



IBM Developer  
SKILLS NETWORK

# Winning Space Race with Data Science

<Ling Wang>  
<Dec 12, 2025>



# Outline

---

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

# Executive Summary

---

- Summary of methodologies:
  - Data collection
  - Data wrangling
  - Exploratory data analysis (EDA)
  - Interactive visual analysis
  - Predictive analysis (classification)
- Summary of all results:
  - EDA results
  - Geospatial analysis
  - Interactive dashboard
  - Predictive analysis of classification models

# Introduction

---

- **Background**

SpaceX advertises Falcon 9 rocket launches on its website with a cost of 62 million dollars whereas other providers cost upward of 165 million dollars each, much of the savings is because Space X can reuse the first stage. Therefore if we can determine if the first stage will land, we can determine the cost of a launch. This information can be used if an alternate company wants to bid against SpaceX for a rocket launch.

- **Problems**

- Any patterns in the data?
- What would be the label for training supervised models.
- Best predictive model for falcon 9 first stage successful landing.





Section 1

# Methodology

# Methodology

---

## Executive Summary

- Data collection methodology:
  - Collection using SpaceX REST API and web scraping from wikipedia page.
- Perform data wrangling
  - Filtering data, dealing with missing data and one hot encoding for categorical data.
  - Preparation for following analysis and modeling.
- Perform exploratory data analysis (EDA) using visualization and SQL
- Perform interactive visual analytics using Folium and Plotly Dash
- Perform predictive analysis using classification models
  - Build logistic regression, decision tree, svm, and knn model using classification method (scikit learn)
  - Evaluate model performance using accuracy, precision, F1, recall (scikit learn)

# Data Collection

---

1. Request from SpaceX API
  - a. Collect past launch data via open-source API (GET request) to get JSON data.
  - b. Filled missing value of payload mass column with mean.
2. Web scraping
  - a. Request Falcon 9 launch data from wikipedia page.
  - b. Extracted all the column names from the HTML table.
  - c. Transformed the table into a Panda dataframe.

# Data Collection – SpaceX API

---

- **GET /launches** → retrieve launch info
- **parse JSON** → extract fields like **flight\_number**, **launch\_site**
- **store in DataFrame** → save data in pandas for analysis

GitHub url:

<https://github.com/lingwanggatech/IBM-SpaceX-Capstone/blob/main/1-spacex-data-collection-api.ipynb>





# Data Collection - Scraping

---

## Workflow

1. Send HTTP request (GET)
2. Receive HTML response
3. Parse HTML (BeautifulSoup)
4. Extract data (soup.title attributes)
5. Extract relevant column from HTML table header
6. Parse the launch HTML tables
7. Store data in Pandas dataframe

GitHub url:

<https://github.com/lingwanggatech/IBM-SpaceX-Capstone/blob/main/2-spacex-web scraping.ipynb>



# Data Wrangling

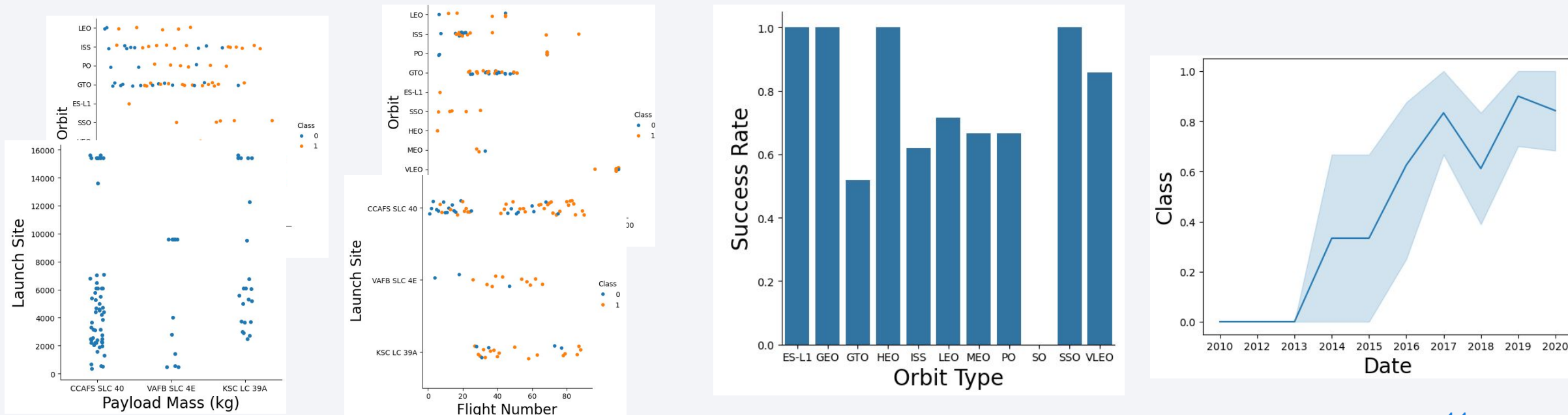
---

1. Calculate the number of launchers on each site(`.value_counts()`)
2. Calculate the occurrence of each orbit (`.value_counts()`)
3. Determine the landing outcomes and create a landing outcome label
4. Determine the success rate (`.mean()`)
5. Save processed data to csv.

# EDA with Data Visualization

**scatter plot, bar plot and line plot** to visualize:

1. Relationship among orbit type, payload mass, flight number, launch site
2. Relationship between success rate of each orbit type
3. Trend of successful launch.



# EDA with SQL

---

- Using bullet point format, summarize the SQL queries you performed
- Find unique launch sites (Distinct)
- Display 5 records where launch site begin with “CCA” (Limit 5, Like ‘CCA%’)
- Total payload mass by NASA (CRS) (sum)
- Average payload mass by booster version is F9 v1.1
- First successful landing date in ground pad (min(Date))
- Listing names of booster have success in drone ship and  $4000 < \text{payload mass} < 6000$
- Total number of successful and failure outcomes (count)
- Find booster version with max(payload mass)
- List records has failure landing in 2025
- Count landing outcome between 06-04-2016 and 03-20-2017

