IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

Andrew Doublard
06-05-2024

Outline

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

Executive Summary

- In this project the following methodologies were used:
  - ➢ Data Collection through API.
  - ➢ Data Collection with Web Scraping.
  - ➢ Data Wrangling.
  - ➢ Exploratory Data Analysis with SQL.
  - ➢ Exploratory Data Analysis with Data Visualization.
  - ➢ Interactive Visual Analytics with Folium.
  - ➢ Machine Learning Prediction.

- Results:
  - ➢ Exploratory Data Analysis to identify key relationships.
  - ➢ Interactive Visual Analytics to show the outcomes.
  - ➢ Predictive Analytics to answer the business question.

Introduction

- Space travel is beginning to become a reality due to companies like SpaceX making it more affordable. SpaceX achieves this via their reusable Falcon 9 rockets which reduces costs to $62m, roughly one third of it’s competitors. In this project, I will analyse SpaceX's data using machine learning, with the goal of predicting whether the Falcon 9 first stage will land and in turn, also predicting its cost.

- Main Objectives:
  - ➢ Identify the key relationships which determine if the rocket will land successfully.
  - ➢ How these key relationships interact to give the success rate of a particular set of conditions.
  - ➢ Which conditions yield the highest chance of a successful outcome.

Section 1

Methodology

Methodology

Executive Summary

- Data collection methodology:
  - Requesting data from SpaceX's API.
  - Web scraping from SpaceX's Wikipedia.
- Perform data wrangling
  - Adding a Landing Outcome label.
- 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

- Part 1 of the Data Collection was to make a request to the SpaceX API:
  - Requested and parsed the SpaceX launch data using the GET request.
  - Used json_normalize method to convert the Json result into a data frame.
  - Filtered the data frame to only include Falcon 9 launches.
  - Dealt with missing values, replacing the missing Payload Mass with the mean value.
- Part 2 was to use web scraping on the SpaceX Wikipedia:
  - Request the Falcon9 Launch Wiki page from its URL.
  - Extract all column/variable names from the HTML table header.
  - Create a data frame by parsing the launch HTML tables.

Data Collection – SpaceX API

- I used the get request to the SpaceX API to collect data, clean the requested data and did some basic data wrangling and formatting.

- GitHub documentation of this stage can be found here.

API Calls
- Rocket Data, Launch Data, Payload Data, Landing Data.

Data Frame
- Converting JSON result to Data Frame, and filtering to Falcon 9 Rockets only.

Data Wrangling
- Replacing missing values for Payload with the mean.

Data Collection - Scraping

- Web scraping from Falcon9 Launch Wikipedia page.

- GitHub documentation of this stage can be found here.

Requesting Data from its URL

- HTTP GET method to request the Falcon9 Launch HTML page, as an HTTP response. Then into a BeautifulSoup object.

Extracting Table and Columns

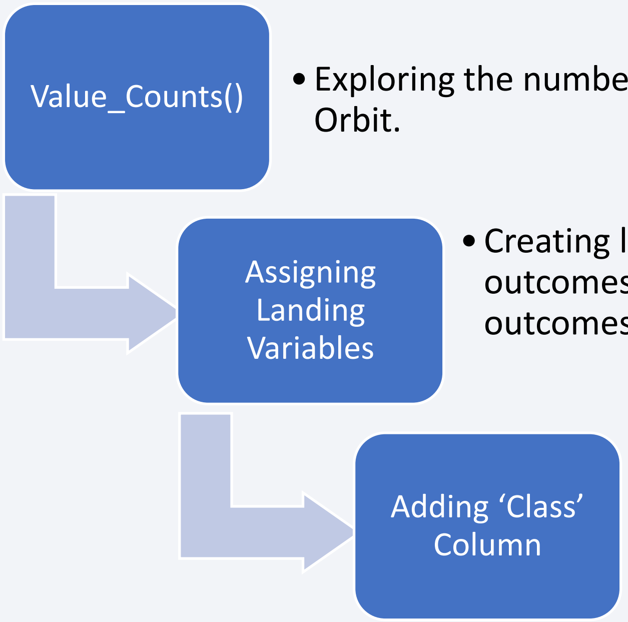
- Using the find_all() function.

Creating Data Frame

- Parsing the HTML tables. Appending the data into a empty dictionary that gets converted into a pandas data frame.

Data Wrangling

- Wrangling the Data to create training labels.



- Exploring the number of Launches per Site and Orbit.
- Creating lists for all landing outcomes and unsuccessful outcomes.
- Using the Landing variables to add a 'Class' column to determine if a landing was successful or not.

- GitHub documentation of this stage can be found here.

EDA with Data Visualization

1) Explored how flight number and payload relate to launch sites using scatter plots.

2) Visualized the success rates of different orbit types using a barplot.

3) Investigated the relationship between flight number and orbit type using a scatter point chart.

4) Explored the relationship between payload and orbit type using a scatter point chart.

5) Plotted the yearly trend of launch success to understand the program's performance over time.

6) Dummy Variables Creation:

Converted categorical columns into dummy variables for further analysis.

GitHub documentation of this stage can be found here.

EDA with SQL

1) A query to identify unique site used for launch by SpaceX.
2) Showed five records from launch sites with names containing string “CCA”.
3) Displayed total payload mass carried by boosters launched by NASA (CRS).
4) Showed the mean payload mass carried by booster version F9 v1.1.
5) Listed the date when the first successful 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) Used a subquery to list the names of the Booster versions which have carried the maximum payload mass.
9) Used Substr() to 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 landing outcomes such as Failure (drone ship) or Success (ground pad) between the date 2010-06-04 and 2017-03-20, in descending order.

- GitHub documentation of this stage can be found here.

Build an Interactive Map with Folium

1) Added map objects such as markers and circles of the launch sites on the folium map.
2) Added additional markers to each launch site to specify:

    a) Successful launches with green markers.

    b) Unsuccessful launches with red markers.

3) Put a marker on nearest coastline, and calculated distance to a launch site.
4) Drew a Polyline between the coastline and the launch site.
5) Put markers on, and then calculated distances to the closest city.

- GitHub documentation of this stage can be found here.

Build a Dashboard with Plotly Dash

1) Displayed the total launches from each site using pie charts, providing an overview of launch distribution.

2) Plotted scatter graphs showing the relationship between outcomes and payload mass for different booster versions, offering insights into mission success relative to payload mass.

3) Integrated a dropdown menu to select launch sites and a range slider to choose payload masses, enhancing user interactivity and data exploration.

4) Implemented callback functions to allow real-time data visualization based on user input.

- GitHub documentation of this stage can be found here.

