IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

Stavros Karakepelis
April 5, 2024

Outline

Executive Summary

- SpaceY is a new commercial rocket launch provider who wants to bid against SpaceX.
- SpaceX advertises launch services starting at $62 million for missions that allow some fuel to be reserved for landing the 1st stage rocket booster, so that it can be reused.
- SpaceX public statements indicate a 1st stage Falcon 9 booster to cost upwards of $15 million to build without including R&D cost recoupment or profit margin.
- Given mission parameters such as payload mass and desired orbit, the models produced in this report were able to predict the first stage rocket booster landing successfully with an accuracy level of 83.3%.
- As a result, SpaceY will be able to make more informed bids against SpaceX by using 1st stage landing predictions as a proxy for the cost of a launch.

Introduction

- SpaceX advertises Falcon 9 rocket launches with a cost of 62 million dollars when the first stage of their rockets can be reused.
- The first stage is estimated to cost upwards of 15 million to build without including R&D cost recoupment or profit margin.
- Sometimes SpaceX will sacrifice the first stage due to mission parameters such as payload, orbit, and customer.
- Therefore, this report aims to accurately predict the likelihood of the first stage rocket landing successfully as a proxy for the cost of a launch.

Section 1

Methodology

Executive Summary

- Data collection
- Data wrangling
- Exploratory data analysis (EDA)
- Data visualization and analytics
- Predictive analysis using ML models
- Report results to stakeholders

Data Collection - SpaceX API

API method

- Acquired historical launch data from Open Source REST API for SpaceX
- Requested and parsed the SpaceX launch data using the GET request
- Filtered the dataframe to only include Falcon 9 launches
- Replaced missing payload mass values from classified missions with mean

Data Collection – SpaceX API

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

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

GitHub:
https://github.com/skarakepelis/edX_DS_Capstone_Project/blob/main/Module%201%20-%20Capstone%20Introduction%20and%20Understanding%20the%20Datasets/jupyter-labs-spacex-data-collection-api.ipynb

	FlightNumber	Date	BoosterVersion	PayloadMass	Orbit	LaunchSite	Outcome	Flights	GridFins	Reused	Legs
4	1	2010-06-04	Falcon 9	6123.547647	LEO	CCSFS SLC 40	None None	1	False	False	False
5	2	2012-05-22	Falcon 9	525.000000	LEO	CCSFS SLC 40	None None	1	False	False	False
6	3	2013-03-01	Falcon 9	677.000000	ISS	CCSFS SLC 40	None None	1	False	False	False
7	4	2013-09-29	Falcon 9	500.000000	PO	VAFB SLC 4E	False Ocean	1	False	False	False
8	5	2013-12-03	Falcon 9	3170.000000	GTO	CCSFS SLC 40	None None	1	False	False	False
...	...	...	...	...	...	...	...	...	...	...	...
89	86	2020-09-03	Falcon 9	15600.000000	VLEO	KSC LC 39A	True ASDS	2	True	True	True
90	87	2020-10-06	Falcon 9	15600.000000	VLEO	KSC LC 39A	True ASDS	3	True	True	True
91	88	2020-10-18	Falcon 9	15600.000000	VLEO	KSC LC 39A	True ASDS	6	True	True	True
92	89	2020-10-24	Falcon 9	15600.000000	VLEO	CCSFS SLC 40	True ASDS	3	True	True	True
93	90	2020-11-05	Falcon 9	3681.000000	MEO	CCSFS SLC 40	True ASDS	1	True	False	True

Data Collection – Web Scraping

Web Scraping

- Acquired historical launch data from Wikipedia page 'List of Falcon 9 and Falcon Heavy Launches' Requested the Falcon9 Launch Wiki page from its Wikipedia URL
- Extracted all column/variable names from the HTML table header
- Parsed the table and converted it into a Pandas data frame

Data Collection – Web Scraping

GitHub:

https://github.com/skarakepelis/edX_DS_Capstone_Project/blob/main/Module%201%20-%20Capstone%20Introduction%20and%20Understanding%20the%20Datasets/jupyter-labs-webscraping.ipynb

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

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

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

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

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>
```

```
# 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

Landing Outcomes
sample size = 90
□= Class 0
□= Class 1

True	ASDS	41
None	None	19
True	RTLS	14
False	ASDS	6
True	Ocean	5
None	ASDS	2
False	Ocean	2
False	RTLS	1

GitHub:
https://github.com/skarakepelis/edX_DS_Capstone_Project/blob/main/Module%201%20-%20Capstone%20Introduction%20and%20Understanding%20the%20Datasets/labs-jupyter-spacex-data_wrangling_jupyterlite.jupyterlite.ipynb

Data Wrangling

- Explored data to determine the label for training supervised models
  - Calculated the number of launches on each site
  - Calculated the number and occurrence of each orbit
  - Calculated the number and occurrence of mission outcome per orbit type
- Created a landing outcome training label from 'Outcome' column
  - Training label: 'Class'
  - Class = 0; first stage booster did not land successfully
    - None None; not attempted
    - None ASDS; unable to be attempted due to launch failure
    - False ASDS; drone ship landing failed
    - False Ocean; ocean landing failed
    - False RTLS; ground pad landing failed
  - Class = 1; first stage booster landed successfully
    - True ASDS; drone ship landing succeeded
    - True RTLS; ground pad landing succeeded
    - True Ocean; ocean landing succeeded

EDA with SQL

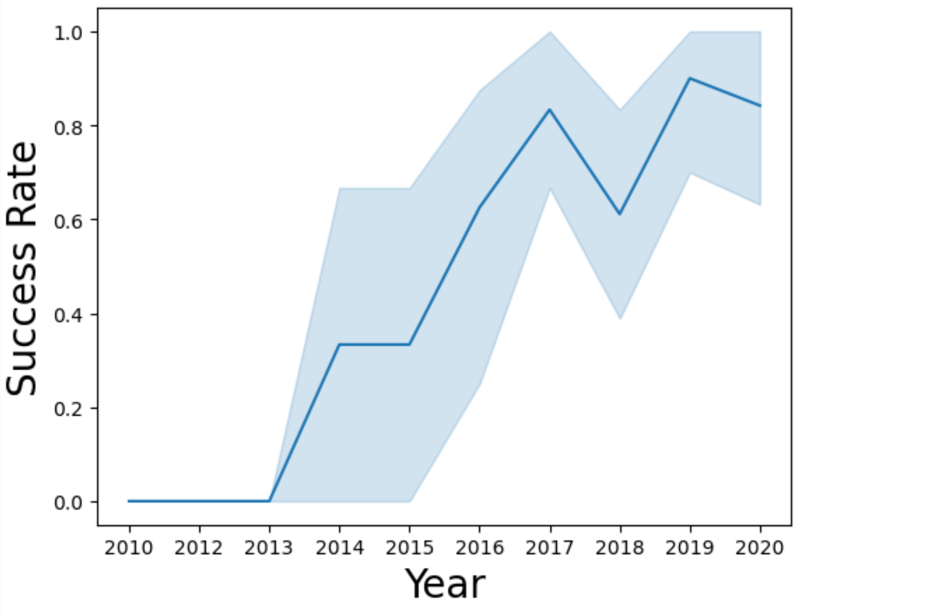
- Loaded data into a db instance called “my_data1.db”
- Ran SQL queries to display and list information about
  - Launch sites
  - Payload masses
  - Booster versions
  - Mission outcomes
  - Booster landings

