IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

Kostiantyn
2023-11-29

Outline

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

Executive Summary

Summary of methodologies used

- CRISP-DM (Cross-Industry Standard Process for Data Mining)
- Feature Engineering

Summary of all results

Introduction

Project background and context

The commercial space age is booming, and companies like Virgin Galactic, Rocket Lab, Blue Origin, and SpaceX are making space travel more accessible and affordable. SpaceX, in particular, has made significant strides in reducing the cost of space launches, primarily through the ability to reuse the first stage of its Falcon 9 rocket.

This capstone project puts you in the role of a data scientist working for Space Y, a new rocket company seeking to compete with SpaceX. Your task is to develop a machine learning model that can predict whether SpaceX will reuse the first stage of its Falcon 9 rocket for a given launch. This information will be crucial in determining the price of each launch for Space Y.

Problems we want to find answers

- Determining the Cost of Launch
- Predicting First Stage Reusability
- Creating Dashboards for Data Analysis

Section 1

Methodology

Methodology

Executive Summary

Data collection methodology

- We obtained data from two distinct sources, drawing information from both Wikipedia and an external API.

Perform data wrangling

- The gathered data underwent enrichment through the generation of a final outcome label, derived from a comprehensive analysis of summarized features and outcome data

Methodology

Executive Summary

Next Steps during EDA was performed

- Calculate the number of launches on each site: This will help us understand the distribution of launches across different launch sites.
- Calculate the number and occurrence of each orbit: This will help us understand the types of orbits that SpaceX typically targets.
- Calculate the number and occurrence of mission outcome of the orbits: This will help us understand the success rate of SpaceX launches for different orbits.
- Create a landing outcome label from Outcome column: This will create a new binary variable that indicates whether the first stage of the Falcon 9 rocket landed successfully or not. This will be the target variable for our machine learning model.

Methodology

Executive Summary

To build, tune, and evaluate classification models, follow these steps:

- Prepare the data: clean, explore, and engineer features.
- Choose a classification algorithm and split data into training and testing sets.
- Train the model and tune hyperparameters for optimal performance.
- Evaluate the model using metrics like accuracy, precision, recall, and F1-score.
- Analyze misclassified examples and error types to improve the model.
- Compare different models and select the best one based on performance trade-offs.
- Use visualizations, feature importance, and cross-validation for deeper understanding.
- Deploy the chosen model for real-world use.

Data Collection

How data sets were collected.

To ensure data accuracy and completeness, we utilized both Wikipedia and an API to collect relevant information.

```
df=pd.read_csv("https://cf-courses-data.s3.us.cloud-object-storage.appdomain.cloud/IBM-DS0321EN-SkillsNetwork/datasets/dataset_part_1.csv")
df.head(10)
```

	FlightNumber	Date	BoosterVersion	PayloadMass	Orbit	LaunchSite	Outcome	Flights	GridFins	Reused	Legs	LandingPad	Block	ReusedCount	Serial	Longitude	Latitude
0	1	2010-06-04	Falcon 9	6104.959412	LEO	CCAFS SLC 40	None None	1	False	False	False	NaN	1.0	0	B0003	-80.577366	28.561857
1	2	2012-05-22	Falcon 9	525.000000	LEO	CCAFS SLC 40	None None	1	False	False	False	NaN	1.0	0	B0005	-80.577366	28.561857
2	3	2013-03-01	Falcon 9	677.000000	ISS	CCAFS SLC 40	None None	1	False	False	False	NaN	1.0	0	B0007	-80.577366	28.561857
3	4	2013-09-29	Falcon 9	500.000000	PO	VAFB SLC 4E	False Ocean	1	False	False	False	NaN	1.0	0	B1003	-120.610829	34.632093
4	5	2013-12-03	Falcon 9	3170.000000	GTO	CCAFS SLC 40	None None	1	False	False	False	NaN	1.0	0	B1004	-80.577366	28.561857
5	6	2014-01-06	Falcon 9	3325.000000	GTO	CCAFS SLC 40	None None	1	False	False	False	NaN	1.0	0	B1005	-80.577366	28.561857
6	7	2014-04-18	Falcon 9	2296.000000	ISS	CCAFS SLC 40	True Ocean	1	False	False	True	NaN	1.0	0	B1006	-80.577366	28.561857
7	8	2014-07-14	Falcon 9	1316.000000	LEO	CCAFS SLC 40	True Ocean	1	False	False	True	NaN	1.0	0	B1007	-80.577366	28.561857
8	9	2014-08-05	Falcon 9	4535.000000	GTO	CCAFS SLC 40	None None	1	False	False	False	NaN	1.0	0	B1008	-80.577366	28.561857
9	10	2014-09-07	Falcon 9	4428.000000	GTO	CCAFS SLC 40	None None	1	False	False	False	NaN	1.0	0	B1011	-80.577366	28.561857

Data Collection – SpaceX API

I created a new repository called "ibm_ds_cert" on GitHub, which contains the Jupyter notebook for the web scraping and data visualization of the SpaceX dataset. You can find the repository [here].

The notebook includes various visualizations, such as scatter plots, bar charts, and line charts, to explore the relationships between different variables and gain insights into the SpaceX dataset.

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

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

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

Data Collection - Scraping

I created a new repository called "ibm_ds_cert" on GitHub, which contains the Jupyter notebook for the web scraping and data visualization of the SpaceX dataset. You can find the repository here.

The notebook includes various visualizations, such as scatter plots, bar charts, and line charts, to explore the relationships between different variables and gain insights into the SpaceX dataset.

TASK 1: Request the Falcon9 Launch Wiki page from its URL

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

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

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

```
# Use BeautifulSoup() to create a BeautifulSoup object from a response text content
soup = BeautifulSoup(response.text, 'html.parser')
```

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

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

```
'List of Falcon 9 and Falcon Heavy launches - Wikipedia'
```

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

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

Let's try to find all tables on the wiki page first. If you need to refresh your memory about `BeautifulSoup`

```
# Use the find_all function in the BeautifulSoup object, with element type `table`
# Assign the result to a list called `html_tables`

html_tables = [item for item in soup.find_all('table')]

# html_tables = [item.text for item in soup.find_all('table')] #readable variant
```

Data Wrangling

Data Wrangling processed steps:

- Analyze Launch Distribution: Determine the number of launches conducted at each launch site.
- Characterize Orbit Types: Quantify the frequency of each orbit type targeted by SpaceX missions.
- Assess Orbit-Specific Mission Outcomes: Evaluate the success rate of SpaceX missions for each orbit type.
- Derive Landing Outcome Indicator: Create a binary variable indicating whether the Falcon 9 rocket's first stage successfully landed.

I created a new repository called "ibm_ds_cert" on GitHub, which contains the Jupyter notebook for the DW and data visualization of the SpaceX dataset. You can find the repository here. The notebook includes various visualizations, such as scatter plots, bar charts, and line charts, to explore the relationships between different variables and gain insights into the SpaceX dataset.

EDA with Data Visualization

Visualizations and Purposes

- Scatter plot: To understand the distribution of flight numbers across different launch sites, payload weight variations, and any patterns or trends between flight numbers and orbit types.
- Bar chart: To compare the success rates of SpaceX missions for different orbit types and assess whether payload weight is influenced by the targeted orbit type.
- Line chart: To analyze the overall trend of SpaceX launch success over the years.
- Data type conversion: To convert categorical variables into numerical representations for machine learning modeling and ensure consistent data type for numerical variables.

I created a new repository called "ibm_ds_cert" on GitHub, which contains the Jupyter notebook for the data visualization of the SpaceX dataset. You can find the repository here. The notebook includes various visualizations, such as scatter plots, bar charts, and line charts, to explore the relationships between different variables and gain insights into the SpaceX dataset.

