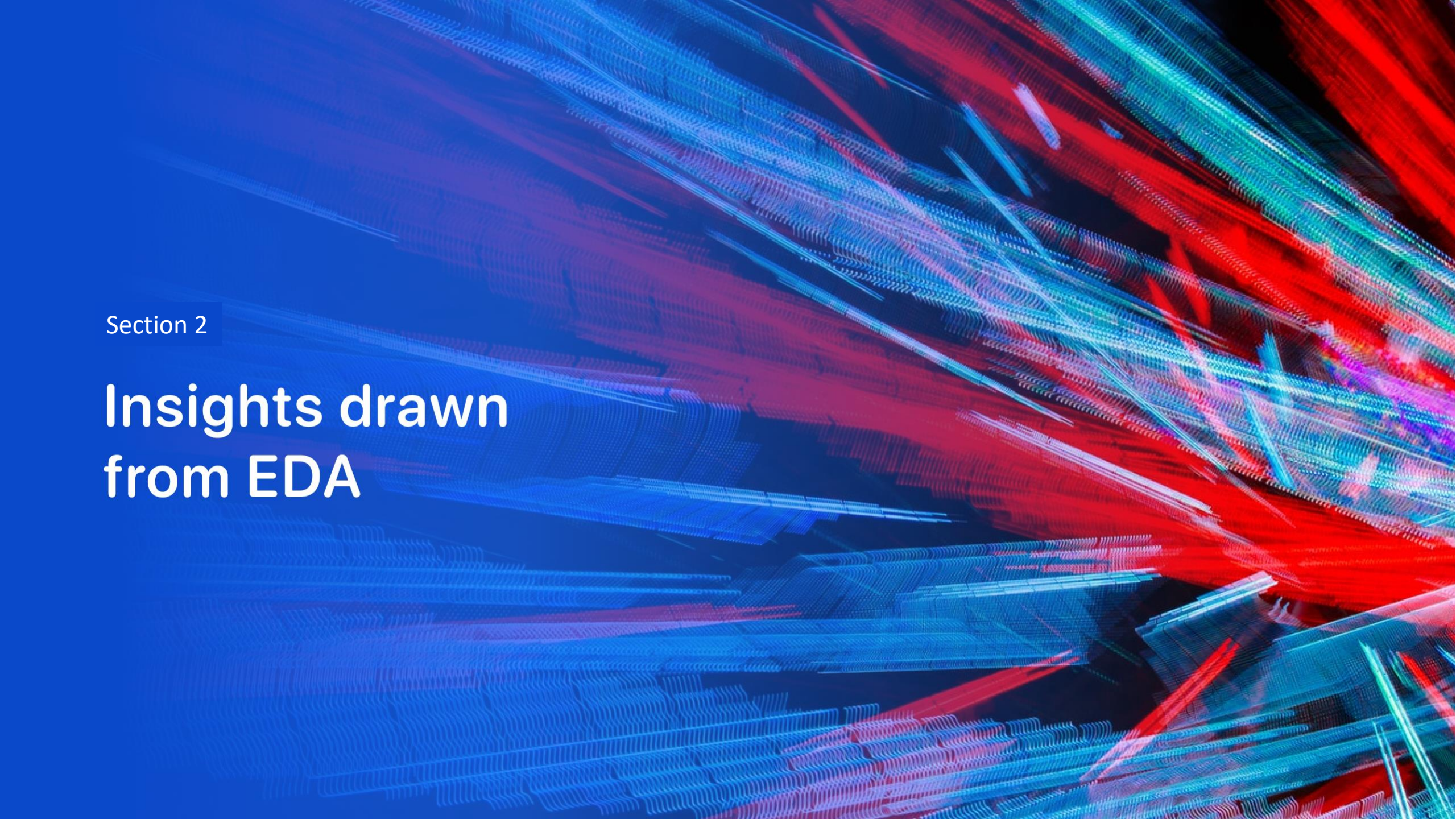
Predictive Analysis (Classification)

- Summarize how you built, evaluated, improved, and found the best performing classification model
  - The model was built using the SpaceX and Wikipedia launch data collected previously.
  - The data was scaled using StandScaler
  - The data was split into training and test sets using the train_test_split method
  - Four predictive models were created. Each model was:
    - Evaluated for accuracy
    - Parameter tuned with GridCV
    - The best performing model was selected based on accuracy with accuracy and best performing parameters provided
- GitHub URL

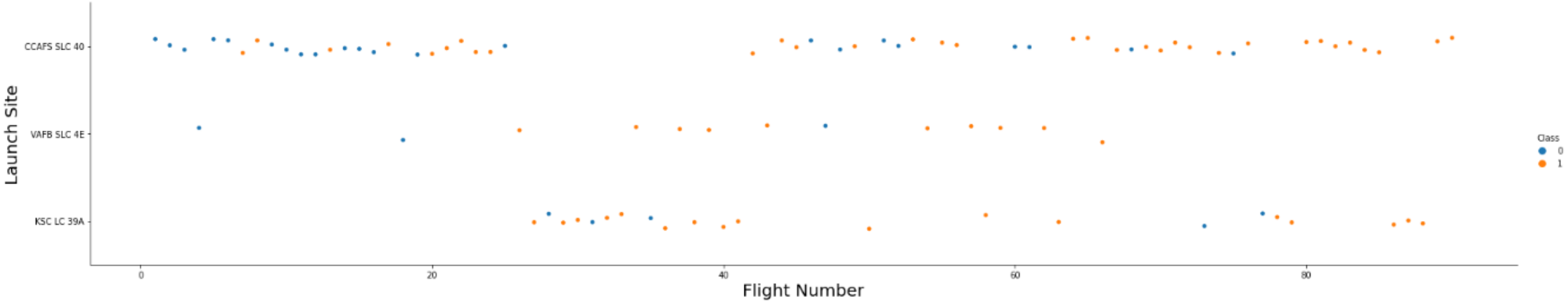
IBM-Data-Science-Capstone/SpaceX_Machine Learning Prediction_Part_5.ipynb at main · Data-Made-Simple/IBM-Data-Science-Capstone (github.com)

Results

- The project resulted in:
  - Exploratory data analysis results including figures and charts presented in this document
  - Interactive analytics demo in screenshots using Plotly and Dash
  - Predictive analysis results from four separate models