- GitHub: https://github.com/skarakepelis/edX_DS_Capstone_Project/blob/main/Module%202%20-%20Exploratory%20Data%20Analysis%20(EDA)/jupyter-labs-eda-sql-edx_sqllite.ipynb

EDA with Data Visualization

- Read the dataset into a Pandas dataframe
- Used Matplotlib and Seaborn visualization libraries to plot
  - FlightNumber x PayloadMass
  - FlightNumber x LaunchSite
  - Payload x LaunchSite
  - Orbit type x Success rate
  - FlightNumber x Orbit type
  - Payload x Orbit type
  - Year x Success Rate

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



GitHub: https://github.com/skarakepelis/edX_DS_Capstone_Project/blob/main/Module%202%20-%20Exploratory%20Data%20Analysis%20(EDA)/jupyter-labs-eda-dataviz.ipynb.jupyterlite.ipynb

Build an Interactive Map with Folium

- Used Python interactive mapping library called Folium
- Marked all launch sites on a map
- Marked the successful/failed launches for each site on map
- Calculated the distances between a launch site to its proximities
    - Railways
    - Highways
    - Coastlines
    - Cities

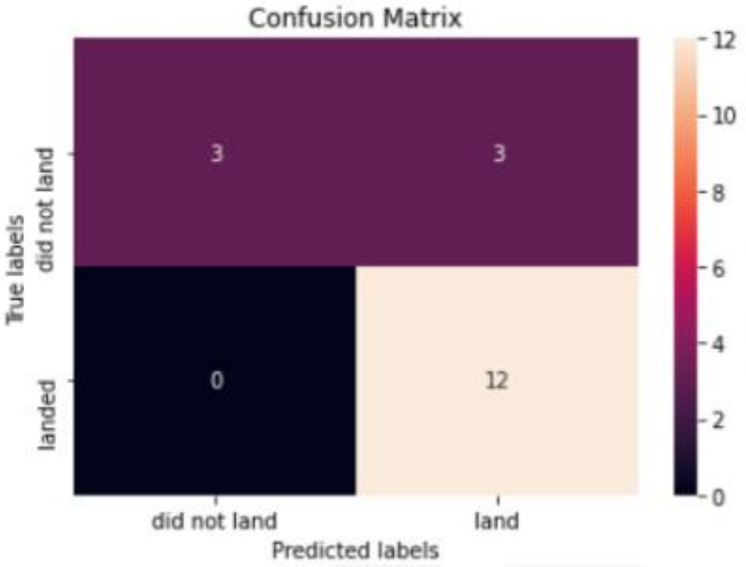
GitHub: https://github.com/skarakepelis/edX_DS_Capstone_Project/blob/main/Module%203%20-%20Interactive%20Visual%20Analytics%20and%20Dashboard/lab_jupyter_launch_site_location.jupyterlite.ipynb

Build a Dashboard with Plotly Dash

- Used Python interactive dashboarding library called Plotly Dash to enable stakeholders to explore and manipulate data in an interactive and real-time way
- Pie chart showing success rate
  - Color coded by launch site
- Scatter chart showing payload mass vs. landing outcome
  - Color coded by booster version
  - With range slider for limiting payload amount
- Drop-down menu to choose between all sites and individual launch sites

- GitHub: https://github.com/skarakepelis/edX_DS_Capstone_Project/blob/main/Module%203%20-%20Interactive%20Visual%20Analytics%20and%20Dashboard/spacex_dash_app.py

Predictive Analysis (Classification)



Confusion matrix of logistic regression model, showing 15 correct predictions and 3 false positives

- Predictive Analysis (Model development)
  - Imported libraries and defined function to create confusion matrix
    - Pandas
    - Numpy
    - Matplotlib
    - Seaborn
    - Sklearn
  - Loaded the dataframe created during data collection
  - Created a column for our training label 'Class' created during data wrangling
  - Standardized the data
  - Split the data into training data and test data
  - Fit the training data to various model types
    - Logistic Regression
    - Support Vector Machine
    - Decision Tree Classifier
    - K Nearest Neighbors Classifier
  - Used a cross-validated grid-search over a variety of hyperparameters to select the best ones for each model
    - Enabled by Scikit-learn library function GridSearchCV
  - Evaluated accuracy of each model using test data to select the best model

Results

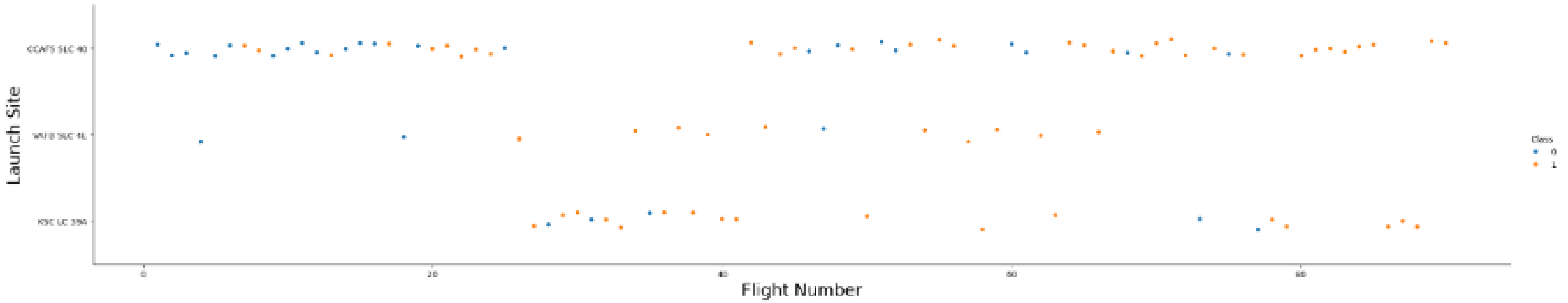
- Exploratory data analysis results
- Interactive analytics demo in screenshots
- Predictive analysis results

