IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

Hector Santacruz
Noviembre 2024

Outline

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

Executive Summary

- Summary of methodologies
  - Data science approach collecting data, as much and relevant as a possible.
  - Statistical analysis and data visualization to better understanding of the results
  - Machine Learning to refine predictive models
- Summary of all results
  - Prediction of land of the Falcon 9 first stage
  - Cost of launch

Introduction

- Project background and context
  - 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
- Problems you want to find answers
  - The successful rate of Falcon 9 landings
  - The cost of the landing process

Section 1

Methodology

Methodology

Executive Summary

- Data collection methodology:
  - Data was collected by Space X API and web scrapping from Wikipedia.
- Perform data wrangling
  - One hot encoding to categorical features
- 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.
    - Get request to the Space X API
    - Encoding the content by a JSON function
    - Call and turn the JSON function into a pandas dataset
    - Transform the data by cleaning, check of the potential missing values
    - Web scrapping from Wikipedia by the help of beautiful soup
- You need to present your data collection process use key phrases and flowcharts

Data Collection – SpaceX API

- With the help of Space X API the data was collected.
- Then the data was transformed to better use an understanding.
- The link to notebook: https://github.com/Patich82/Applied-Data-Science-Capstone/blob/main/jupyter-labs-spacex-data-collection-api_SantacruzH.ipynb

[45]:
```python
# Takes the dataset and uses the cores column to call the API and append the data to the lists
def getCoreData(data):
    for core in data['cores']:
            if core['core'] != None:
                response = requests.get("https://api.spacexdata.com/v4/cores/"+core['core']).json()
                Block.append(response['block'])
                ReusedCount.append(response['reuse_count'])
                Serial.append(response['serial'])
            else:
                Block.append(None)
                ReusedCount.append(None)
                Serial.append(None)
            Outcome.append(str(core['landing_success'])+' '+str(core['landing_type']))
            Flights.append(core['flight'])
            GridFins.append(core['gridfins'])
            Reused.append(core['reused'])
            Legs.append(core['legs'])
            LandingPad.append(core['landpad'])
```

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

[46]:
```python
spacex_url="https://api.spacexdata.com/v4/launches/past"
```

[47]:
```python
response = requests.get(spacex_url)
```

Check the content of the response

[48]:
```python
print(response.content)
```

	FlightNumber	Date	BoosterVersion	PayloadMass	Orbit	LaunchSite	Outcome	Flights	GridFins	Reused	Legs
4	1	2010-06-04	Falcon 9	NaN	LEO	CCSFS SLC 40	None None	1	False	False	False
5	2	2012-05-22	Falcon 9	525.0	LEO	CCSFS SLC 40	None None	1	False	False	False
6	3	2013-03-01	Falcon 9	677.0	ISS	CCSFS SLC 40	None None	1	False	False	False
7	4	2013-09-29	Falcon 9	500.0	PO	VAFB SLC 4E	False Ocean	1	False	False	False
8	5	2013-12-03	Falcon 9	3170.0	GTO	CCSFS SLC 40	None None	1	False	False	False
...	...	...	...	...	...	...	...	...	...	...	...
89	86	2020-09-03	Falcon 9	15600.0	VLEO	KSC LC 39A	True ASDS	2	True	True	True
90	87	2020-10-06	Falcon 9	15600.0	VLEO	KSC LC 39A	True ASDS	3	True	True	True
91	88	2020-10-18	Falcon 9	15600.0	VLEO	KSC LC 39A	True ASDS	6	True	True	True
92	89	2020-10-24	Falcon 9	15600.0	VLEO	CCSFS SLC 40	True ASDS	3	True	True	True
93	90	2020-11-05	Falcon 9	3681.0	MEO	CCSFS SLC 40	True ASDS	1	True	False	True

90 rows × 17 columns

Data Collection - Scraping

- Extracted a Falcon 9 launch records HTML table from Wikipedia
- Parsed the table and convert it into a Pandas data frame
- The notebook link: https://github.com/Patich82/Applied-Data-Science-Capstone/blob/main/jupyter-labs-webscraping_SantacruzH.ipynb

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

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

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

```
[28]: # Use BeautifulSoup() to create a BeautifulSoup object from a response text content

      soup = BeautifulSoup(response.text, "html5lib")
```