Section 2
Insights drawn from EDA

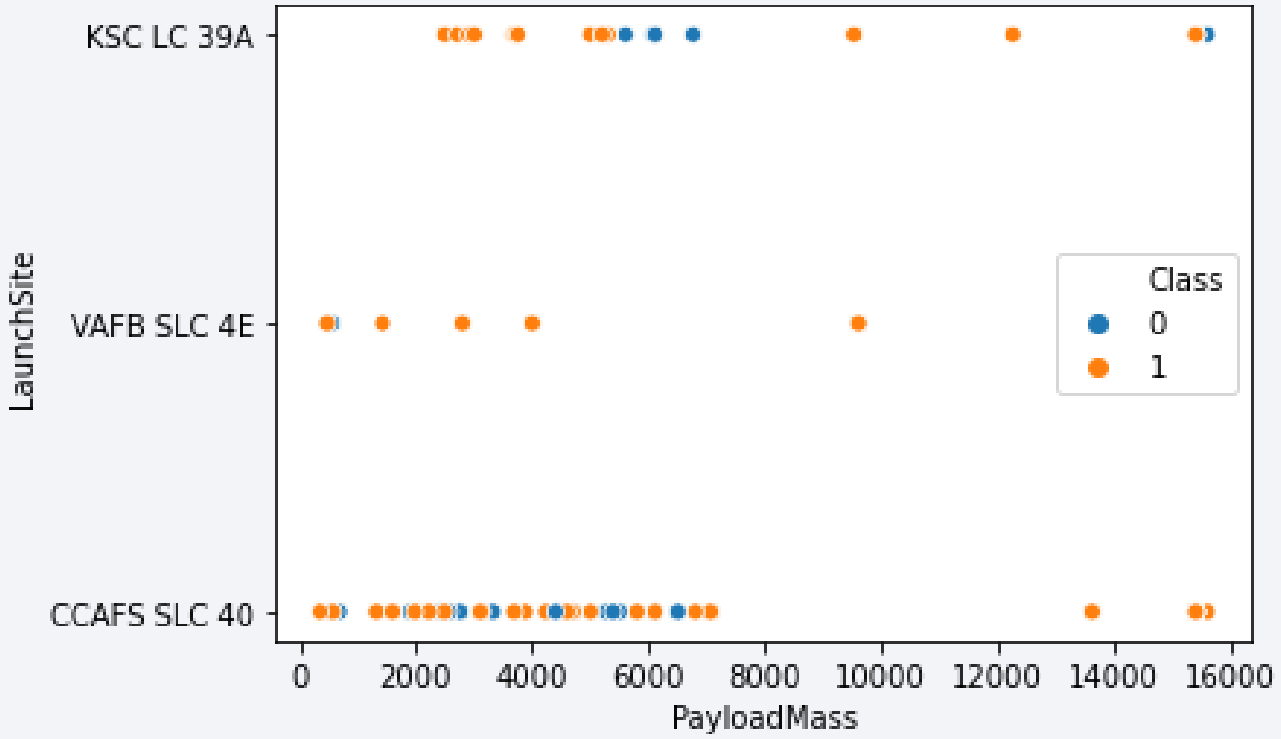
Flight Number vs. Launch Site



- The scatter plot displays the flight number vs launch site. The plot displays all launch sites in the Y axis and the flight number in the X axis. The chart shows the utilization of specific launch sites over time.

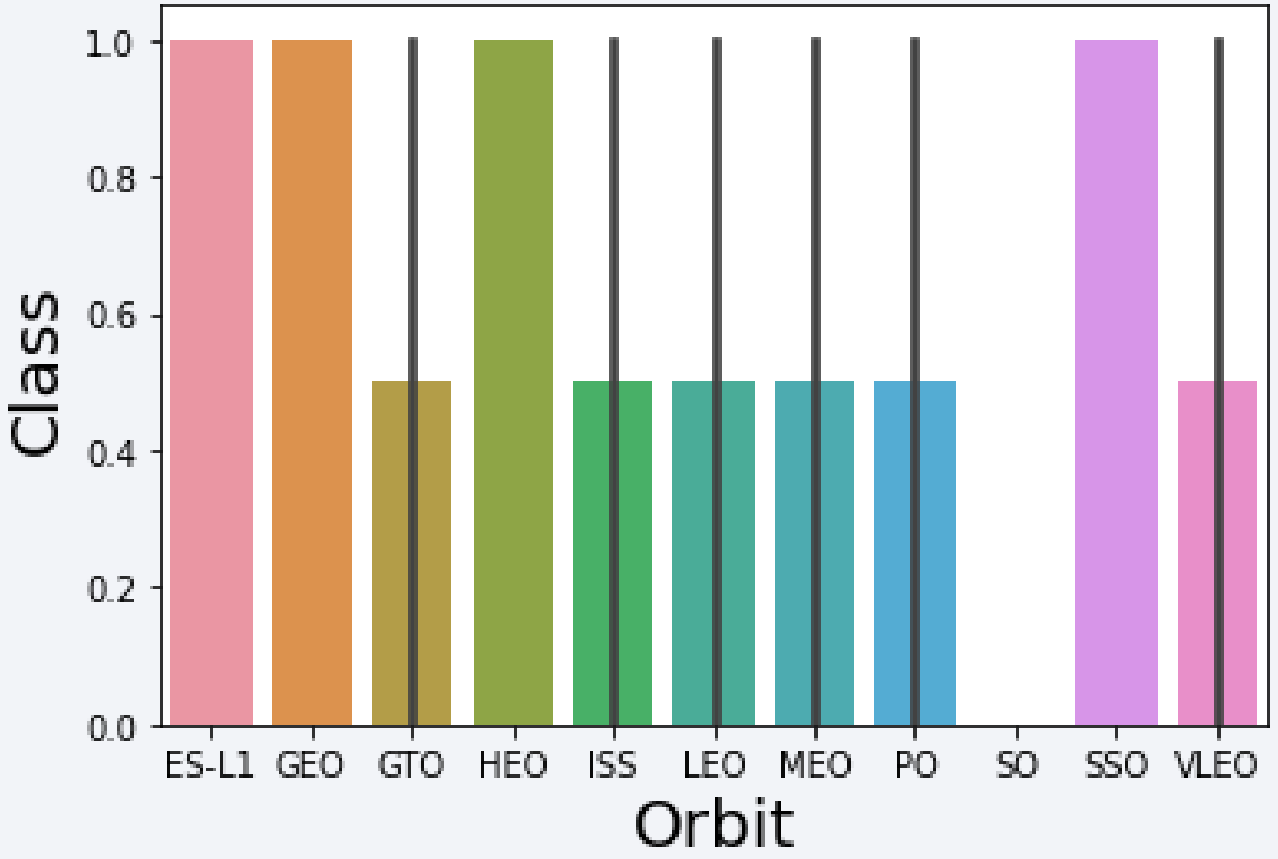
Payload vs. Launch Site

- The scatter plot shows the payload mass of each launch from each launch site.
- The chart shows that SpaceX typically launched lower mass missions from the CCAFS SLC 40 site but, that launch site is still capable of launching high payloads.