Section 2

Insights drawn from EDA

Flight Number vs. Launch Site

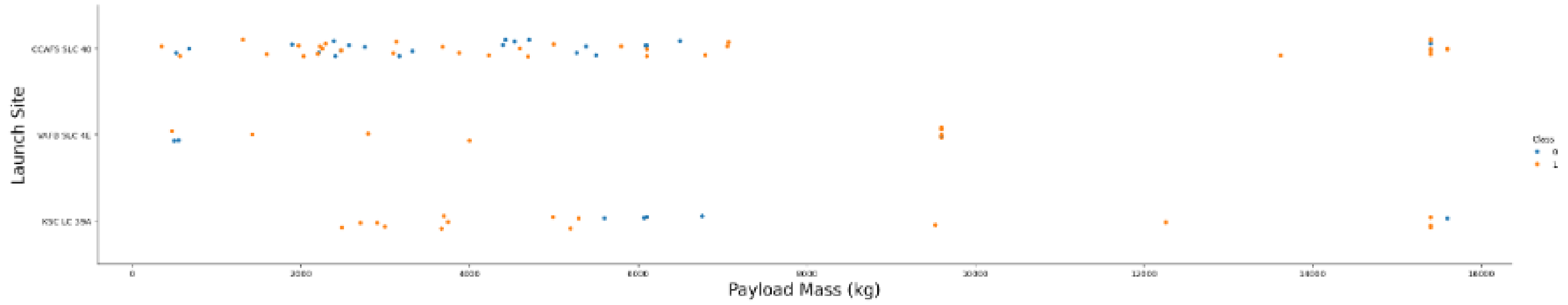
```
In [5]:
# Plot a scatter point chart with x axis to be Flight Number and y axis to be the launch site, and hue to be the class va
sns.catplot(y="LaunchSite", x="FlightNumber", hue="Class", data=df, aspect = 5)
plt.xlabel("Flight Number",fontsize=20)
plt.ylabel("Launch Site",fontsize=20)
plt.show()
```



Payload vs. Launch Site

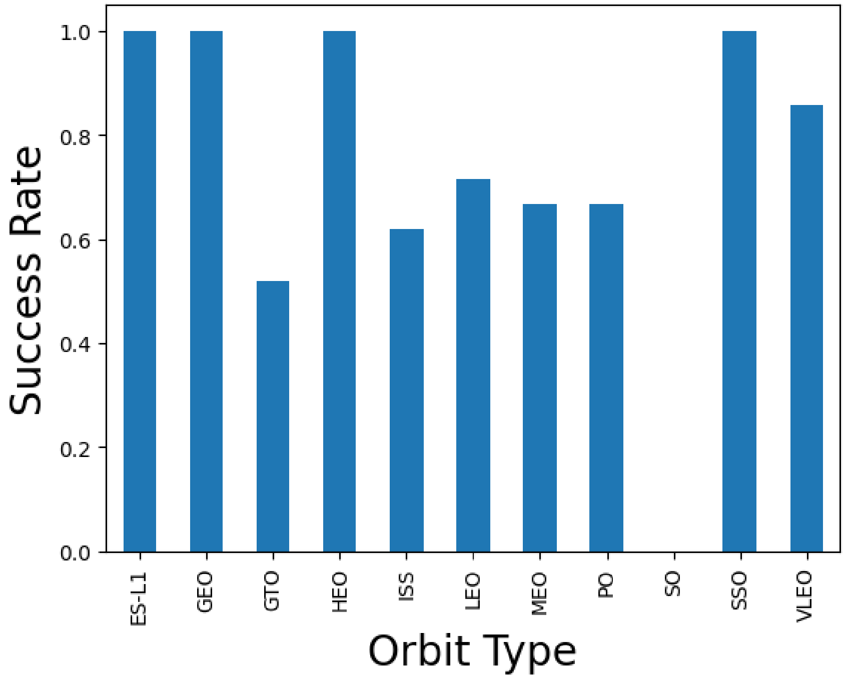
In [8]:

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



Success Rate vs. Orbit Type

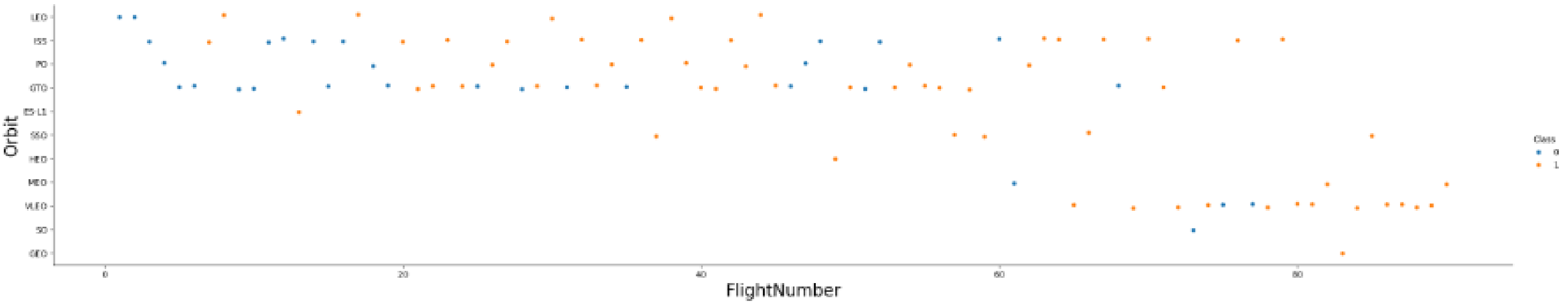
```
In [12]: # HINT use groupby method on Orbit column and get the mean of Class column
df.groupby("Orbit").mean(numeric_only=True)['Class'].plot(kind='bar')
plt.xlabel("Orbit Type",fontsize=20)
plt.ylabel("Success Rate",fontsize=20)
plt.show()
```



Flight Number vs. Orbit Type

In [13]:

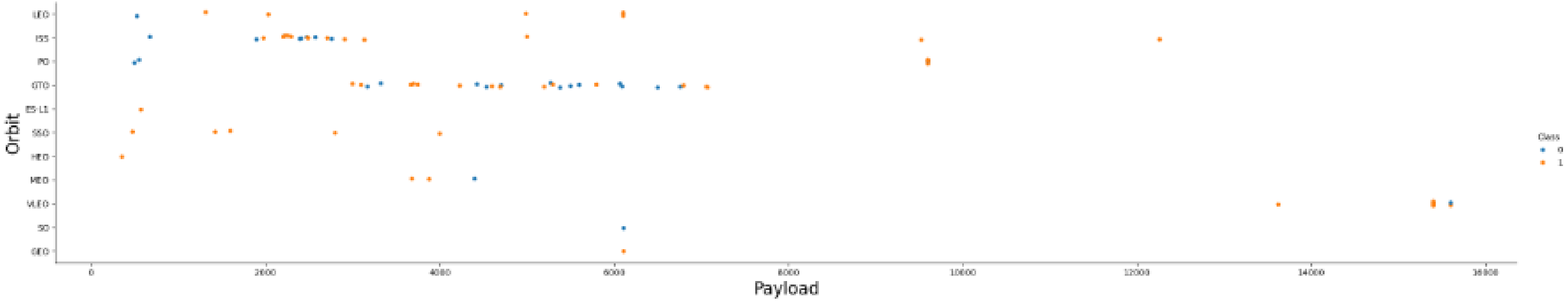
```python
# Plot a scatter point chart with x axis to be FlightNumber and y axis to be the Orbit, and hue to be the class value
sns.catplot(y="Orbit", x="FlightNumber", hue="Class", data=df, aspect = 5)
plt.xlabel("FlightNumber",fontsize=20)
plt.ylabel("Orbit",fontsize=20)
plt.show()
```