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

```
[29]: # Use soup.title attribute
      print(soup.title)

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

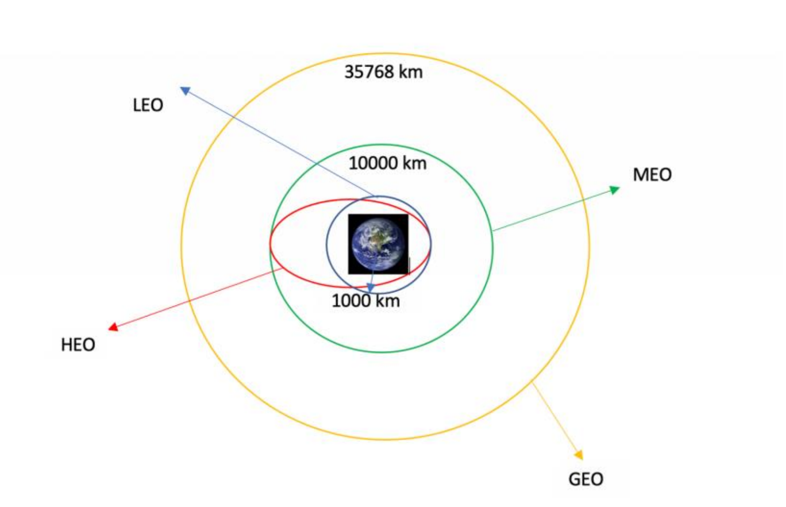
[42]:

	Flight No.	Launch site	Payload	Payload mass	Orbit	Customer	Launch outcome	Version Booster	Booster landing	Date	Time
0	1	CCAFS	Dragon Spacecraft Qualification Unit	0	LEO	SpaceX	Success\n	F9 v1.07B0003.18	Failure	4 June 2010	18:45
1	2	CCAFS	Dragon	0	LEO	NASA	Success	F9 v1.07B0004.18	Failure	8 December 2010	15:43
2	3	CCAFS	Dragon	525 kg	LEO	NASA	Success	F9 v1.07B0005.18	No attempt\n	22 May 2012	07:44
3	4	CCAFS	SpaceX CRS-1	4,700 kg	LEO	NASA	Success\n	F9 v1.07B0006.18	No attempt	8 October 2012	00:35
4	5	CCAFS	SpaceX CRS-2	4,877 kg	LEO	NASA	Success\n	F9 v1.07B0007.18	No attempt\n	1 March 2013	15:10
...	...	...	...	...	...	...	...	...	...	...	...
222	117	CCSFS	Starlink	15,600 kg	LEO	SpaceX	Success\n	F9 B5B1051.10657	Success	9 May 2021	06:42
223	118	KSC	Starlink	~14,000 kg	LEO	SpaceX	Success\n	F9 B5B1058.8660	Success	15 May 2021	22:56
224	119	CCSFS	Starlink	15,600 kg	LEO	NASA	Success\n	F9 B5B1063.2665	Success	26 May 2021	18:59
225	120	KSC	SpaceX CRS-22	3,328 kg	LEO	Sirius XM	Success\n	F9 B5B1067.1668	Success	3 June 2021	17:29
226	121	CCSFS	SXM-8	7,000 kg	GTO	NaN	NaN	F9 B5	NaN	6 June 2021	04:26

227 rows × 11 columns

Data Wrangling

- Calculated the number of launches by site
- Calculated the number of occurrences of each orbit
- Calculate the number and occurence of mission outcome of the orbits
- Link to the notebook: https://github.com/Patich82/Applied-Data-Science-Capstone/blob/main/labs-jupyter-spacex-Data%20wrangling_SantacruzH.ipynb



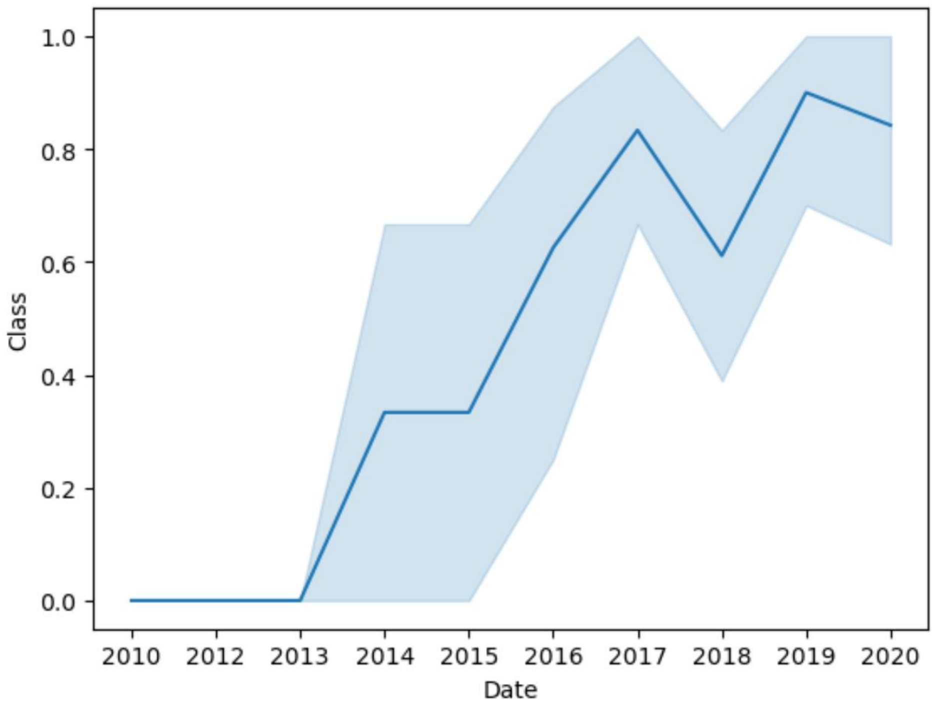
FlightNumber	Date	BoosterVersion	PayloadMass	Orbit	LaunchSite	Outcome	Flights
1	2010-06-04	Falcon 9	6104.959412	LEO	CCAFS SLC 40	None None	1
2	2012-05-22	Falcon 9	525.000000	LEO	CCAFS SLC 40	None None	1
3	2013-03-01	Falcon 9	677.000000	ISS	CCAFS SLC 40	None None	1
4	2013-09-29	Falcon 9	500.000000	PO	VAFB SLC 4E	False Ocean	1
5	2013-12-03	Falcon 9	3170.000000	GTO	CCAFS SLC 40	None None	1

Flights	GridFins	Reused	Legs	LandingPad	Block	ReusedCount	Serial	Longitude	Latitude	Class
1	False	False	False	NaN	1.0	0	B0003	-80.577366	28.561857	0
1	False	False	False	NaN	1.0	0	B0005	-80.577366	28.561857	0
1	False	False	False	NaN	1.0	0	B0007	-80.577366	28.561857	0
1	False	False	False	NaN	1.0	0	B1003	-120.610829	34.632093	0
1	False	False	False	NaN	1.0	0	B1004	-80.577366	28.561857	0

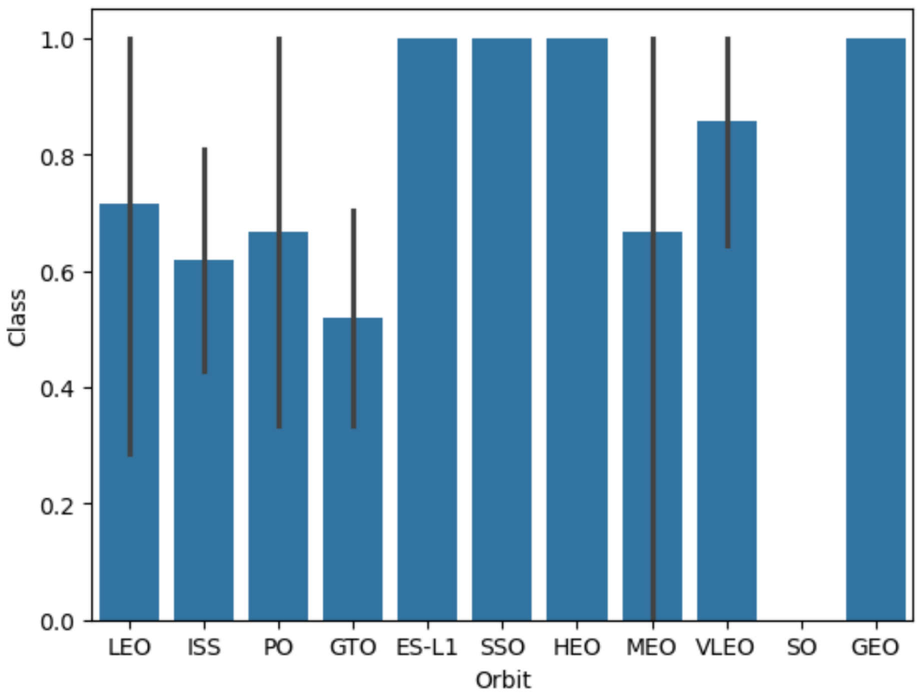
EDA with Data Visualization

The data were analyzed by visualizing the following relationships:

- The relationship between flight number and launch site
- The payload and launch site
- The success rate of each orbit type