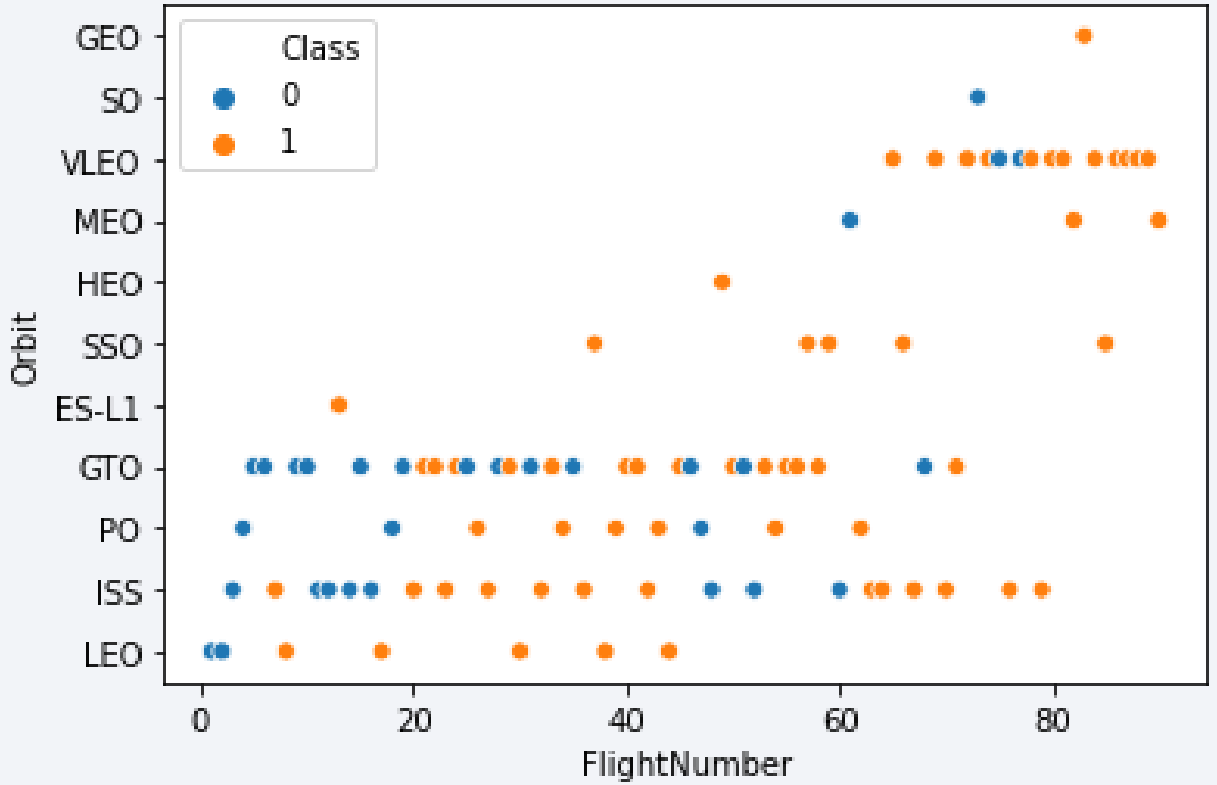
Success Rate vs. Orbit Type

- The bar chart shows the successful outcomes of missions per orbit.



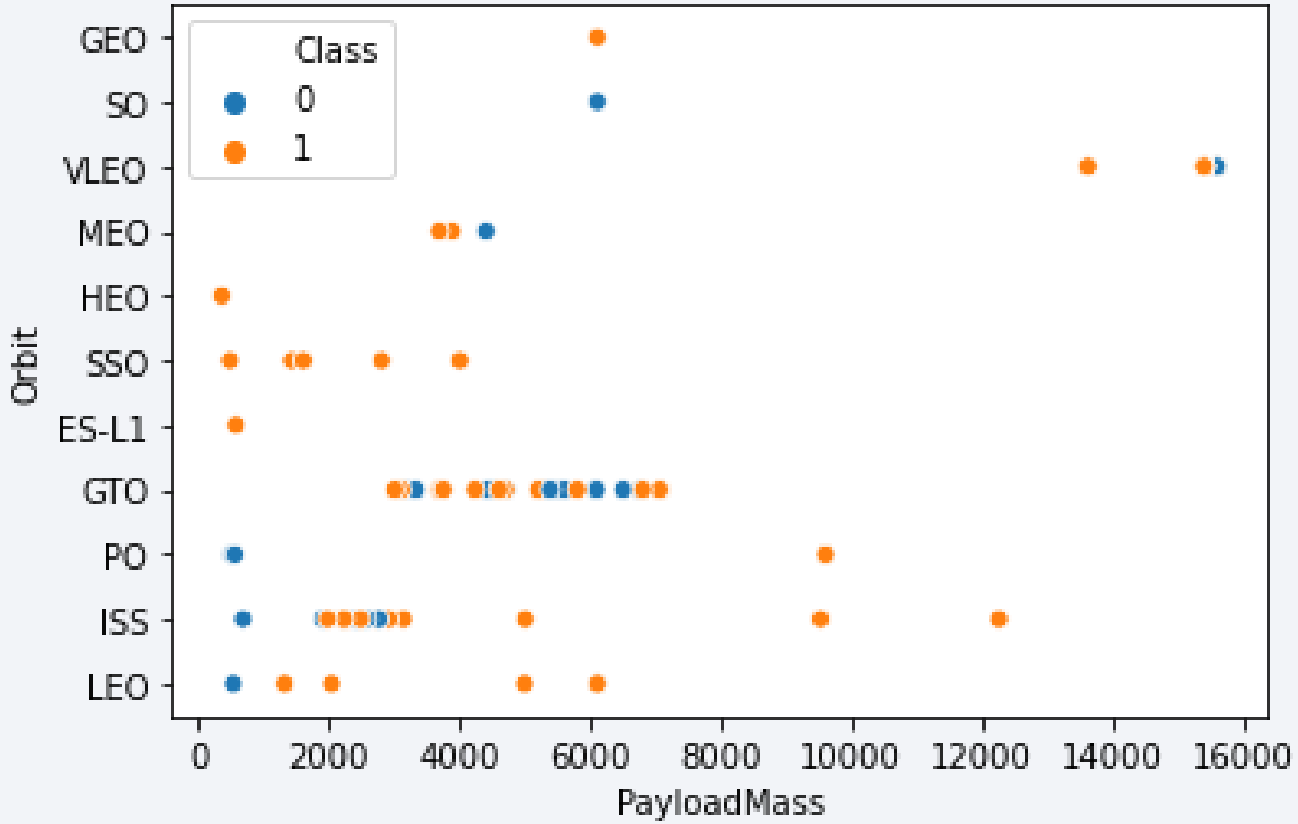
Flight Number vs. Orbit Type

- The scatter plot shows the flight number versus orbit type. Mission outcome failures are displayed as blue dots and the chart indicates that mission outcome success has improved over time.