Payload vs. Orbit Type

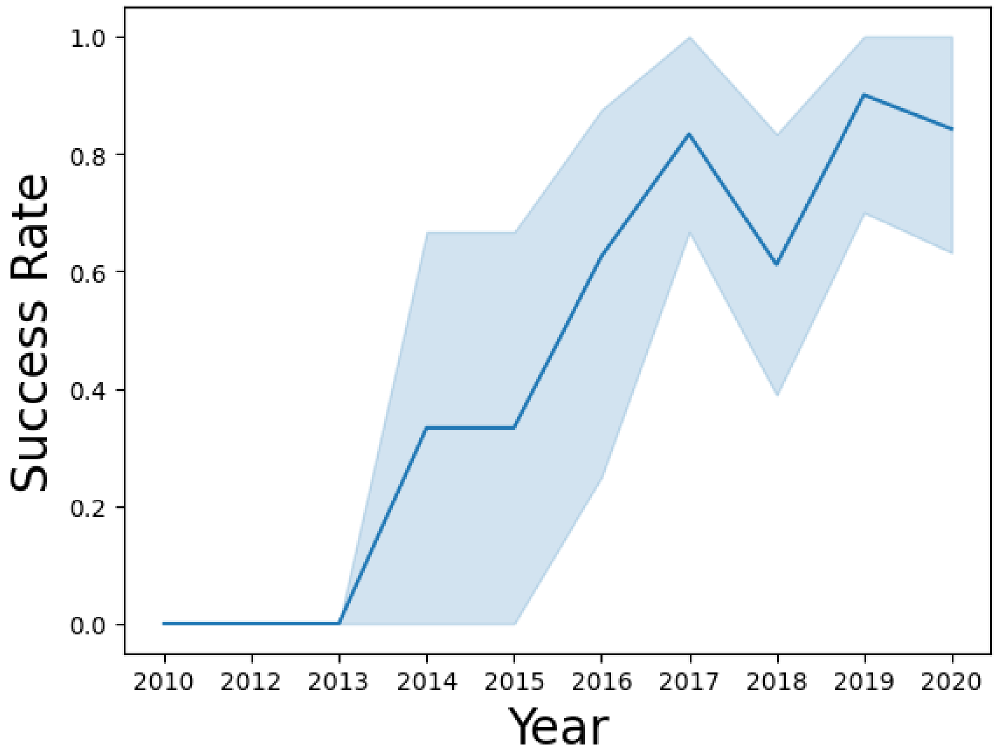
In [14]:

```
# Plot a scatter point chart with x axis to be Payload and y axis to be the Orbit, and hue to be the class value
sns.catplot(y="Orbit", x="PayloadMass", hue="Class", data=df, aspect = 5)
plt.xlabel("Payload",fontsize=20)
plt.ylabel("Orbit",fontsize=20)
plt.show()
```



Launch Success Yearly Trend

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



All Launch Site Names

In [5]:
```
%%sql SELECT DISTINCT LAUNCH_SITE
FROM SPACEXTABLE;
```

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

Out[5]:

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

Launch Site Names Begin with 'KSC'

In [6]:

```
%%sql
SELECT *
FROM SPACEXTBL
WHERE LAUNCH_SITE LIKE 'KSC%'
LIMIT 5;
```

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

Out[6]:

Date	Time (UTC)	Booster_Version	Launch_Site	Payload	PAYLOAD_MASS__KG_	Orbit	Customer	Mission_Outcome	Landing_Outcome
2017-02-19	14:39:00	F9 FT B1031.1	KSC LC-39A	SpaceX CRS-10	2490	LEO (ISS)	NASA (CRS)	Success	Success (ground pad)
2017-03-16	6:00:00	F9 FT B1030	KSC LC-39A	EchoStar 23	5600	GTO	EchoStar	Success	No attempt
2017-03-30	22:27:00	F9 FT B1021.2	KSC LC-39A	SES-10	5300	GTO	SES	Success	Success (drone ship)
2017-05-01	11:15:00	F9 FT B1032.1	KSC LC-39A	NROL-76	5300	LEO	NRO	Success	Success (ground pad)
2017-05-15	23:21:00	F9 FT B1034	KSC LC-39A	Inmarsat-5 F4	6070	GTO	Inmarsat	Success	No attempt

Total Payload Mass

In [7]:
```
%%sql
SELECT SUM(PAYLOAD_MASS__KG_)
FROM SPACEXTABLE
WHERE Customer = 'NASA (CRS)';
```

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

Out[7]:

SUM(PAYLOAD_MASS__KG_)
45596

Average Payload Mass by F9 v1.1

```
In [8]:  %%sql
         SELECT AVG(PAYLOAD_MASS__KG_)
         FROM SPACEXTBL
         WHERE Booster_Version LIKE 'F9 v1.1%';

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

Out[8]:

AVG(PAYLOAD_MASS__KG_)
2534.6666666666665

First Successful Ground Landing Date

In [9]:
```
%%sql
SELECT MIN(Date)
FROM SPACEXTABLE
WHERE Landing_Outcome = 'Success (ground pad)';
```

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

Out[9]:

MIN(Date)
2015-12-22

Successful Drone Ship Landing with Payload between 4000 and 6000

In [10]:
```
%%sql
SELECT BOOSTER_VERSION
FROM SPACEXTABLE
WHERE LANDING_OUTCOME = 'Success (drone ship)'
    AND 4000 < PAYLOAD_MASS__KG_ < 6000;
```

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

Out[10]:

Booster_Version
F9 FT B1021.1
F9 FT B1022
F9 FT B1023.1
F9 FT B1026
F9 FT B1029.1
F9 FT B1021.2
F9 FT B1029.2
F9 FT B1036.1
F9 FT B1038.1
F9 B4 B1041.1
F9 FT B1031.2
F9 B4 B1042.1
F9 B4 B1045.1
F9 B5 B1046.1

Total Number of Successful and Failure Mission Outcomes

In [11]:

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

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

Out[11]:

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

Boosters Carried Maximum Payload

```
%%sql
SELECT DISTINCT BOOSTER_VERSION
FROM SPACEXTABLE
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

2017 Launch Records

```sql
%%sql
SELECT
    CASE substr(DATE, 6, 2)
        WHEN '01' THEN 'January'
        WHEN '02' THEN 'February'
        WHEN '03' THEN 'March'
        WHEN '04' THEN 'April'
        WHEN '05' THEN 'May'
        WHEN '06' THEN 'June'
        WHEN '07' THEN 'July'
        WHEN '08' THEN 'August'
        WHEN '09' THEN 'September'
        WHEN '10' THEN 'October'
        WHEN '11' THEN 'November'
        WHEN '12' THEN 'December'
        ELSE NULL
    END AS Month,
    LANDING_OUTCOME,
    BOOSTER_VERSION,
    LAUNCH_SITE
FROM
    SPACEXTABLE
WHERE
    substr(DATE, 0, 5) = '2017' AND LANDING_OUTCOME = 'Success (ground pad)';
```