- The relationship between flight number and orbit type
- The yearly trend in launch success
- The notebook link: https://github.com/Patich82/Applied-Data-Science-Capstone/blob/main/jupyter-labs-edadataviz_Santacruz_H.ipynb



you can observe that the sucess rate since 2013 kept increasing till 2020



Analyze the plotted bar chart to identify which orbits have the highest success rates.

EDA with SQL

- Load the Space X data to a Sqlite database
- Executed tasks involve understanding the data, loading it into a database, and executing SQL queries for analysis.
- Executed tasks:
  - **Task 1**: Find unique launch sites.
  - **Task 3**: Calculate total payload mass for missions launched by NASA (CRS).
  - **Task 5**: Discover the date of the first successful ground pad landing.
  - **Task 7**: Count successful and failure mission outcomes.
  - **Task 8**: Identify booster versions with the maximum payload mass.
  - **Task 10**: Rank landing outcome counts (Success/Failure) between specific dates.
- The notebook link: https://github.com/Patich82/Applied-Data-Science-Capstone/blob/main/jupyter-labs-eda-sql-coursera_sqllite_SantacruzH.ipynb

Build an Interactive Map with Folium

Key Functionalities:

1. Visualization:
    1. Plots launch site locations on a map.
    2. Employs color-coded markers to distinguish successful and failed launches.
2. Distance Calculation:
    1. Utilizes a distance function to calculate the distance between a launch site and a user-selected point of interest (e.g., coastline).

Interactive Exploration:

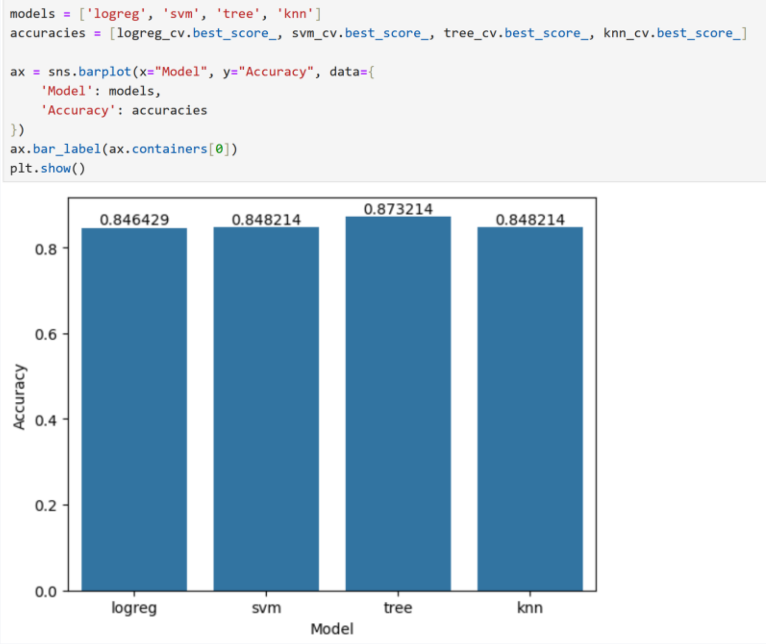
- Zoom in and out for detailed inspection of launch site locations.
- Identify successful and failed launches through color-coded markers.
- Discover the geographic coordinates of any map location (useful for pinpointing points of interest like railways or highways).
- Calculate the distance between a launch site and a chosen point of interest using the implemented distance function.
- Link to the notebook: https://github.com/Patich82/Applied-Data-Science-Capstone/blob/main/lab_jupyter_launch_site_location_SantacruzH.ipynb

Build a Dashboard with Plotly Dash

- Creation of an interactive dashboard using Plotly Dash.
- Visualization of the total launches by site using pie charts.
- Building the relationship between launch outcome and payload mass for different booster versions using scatter plots could provide a visual analysis for each situation.
- The notebook link: https://github.com/Patich82/Applied-Data-Science-Capstone/blob/main/spacex_dash_app_SantacruzH.py

Predictive Analysis (Classification)

- We utilized numpy and pandas for data loading and trans formation, followed by splitting the dataset into training a nd testing subsets.