EDA with SQL

Summary of SQL Queries without Code

- Extracted a list of all distinct launch sites from the space mission data.
- Identified five records where the launch site name began with the string 'CCA'.
- Calculated the total payload mass carried by boosters launched by NASA (CRS) missions.
- Retrieved the date of the first successful landing outcome achieved on a ground pad.
- Listed the booster names associated with successful drone ship landings and payload masses between 4000 and 6000.
- Counted the total number of successful and failure mission outcomes across all launches.
- Identified the booster versions that had carried the maximum payload mass.

I created a new repository called "ibm_ds_cert" on GitHub, which contains the Jupyter notebook for the queering of the SpaceX dataset. You can find the repository [here](). The notebook includes codes to explore the relationships between different variables and gain insights into the SpaceX dataset.

Build an Interactive Map with Folium

The summary of the map objects I created and added to a folium map.

Launch Sites:

- Blue circles with popup labels and icons.
- Green for successful, red for failed launches.

Proximity Distance:

- Marker clusters and polylines.
- Distance displays on markers and polylines.

I created a new repository called "ibm_ds_cert" on GitHub, which contains the Jupyter notebook for the data visualization of the SpaceX dataset. You can find the repository here. The notebook includes various visualizations, such as scatter plots, bar charts, and line charts, to explore the relationships between different variables and gain insights into the SpaceX dataset.

Build a Dashboard with Plotly Dash

What plots/graphs and interactions you have added to a dashboard

- Added a drop-down input component to select a launch site.
- Added a callback function to render a pie chart visualizing launch success counts based on the selected site.
- Added a range slider to select a payload range.
- Added a callback function to render a scatter plot visualizing the relationship between payload and launch outcome.

I created a new repository called "ibm_ds_cert" on GitHub, which contains the Jupyter notebook for the data visualization of the SpaceX dataset. You can find the repository here. The notebook includes various visualizations, such as scatter plots, bar charts, and line charts, to explore the relationships between different variables and gain insights into the SpaceX dataset.

Predictive Analysis (Classification)

Summary on how we built, evaluated, improved, and found the best performing classification model

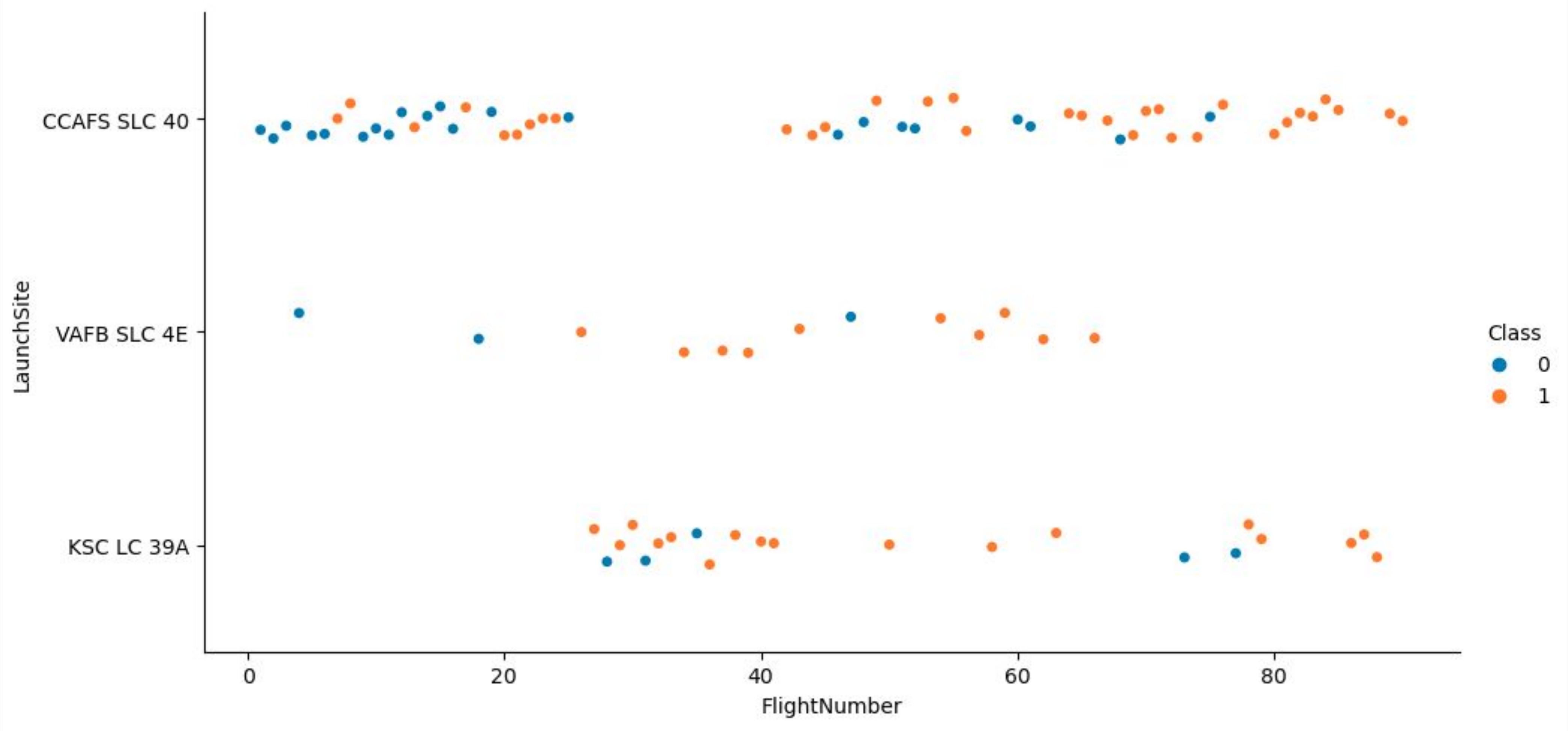
- Convert a data column to a NumPy array and Pandas Series
- Standardize data and split it into training and test sets
- Train and evaluate logistic regression, support vector machine, decision tree classifier, and K nearest neighbors models

I created a new repository called "ibm_ds_cert" on GitHub, which contains the Jupyter notebook for the data visualization of the SpaceX dataset. You can find the repository here. The notebook includes various visualizations, such as scatter plots, bar charts, and line charts, to explore the relationships between different variables and gain insights into the SpaceX dataset.