# Build an Interactive Map with Folium

---

- 

circle,marker

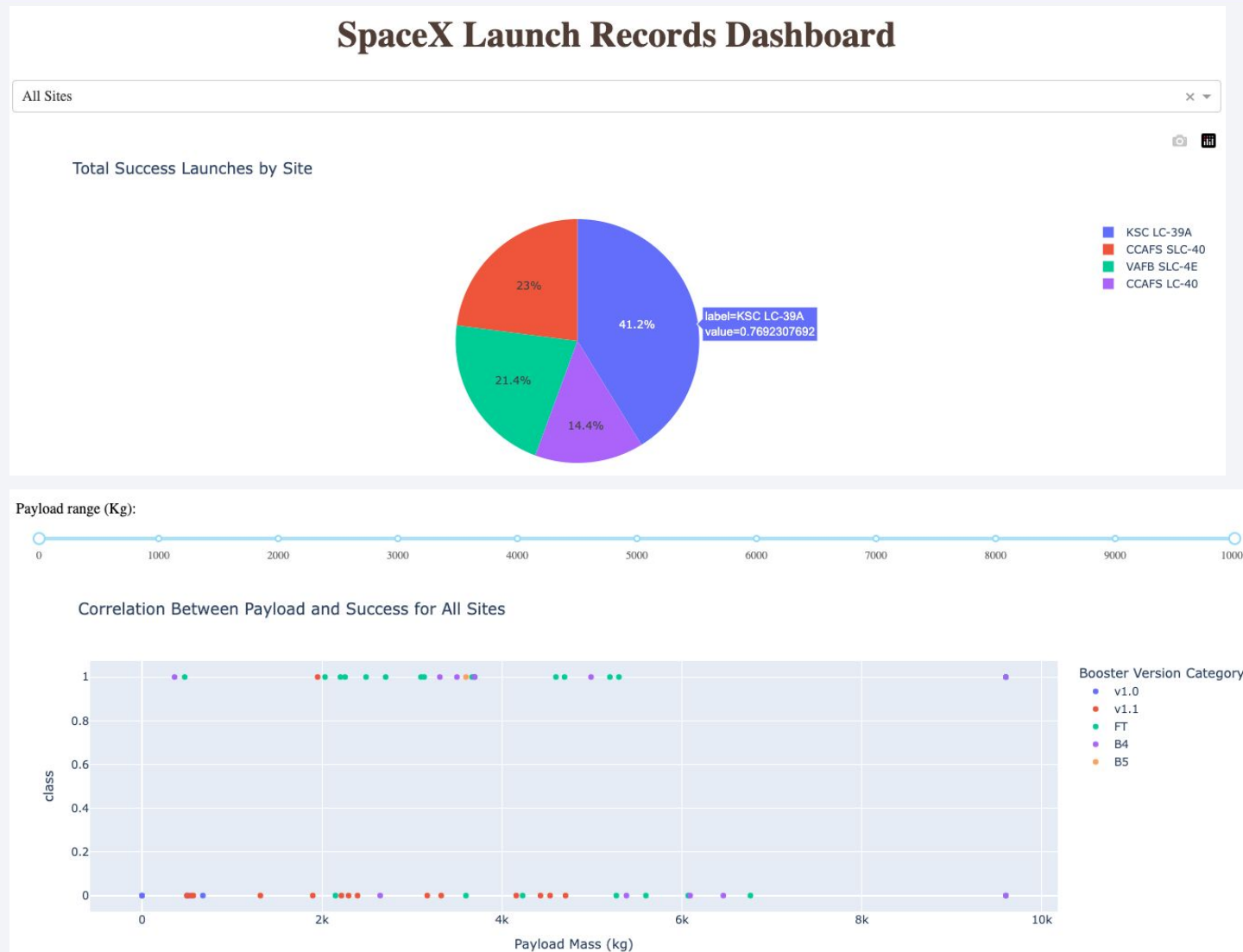
Mark all launch sites on map

Mark the success/failed launches for each sites on map

Calculate the distances between a launch site to its proximities.



# Build a Dashboard with Plotly Dash



# Predictive Analysis (Classification)

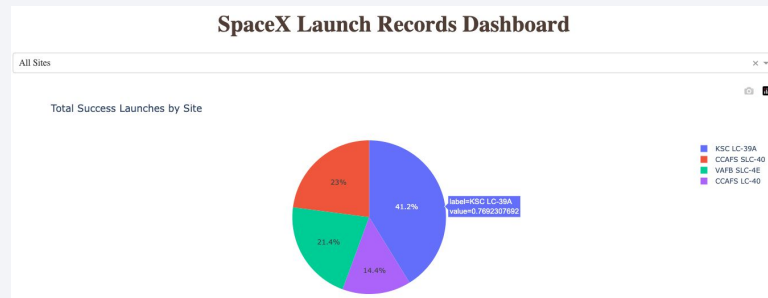
---

- Split data into testing (20%) and training set (80%)
- Grid search to find the best hyperparameters
- Make prediction
- Evaluate the model using accuracy, confusion matrix.
- Compare models and find the best one.



# Results

- Exploratory data analysis results
  1. Launch success has improved over time
  2. KSC LC-39A has the highest success rate
- Interactive analytics demo in screenshots



- Predictive analysis results
  1. All four models have similar performance.
  2. All models' accuracy is 83%, so I'd choose the logistic regression for less time complexity.



The background of the slide is an abstract composition. It features a solid blue area on the left side, which transitions into a dynamic pattern of diagonal streaks in shades of blue and red on the right. Overlaid on these streaks is a faint, light-blue grid pattern, giving the impression of a digital or data-driven environment.