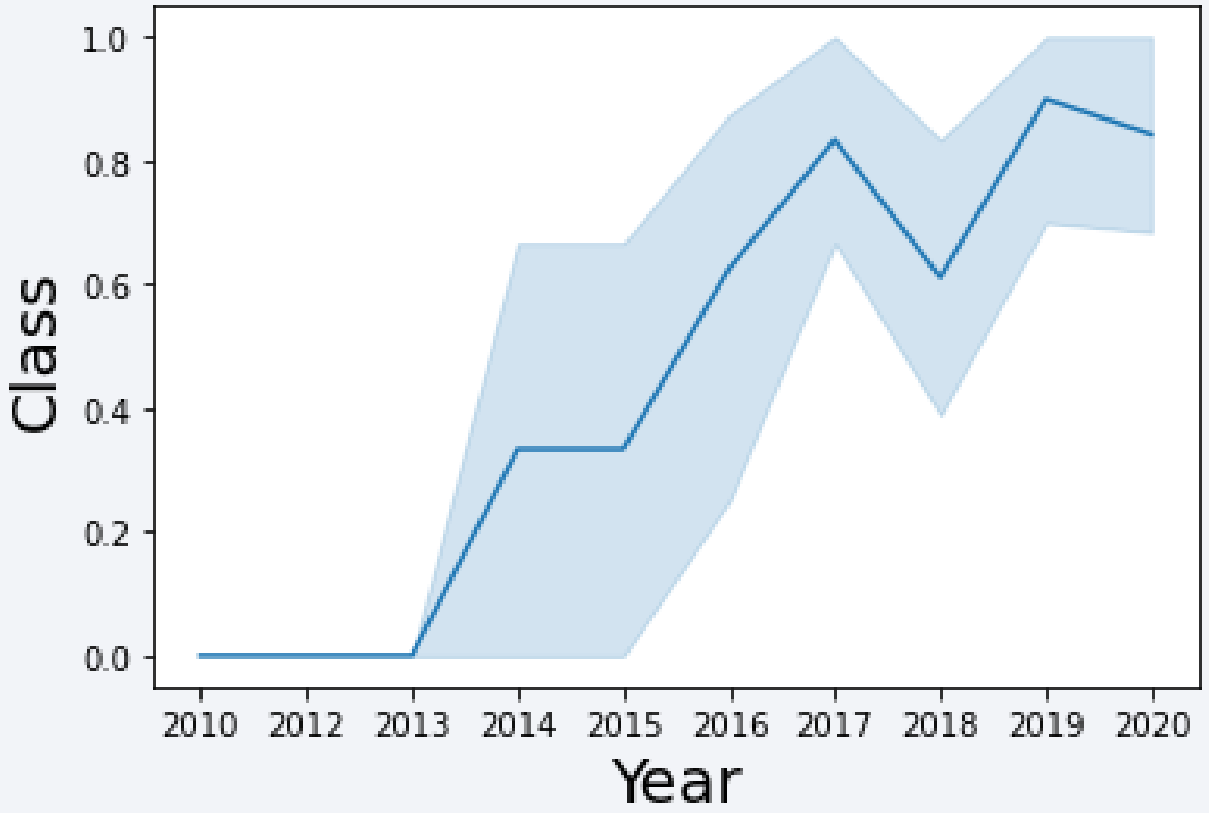
Payload vs. Orbit Type

- The scatter plot shows the payload mass versus orbit type. The chart shows the lighter payloads are more typical than higher payloads.



Launch Success Yearly Trend

- Show a line chart of yearly average success rate
- Show the screenshot of the scatter plot with explanations



All Launch Site Names

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

In [9]:
```
%sql select distinct(LAUNCH_SITE) from SPACEX
```

```
 * ibm_db_sa://zyl20881:***@6667d8e9-9d4d-4ccb-ba32-21da3bb5aafc.c1ogj3sd0tgtu0lqde00.databases.appdomain.cloud:30376/bludb
Done.
```

Out[9]:

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

- Query used: “SELECT DISTINCT(LAUNCH_SITE) from SPACEX”
- The query results in four unique launch sites contained in the dataset.

Launch Site Names Begin with 'CCA'

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

In [12]:
```
%sql select * from SPACEX where LAUNCH_SITE LIKE 'CCA%' LIMIT (5)
```

```
 * ibm_db_sa://zyl20881:***@6667d8e9-9d4d-4ccb-ba32-21da3bb5aafc.c1ogj3sd0tgtu0lqde00.databases.appdomain.cloud:30376/bludb
Done.
```

Out[12]:

DATE	time__utc_	booster_version	launch_site	payload	payload_mass__kg_	orbit	customer	mission_outcome	landing__outcome
2010-06-04	18:45:00	F9 v1.0 B0003	CCAFS LC-40	Dragon Spacecraft Qualification Unit	0	LEO	SpaceX	Success	Failure (parachute)
2010-12-08	15:43:00	F9 v1.0 B0004	CCAFS LC-40	Dragon demo flight C1, two CubeSats, barrel of Brouere cheese	0	LEO (ISS)	NASA (COTS) NRO	Success	Failure (parachute)
2012-05-22	07:44:00	F9 v1.0 B0005	CCAFS LC-40	Dragon demo flight C2	525	LEO (ISS)	NASA (COTS)	Success	No attempt
2012-10-08	00:35:00	F9 v1.0 B0006	CCAFS LC-40	SpaceX CRS-1	500	LEO (ISS)	NASA (CRS)	Success	No attempt
2013-03-01	15:10:00	F9 v1.0 B0007	CCAFS LC-40	SpaceX CRS-2	677	LEO (ISS)	NASA (CRS)	Success	No attempt

- Query Used: “SELECT * FROM SPACEX where LAUNCH_SITE LIKE ‘CCA%’ LIMIT (5)”
- The query displays the first 5 records containing launch sites that start with ‘CCA’.

Total Payload Mass

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

```
In [30]:  1 %sql select SUM(payload_mass__kg_) from SPACEX where CUSTOMER = 'NASA (CRS)'
 * ibm_db_sa://zyl20881:***@6667d8e9-9d4d-4ccb-ba32-21da3bb5aafc.c1ogj3sd0tgtu0lqde00.databases.appdomain.cloud:30376/bludb
Done.
```

Out[30]:

1
45596

- Query Used: “SELECT SUM(payload_mass__kg_) from SPACEX where CUSTOMER = ‘NASA (CRS)”
- The query returns the sum of all payloads where the customer column equals ‘NASA (CRS)’.

Average Payload Mass by F9 v1.1

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

In [31]:
```
%sql select avg(CAST(payload_mass__kg_ AS DECIMAL(7,2))) from SPACEX where booster_version = 'F9 v1.1'
```

```
 * ibm_db_sa://zyl20881:***@6667d8e9-9d4d-4ccb-ba32-21da3bb5aafc.c1ogj3sd0tgtu0lqde00.databases.appdomain.cloud:30376/bludb
Done.
```

Out[31]:

1
2928.4000000000000000000000000

- Query Used: “SELECT AVG(CAST(payload_mass__kg_ AS DECIMAL(7,2))) from SPACEX where booster_version = ‘F9 v1.1’”.
- The query returns the average payload mass for booster version F9 v1.1.