- Multiple machine learning models were developed, with h yperparameter optimization conducted via GridSearchCV.
- Accuracy was employed as the evaluation metric for mode l performance,
  through iterative feature engineering and algorithm refine ment.
- Finally, the most effective classification model was identifi ed
- The notebook link: https://github.com/Patich82/Applied-Data-Science-Capstone/blob/main/SpaceX_Machine%20Learning%20Prediction_Part_5_SantacruzH.ipynb



Results

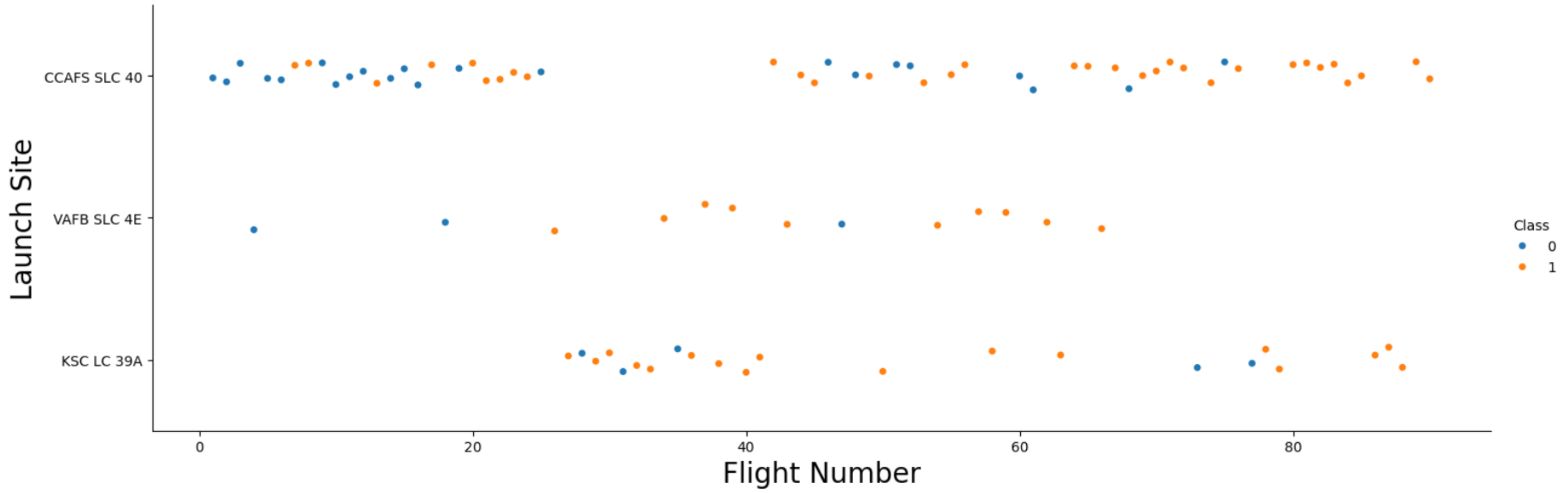
- Exploratory data analysis results
- Interactive analytics demo in screenshots
- Predictive analysis results

Section 2

Insights drawn from EDA

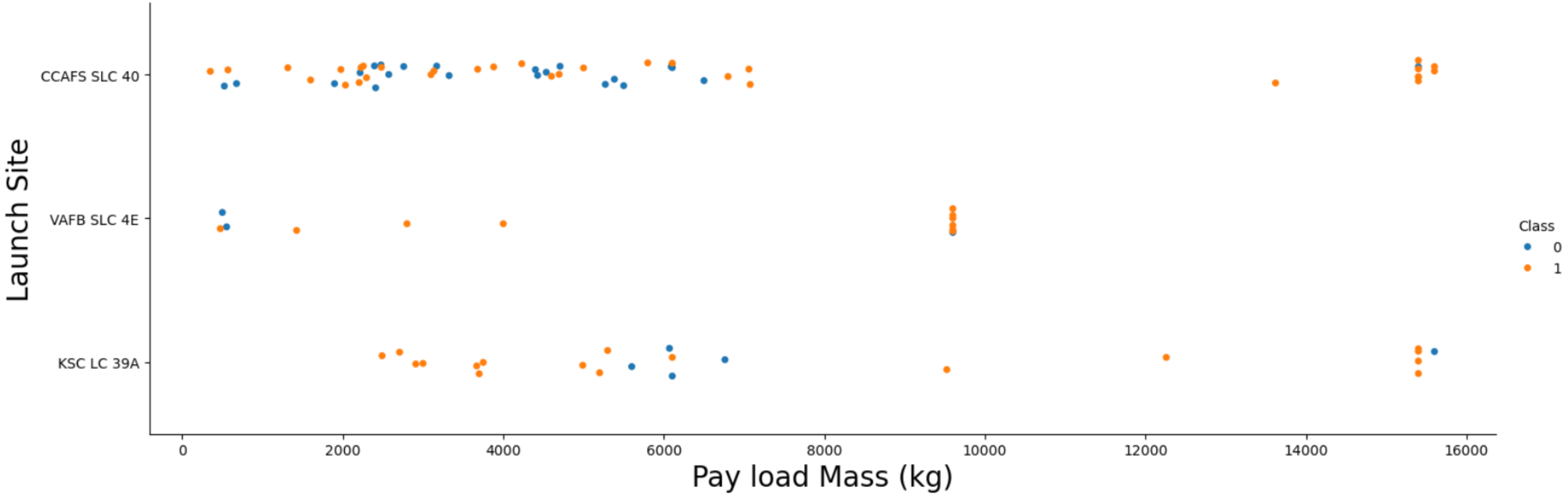
Flight Number vs. Launch Site

Launch_sites with a higher volume of flights tend to achieve greater success rates. This suggests that increased operational activity at th ese sites is linked to more successful launch outcomes.



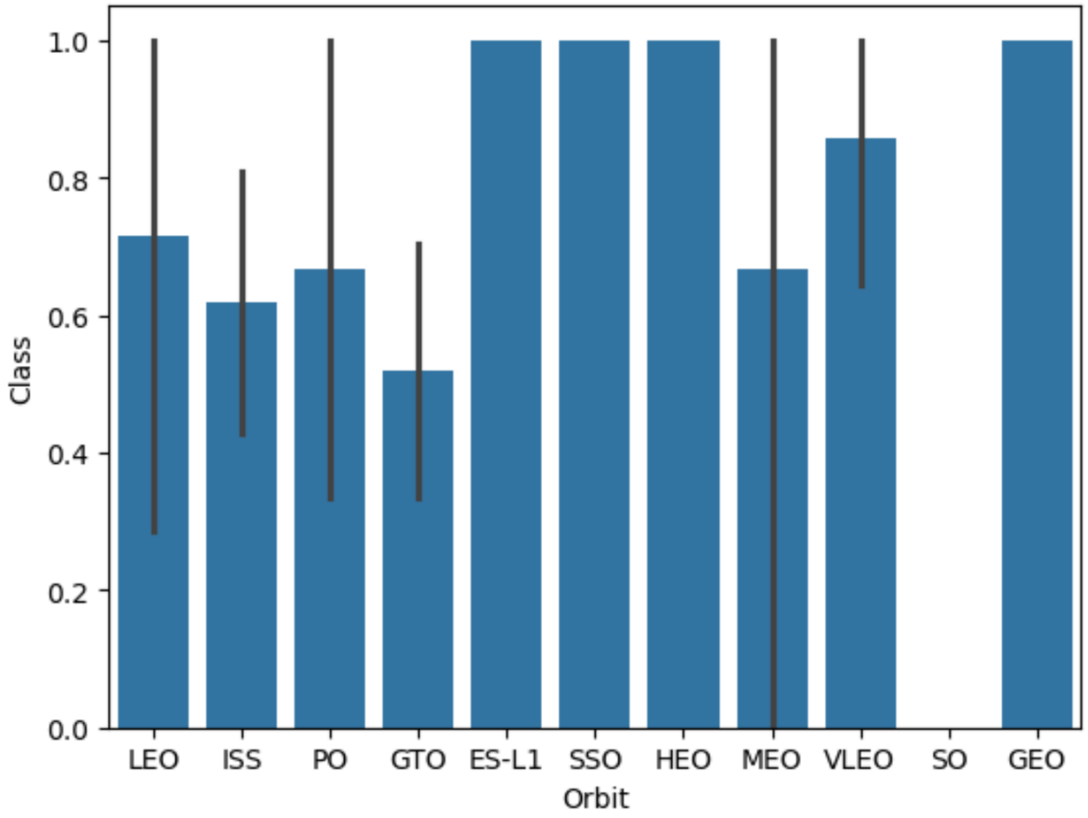
Payload vs. Launch Site

- Each launch site shows a mix of successful and failed landings, suggesting that factors other than payload mass might influence landing success.



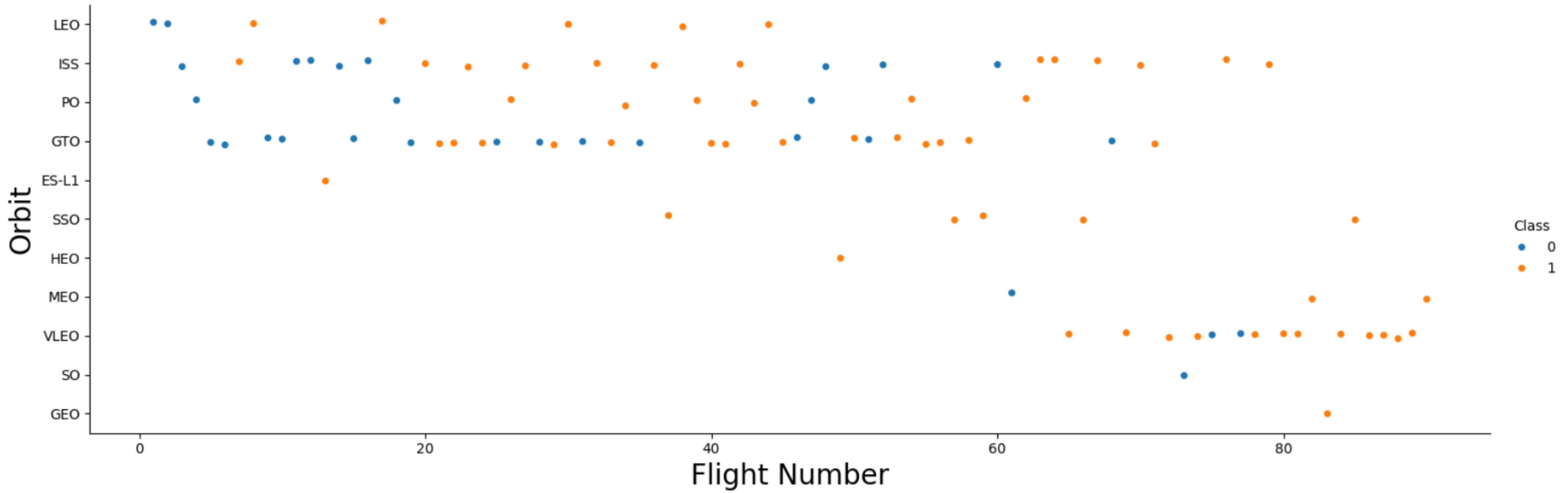
Success Rate vs. Orbit Type

- ES-L1, GEO, HEO, SSO, VLEO had the better success rate



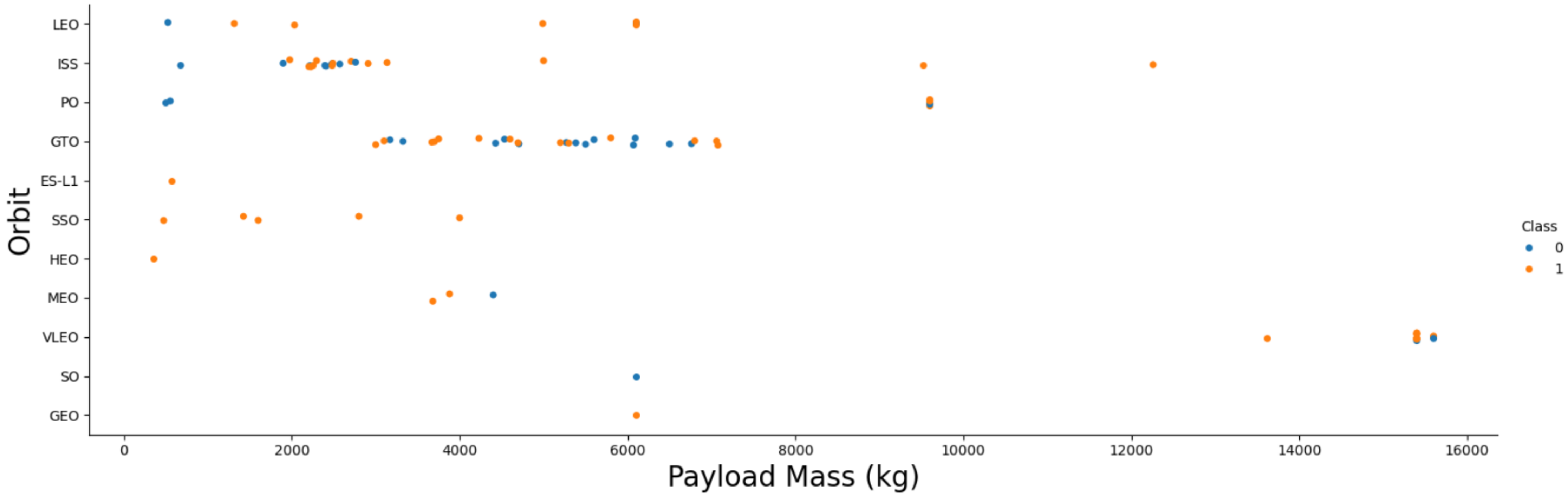
Flight Number vs. Orbit Type

- In Low Earth Orbit (LEO), a higher number of flights is associated with a higher success rate. However, in Geostationary Transfer Orbit (GTO), there doesn't seem to be a link between the number of flights and the success rate, suggesting that different factors may be at play in inf luencing outcomes in GTO.



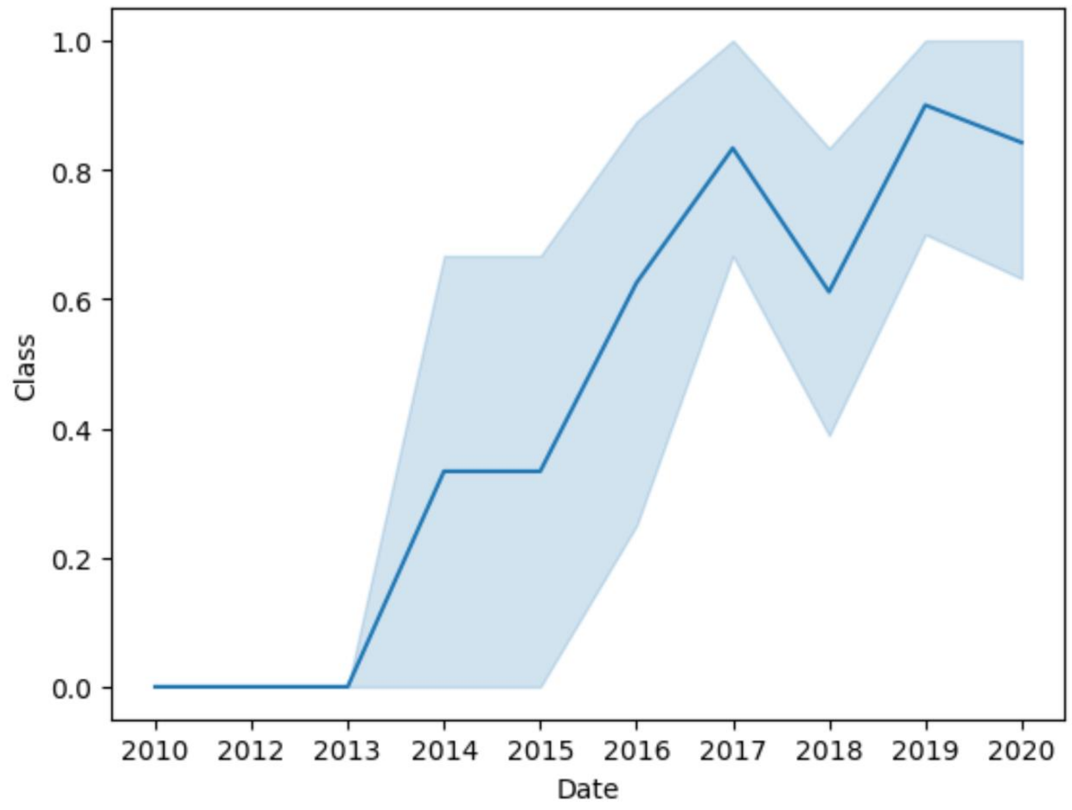
Payload vs. Orbit Type

- Booster landing success rates are generally higher for Polar, LEO, and ISS missions
- GTO missions prioritize payload deployment, making booster landing a secondary or non-existent goal.



Launch Success Yearly Trend

- The success rate increases significantly from 2013 to 2017, then there is a drop in the success rate until 2018, increasing again until 2019 and dropping slightly until 2020.
- While the success rate generally increases from 2013 to 2020, declines in specific years could be due to the development of new technology that is tested, fails, and tweaked until it starts working correctly.



All Launch Site Names

- We extracted unique launch site information from the original spacex_df DataFrame by grouping the data by launch site, selecting the first row of each group and then creating a new DataFrame with only the desired columns. This new DataFrame, launch_sites_df, can then be used for further analysis or visualization.nd the names of the unique launch sites