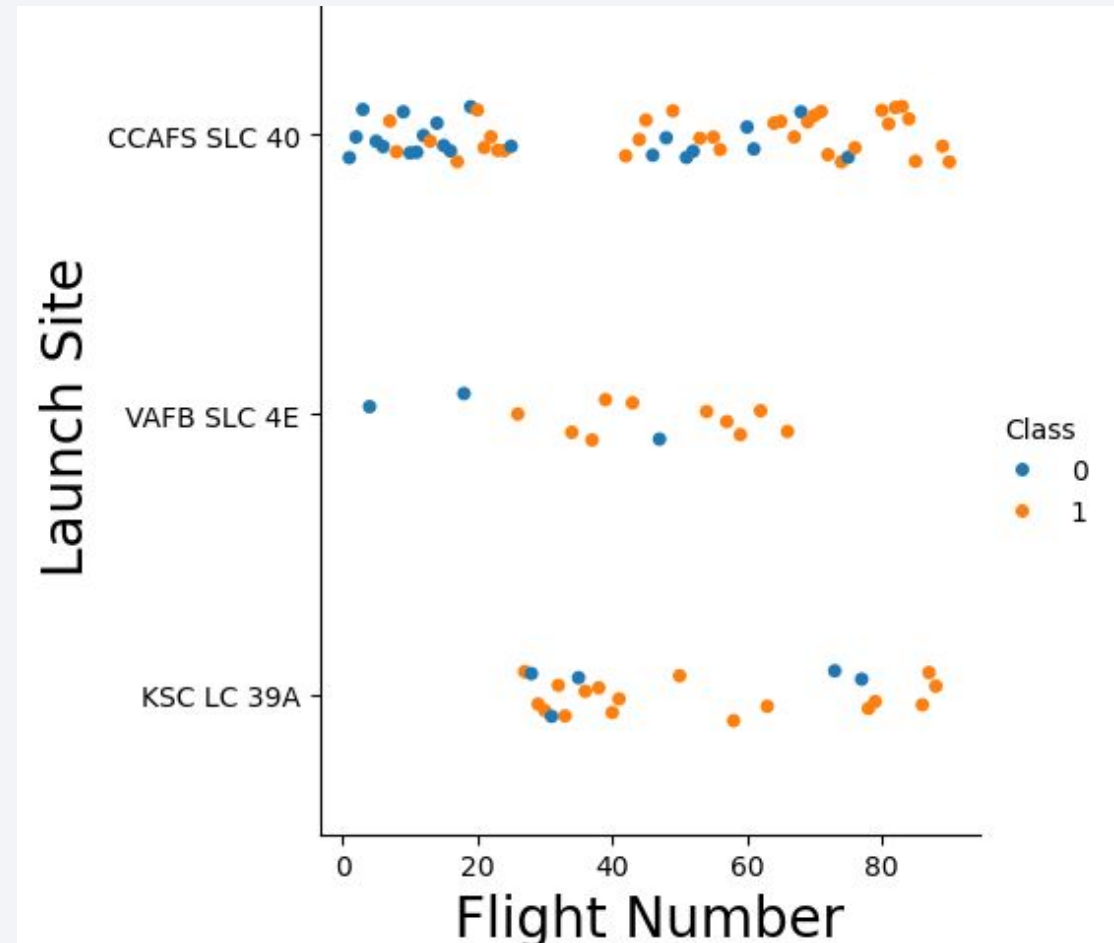
Section 2

# Insights drawn from EDA



# Flight Number vs. Launch Site

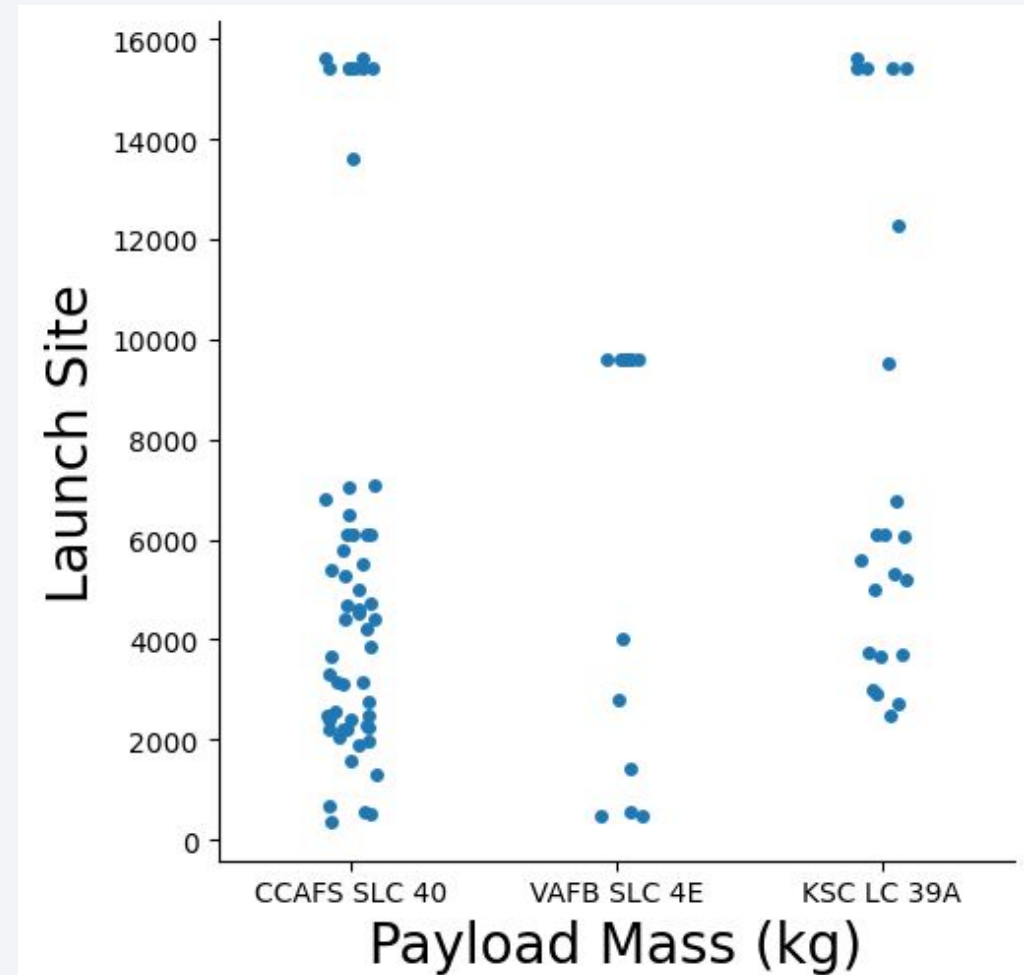
- There is not clear pattern of two classes among Launch site and Flight number.





# Payload vs. Launch Site

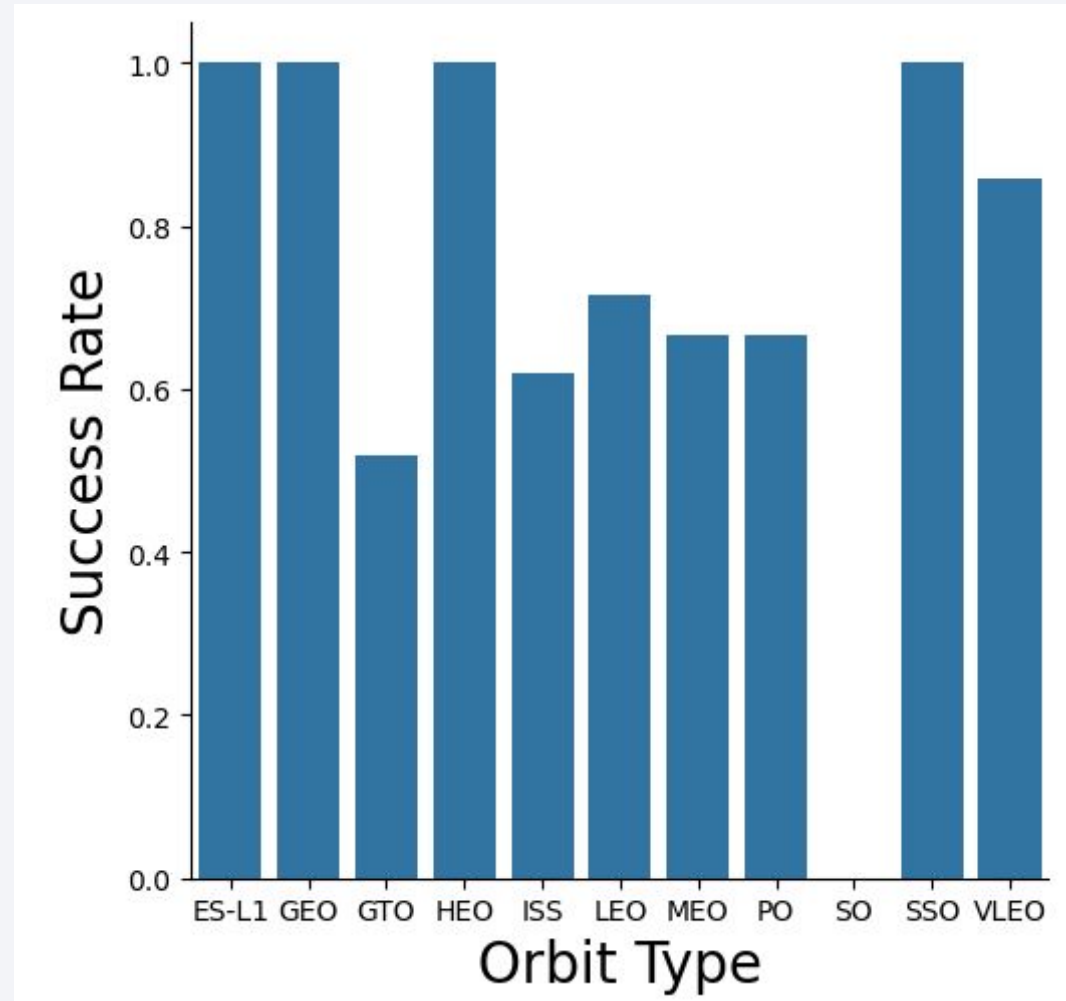
- For VAFB-SLC launchsite there are no rockets launched for heavy payload mass ( $> 100,000$ )