First Successful Ground Landing Date

```
In [38]:  1 %sql select min(date) from SPACEX where Landing__Outcome = 'Success (ground pad)'
 * ibm_db_sa://zyl20881:***@6667d8e9-9d4d-4ccb-ba32-21da3bb5aafc.c1ogj3sd0tgtu0lqde00.databases.appdomain.cloud:30376/bludb
Done.
Out[38]:
1
2015-12-22
```

- Query Used: “SELECT MIN(date) from SPACEX where Landing__Outcome = ‘Success (ground pad)”
- The query returns the earliest date, via the MIN function, for which there was a successful ground pad mission outcome.

Successful Drone Ship Landing with Payload between 4000 and 6000

List the names of the boosters which have success in drone ship and have payload mass greater than 4000 but less than 6000

In [4]:
```
%sql select booster_version from spacex where Landing__Outcome = 'Success (drone ship)'\
        and payload_mass__kg_ > 4000 and payload_mass__kg_ < 6000
```

```
 * ibm_db_sa://zyl20881:***@6667d8e9-9d4d-4ccb-ba32-21da3bb5aafc.c1ogj3sd0tgtu0lqde00.databases.appdomain.cloud:30376/bludb
Done.
```

Out[4]:

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

- Query Used: “SELECT booster_version from SPACEX where Landing__Outcome = ‘Success (drone ship)’ and payload_mass__kg_ > 4000 and payload_mass__kg_ < 6000

- The query returns the booster versions for successful mission outcomes and a payload mass between 4000 and 6000.

Total Number of Successful and Failure Mission Outcomes

```
In [5]:  1 %sql select DISTINCT(mission_outcome), COUNT(Mission_outcome) from spacex group by mission_outcome
 * ibm_db_sa://zyl20881:***@6667d8e9-9d4d-4ccb-ba32-21da3bb5aafc.c1ogj3sd0tgtu0lqde00.databases.appdomain.cloud:30376/bludb
Done.
```

Out[5]:

mission_outcome	2
Failure (in flight)	1
Success	99
Success (payload status unclear)	1

- Query Used: “SELECT DISTINCT(mission_outcome), COUNT(mission_outcome) from SPACEX GROUP BY mission_outcome”
- The query returns the count of mission outcomes grouped by unique mission outcome.

Boosters Carried Maximum Payload

List the names of the booster_versions which have carried the maximum payload mass. Use a subquery

In [6]:
```
%sql select booster_version from spacex where payload_mass__kg_ = (select max(payload_mass__kg_) from spacex)
```

```
 * ibm_db_sa://zyl20881:***@6667d8e9-9d4d-4ccb-ba32-21da3bb5aafc.c1ogj3sd0tgtu0lqde00.databases.appdomain.cloud:30376/bludb
Done.
```

Out[6]:

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

- Query Used: “SELECT booster_version from SPACEX where payload_mass__kg_ = (SELECT MAX(payload_mass__kg_) from SPACEX)
- A subquery is used to query all results with the max payload. The main query selects the booster version from that subset of data.

2015 Launch Records

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

In [7]:
```
%sql select booster_version, launch_site from spacex where date LIKE '%2015%' and Landing__Outcome LIKE '%Failure%'
```

```
 * ibm_db_sa://zyl20881:***@6667d8e9-9d4d-4ccb-ba32-21da3bb5aafc.c1ogj3sd0tgtu0lqde00.databases.appdomain.cloud:30376/bludb
Done.
```

Out[7]:

booster_version	launch_site
F9 v1.1 B1012	CCAFS LC-40
F9 v1.1 B1015	CCAFS LC-40

- Query Used: “SELECT booster_version, launch_site from SPACEX where date LIKE ‘%2015%’ and Landing__Outcome LIKE ‘%Failure%’”

- The query returns the booster version and launch site for all failed landing outcomes experienced during 2015.

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

In [9]:
```
%sql select landing__outcome, count(landing__outcome) from spacex where date > '06-04-2010'\
            and date < '03-20-2017' group by landing__outcome order by count(landing__outcome) DESC
```

```
 * ibm_db_sa://zyl20881:***@6667d8e9-9d4d-4ccb-ba32-21da3bb5aafc.c1ogj3sd0tgtu0lqde00.databases.appdomain.cloud:30376/bludb
Done.
```

Out[9]:

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

- Query Used: “SELECT landing__outcome, count(landing__outcome) from SPACEX where date > ‘06-04-2010’ and date < ‘03-20-2017’ GROUP BY landing__outcome ORDER BY count(landing__outcome) DESC
- The query returns the count of landing outcomes between 6/4/2010 and 3/20/2017. The result is grouped by unique landing outcome and displayed in descending order.