Month	Landing_Outcome	Booster_Version	Launch_Site
February	Success (ground pad)	F9 FT B1031.1	KSC LC-39A
May	Success (ground pad)	F9 FT B1032.1	KSC LC-39A
June	Success (ground pad)	F9 FT B1035.1	KSC LC-39A
August	Success (ground pad)	F9 B4 B1039.1	KSC LC-39A
September	Success (ground pad)	F9 B4 B1040.1	KSC LC-39A
December	Success (ground pad)	F9 FT B1035.2	CCAFS SLC-40

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

In [49]:

```
%%sql
SELECT LANDING_OUTCOME, COUNT(LANDING_OUTCOME) AS TOTAL_NUMBER
FROM SPACEXTABLE
WHERE DATE BETWEEN '2010-06-04' AND '2017-03-20'
GROUP BY LANDING_OUTCOME
ORDER BY TOTAL_NUMBER DESC
```

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

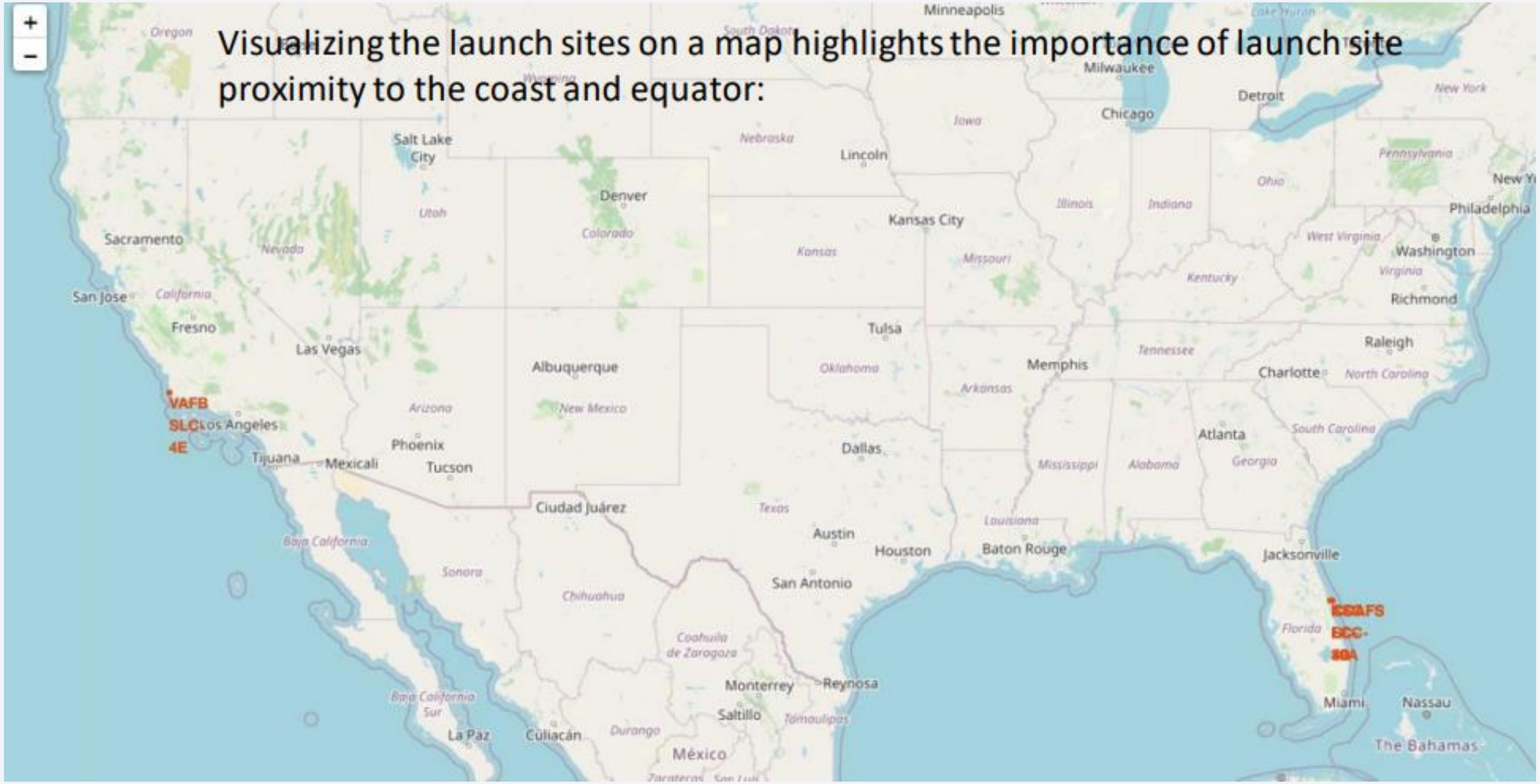
Out[49]:

Landing_Outcome	TOTAL_NUMBER
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