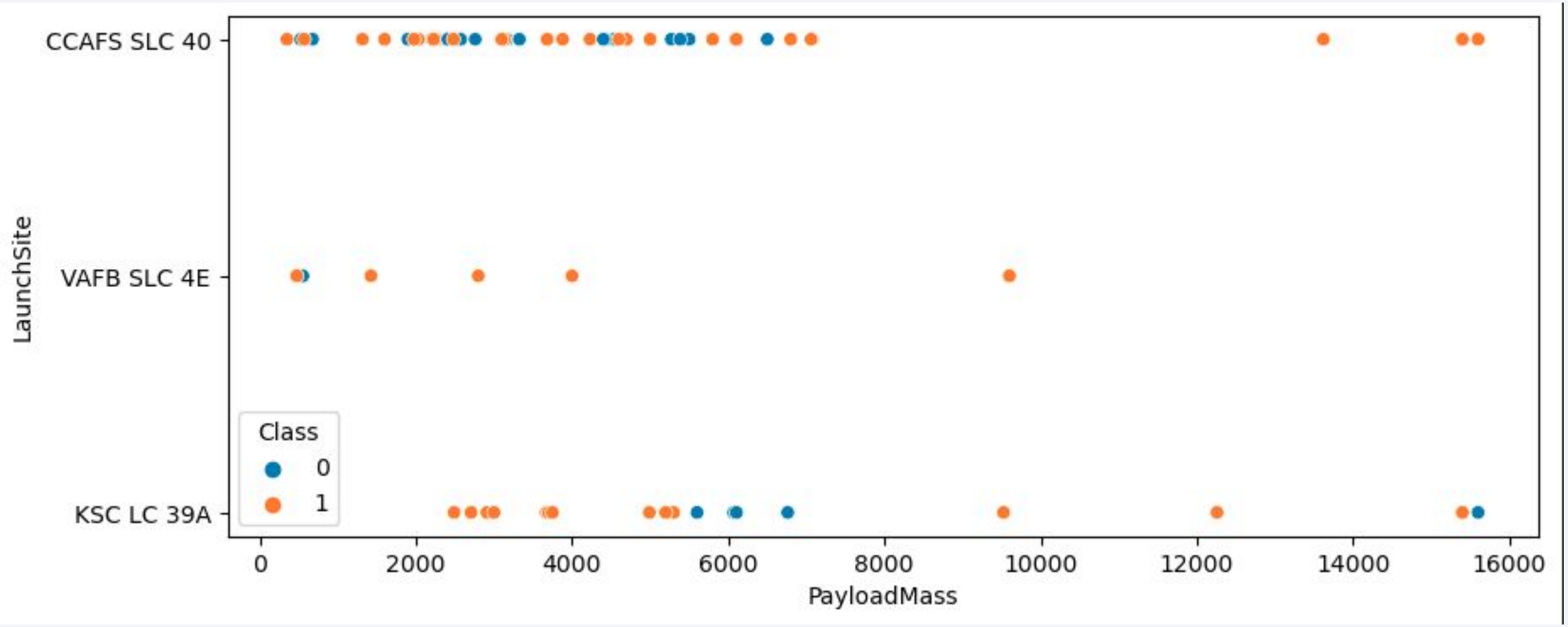
Section 2

Insights drawn from EDA

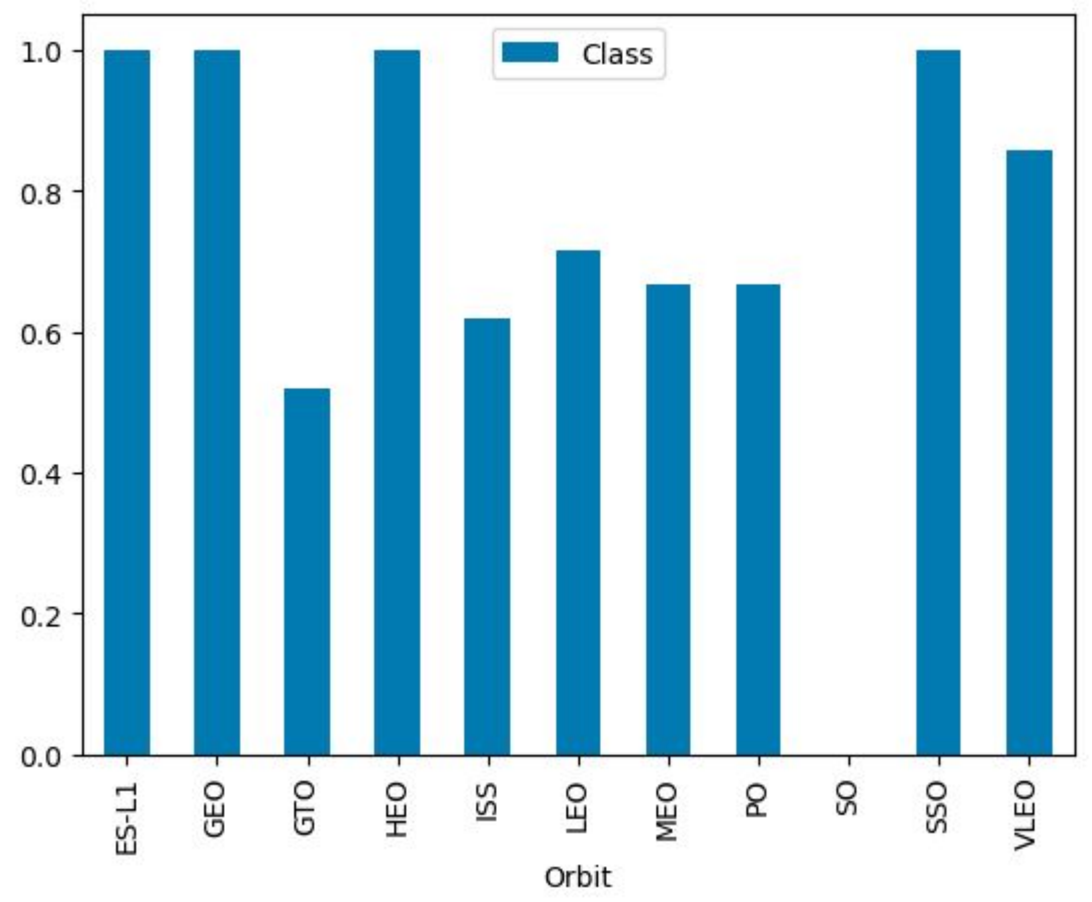
Flight Number vs. Launch Site



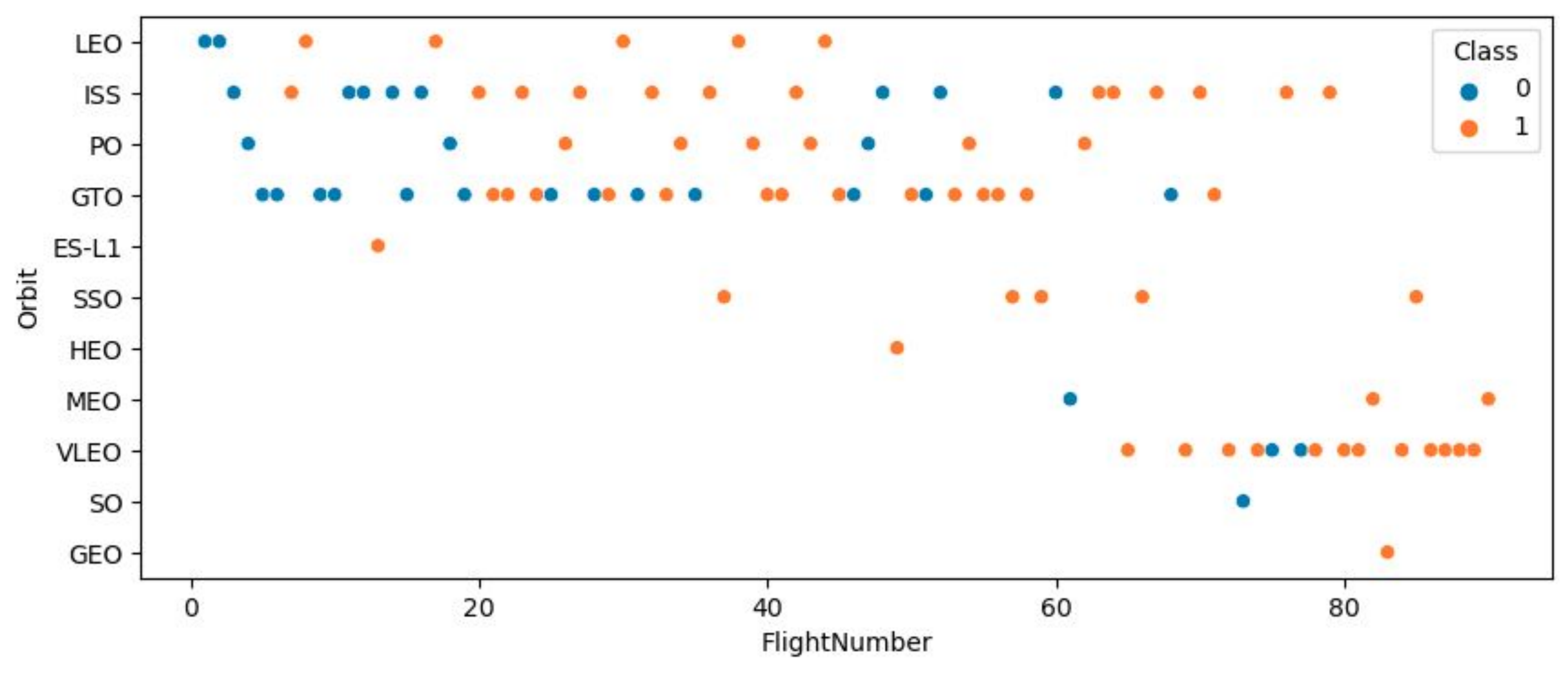
Payload vs. Launch Site



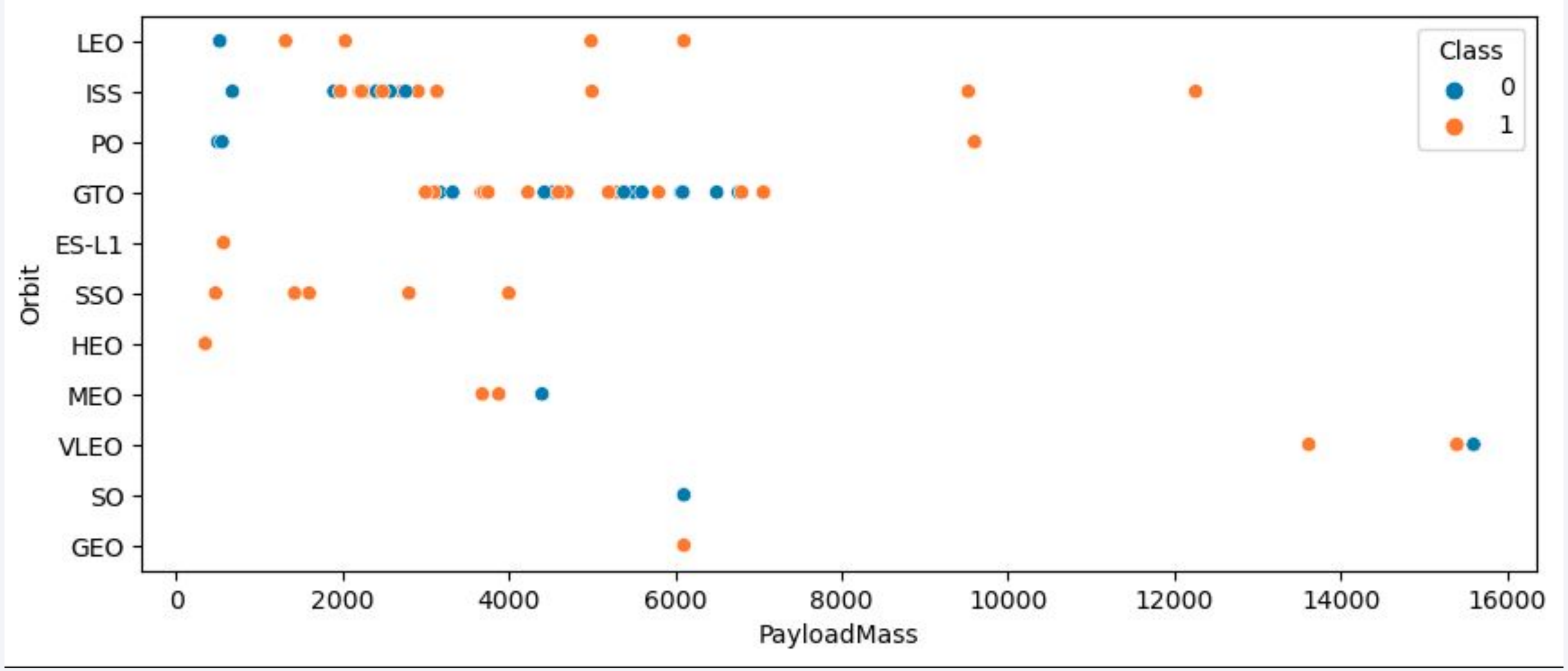
Success Rate vs. Orbit Type



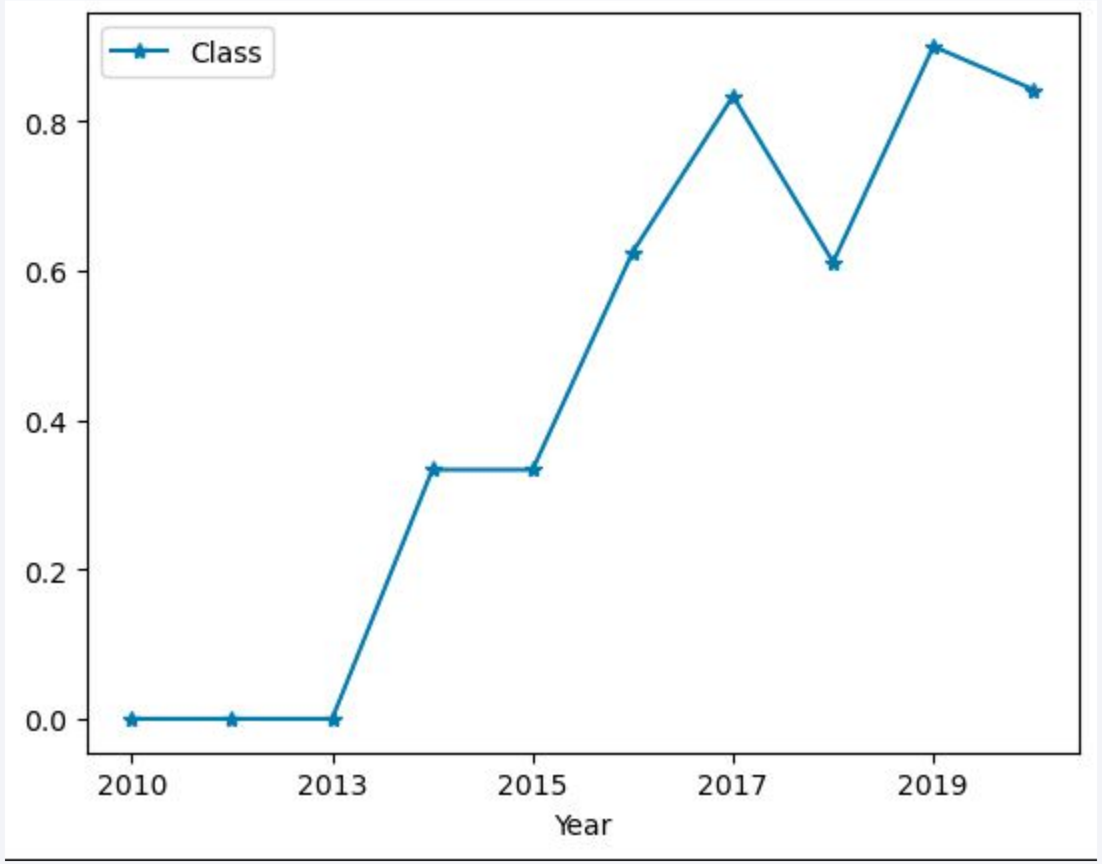
Flight Number vs. Orbit Type



Payload vs. Orbit Type



Launch Success Yearly Trend



All Launch Site Names

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

Launch Site Names Begin with 'CCA'

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

Total Payload Mass

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

```
%%sql

select
  sum(PAYLOAD_MASS__KG_) total_payload_mass
from SPACEXTBL
where Customer = 'NASA (CRS)'
```

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

total_payload_mass
45596

Average Payload Mass by F9 v1.1

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

```
%%sql

select
  avg(PAYLOAD_MASS__KG_) avg_payload_mass
from SPACEXTBL
where Booster_Version like '%F9 v1.1%'
```

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

avg_payload_mass
2534.6666666666665

First Successful Ground Landing Date

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

Hint:Use min function

```
%%sql

select
  min(Date) date
from SPACEXTBL
where Landing_Outcome='Success (ground pad)'
```

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

date
2015-12-22

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

```
%%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.
```