Section 3

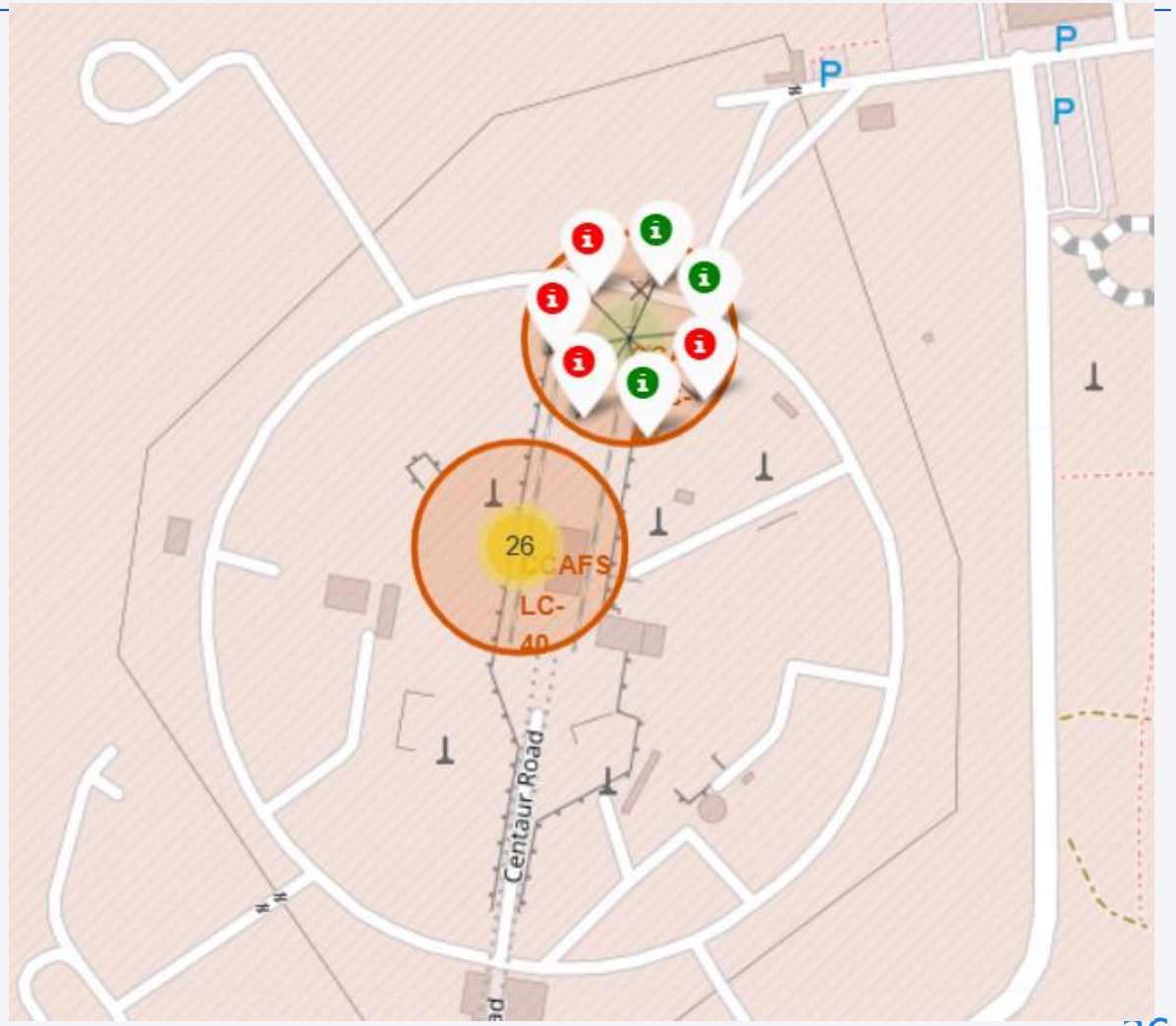
Launch Sites Proximities Analysis

SpaceX Launch Sites

- The screenshot shows the SpaceX launch sites depicted on a map.

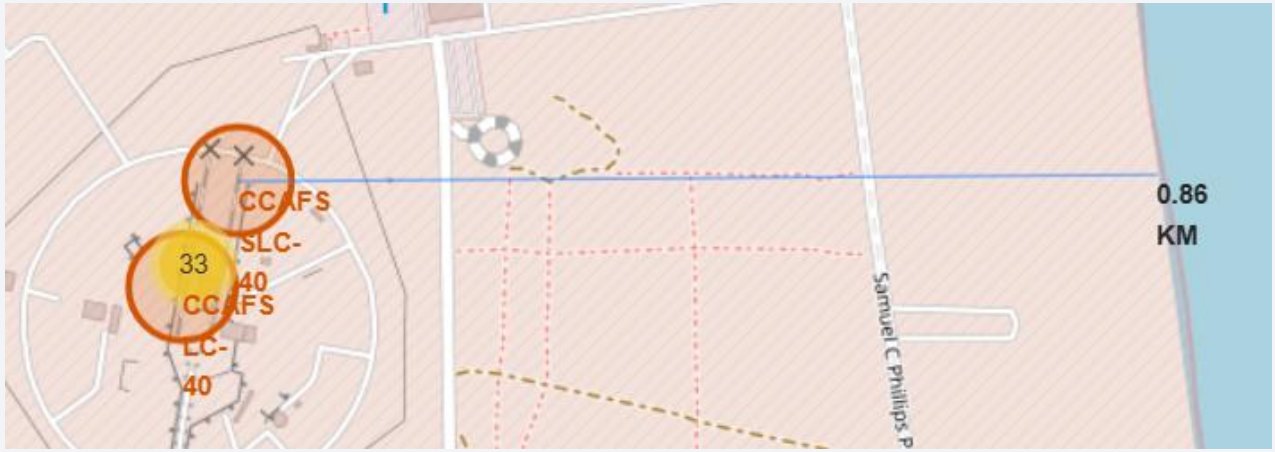
Launch Site CCAFS SLC-40 Labeled Launch Outcomes

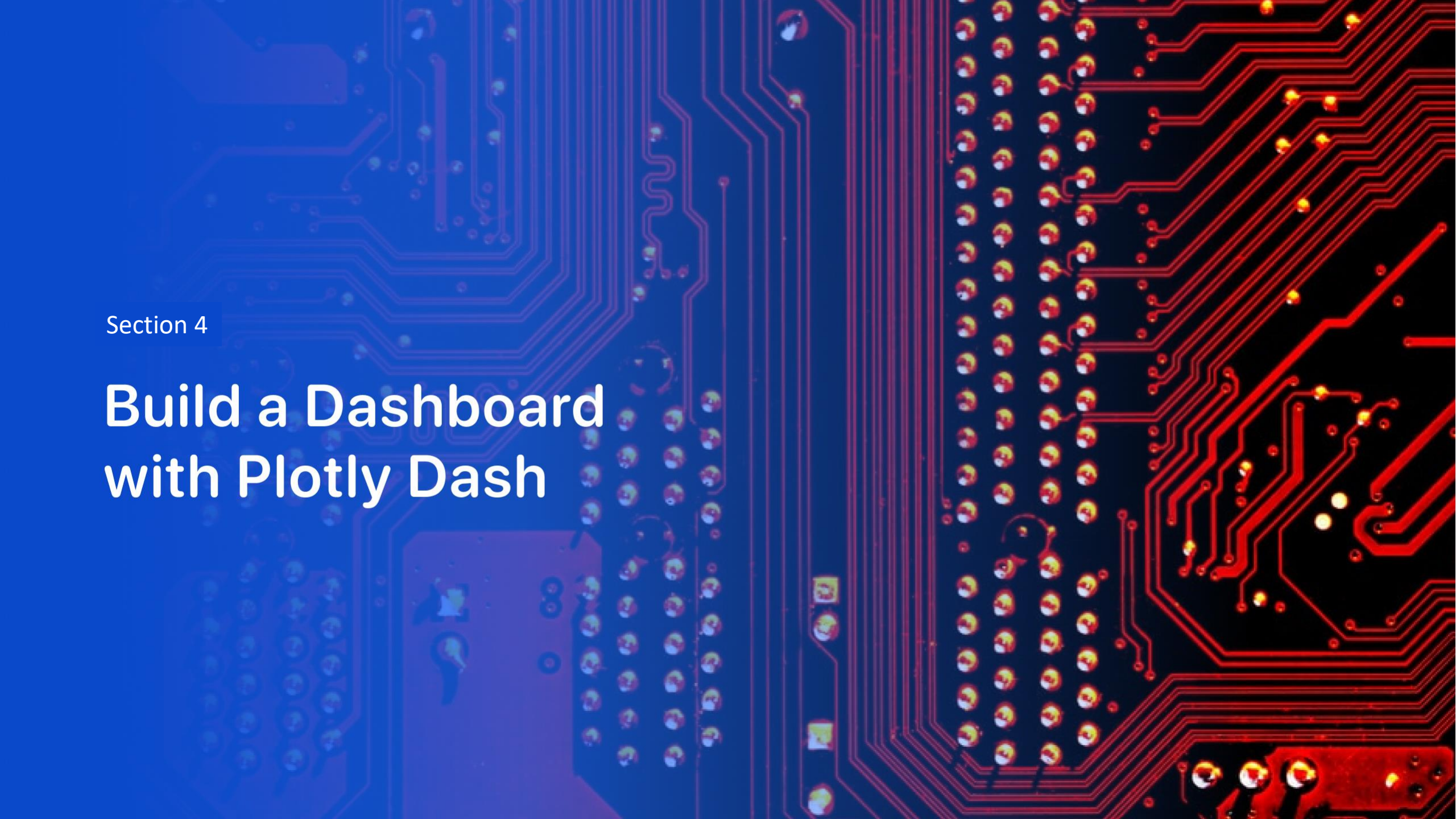
- The screenshot shows circles displaying launch sites CCAFS SLC-40 and CCAFS LC-40.
- CCSFA SLC-40 is expanded and shows the launch outcome results expressed as colors. Red is a failed outcome and green is a successful outcome.