# Success Rate vs. Orbit Type

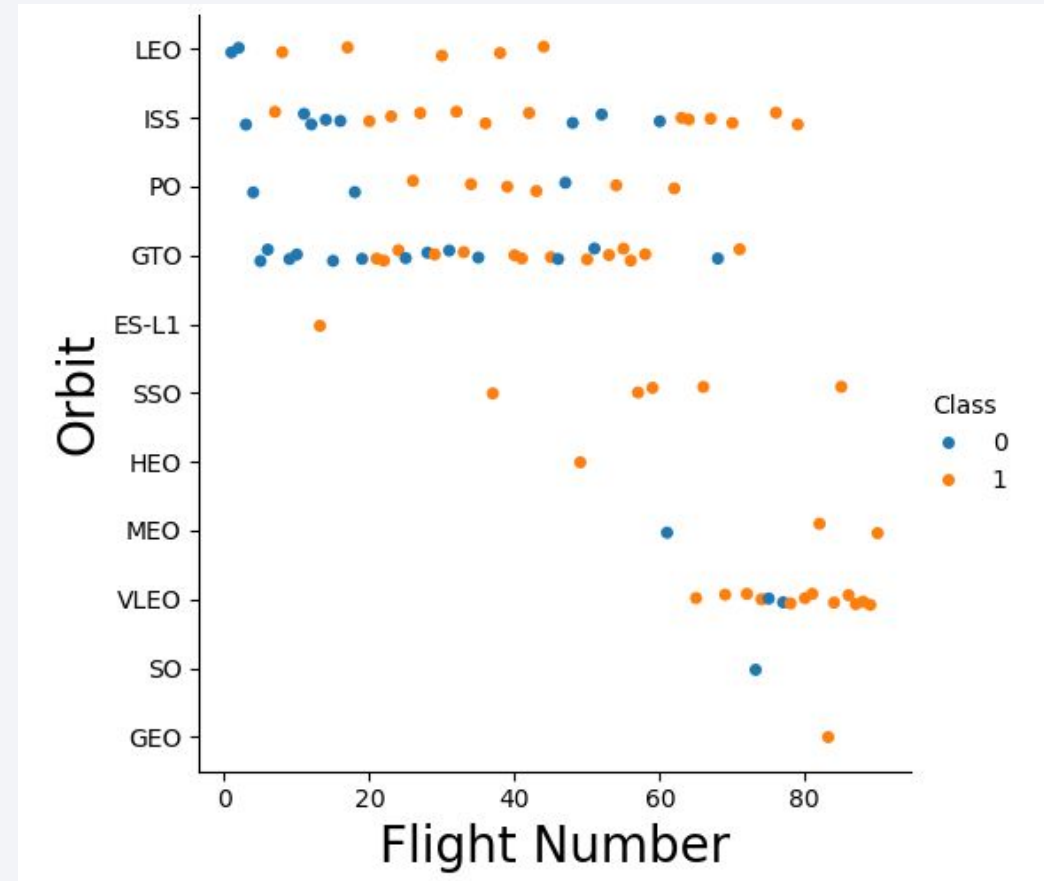
---

- ES\_L1, GEO, HEO and SSO have the highest (100%) success rate.



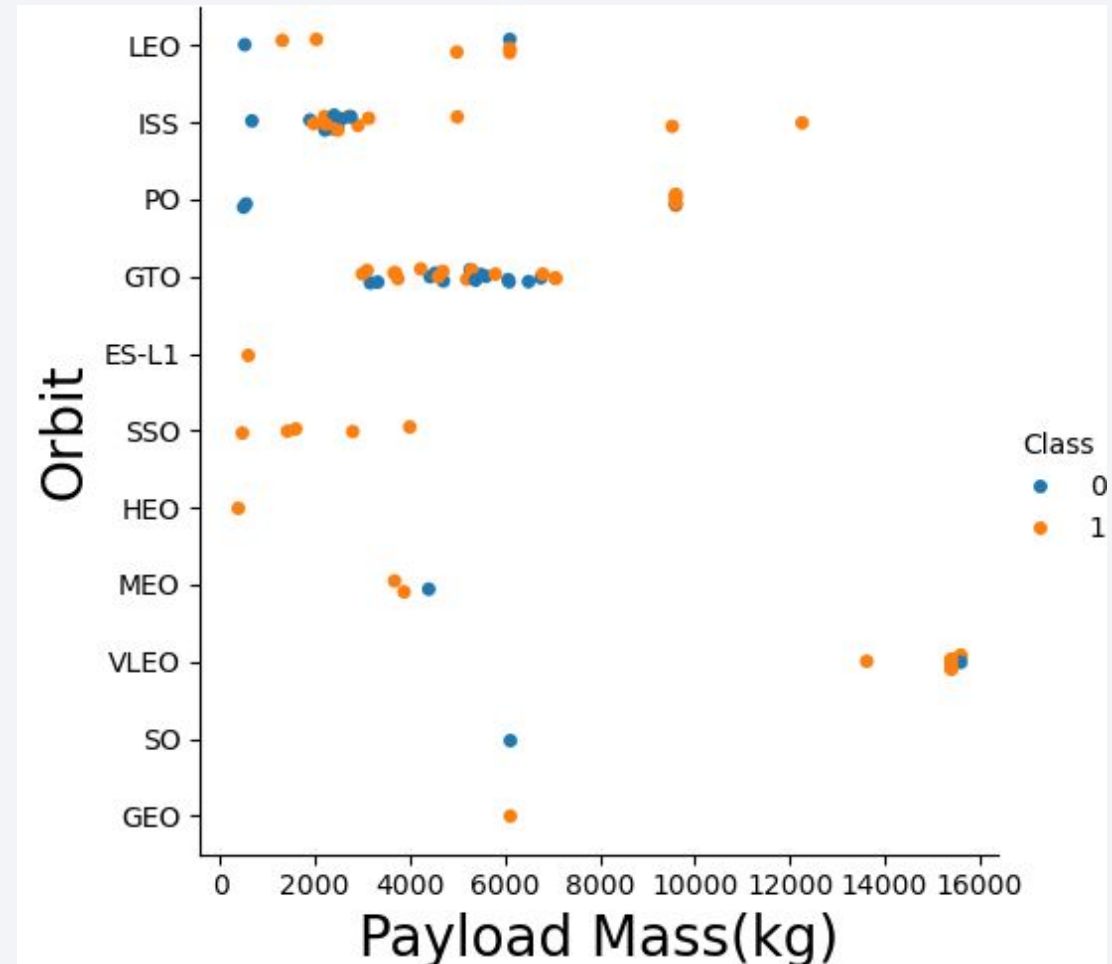
# Flight Number vs. Orbit Type

- In LEO orbit, success seems to be related to the number of flights.
- In GTO orbit, there appears to be no relationship between flight number and success