Predictive Analysis (Classification)

Created a Numpy array from "Class“ variable. → Used StandardScaler to standardize the data. → Split into an 80:20 Train-Test set. → Identified best parameters → Assessed them accuracy using test data → Plotted Confusion Matrix

- Approaches:
  1) Logistic Regression
  2) Support Vector Machine
  3) Decision Tree Classifier
  4) K-Nearest Neighbour

- GitHub documentation of this stage can be found [here].

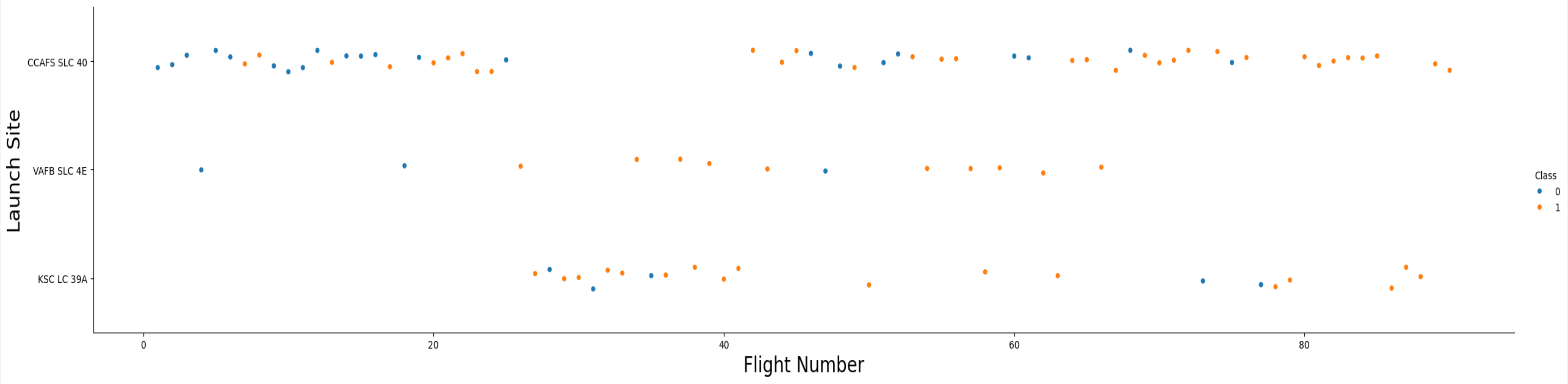
Results

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

Section 2

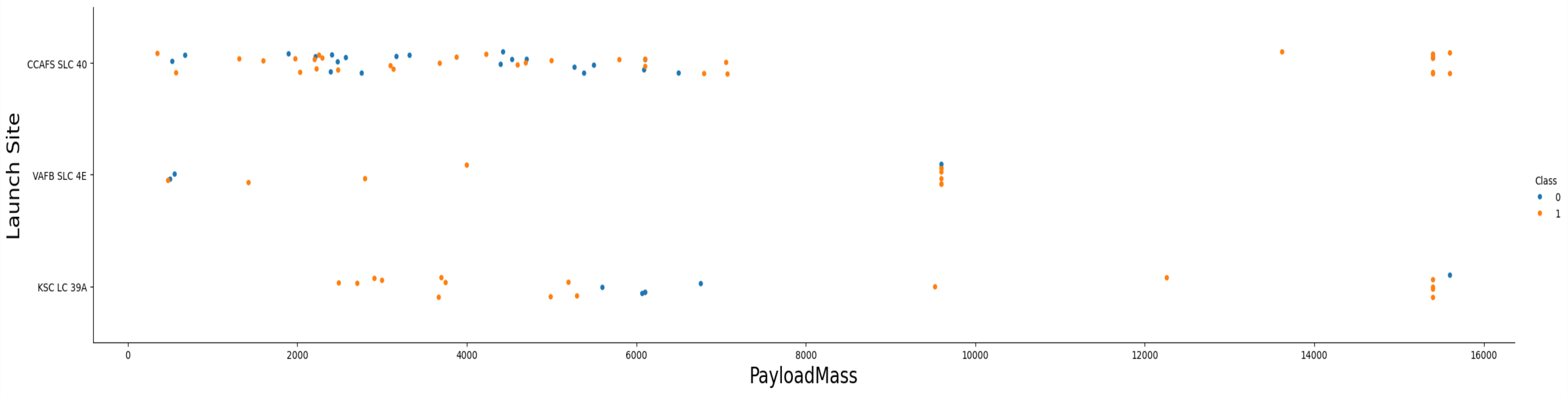
Insights drawn from EDA

Flight Number vs. Launch Site



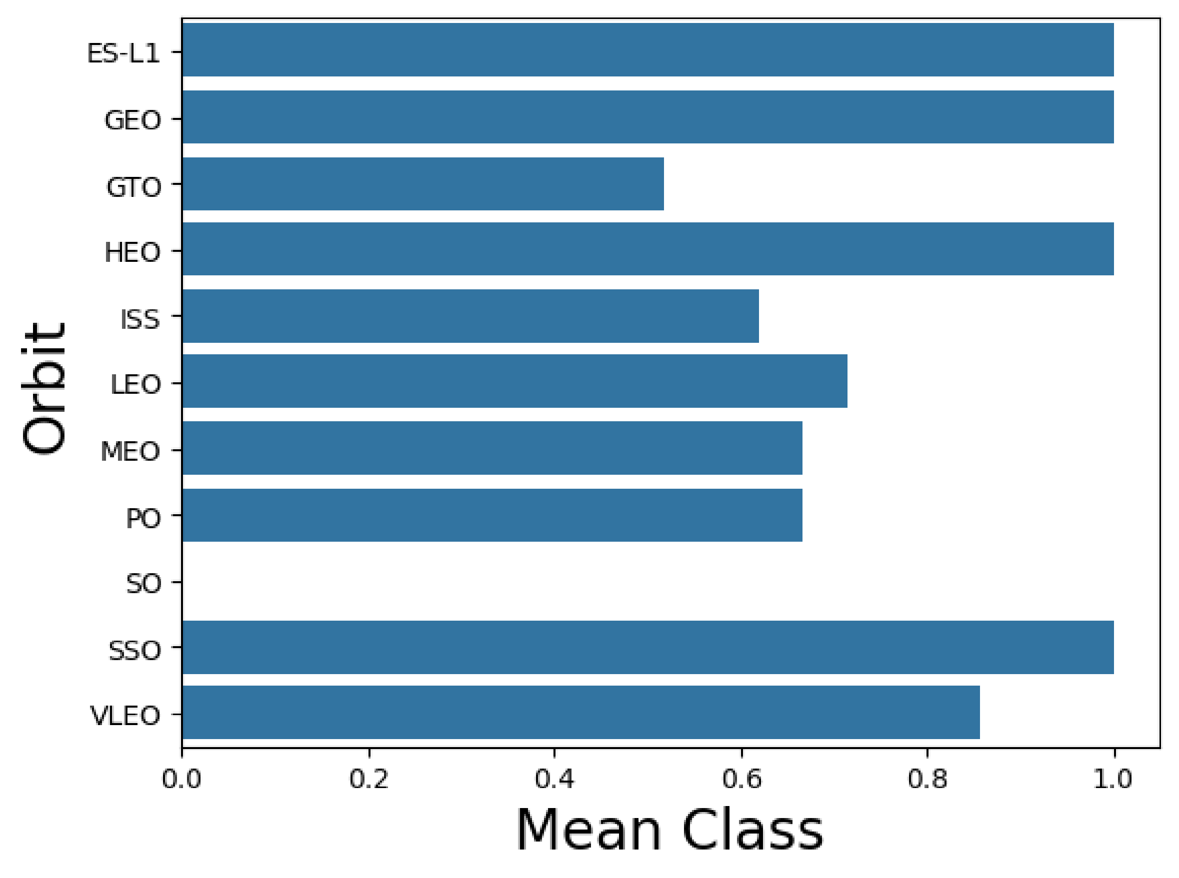
1) Flight success becomes more frequent as the number of flights increase.
2) Launch site CCAF SLC 40 is the most used launch site.