```
# Select relevant sub-columns: `Launch Site`, `Lat(Latitude)`, `Long(Longitude)`, `class`
spacex_df = spacex_df[['Launch Site', 'Lat', 'Long', 'class']]
launch_sites_df = spacex_df.groupby(['Launch Site'], as_index=False).first()
launch_sites_df = launch_sites_df[['Launch Site', 'Lat', 'Long']]
launch_sites_df
```

	Launch Site	Lat	Long
0	CCAFS LC-40	28.562302	-80.577356
1	CCAFS SLC-40	28.563197	-80.576820
2	KSC LC-39A	28.573255	-80.646895
3	VAFB SLC-4E	34.632834	-120.610745

Launch Site Names Begin with 'CCA'

- The task is to identify and extract rows from the df DataFrame where the launch site starts with "CCA".
- This filtered DataFrame can then be used for further analysis, visualization, or other tasks.

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

```
df[df["Launch_Site"].str.startswith("CCA")].head()
```

	Date	Time (UTC)	Booster_Version	Launch_Site	Payload	PAYLOAD_MASS__KG_	Orbit	Customer	Mission_Outcome	Landing_Outcome
0	2010-06-04	18:45:00	F9 v1.0 B0003	CCAFS LC-40	Dragon Spacecraft Qualification Unit	0	LEO	SpaceX	Success	Failure (parachute)
1	2010-12-08	15:43:00	F9 v1.0 B0004	CCAFS LC-40	Dragon demo flight C1, two CubeSats, barrel of...	0	LEO (ISS)	NASA (COTS) NRO	Success	Failure (parachute)
2	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
3	2012-10-08	0:35:00	F9 v1.0 B0006	CCAFS LC-40	SpaceX CRS-1	500	LEO (ISS)	NASA (CRS)	Success	No attempt
4	2013-03-01	15:10:00	F9 v1.0 B0007	CCAFS LC-40	SpaceX CRS-2	677	LEO (ISS)	NASA (CRS)	Success	No attempt

Total Payload Mass

- The task is to calculate the total payload mass launched by NASA (CRS) as per the data in the df DataFrame.
- This can be useful for various analyses, such as understanding the payload capacity or mission profile of NASA's CRS missions.

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