# Payload vs. Orbit Type

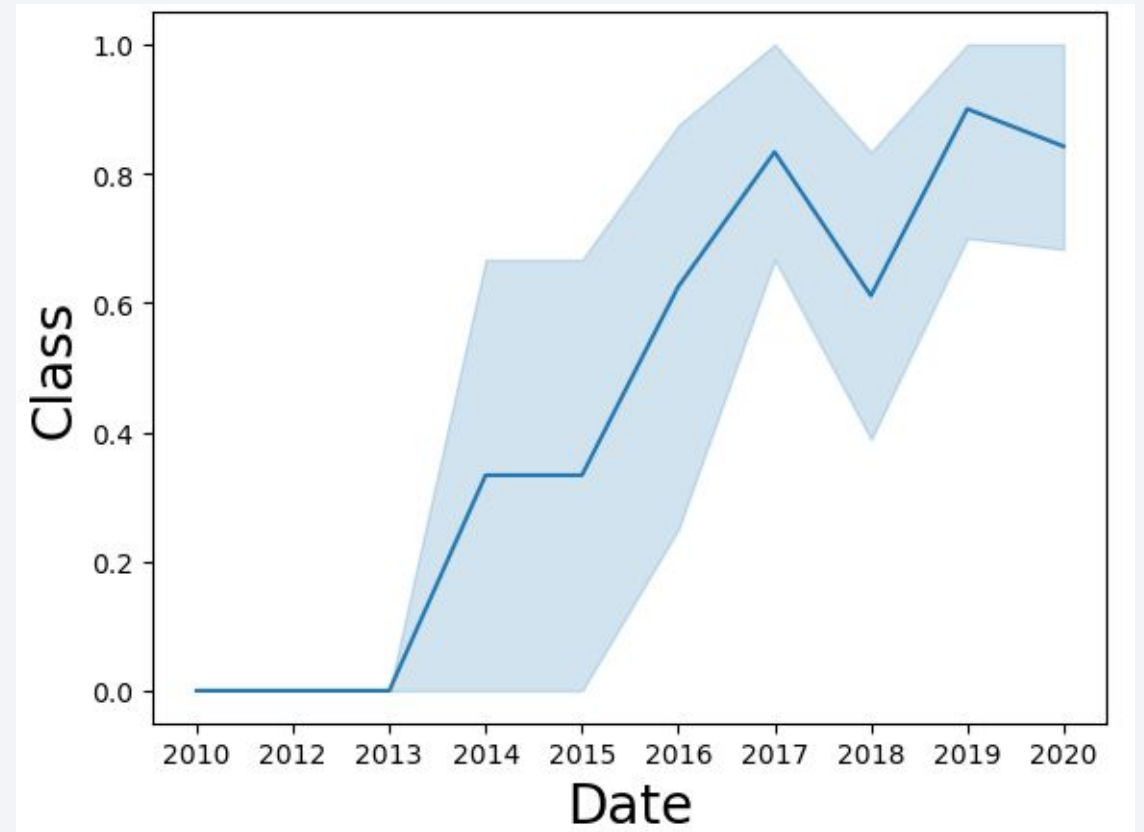
- For Polar, LEO and ISS, with heavy payloads the successful landing rate are more.



# Launch Success Yearly Trend

---

- Success rate kept increasing since 2013 until 2020.





# All Launch Site Names

---

- Names of the unique launch sites
  - CCAFS LC-40
  - CCAFS SLC-40
  - KSC LC-39A
  - VAFB SLC-4E
- USing **DISTINCT** to query the unique launch site name

```
[20]: %sql SELECT DISTINCT LAUNCH_SITE FROM SPACEXTBL
```

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

```
Done.
```

```
[20]: Launch_Site
```

```
CCAFS LC-40
```

```
VAFB SLC-4E
```

```
KSC LC-39A
```

```
CCAFS SLC-40
```

# Launch Site Names Begin with 'CCA'

- Find 5 records where launch sites begin with `CCA`
- Limit 5 to extract 5 records, LIKE in where clause to filter column having 'CCA'

```
In [11]: %%sql SELECT *
          FROM SPACEXTBL
          WHERE LAUNCH_SITE LIKE 'CCA%' LIMIT 5;
```

\* sqlite:///my\_data1.db  
Done.

Out[11]:

Date	Time (UTC)	Booster_Version	Launch_Site	Payload	PAYLOAD_MASS_KG_	Orbit	Customer	Mission_Outcome	Landing_
2010-06-04	18:45:00	F9 v1.0 B0003	CCAFS LC-40	Dragon Spacecraft Qualification Unit	0	LEO	SpaceX	Success	Failure (p
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 (p
2012-05-22	7:44:00	F9 v1.0 B0005	CCAFS LC-40	Dragon demo flight C2	525	LEO (ISS)	NASA (COTS)	Success	N
2012-10-08	0:35:00	F9 v1.0 B0006	CCAFS LC-40	SpaceX CRS-1	500	LEO (ISS)	NASA (CRS)	Success	N
2013-03-01	15:10:00	F9 v1.0 B0007	CCAFS LC-40	SpaceX CRS-2	677	LEO (ISS)	NASA (CRS)	Success	N

# Total Payload Mass

---

- Sum to calculate total payload mass, where to find certain condition (customer = 'NASA (CRS)')
- Total is 45,596 kg.

```
In [12]: %sql select sum(PAYLOAD_MASS_KG_) from SPACEXTBL where CUSTOMER = 'NASA (CRS)'
```

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

```
Out[12]: sum(PAYLOAD_MASS_KG_)
         45596
```

# Average Payload Mass by F9 v1.1

---

- avg() to calculate the average payload mass carried by booster version F9 v1.1 using **where** condition clause
- Average is 2928.4 kg.

```
In [13]: %sql select avg(PAYLOAD_MASS__KG_) from SPACEXTBL where BOOSTER_VERSION like '%F9 v1.1';
* sqlite:///my_data1.db
Done.
Out[13]: avg(PAYLOAD_MASS__KG_)
          2928.4
```

# First Successful Ground Landing Date

---

- min(Date)
- The first date of successful landing is 2015-12-12.
-

## Successful Drone Ship Landing with Payload between 4000 and 6000

---

- List the names of boosters which have successfully landed on drone ship and had payload mass greater than 4000 but less than 6000

- ☐ F9 FT B1022
- ☐ F9 FT B1026
- ☐ F9 FT B1021.2
- ☐ F9 FT B1031.2

```
In [15]: %sql select Booster_Version from SPACEXTBL where Landing_Outcome = "Success (drone ship)" and PAYLOAD_MASS__KG_
```

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

```
Out[15]: Booster_Version
```

```
F9 FT B1022
```

```
F9 FT B1026
```

```
F9 FT B1021.2
```

```
F9 FT B1031.2
```

# Total Number of Successful and Failure Mission Outcomes

---

- 1 failed in flight, 99 succeeded.

```
In [16]: %sql select Mission_Outcome, count(Mission_Outcome) as number_mission from SPACEXTBL group by Mission_Outcome
```

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

```
Out[16]:
```

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



# Boosters Carried Maximum Payload

---

- List the names of the booster which have carried the maximum payload mass

```
In [17]: %%sql SELECT BOOSTER_VERSION
FROM SPACEXTBL
WHERE PAYLOAD_MASS_KG_ = (SELECT MAX(PAYLOAD_MASS_KG_) FROM SPACEXTBL);
```

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

```
Out[17]: 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 failed landing\_outcomes in drone ship, their booster versions, and launch site names for in year 2015

```
In [18]: %%sql SELECT substr(Date, 6,2) as Month, BOOSTER_VERSION, Launch_Site
FROM SPACEXTBL
WHERE substr(Date,0,5) = '2015' and Landing_Outcome = "Failure (drone ship)";
```

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

```
Out[18]:
```

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

---

- 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

```
In [19]: %%sql SELECT Landing_Outcome, count(Landing_Outcome) as cnt_landing
FROM SPACEXTBL
where Date between '2010-06-04' and '2017-03-20'
group by landing_outcome
Order by cnt_landing DESC
```

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