Payload vs. Launch Site



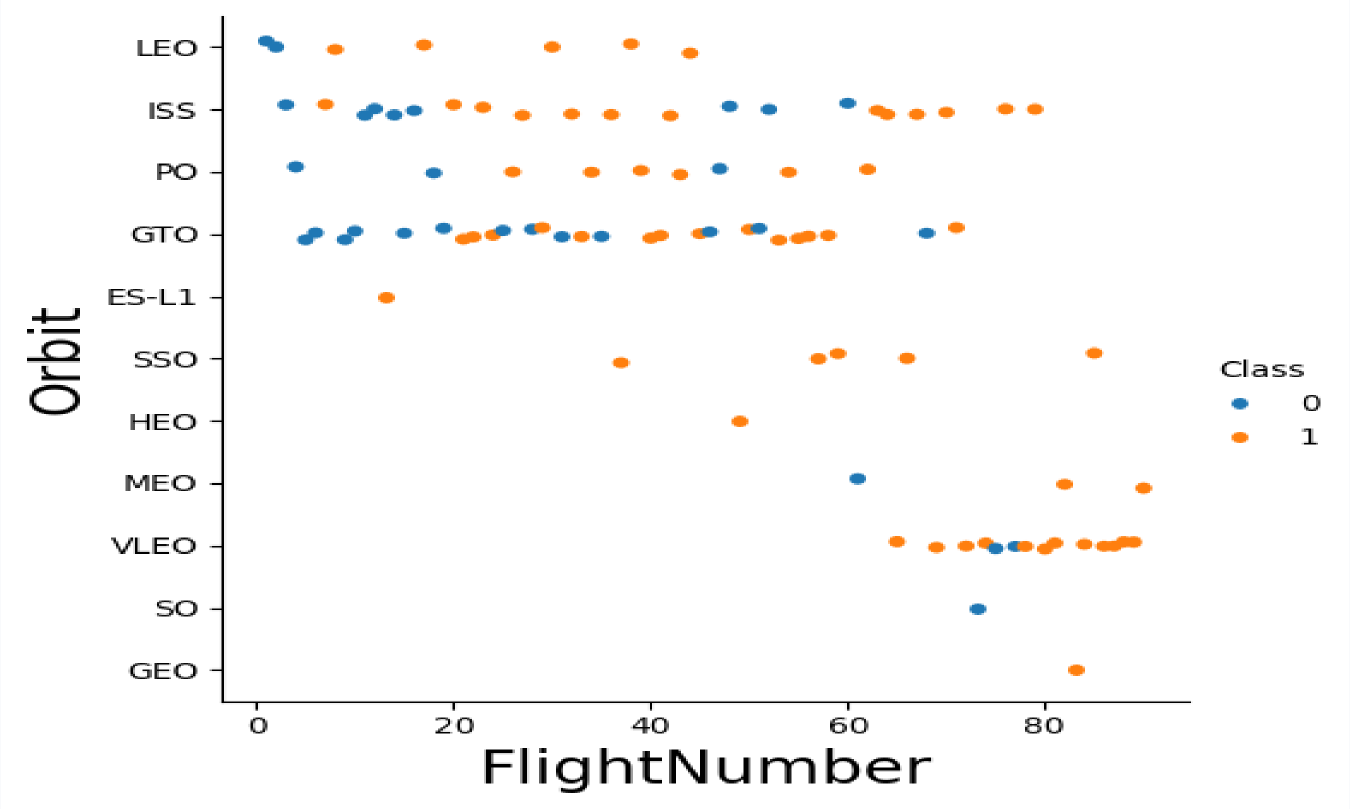
1) Greater payload masses yield more frequent successful landings.