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

Total Number of Successful and Failure Mission Outcomes

List the total number of successful and failure mission outcomes

```
%%sql

select
    case
      when Mission_Outcome like '%success%' then 'success'
      else 'failure'
    end outcome,
    count(1) c
from SPACEXTBL
group by 1
```

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

outcome	c
failure	1
success	100

Boosters Carried Maximum Payload

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

```
%%sql

select distinct
    Booster_Version
from SPACEXTBL
where PAYLOAD_MASS__KG_= (select max(PAYLOAD_MASS__KG_) max_mass 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

2015 Launch Records

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) month,
    Landing_Outcome,
    Booster_Version,
    Launch_Site
from SPACEXTBL
where Date like '2015%'
 and Landing_Outcome = 'Failure (drone ship)'
```

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

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

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

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(*) c
from SPACEXTBL
where Date between '2010-06-04' and '2017-03-20'
group by 1 order by 2 desc
```

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

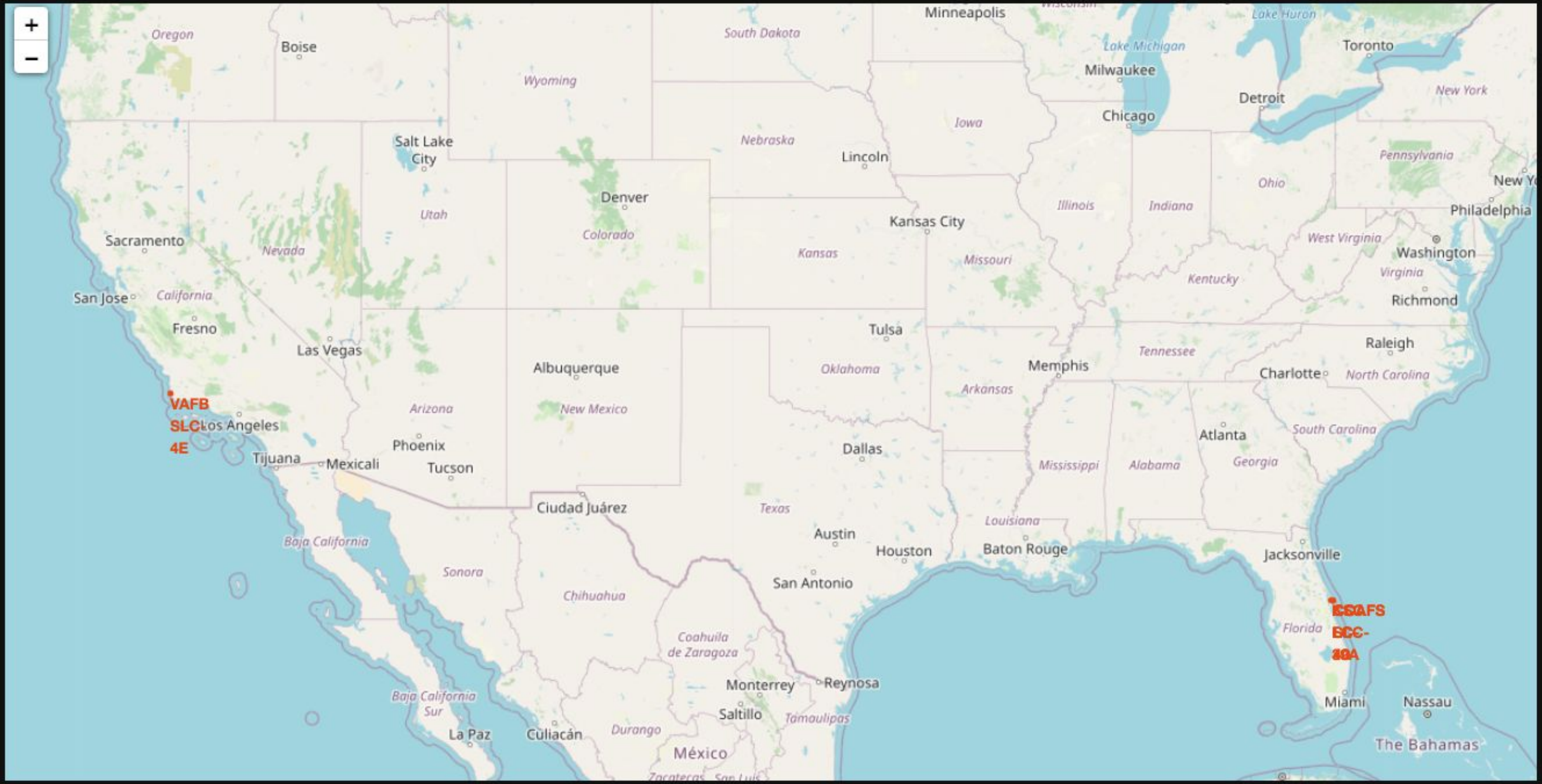
Landing_Outcome	c
No attempt	10
Success (drone ship)	5
Failure (drone ship)	5
Success (ground pad)	3
Controlled (ocean)	3
Uncontrolled (ocean)	2
Failure (parachute)	2
Precluded (drone ship)	1

Section 3

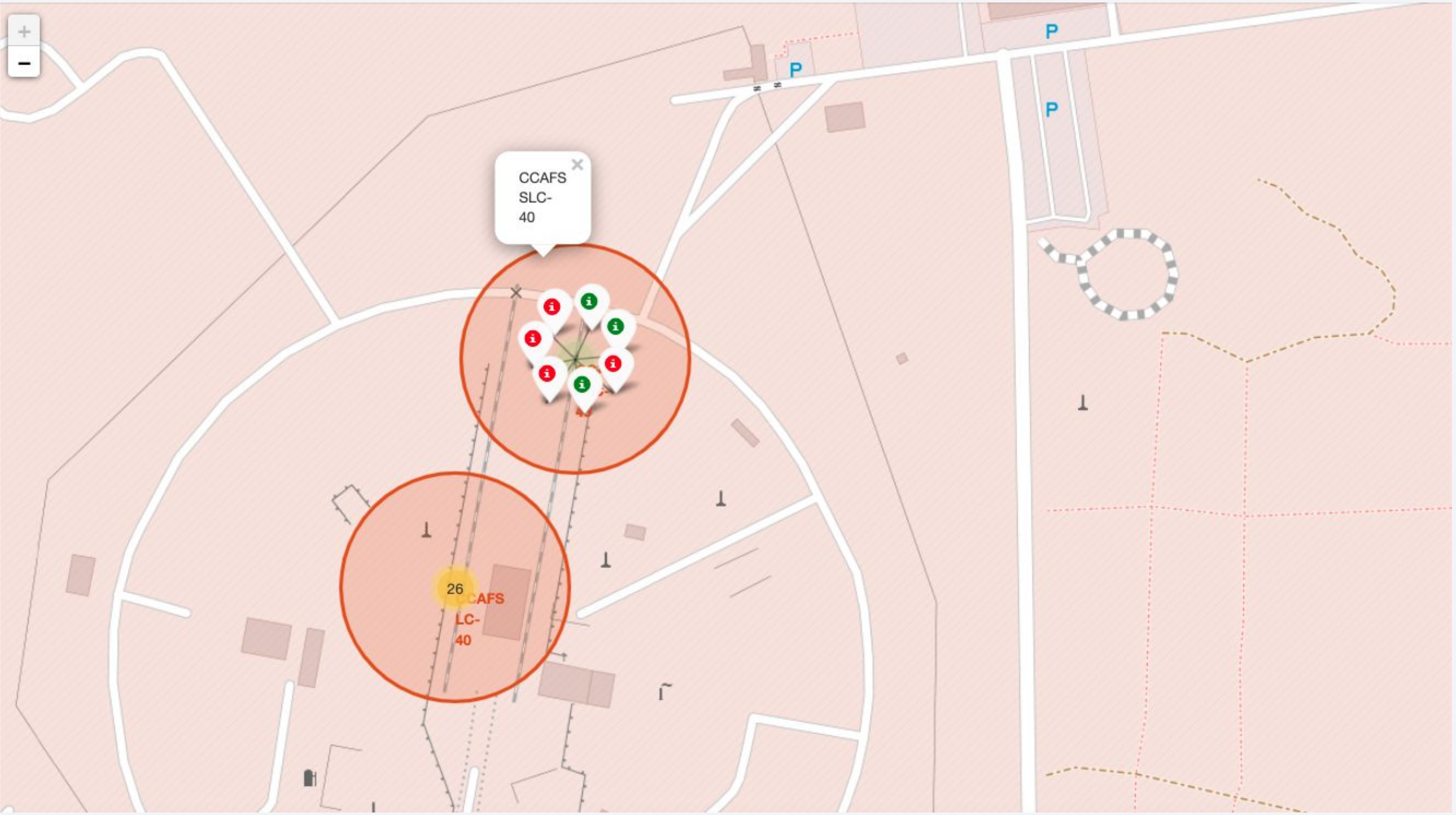