Launch Site CCAFS SLC-40 Distance to Coastline

- The screenshot shows the distance from the CCAFS SLC-40 launch site to the nearest coastline. The image displays the distance as 0.86 KM.

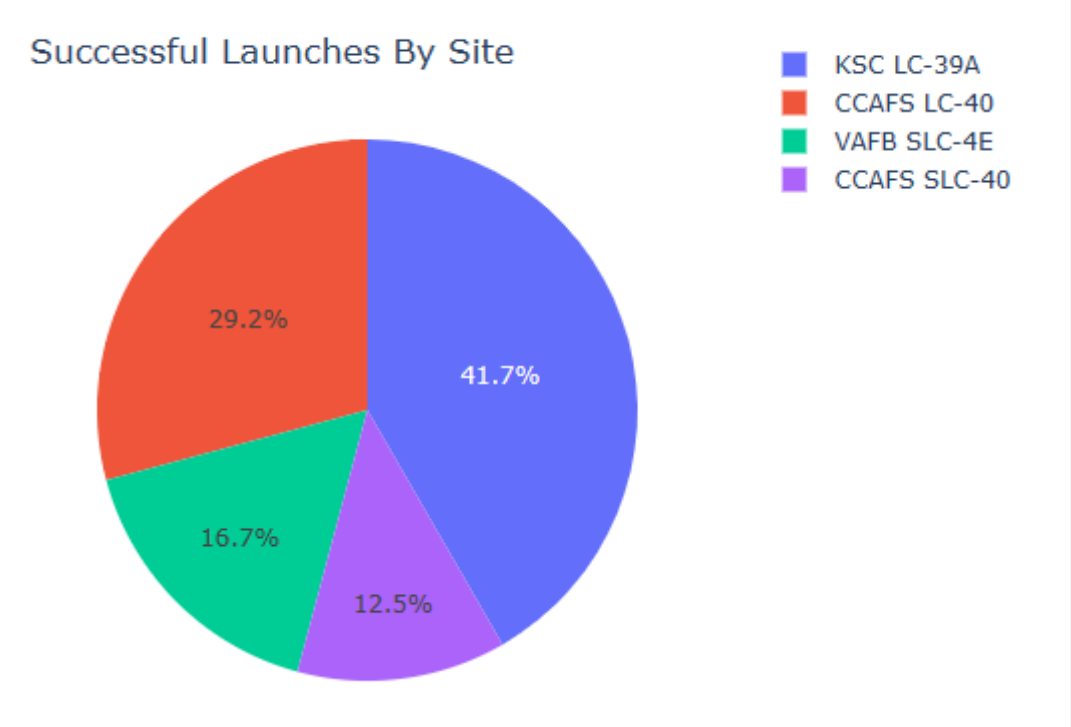



Section 4
Build a Dashboard with Plotly Dash

Launch Success for All Sites

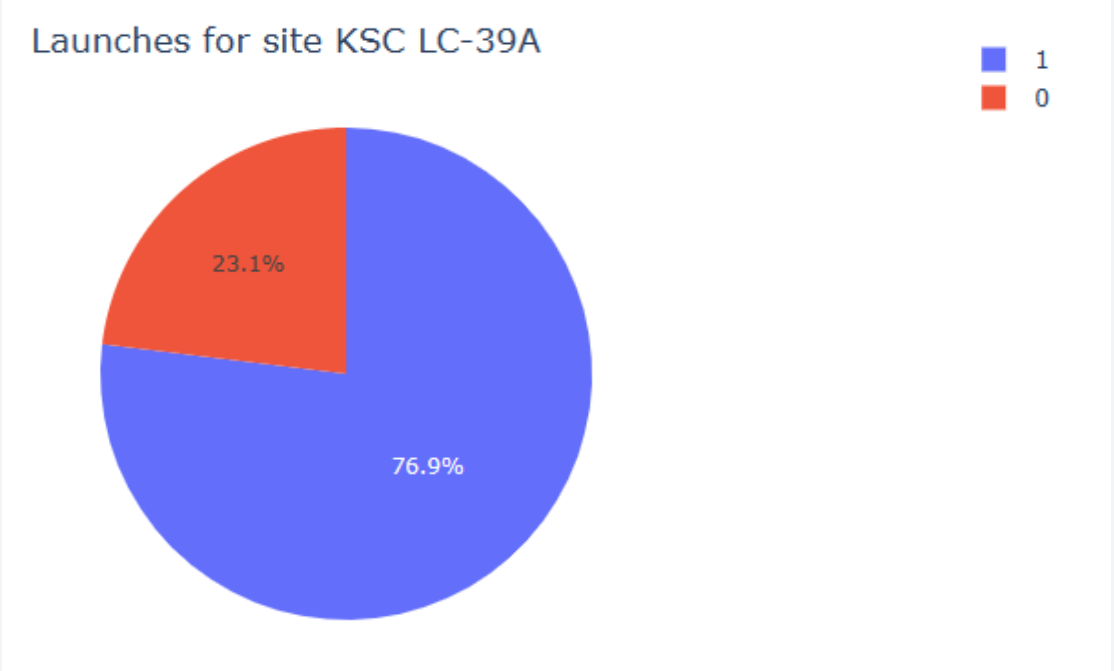
- The screenshot shows a pie chart of all successful launches by site.

- The chart contains a title, legend indicating the color code for each site, and the pie chart with successful launches displayed as a percentage.