```
df[df["Customer"] == "NASA (CRS)"]["PAYLOAD_MASS__KG_"].sum()
```

```
45596
```

Average Payload Mass by F9 v1.1

- The task is to calculate the average payload mass launched by the F9 v1.1 booster as per the data in the df DataFrame.
- This can be useful for various analyses, such as comparing the performance of different booster versions or understanding the payload capacity of the F9 v1.1.

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

```
df[df["Booster_Version"] == "F9 v1.1"]["PAYLOAD_MASS__KG_"].mean()
```

```
2928.4
```

First Successful Ground Landing Date

- The task is to find the earliest date when a SpaceX rocket successfully landed on a ground pad.
- It does this by selecting the minimum date from the SPACEXTBL table, but only considering records where the landing outcome was a successful ground pad landing.

```
%sql SELECT MIN(DATE) AS FIRST_SUCCESS_GP FROM SPACEXTBL WHERE LANDING_OUTCOME = 'Success (ground pad)';
```

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

FIRST_SUCCESS_GP
2015-12-22

Successful Drone Ship Landing with Payload between 4000 and 6000

- The task is to identify the unique booster versions that have successfully landed on a drone ship while carrying payloads between 4000 and 6000 kilograms.
- By using the DISTINCT keyword, the query ensures that each booster version is listed only once, even if it has been used multiple times.

```
%sql SELECT DISTINCT BOOSTER_VERSION FROM SPACEXTBL WHERE PAYLOAD_MASS__KG_ BETWEEN 4000 AND 6000 AND LANDING_OUTCOME = 'Success (drone ship)';
```

```
 * 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

- The task is to provide a summary of the mission outcomes in the SPACEXTBL table. It counts the number of missions for each outcome and presents the results in a clear and organized manner.
- This information can be useful for analyzing the overall success rate of SpaceX missions and identifying trends or patterns in mission outcomes.

List the total number of successful and failure mission outcomes

```
%sql SELECT MISSION_OUTCOME, COUNT(*) AS QTY FROM SPACEXTBL GROUP BY MISSION_OUTCOME ORDER BY MISSION_OUTCOME;
```

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

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

Boosters Carried Maximum Payload

- The task is to identify the booster versions that have been used to launch the heaviest payloads.
- By filtering the data based on the maximum payload mass and selecting distinct booster versions, we can pinpoint the specific boosters that have been capable of handling such heavy loads.

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

```
#df.loc[(df["Mission_Outcome"] == "Success") & (df["PAYLOAD_MASS__KG_"] > 4000) & (df["PAYLOAD_MASS__KG_"] < 6000)] #["Booster_Version"].unique()
%sql SELECT DISTINCT BOOSTER_VERSION FROM SPACEXTBL WHERE PAYLOAD_MASS__KG_ = (SELECT MAX(PAYLOAD_MASS__KG_) FROM SPACEXTBL) ORDER BY BOOSTER_VERSION;
```

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

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

2015 Launch Records

- The task is to identify the booster versions and launch sites associated with failed drone ship landings in 2015.
- By extracting the month from the date and filtering the data based on the landing outcome and year, we can gain insights into the specific circumstances of these failures and potentially identify trends or patterns.

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.

Note: SQLLite does not support monthnames. So you need to use substr(Date, 6,2) as month to get the months and substr(Date,0,5)='2015' for year.

```
%sql SELECT substr(Date, 6, 2) AS Month, BOOSTER_VERSION, LAUNCH_SITE FROM SPACEXTBL WHERE LANDING_OUTCOME = 'Failure (drone ship)' AND substr(Date,0,5)='2015' ORDER BY Month;
```

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

Month	Booster_Version	Launch_Site
01	F9 v1.1 B1012	CCAFS LC-40
04	F9 v1.1 B1015	CCAFS LC-40

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

- The task is to analyze the success rate of SpaceX landings within a specific date range. By filtering the data based on the landing outcome and date range, and then grouping and counting the successful outcomes, we can gain insights into the frequency of different types of successful landings during this period.
- This information can be useful for tracking the progress of SpaceX's landing technology and identifying trends in landing outcomes.