Success Rate vs. Orbit Type



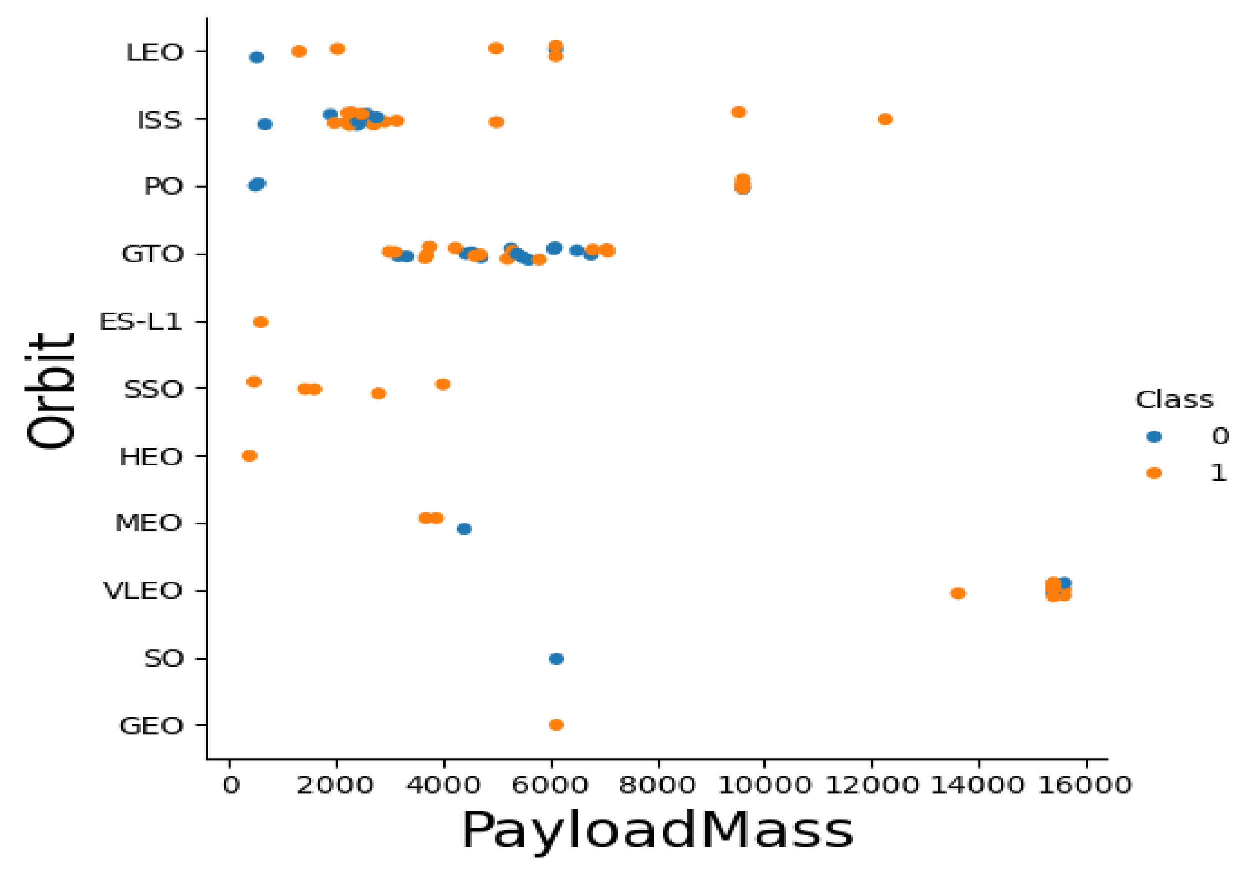
1) Launches to orbits ES-L1, GEO, HEO, and SSO have the most successful landing outcomes with the mean class = 1 each, showing all landings were successful at these orbit types.

Flight Number vs. Orbit Type



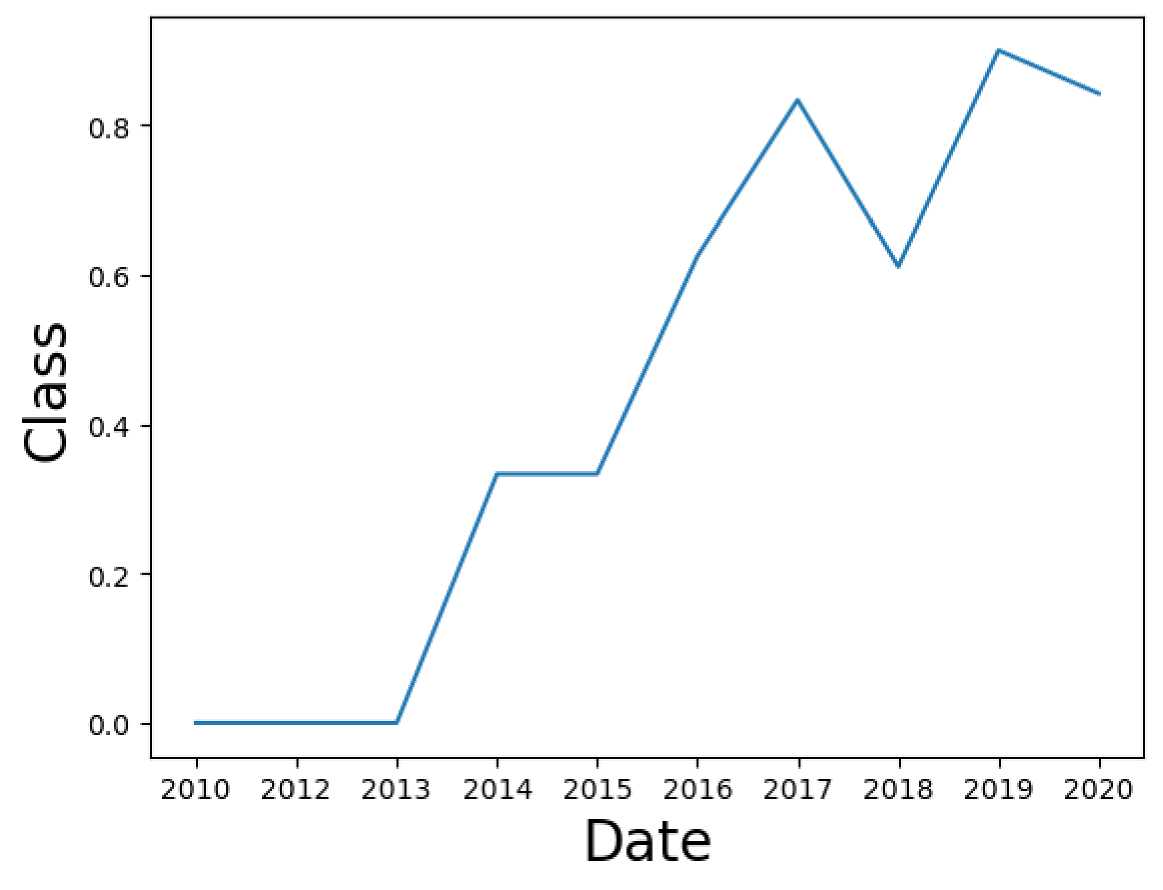
1) Flights to orbit VLEO became more frequent with flight number, and were also more successful. Additionally, LEO orbit flights had a high success rate.

Payload vs. Orbit Type



1) Heavier payloads were more frequent at VLEO orbit. Orbits PO and ISS also had some heavier payloads. Heavier payloads also had more success.

Launch Success Yearly Trend



1) Launch success improved as the years passed, with a dip in 2018 before peaking in 2019.

All Launch Site Names

Task 1

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

```
%sql Select distinct Launch_Site from SPACEXTBL
```

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

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

1) SpaceX have used 4 launch sites, query was executed using ‘distinct’ to return unique cases.