```
Out[19]:
```

Landing_Outcome	cnt_landing
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

A satellite view of Earth from space, showing the curvature of the planet and city lights at night. The image is a composite of a dark blue sky with stars and a view of the Earth's surface from space. The Earth's surface is mostly dark, with a thin layer of atmosphere visible along the horizon. The city lights are concentrated in the lower right quadrant, showing a dense network of urban areas. The text "Section 3" is overlaid on the left side of the image.

Section 3

# Launch Sites Proximities Analysis

# Launch Sites

---

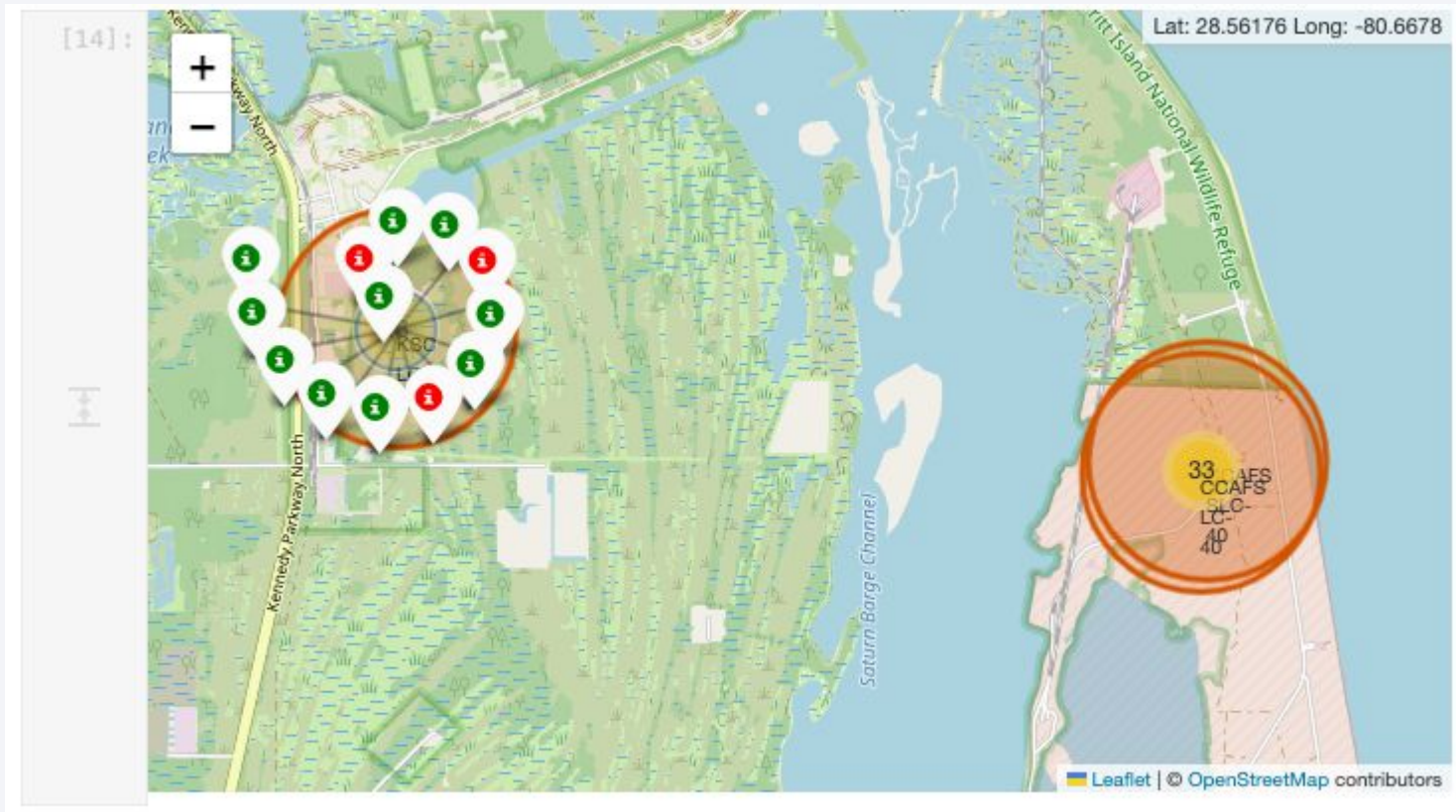
- The closer to equator, the easier to launch to equatorial orbit.





# Launch Outcomes

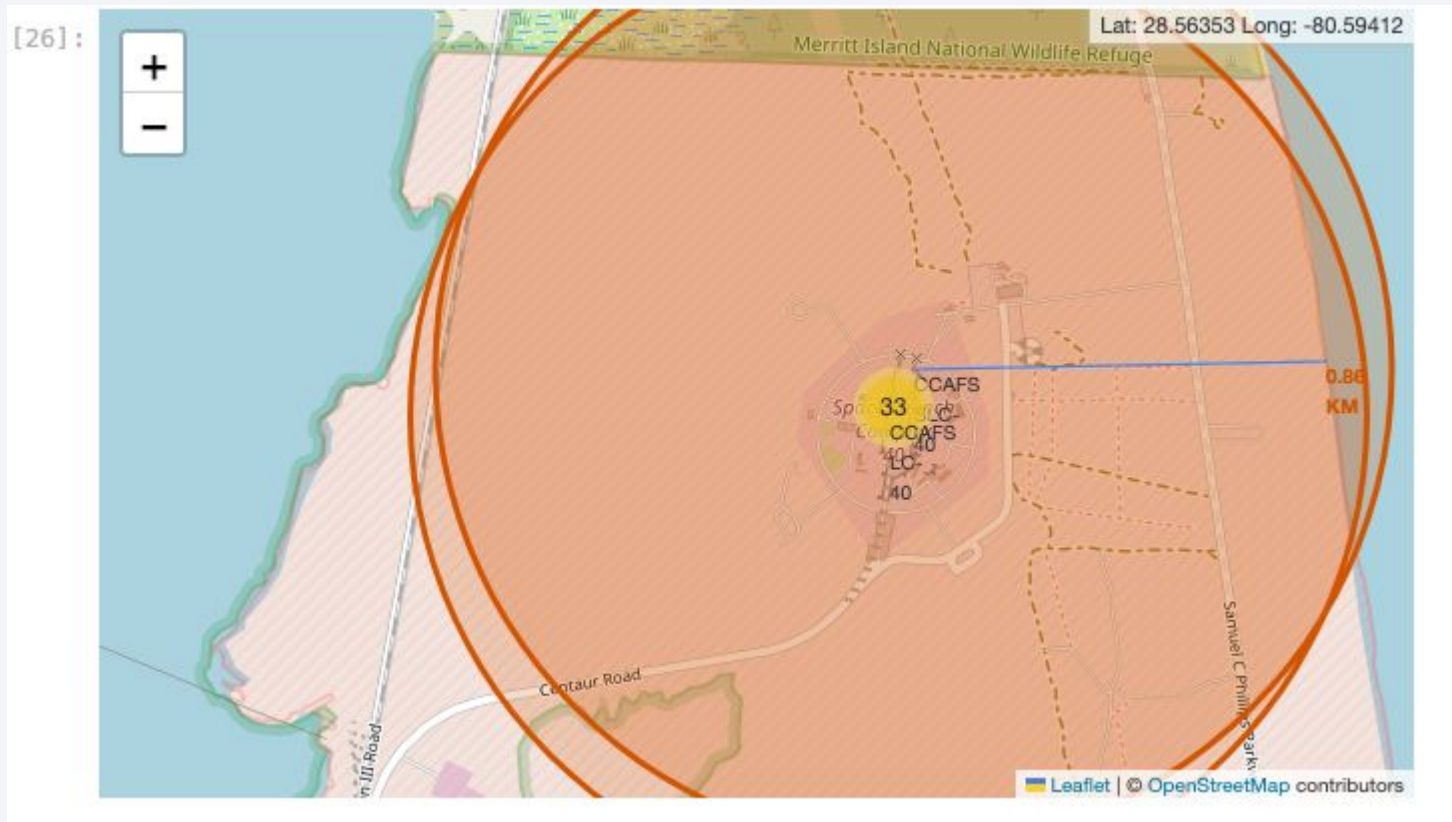
- Green markers for successful launches
- Red markers for failed launches





# Add distance to Proximities

- For CCAFS SLC-40, .0.86 km from nearest coastline



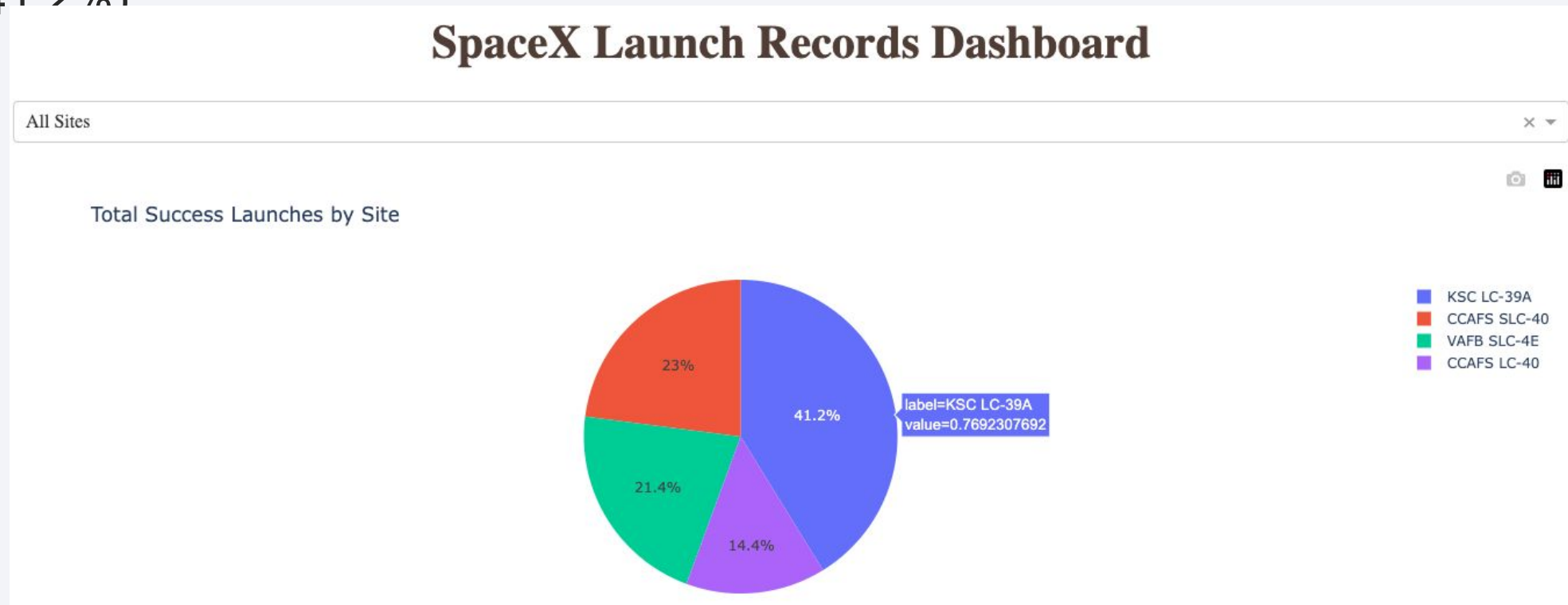


Section 4

# Build a Dashboard with Plotly Dash

# Dashboard: Launch Success of All Sites

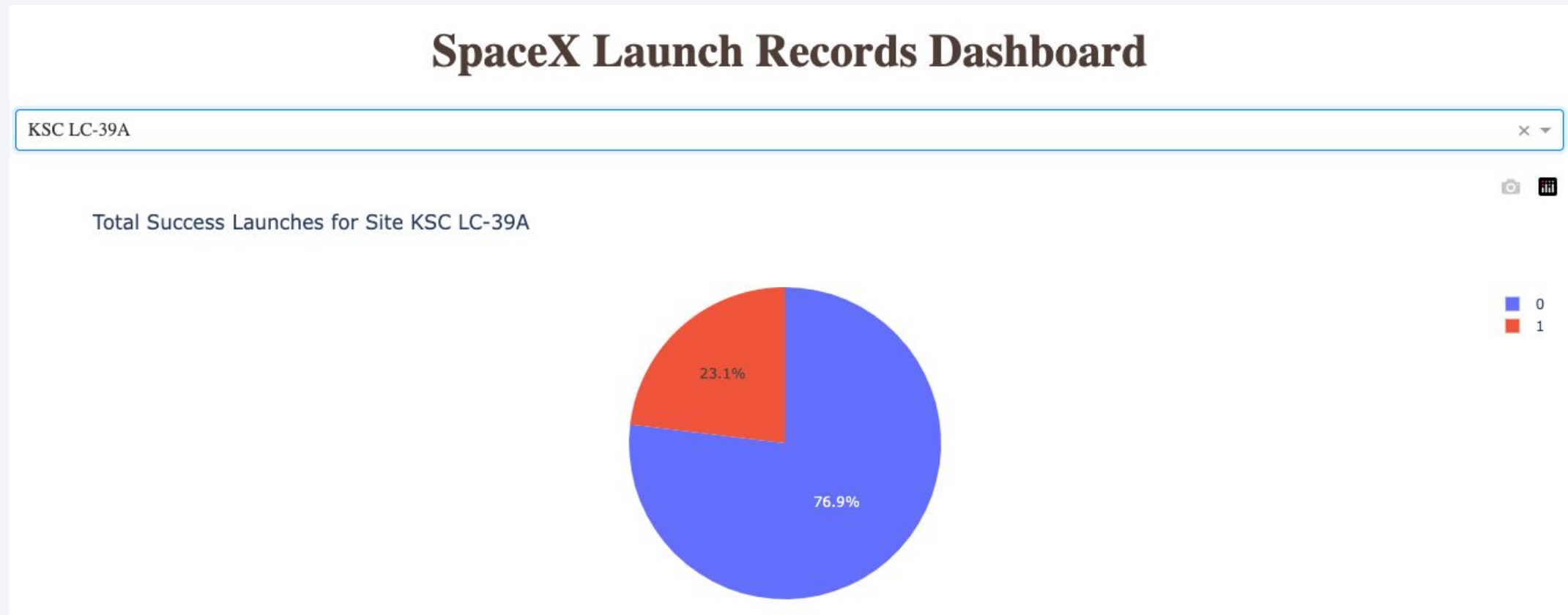