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.

```
[65]: ECT LANDING_OUTCOME, COUNT(*) AS QTY FROM SPACEXTBL WHERE LANDING_OUTCOME LIKE '%Success%' AND Date BETWEEN '2010-06-04' AND '2017-03-20' GROUP BY LANDING_OUTCOME ORDER BY QTY DESC;

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

[65]:

Landing_Outcome	QTY
Success (drone ship)	5
Success (ground pad)	3

Section 3

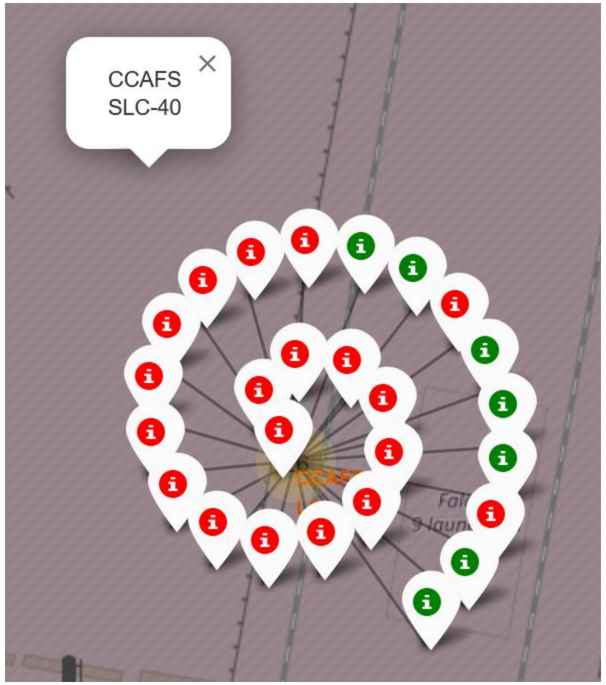
Launch Sites Proximities Analysis

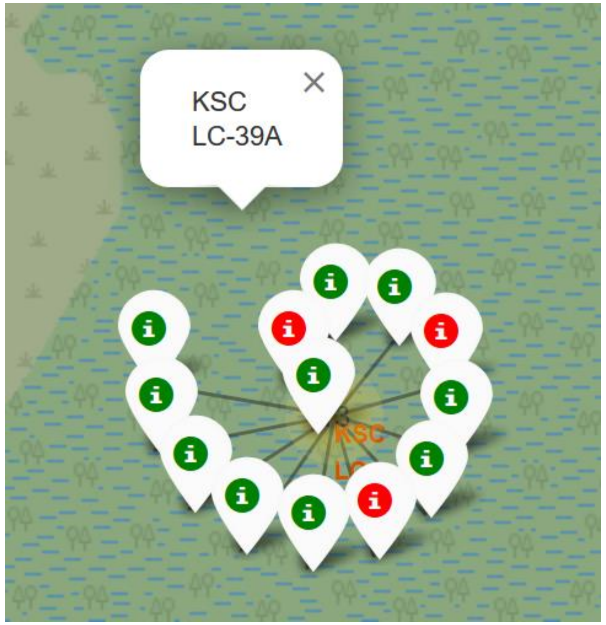
Global Map Launch Sites

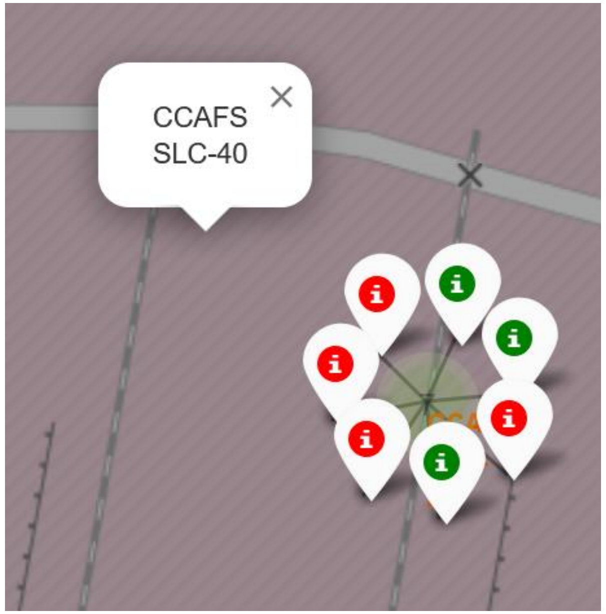
- Space X sites are located only in United States of America.
- The launch sites are near the coast border.
- The launch sites are focused on Florida and California.

Launch sites with failure/success labels

- The 3 firsts captions from left to right are Florida Sites
- The last caption from left to right is California site
- The green markers show successfully launches
- The red markers show failure launches

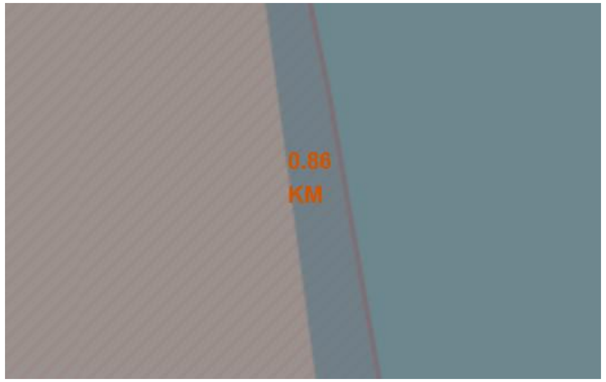


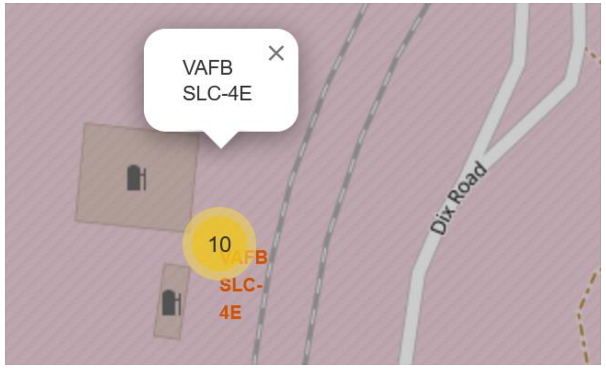


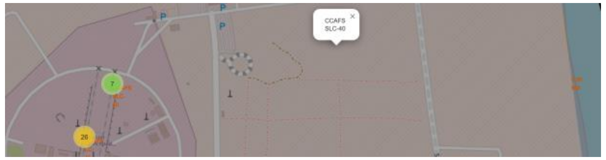


Launch sites vs Landmarks

- The launch sites:
  - Are no close to railways
  - Are no close to highways
  - Keep certain distance away from cities
  - Are close to the coastlines





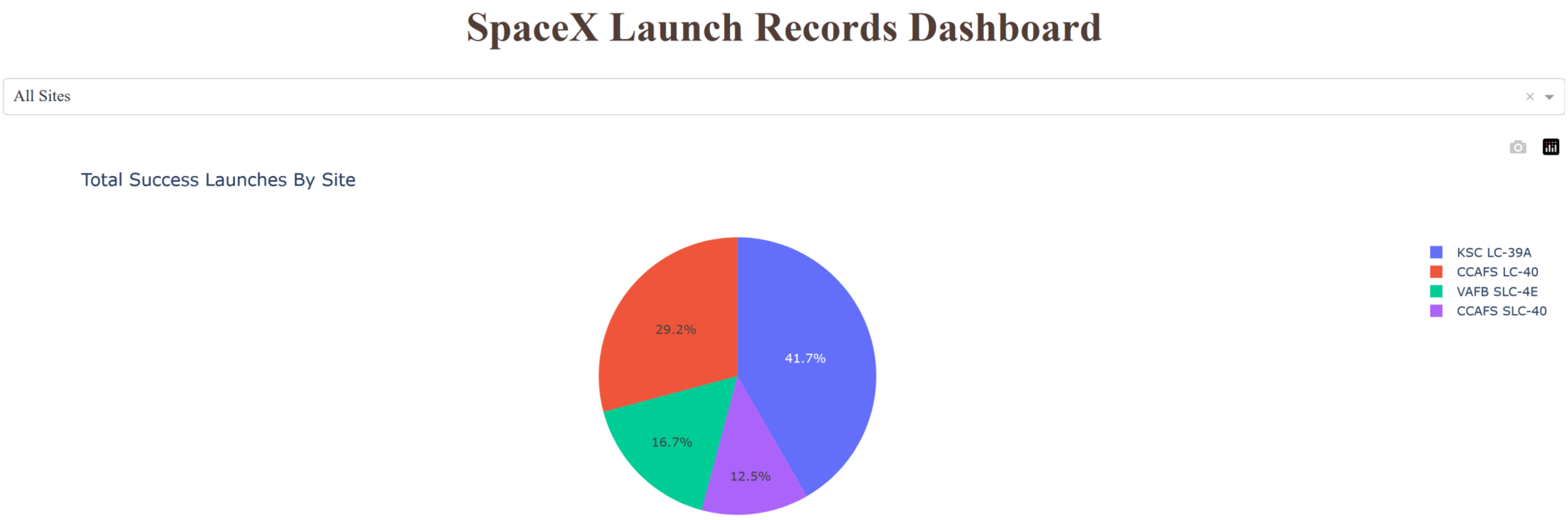


Section 4

Build a Dashboard with Plotly Dash

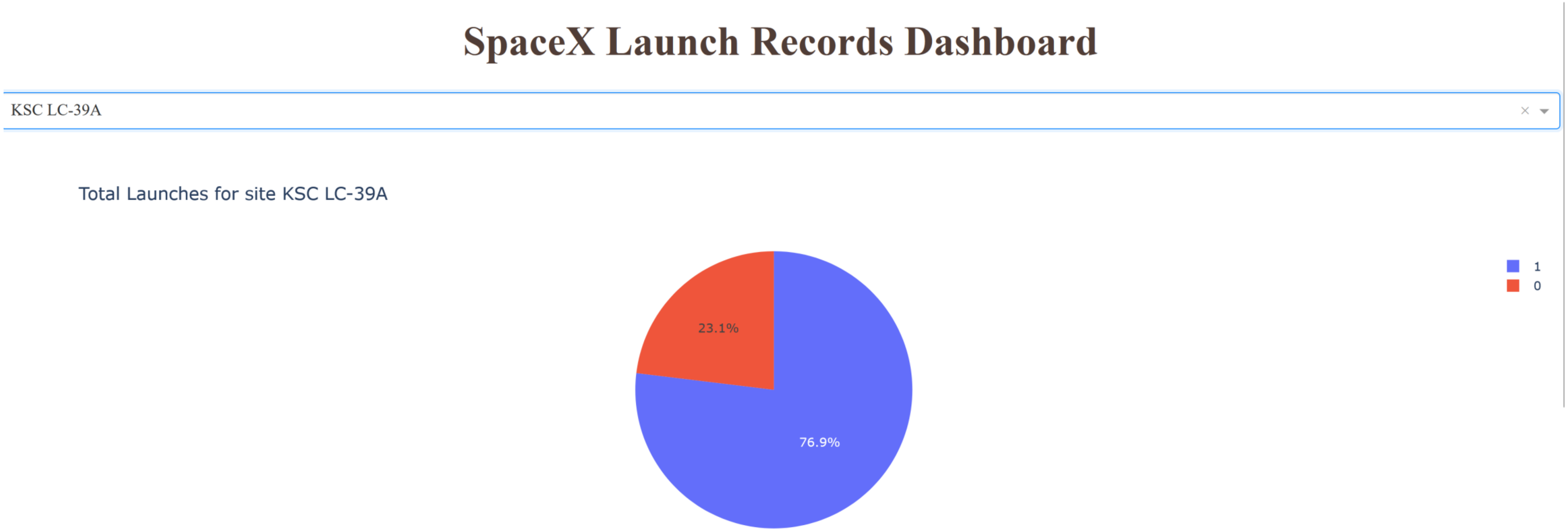
Total Success Launches by Site

- KSC LC 39 A had the most successful launches from all sites
- CCAFS SLC 40 had the lowest successful launches from all sites
- The pie gives us a general vision of the successful rate but we need more data to understand why is this happening



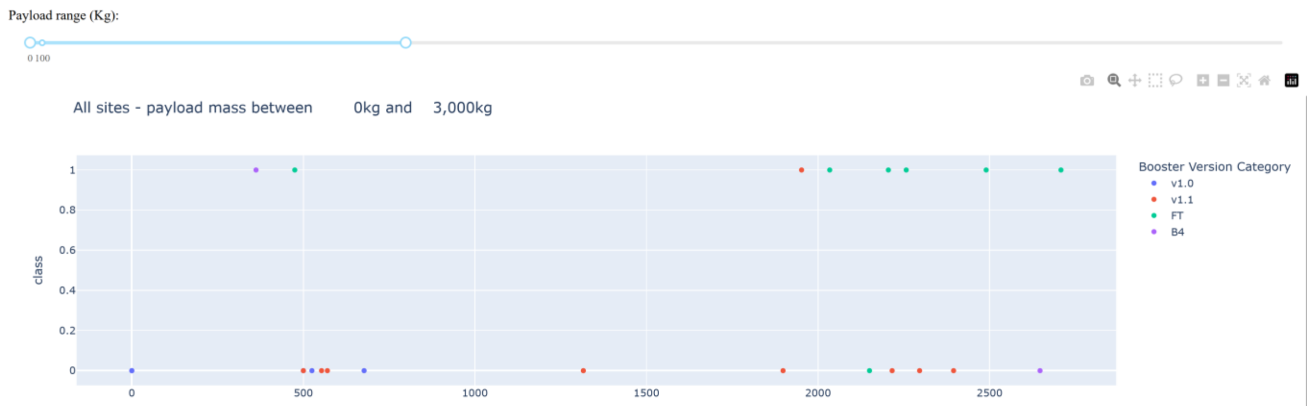
Highest lunch success ratio

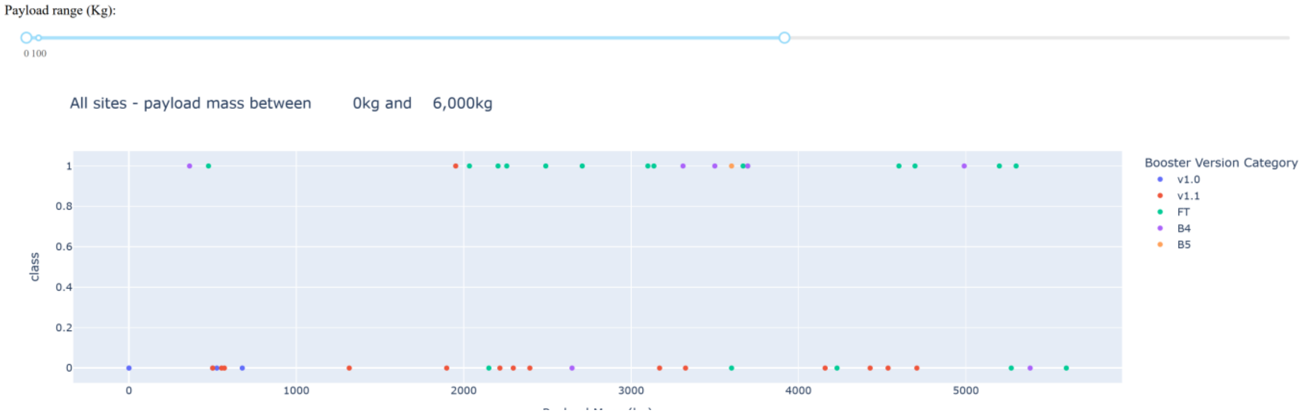
- KSC LC 39A achieved a 76.9% success rate.
- KSC LC 39A achieved a 23.1% failure rate.
- The pie gives us a general vision of the successful rate but we need more data to understand why is this happening

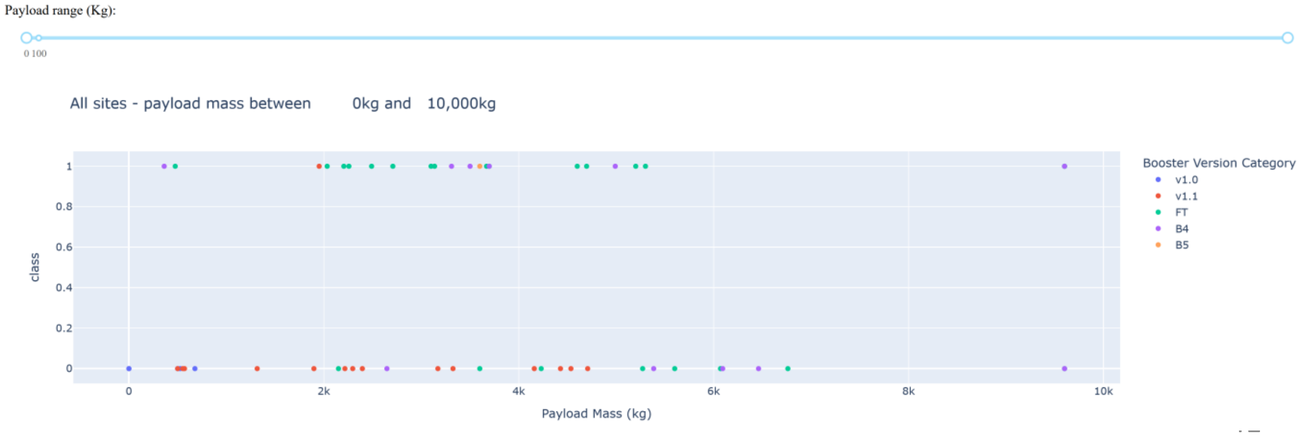


Scatter Plot Payload vs Launch Outcome (All Sites)

- Different booster versions have different payload capacities. Some versions can handle heavier payloads than others.
- More data is needed to get a complete understanding of each booster performance