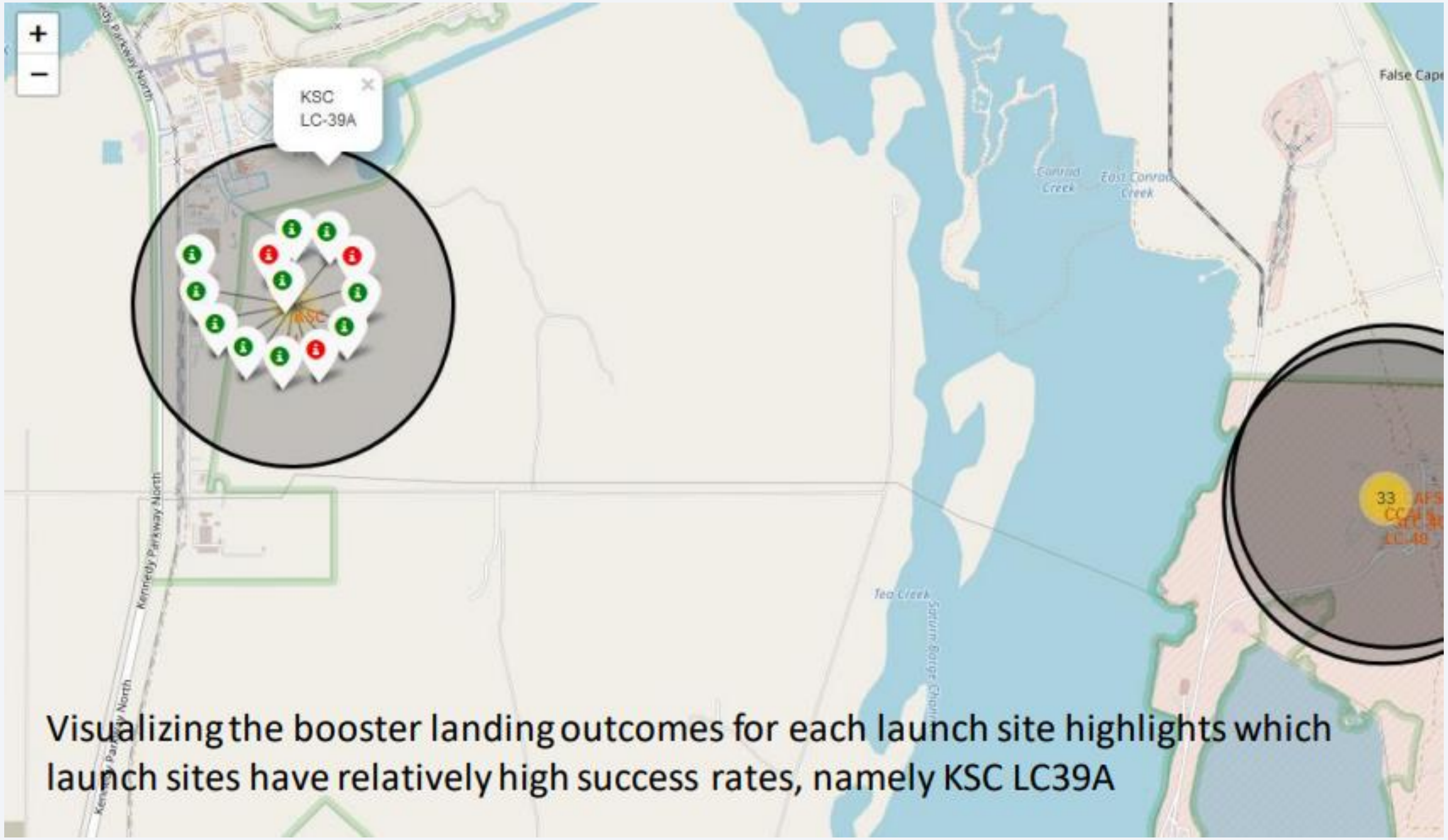
Section 3

Launch Sites Proximities Analysis

Launch Site Location Analysis (1/3)



Launch Site Location Analysis (2/3)



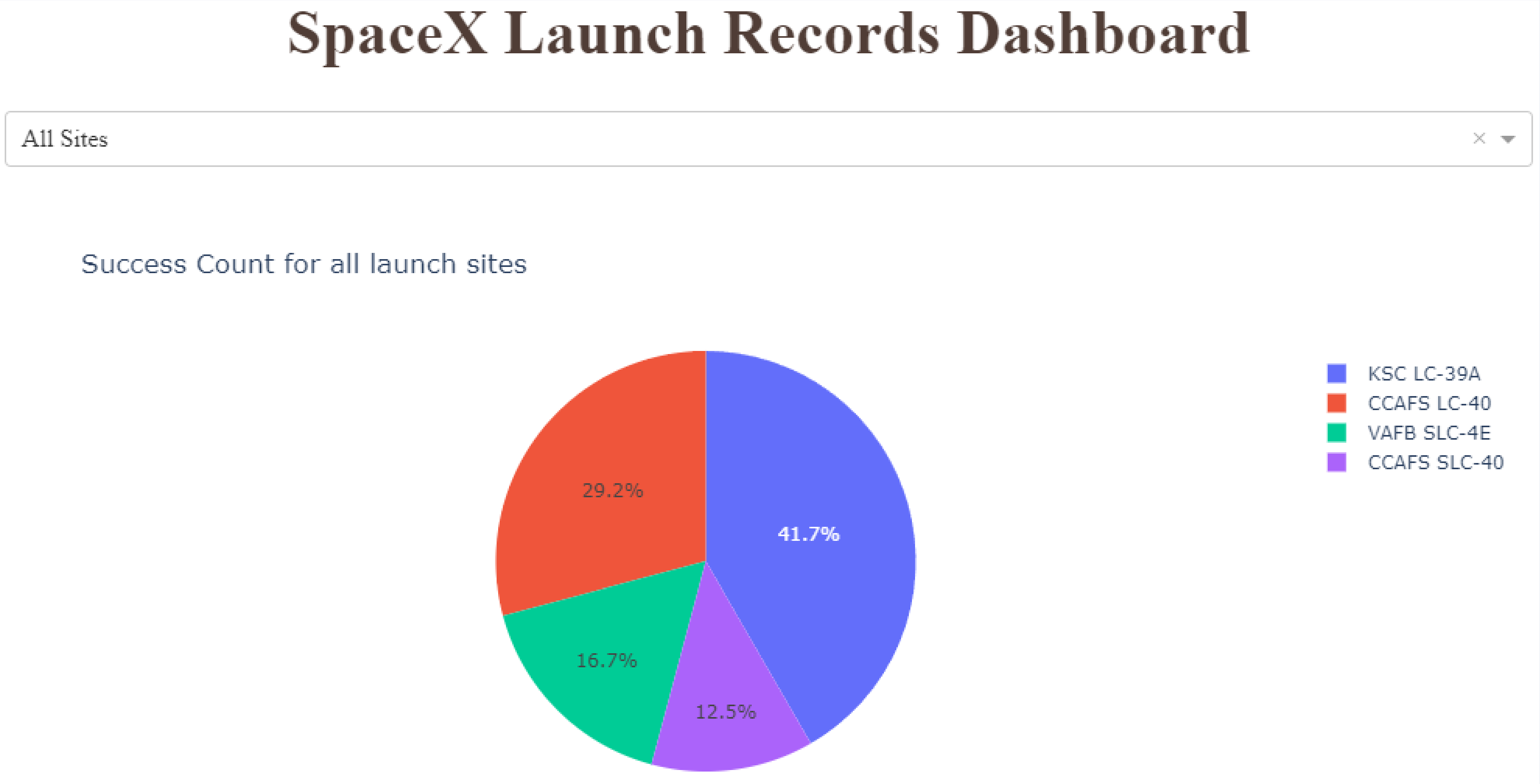
Visualizing the booster landing outcomes for each launch site highlights which launch sites have relatively high success rates, namely KSC LC39A

Launch Site Location Analysis (3/3)