Launch Site Names Begin with 'CCA'

Task 2

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

```
%sql Select * from SPACEXTBL where Launch_Site like 'CCA%' limit 5
```

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

Date	Time (UTC)	Booster_Version	Launch_Site	Payload	PAYLOAD_MASS__KG_	Orbit	Customer	Mission_Outcome	Landing_Outcome
2010-06-04	18:45:00	F9 v1.0 B0003	CCAFS LC-40	Dragon Spacecraft Qualification Unit	0	LEO	SpaceX	Success	Failure (parachute)
2010-12-08	15:43:00	F9 v1.0 B0004	CCAFS LC-40	Dragon demo flight C1, two CubeSats, barrel of Brouere cheese	0	LEO (ISS)	NASA (COTS) NRO	Success	Failure (parachute)
2012-05-22	7:44:00	F9 v1.0 B0005	CCAFS LC-40	Dragon demo flight C2	525	LEO (ISS)	NASA (COTS)	Success	No attempt
2012-10-08	0:35:00	F9 v1.0 B0006	CCAFS LC-40	SpaceX CRS-1	500	LEO (ISS)	NASA (CRS)	Success	No attempt
2013-03-01	15:10:00	F9 v1.0 B0007	CCAFS LC-40	SpaceX CRS-2	677	LEO (ISS)	NASA (CRS)	Success	No attempt

1) Query above shows five records from the database where launch sites begin with 'CCA', using a 'where' clause and limiting to 5 rows.

Total Payload Mass

Task 3

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

```
%sql Select Sum(PAYLOAD_MASS__KG_) from SPACEXTBL where Customer = 'NASA (CRS)'
```

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

Sum(PAYLOAD_MASS__KG_)
45596

1) Total payload mass carried by boosters launched by NASA (CRS), using a ‘where’ clause and the sum() function.

Average Payload Mass by F9 v1.1

Task 4

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

```
%sql Select AVG(PAYLOAD_MASS__KG_) from SPACEXTBL where Booster_Version = 'F9 v1.1'
```

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

AVG(PAYLOAD_MASS__KG_)
2928.4

1) Query showing the average payload mass carried by booster version F9 v1.1 via an AVG() function and 'where' clause.

First Successful Ground Landing Date

Task 5

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

Hint:Use min function

```
%sql Select min(date) from SPACEXTBL where landing_outcome like 'Success%'
```

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

min(date)
2015-12-22

1) Query using the min() function and a 'where' clause to return the date of the first successful landing outcome on the ground pad, which was 22nd December 2015.

Successful Drone Ship Landing with Payload between 4000 and 6000

Task 6

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

In [14]:

```
%%sql
Select distinct Booster_Version
from SPACEXTBL
where landing_outcome = 'Success (drone ship)'
and PAYLOAD_MASS__KG_ between 4000 and 6000
```

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

Out[14]:

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

1) Query using a where clause to return the names of boosters which have successfully landed on the drone ship, and had a payload mass greater than 4000 but less than 6000.

Total Number of Successful and Failure Mission Outcomes

Task 7

List the total number of successful and failure mission outcomes

```
%%sql
Select Mission_Outcome,
Count (*) as No_Mission_outcomes
from SPACEXTBL
group by Mission_Outcome
```

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

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

1) Query to return the number of successful and failed missing outcomes, using a count() function and group by.

Boosters Carried Maximum Payload

Task 8

List the names of the booster_versions which have carried the maximum payload m

```
%%sql
Select distinct Booster_Version
from SPACEXTBL
where PAYLOAD_MASS__KG_ = (Select max(PAYLOAD_MASS__KG_) from SPACEXTBL)
```

```
* 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

1) Query returns the boosters that have carried the maximum payload using a subquery in the 'where' clause and the max() function.

2015 Launch Records

Task 9

List the records which will display the month names, failure landing_outcomes in drone ship ,
months in year 2015.

Note: SQLLite does not support monthnames. So you need to use substr(Date, 6,2) as mo

substr(Date,0,5)='2015' for year.

```
%%sql
Select substr(Date, 6,2) as Month, landing_outcome, Booster_Version, Launch_site
from SPACEXTBL
where landing_outcome = 'Failure (drone ship)'
and substr(Date,0,5)='2015'
```

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

Month	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

1) List of the failed landing outcomes on the drone ship, their booster versions, and launch site names in the year 2015. Query uses substr() function and 'where' clause.

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

Task 10

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.