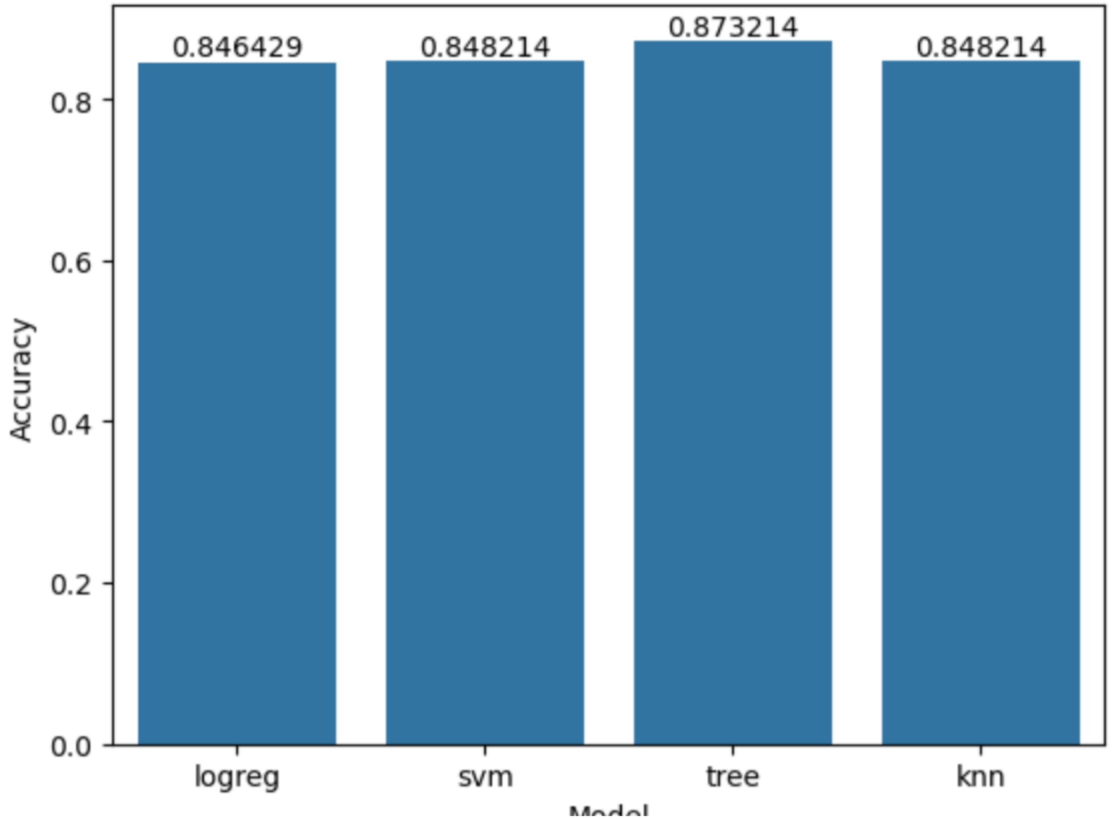


Section 5

Predictive Analysis (Classification)

Classification Accuracy

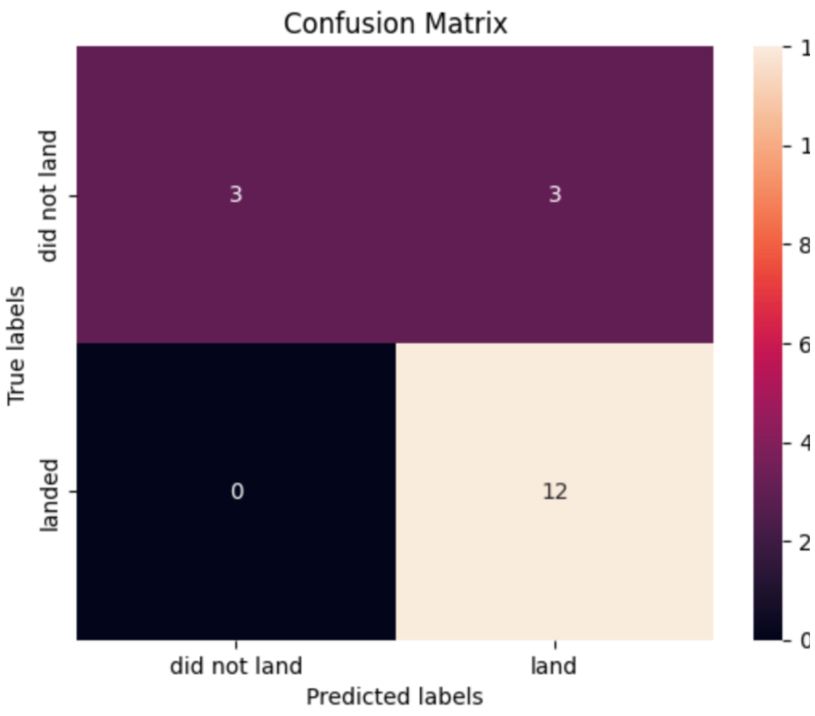
- The decision tree classifier is the model with the highest classification accuracy



Confusion Matrix

- True Positives (TP): 12 instances were correctly predicted as "Landed".
- True Negatives (TN): 3 instances were correctly predicted as "Did Not Land".
- False Positives (FP): 3 instances were incorrectly predicted as "Landed" when they actually "Did Not Land".
- False Negatives (FN): 0 instances were incorrectly predicted as "Did Not Land" when they actually "Landed".

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

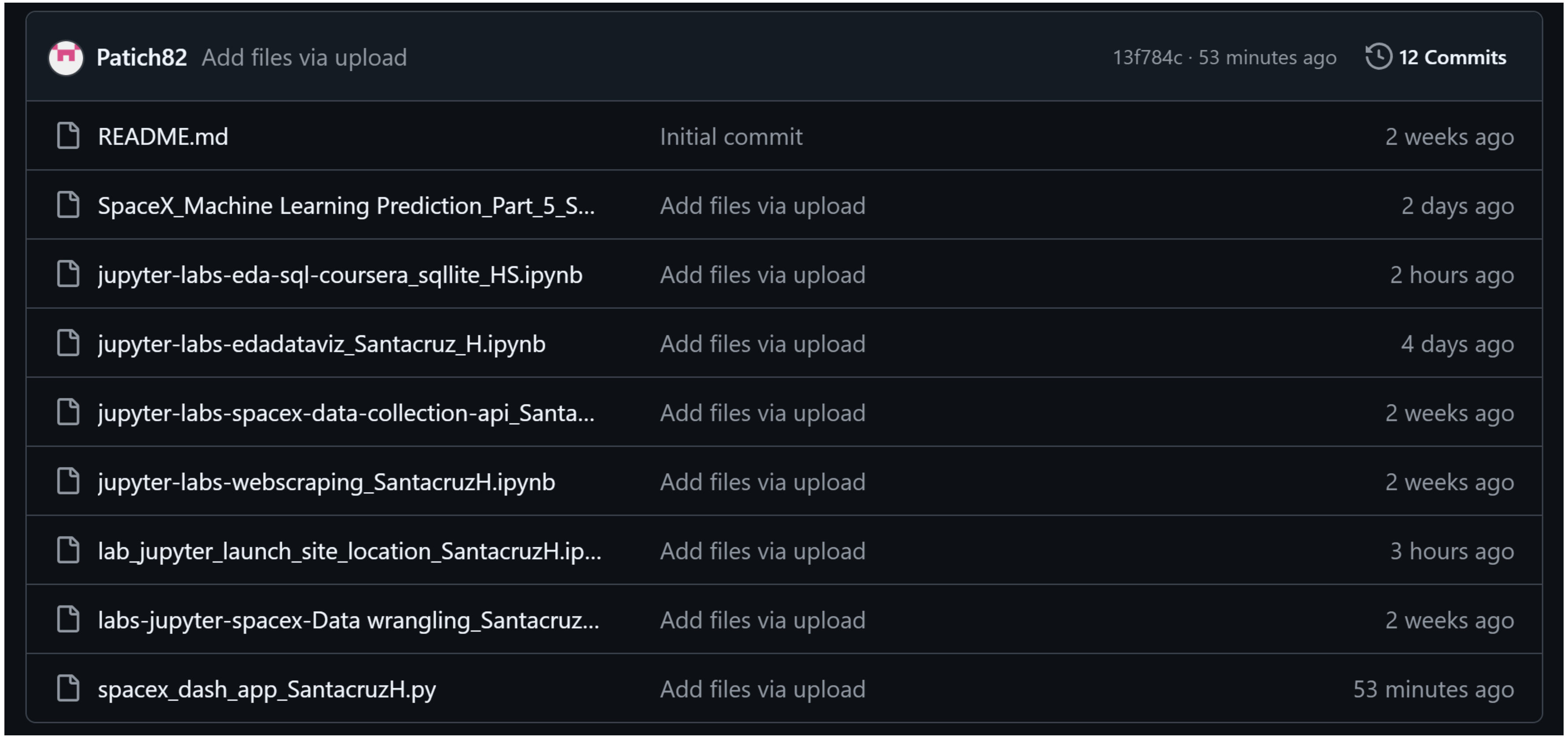


Conclusions

- SpaceX has significantly advanced booster reusability, particularly with the Falcon 9 rocket. This technological achievement has led to substantial cost reductions and increased launch frequency.
- SpaceX has demonstrated the ability to launch a wide range of payloads.
- SpaceX has precise landing techniques, successfully landing rocket boosters both on land and on autonomous drones at sea.
- Orbits ES-L1, GEO, HEO, SSO, VLEO had the most success rate.
- KSC LC-39A had the most successful launches.
- SpaceX has consistently maintained a high mission success rate since 2013
- The Decision tree classifier is the best machine learning algorithm for this task.

Appendix

- You could access to all the codes in GitHub: https://github.com/Patich82/Applied-Data-Science-Capstone/tree/main



Patich82 Add files via upload 13f784c · 53 minutes ago 12 Commits

README.md	Initial commit	2 weeks ago
SpaceX_Machine Learning Prediction_Part_5_S...	Add files via upload	2 days ago
jupyter-labs-eda-sql-coursera_sqllite_HS.ipynb	Add files via upload	2 hours ago
jupyter-labs-edadataviz_Santacruz_H.ipynb	Add files via upload	4 days ago
jupyter-labs-spacex-data-collection-api_Santa...	Add files via upload	2 weeks ago
jupyter-labs-webscraping_SantacruzH.ipynb	Add files via upload	2 weeks ago
lab_jupyter_launch_site_location_SantacruzH.ip...	Add files via upload	3 hours ago
labs-jupyter-spacex-Data wrangling_Santacruz...	Add files via upload	2 weeks ago
spacex_dash_app_SantacruzH.py	Add files via upload	53 minutes ago

Thank you!