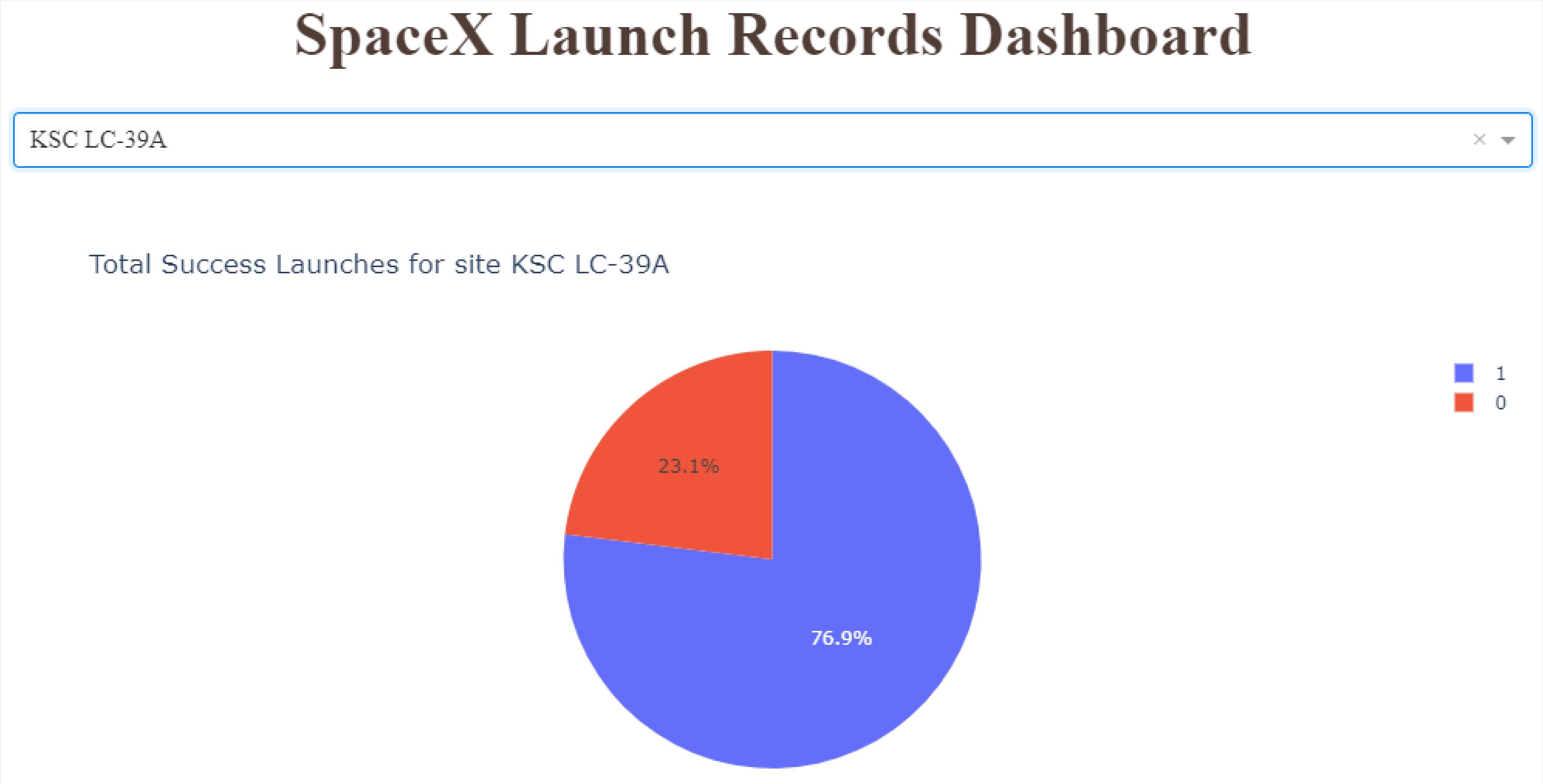
Section 4

Build a Dashboard with Plotly Dash

SpaceX Launch Records Dashboard – All Sites



SpaceX Launch Records Dashboard – KSC LC-39A

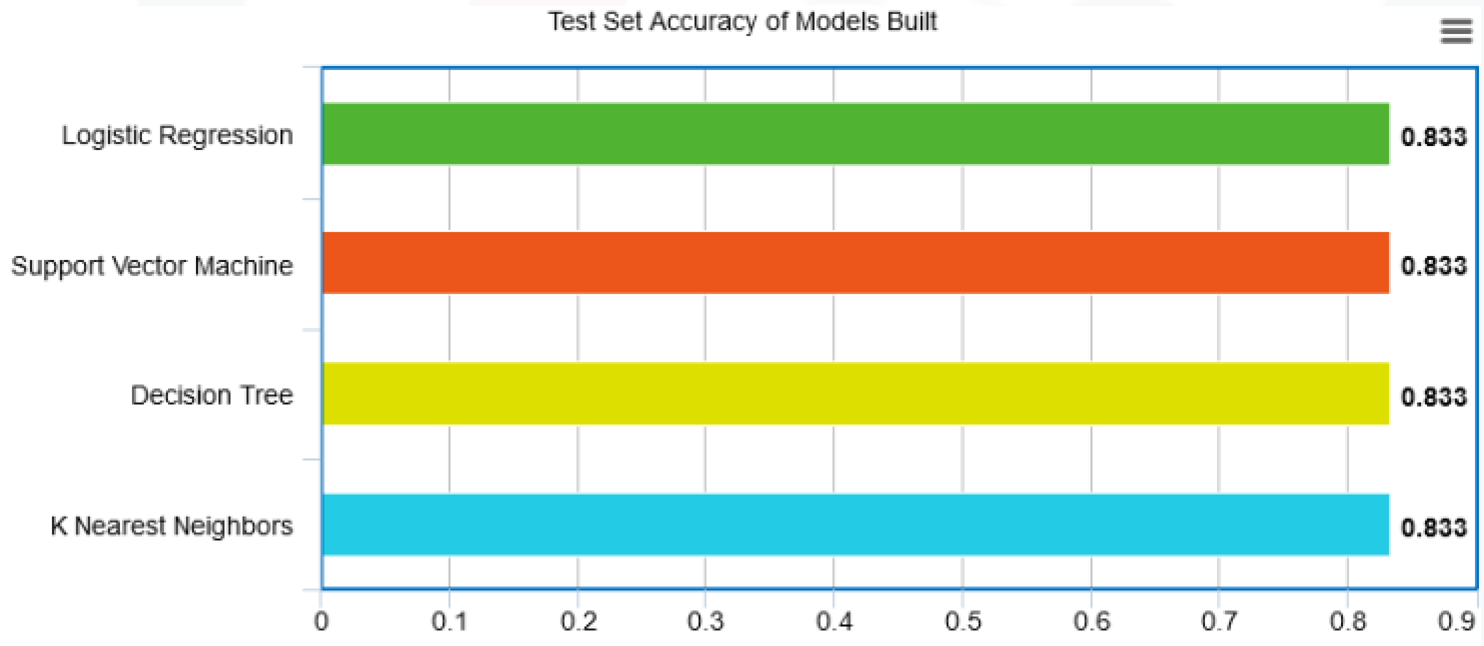


SpaceX Launch Records Dashboard – Payload vs Launch Outcome

Section 5

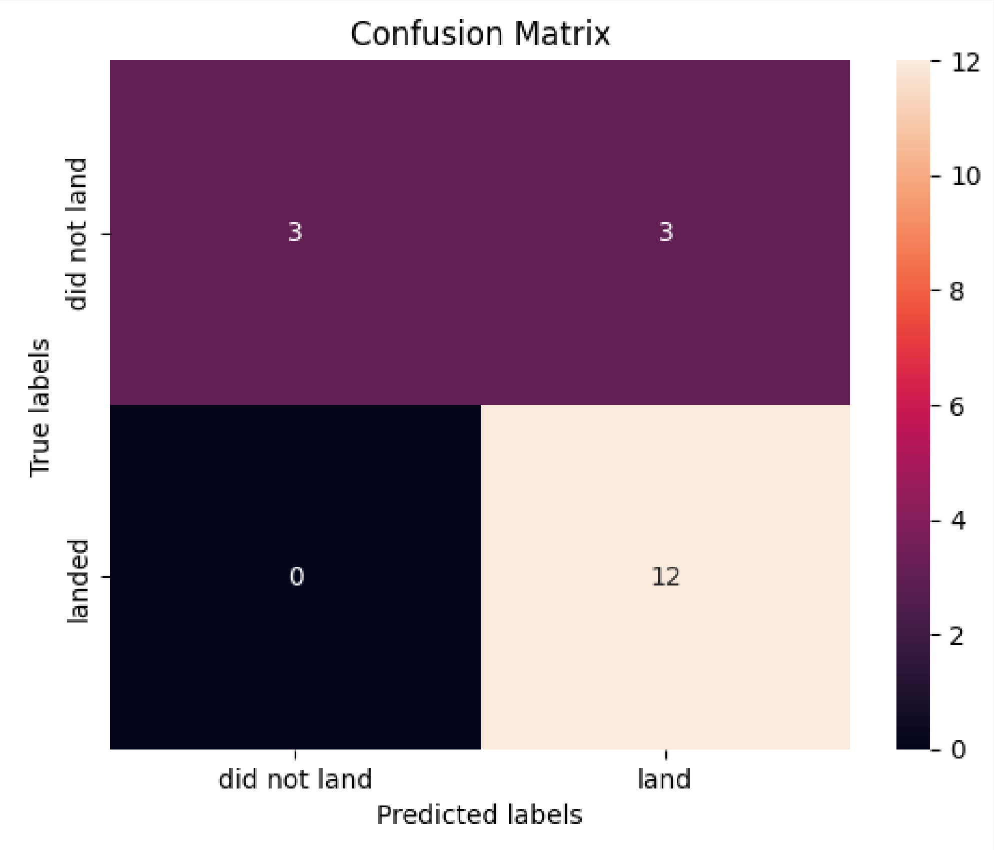
Predictive Analysis (Classification)

Classification Accuracy



- Each of the four models built came back with the same accuracy score, 83.33%

Confusion Matrix



- Pretty much all the models return an accuracy score of 83.33%
- The major problem is false positives as evidenced by the models, incorrectly predicting the first stage booster to land in 3/18 samples in the test set

Conclusions

- Using the models from this report SpaceY can predict when SpaceX will successfully land the 1st stage booster with 83.3% accuracy
- SpaceX public statements indicate the 1st stage booster costs upwards of $15 million to build
- This will enable SpaceY to make more informed bids against SpaceX, since they will have a good idea when to expect the SpaceX bid to include the cost of a sacrificed 1st stage booster
- With a list price of $62 million per launch, sacrificing the $15+ million 1st stage, would put the SpaceX bid at upwards of $77 million
- Biggest opportunities going forward to make even more informed bids:
  - Freeze the best performing combination of model and hyperparameters and re-fit using the whole dataset instead of just the training data
    - Potentially better than using only part of the data to fit the model, but you would no longer be able to measure the accuracy of the resulting model
  - Incorporate additional launch data to the dataset and model as it becomes available
  - Subdivide the current model into two models
    - Predict if SpaceX will ATTEMPT to land the 1st stage
    - Predict if SpaceX will SUCCEED in their attempt
  - Create a related model that predicts if SpaceX will launch using a previously-flown 1st stage booster
    - Would enable SpaceY to take into account when the SpaceX bid would likely include a discount

Appendix

- Notebooks to recreate dataset, analysis, and models:

Module 1 - Capstone Introduction and Understanding the Datasets