Launch Sites Proximities Analysis

Global map markers denoting the locations of all launch sites.

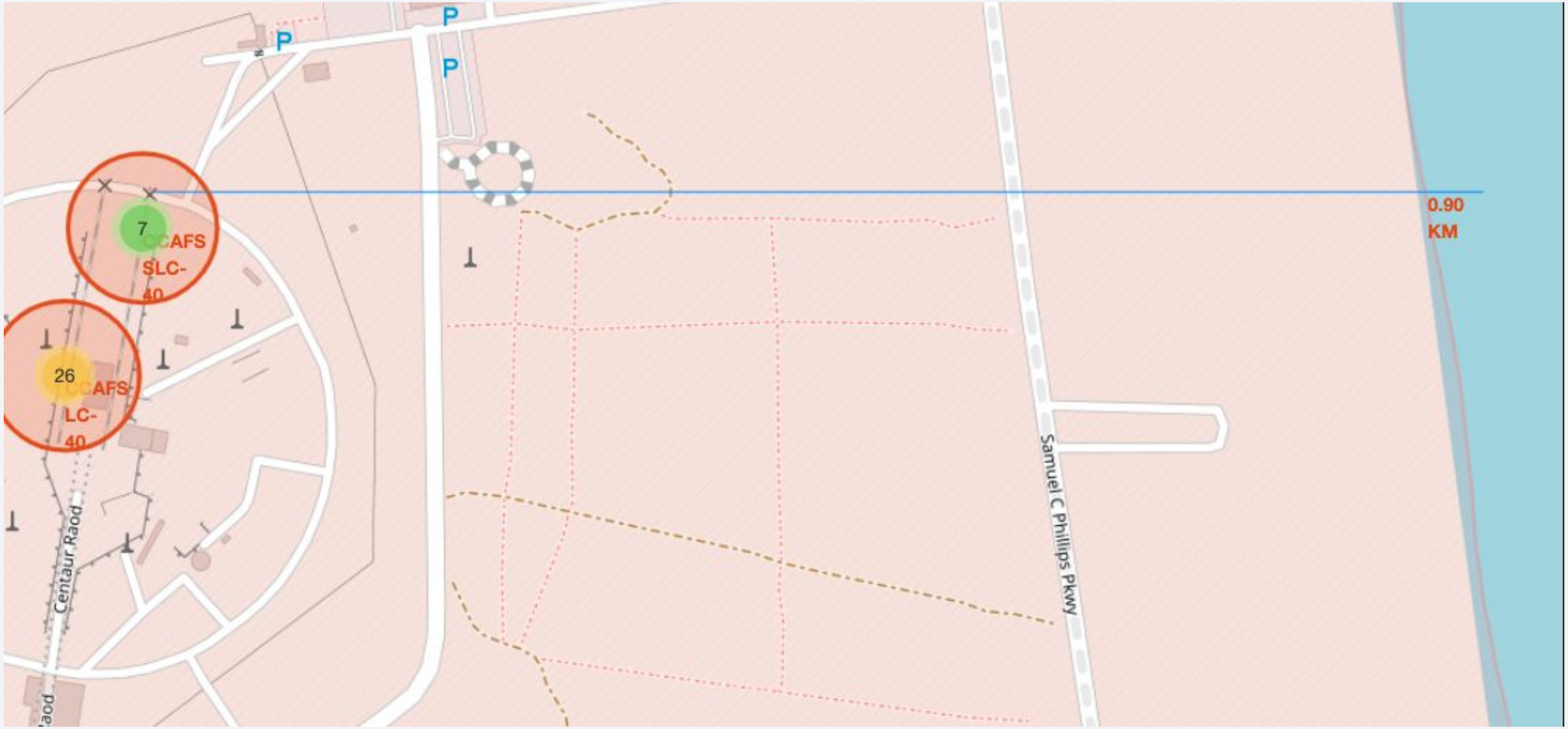
The generated map with marked launch sites should look similar to the following:



Color-labeled markers indicating different launch sites on a map.



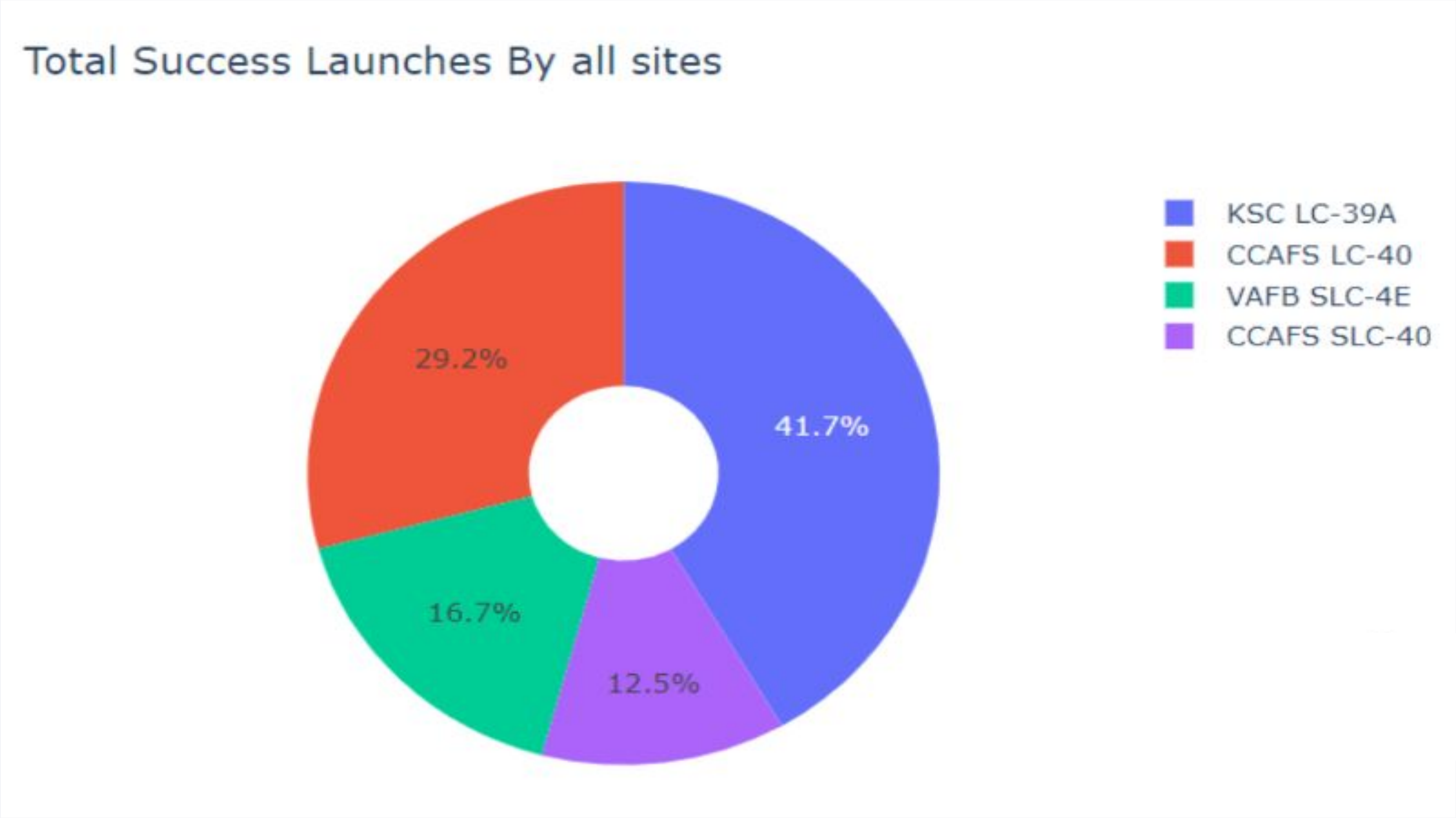
Distance from Launch Sites to Landmarks



Section 4

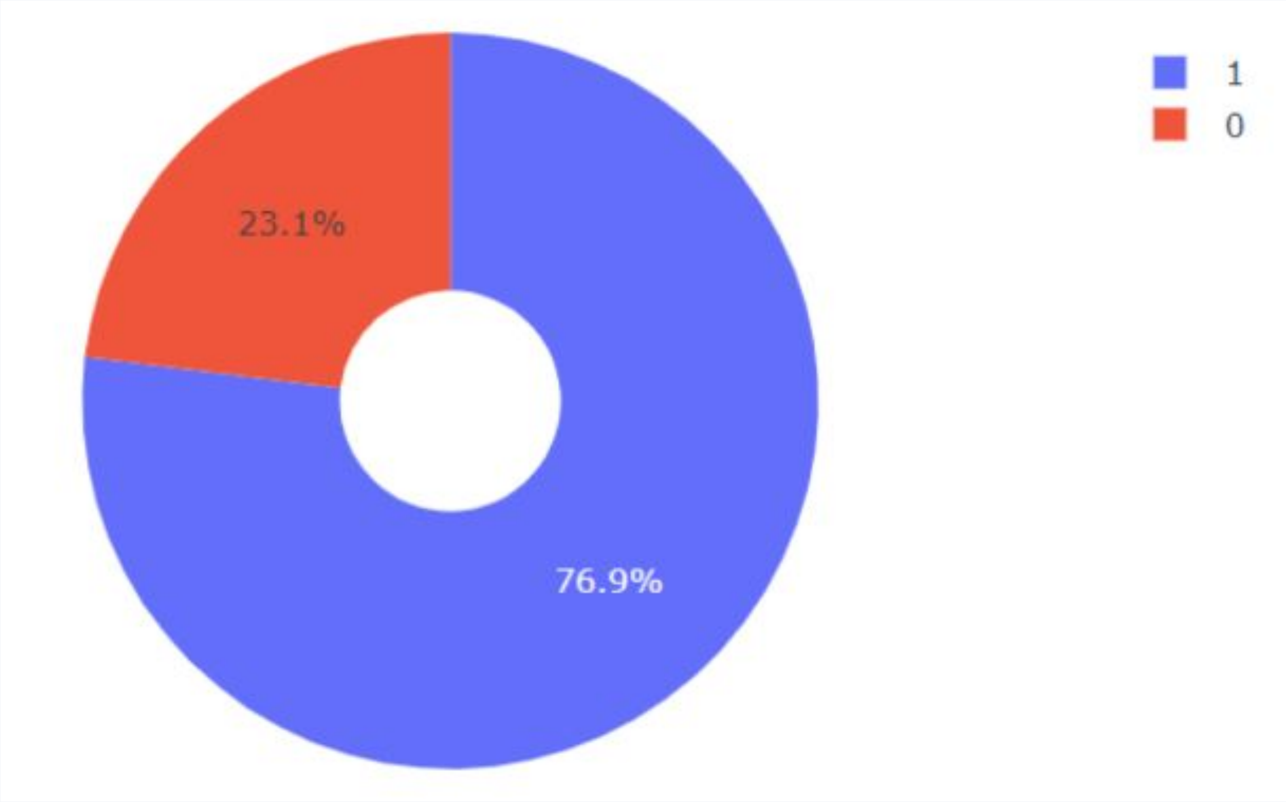
Build a Dashboard with Plotly Dash

A visually engaging pie chart illustrating the success percentages attained by individual launch sites.

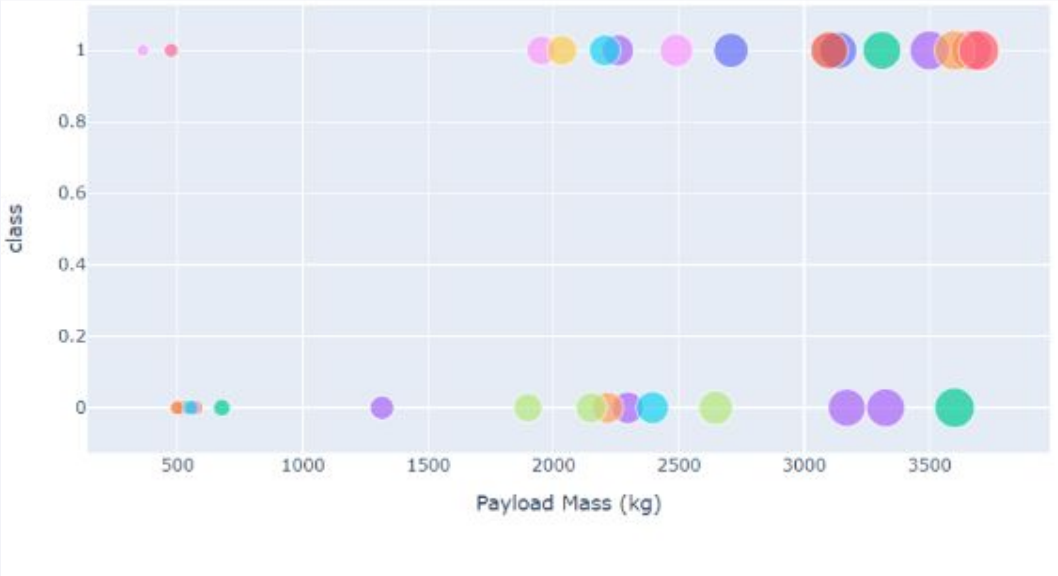


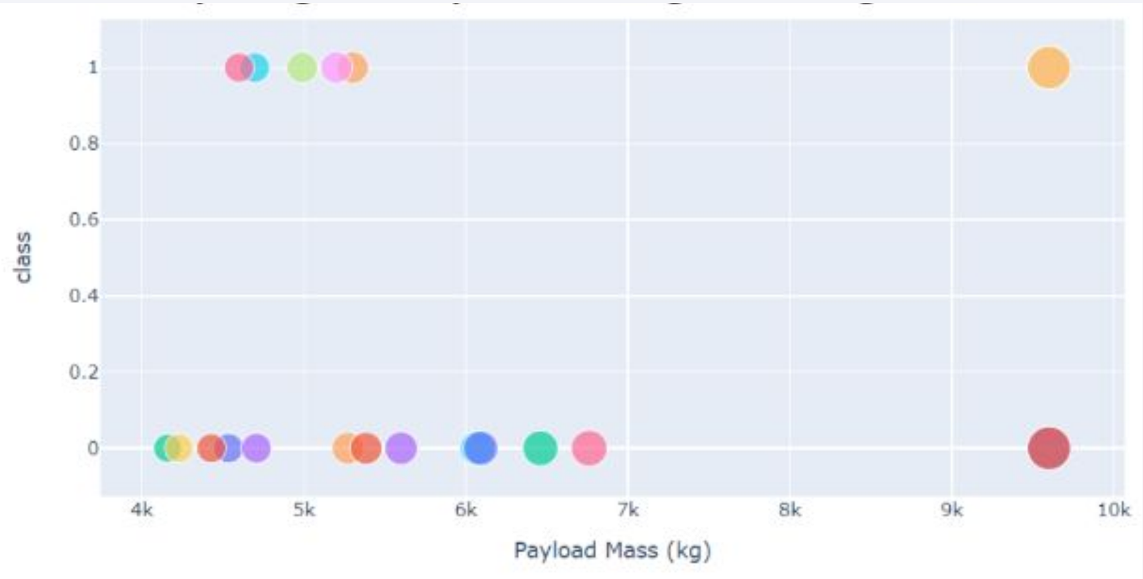
A pie chart highlighting the launch site with the highest success ratio.

Kennedy Space Center Launch Complex 39A (KSC LC-39A) boasts the highest success rate among launch sites.