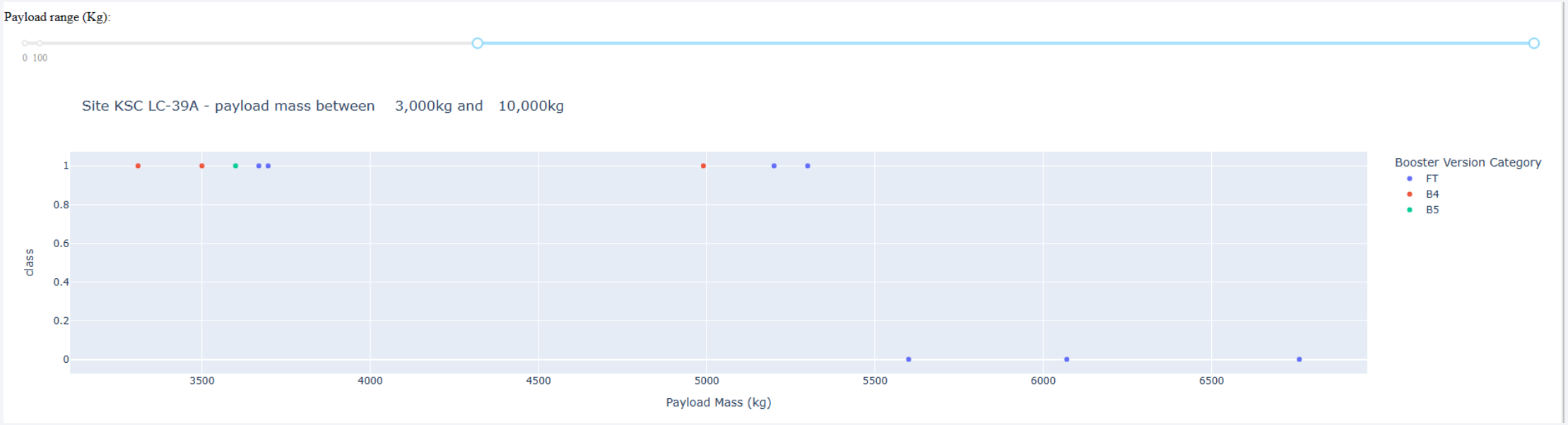


KSC LC-39A Successful Launch Rate

- The pie chart shows the successful launch rate for site KSC LC-39A.
- The screenshot shows the chart title, a legend where 1 denotes a successful launch and 0 denotes a failed launch, and the pie chart.
- The pie chart shows that 76.9% percent of all launches from KSC LC-39A were successful



Payload vs Launch for All Sites (3,000 – 10,000 KG)



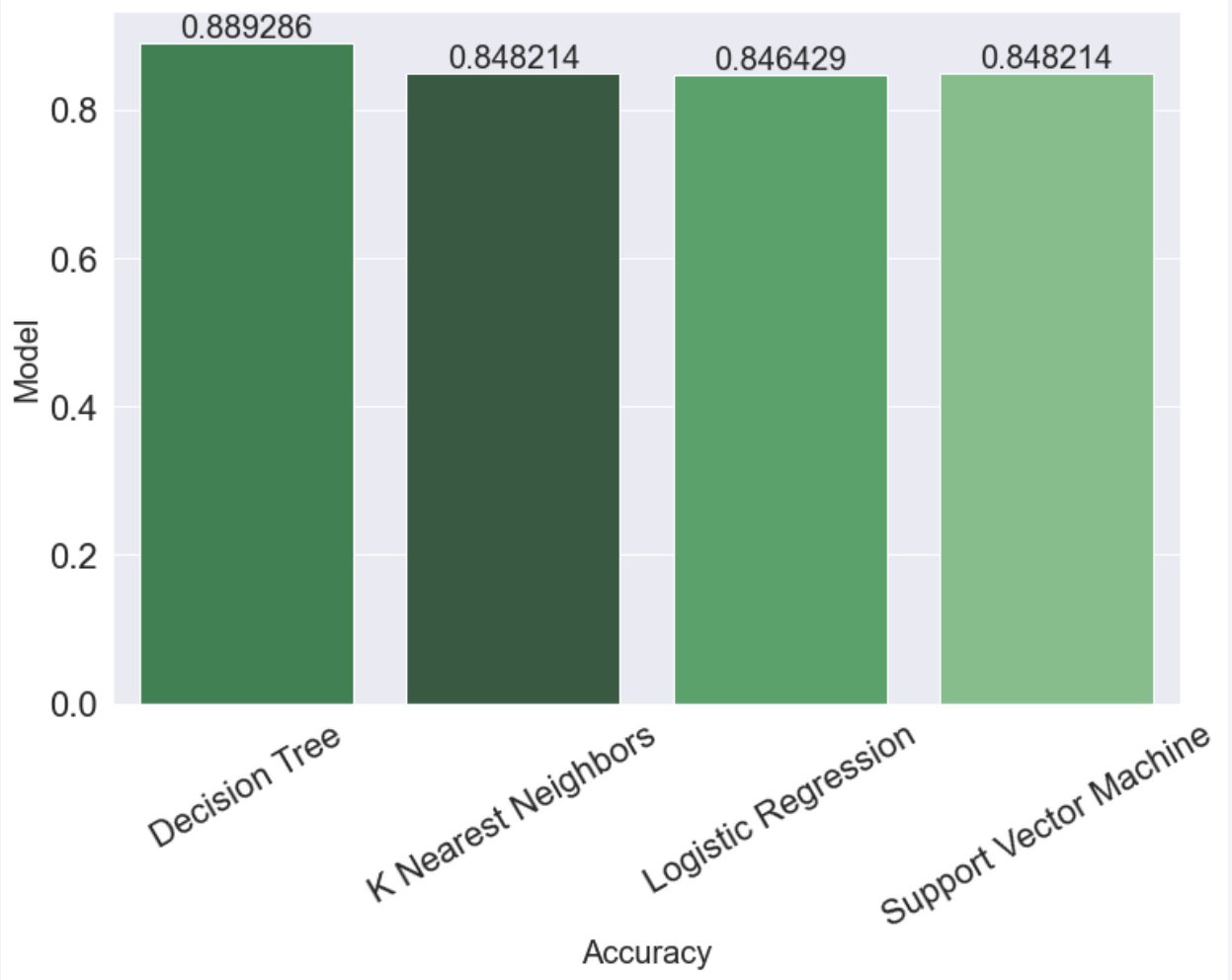
- The scatter plot shows the payload vs launch outcome (1 for success, 0 for failure) for payload masses between 3,000 and 10,000 KG.
- The screenshot contains a title, legend, and scatter plot.

Section 5

Predictive Analysis (Classification)

Classification Accuracy

- The chart shows the best score of each model
- The decision tree model has the best accuracy at 0.889.



Confusion Matrix

- The confusion matrix of the decision tree model is shown to the right
- The confusion matrix plots the test results (true labels) against the model predicted results

Conclusions

- The datasets collected from both SpaceX and Wikipedia were generally complete
- SpaceX has a relatively high success rate and improving with time
- Other factors such as orbit and payload do have an impact on success rate
- The decision tree classifier was the best algorithm for this specific dataset

Appendix

- No other data was used to complete this project.

Thank you!