  - https://github.com/skarakepelis/edX_DS_Capstone_Project/blob/main/Module%201%20-%20Capstone%20Introduction%20and%20Understanding%20the%20Datasets/jupyter-labs-spacex-data-collection-api.ipynb
  - https://github.com/skarakepelis/edX_DS_Capstone_Project/blob/main/Module%201%20-%20Capstone%20Introduction%20and%20Understanding%20the%20Datasets/jupyter-labs-webscraping.ipynb
  - https://github.com/skarakepelis/edX_DS_Capstone_Project/blob/main/Module%201%20-%20Capstone%20Introduction%20and%20Understanding%20the%20Datasets/labs-jupyter-spacex-data_wrangling_jupyterlite.jupyterlite.ipynb

Module 2 - Exploratory Data Analysis (EDA)

  - https://github.com/skarakepelis/edX_DS_Capstone_Project/blob/main/Module%202%20-%20Exploratory%20Data%20Analysis%20(EDA)/jupyter-labs-eda-dataviz.ipynb.jupyterlite.ipynb
  - https://github.com/skarakepelis/edX_DS_Capstone_Project/blob/main/Module%202%20-%20Exploratory%20Data%20Analysis%20(EDA)/jupyter-labs-eda-sql-edx_sqllite.ipynb

Module 3 - Interactive Visual Analytics and Dashboard

  - https://github.com/skarakepelis/edX_DS_Capstone_Project/blob/main/Module%203%20-%20Interactive%20Visual%20Analytics%20and%20Dashboard/lab_jupyter_launch_site_location.jupyterlite.ipynb
  - https://github.com/skarakepelis/edX_DS_Capstone_Project/blob/main/Module%203%20-%20Interactive%20Visual%20Analytics%20and%20Dashboard/spacex_dash_app.py

Module 4 - Predictive Analysis (Classification)

  - https://github.com/skarakepelis/edX_DS_Capstone_Project/blob/main/Module%204%20-%20Predictive%20Analysis%20(Classification)/SpaceX_Machine_Learning_Prediction_Part_5.jupyterlite.ipynb

- Acknowledgments
  - Thank you to Joseph Santarcangelo at IBM for creating the course and materials
  - Thank you to Lakshmi Holla at IBM for the assistance provided

- References

  - https://aviationweek.com/defense-space/space/podcast-interview-spacexs-elon-musk (Interview with Elon Musk where he discloses the 1st stage booster to cost upwards of $15 million)
  - https://datascience.stackexchange.com/a/33050 (Explanation of why you would rebuild your model using the full dataset)
  - https://www.spacex.com/vehicles/falcon-9/ (Source of SpaceX's advertised $62 million launch price)

Thank you!