```
%%sql
Select  landing_outcome,
Count (*) as landing_outcomes
from SPACEXTBL
where Date between '2010-06-04' and '2017-03-20'
Group by landing_outcome
order by count(*) desc
```

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

Landing_Outcome	landing_outcomes
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

1) Query returns the landing outcomes in order of occurrence, between 2010-06-04 and 2017-03-20. Query uses the count() function, a group by, a ‘where’ clause, and an ‘order by’.

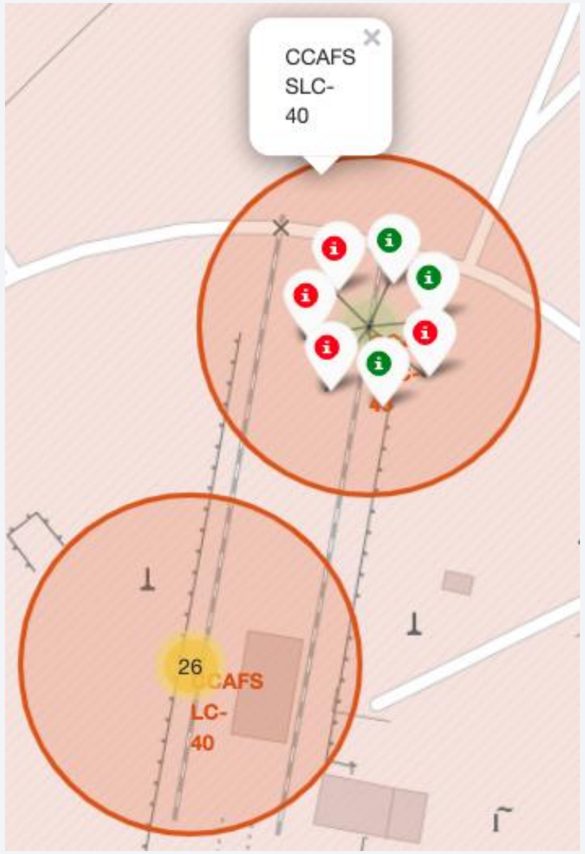
Section 3

Launch Sites Proximities Analysis

Folium Map of Launch Sites

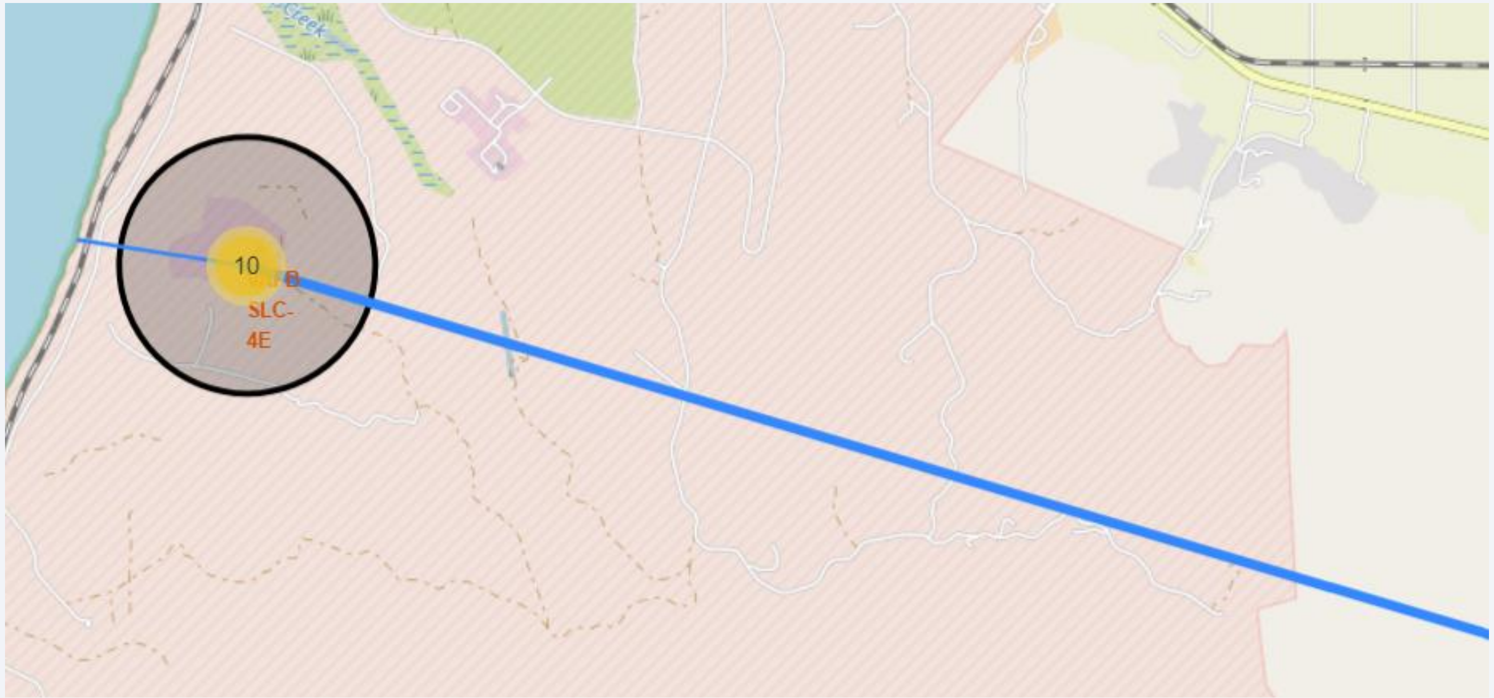
1) Adding Folium Circles and Markers for SpaceX launch sites.

Launch Site Success using Folium Marker Clusters



1) Adding marker clusters to identify which launch sites have relatively high success rates. Green markers show successful flights, Red shows the unsuccessful flights.

Folium Map Polyline to Local Landmarks

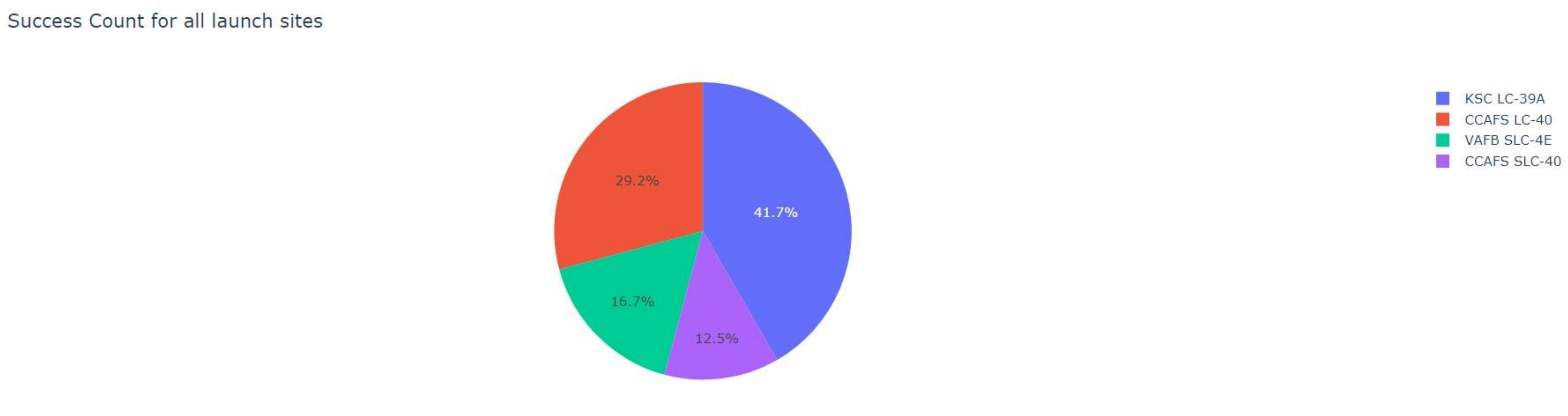


1) Folium Map Polyline to local landmarks, such as a nearby coastline (West) and nearby city (South East).