A scatter plot illustrating the relationship between Payload and Launch Outcome for all launch sites, with the ability to select different payload ranges using a range slider.



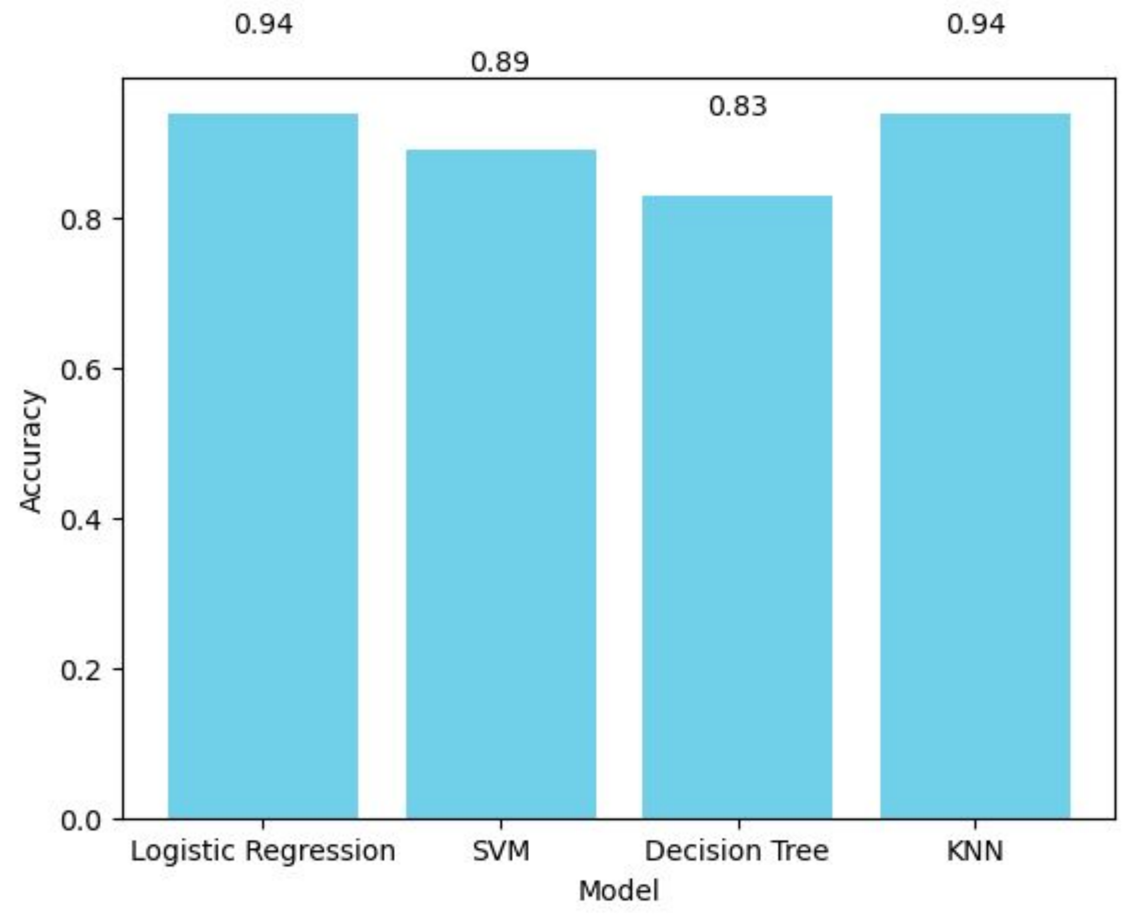


Section 5

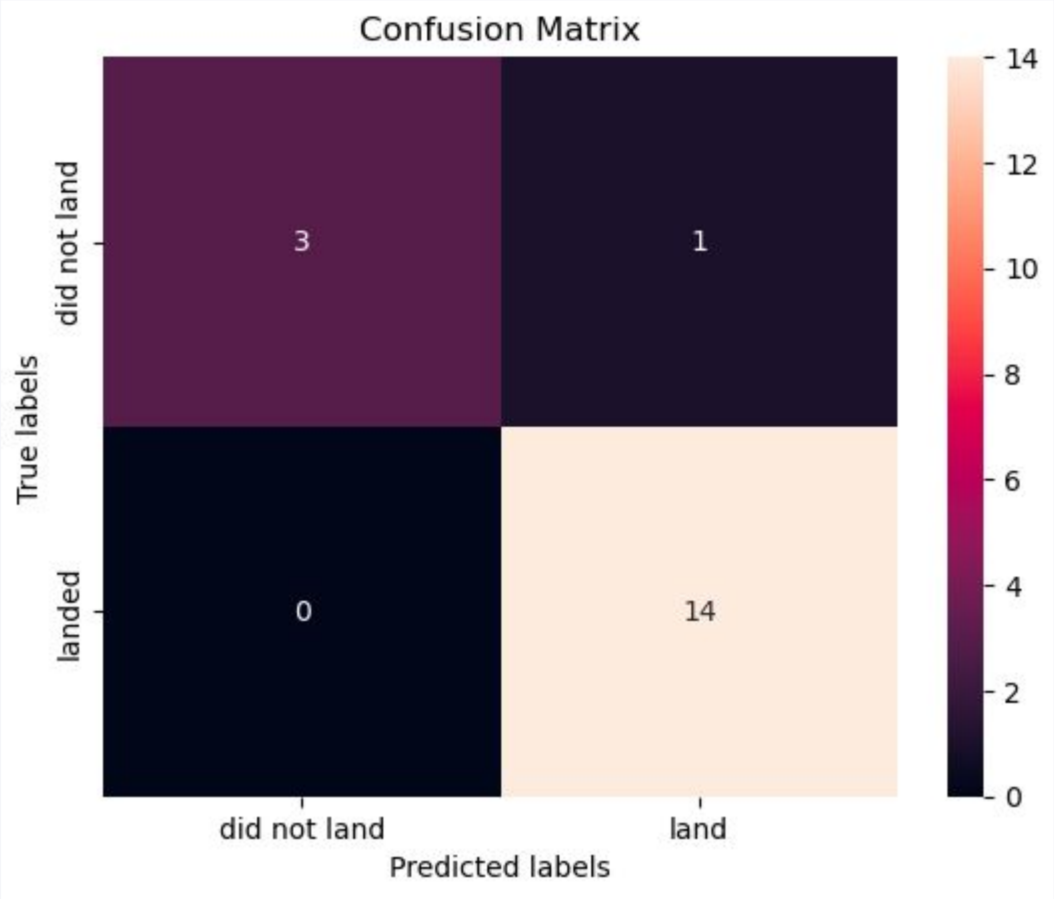
Predictive Analysis (Classification)

Classification Accuracy

LR and KNN exhibit identical levels of accuracy.



Confusion Matrix



Conclusions

- The greater the flight count at a launch site, the higher its success rate.
- The launch success rate showed an upward trend from 2013 to 2020.
- Among all sites, KSC LC-39A recorded the highest number of successful launches.
- LR and KNN exhibit identical levels of accuracy and emerged as the most effective machine learning algorithm in our case.

Thank you!