Section 4

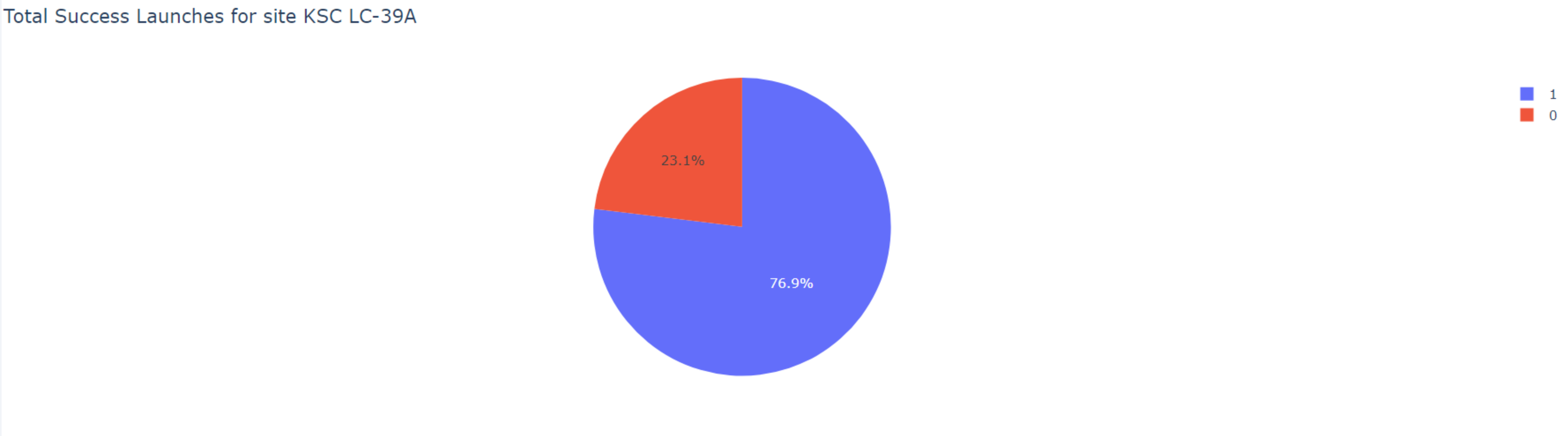
Build a Dashboard with Plotly Dash

Interactive Dashboard: Launch Success Pie Chart 1



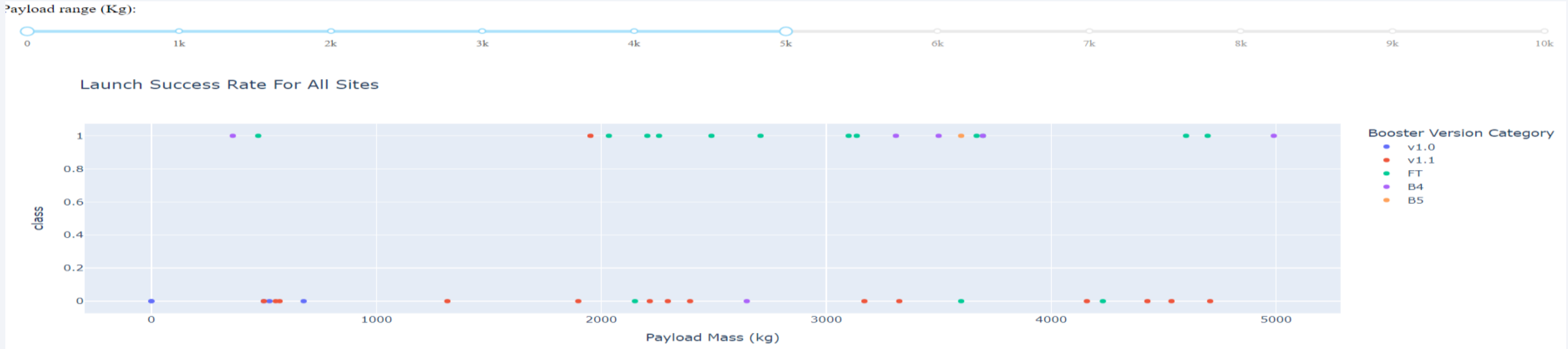
1) The Pie Chart above shows the success count for all the launch sites. It is evident that launch site KSC LC-39A was the most successful with 41.7% of the total successful flights.

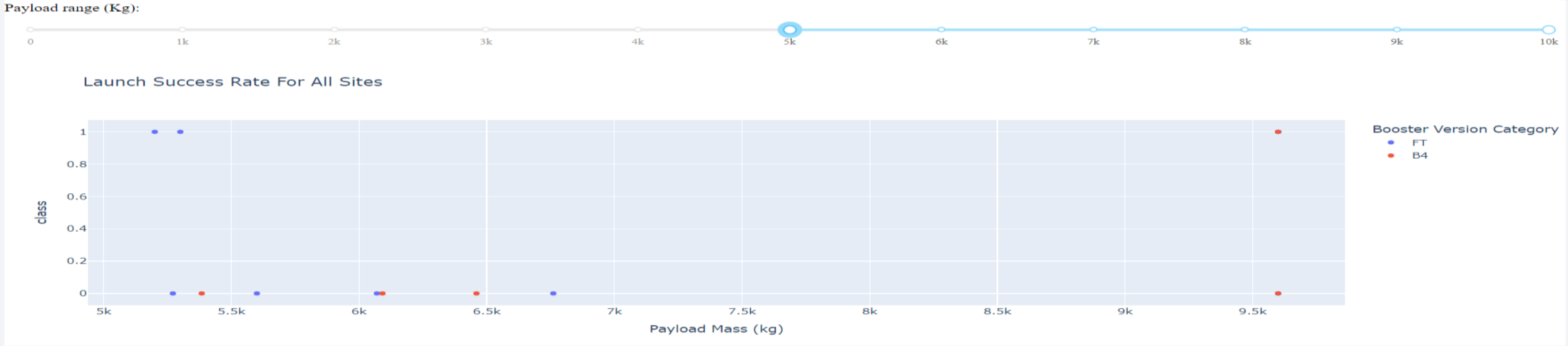
Interactive Dashboard: Launch Success Pie Chart 2



1) The Pie Chart above is filtered onto the most successful launch site KSC LC-39A. We can see that 76.9% of the flights from this launch site were successful.

Interactive Dashboard: Payload vs Launch Outcome



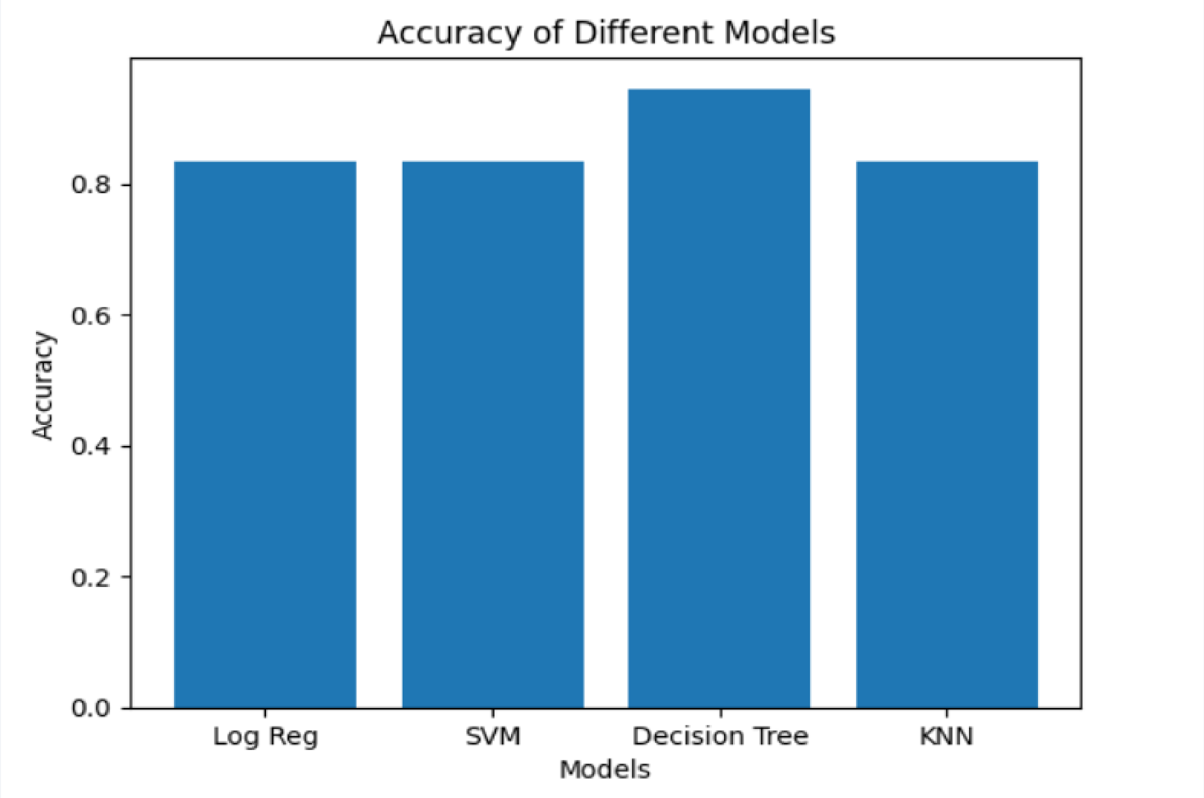


1) Using the filters on the Scatter Plots above, we can see that payloads < 5000kg had more success than the payloads > 5000kg. However, payloads < 5000kg were more frequent.

Section 5

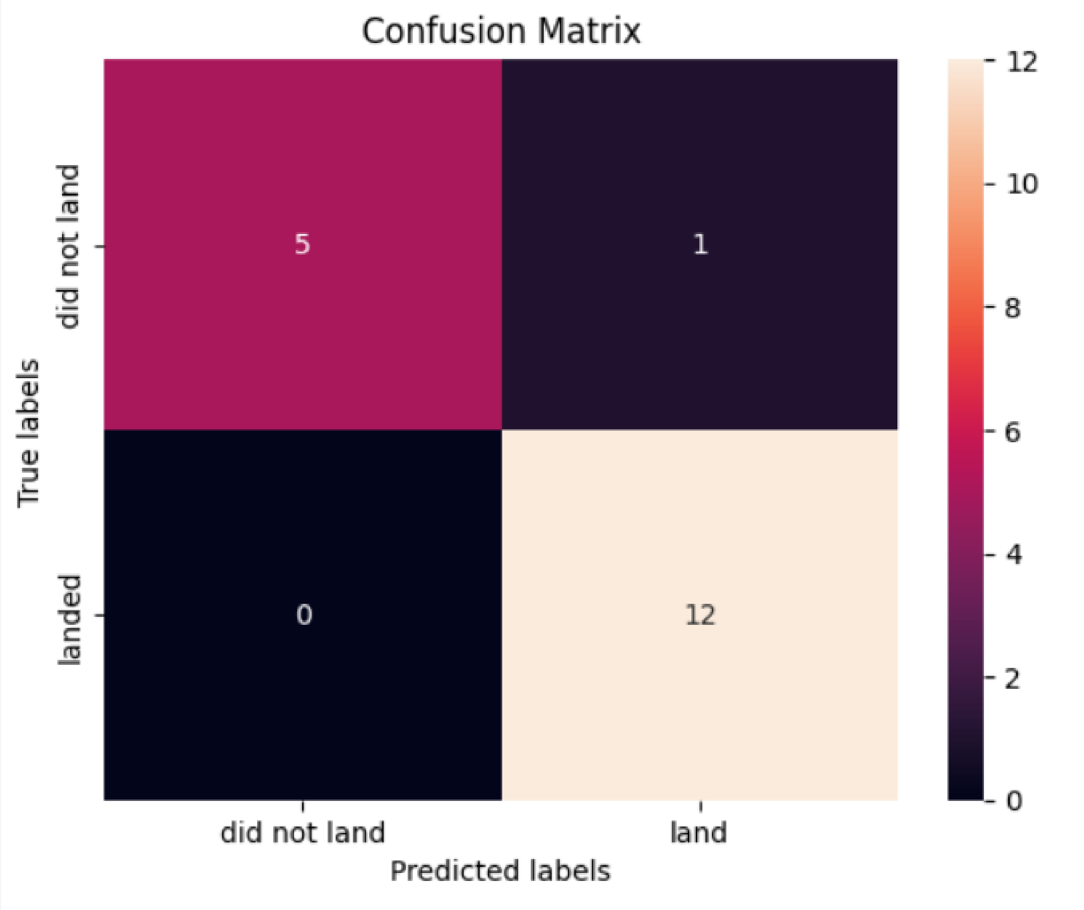
Predictive Analysis (Classification)

Classification Accuracy



1) Using MatPlotLib we can see that the Decision Tree method had the highest accuracy of roughly 0.9 and is therefore the best performing.

Confusion Matrix



1) The Confusion Matrix for the Decision Tree method shows that it only produced 1 false positive and no false negatives.

Conclusions

1) Machine Learning can be used to predict the success of a SpaceX Falcon9 Rocket flight.
2) The Decision Tree Method proved to be the most accurate, at roughly 90% accuracy.
3) Flight success has increased between the years 2013 and 2020, showing that space travel is getting closer to saving money through reusable rockets.
4) Orbit reached has shown to have a key relationship with flight success, with orbits ES-L1, GEO, HEO, and SSO having 100% success rate.
5) Launch site also has demonstrated a key relationship with flight success with launch site KSC LC-39A having the most successful launches of any site.

Appendix

- GitHub repository: https://github.com/andrewdoublard/IBM-Capstone
- SpaceX API: https://api.spacexdata.com/v4/launches/past
- Falcon9 and Falcon Heavy Wikipedia page: https://en.wikipedia.org/wiki/List_of_Falcon_9_and_Falcon_Heavy_launches

Thank you!