- KSC LC-39A has the most successful launches among all launch sites.  
(41.2%)



# Dashboard: KSC LC-39A

---

- KSC LC-39A has the highest success rate (76.9%)





# Dashboard: PayloadMass and Success

- Payload mass between 2,000 kg and 5,000 kg have the highest success rate.



Section 5

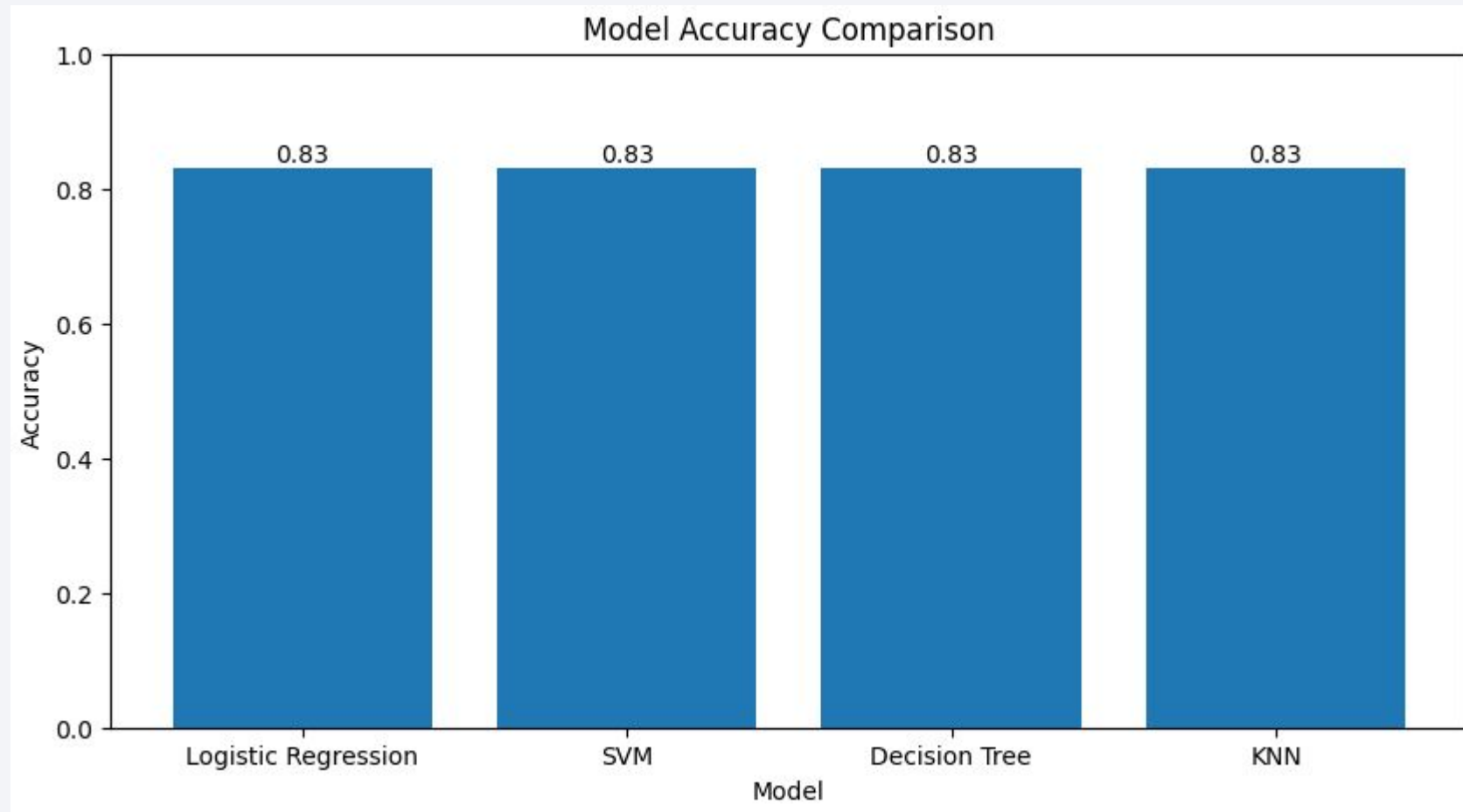
# Predictive Analysis (Classification)



# Classification Accuracy

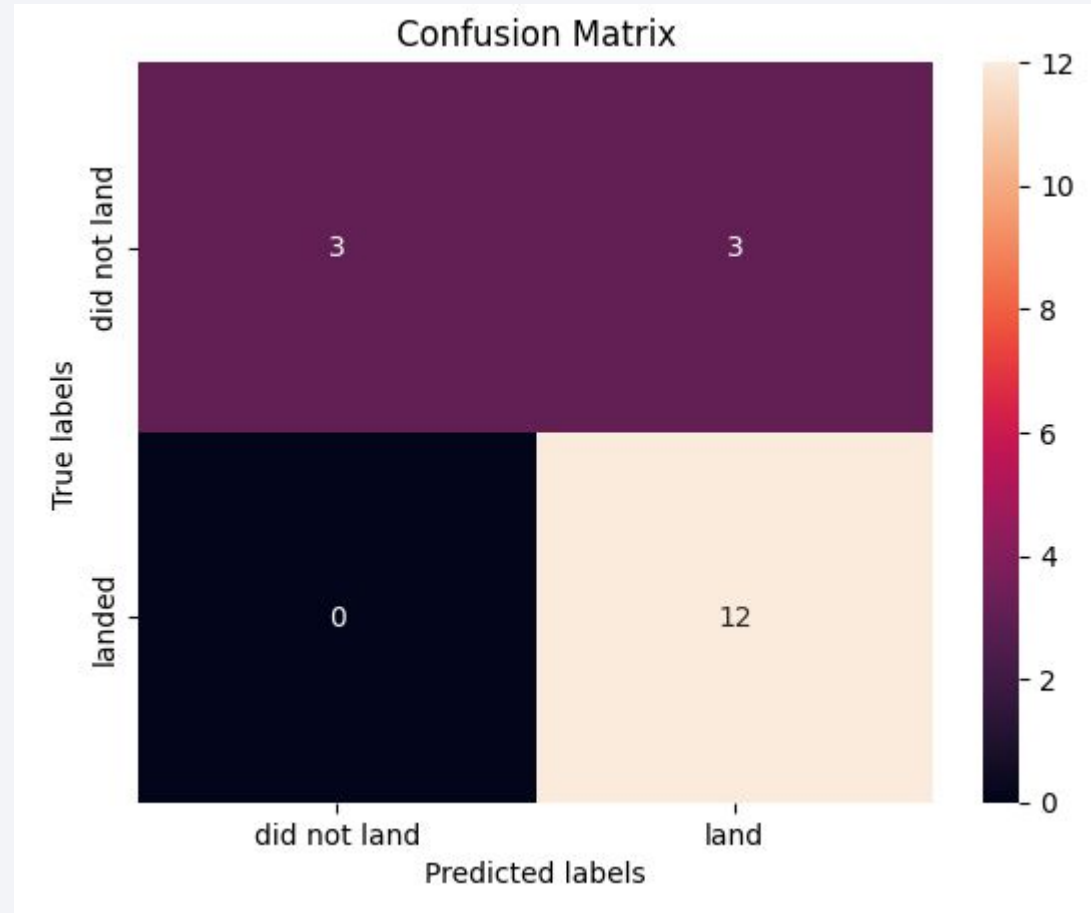
---

- Models performed similarly on the test set with accuracy 83.33%



# Confusion Matrix

- TP: 12
- TN: 3
- FP: 3
- FN: 3



# Conclusions

---

- Models performed similarly on the test set with accuracy 83.33%
- Collecting of more data might be worth to consider
- Try other predictive model, like model employed gradient boosting algorithm.

# Appendix

```
@app.callback( Output(component_id='success-pie-chart', component_property='figure'),
               Input(component_id='site-dropdown', component_property='value'))
✓ def get_pie_chart(launch_site):
    if launch_site == 'All Sites':
        fig = px.pie(values=spacex_df.groupby('Launch Site')['class'].mean(),
                     names=spacex_df.groupby('Launch Site')['Launch Site'].first(),
                     title='Total Success Launches by Site')
    else:
        fig = px.pie(values=spacex_df[spacex_df['Launch Site']==str(launch_site)]['class'].value_counts(normalize=True),
                     names=spacex_df['class'].unique(),
                     title='Total Success Launches for Site {}'.format(launch_site))
    return(fig)
```

```
@app.callback( Output(component_id='success-payload-scatter-chart', component_property='figure'),
               [Input(component_id='site-dropdown', component_property='value'),
                Input(component_id='payload-slider', component_property='value')])
✓ def get_payload_chart(launch_site, payload_mass):
    if launch_site == 'All Sites':
        fig = px.scatter(spacex_df[spacex_df['Payload Mass (kg)'].between(payload_mass[0], payload_mass[1])],
                         x="Payload Mass (kg)",
                         y="class",
                         color="Booster Version Category",
                         hover_data=['Launch Site'],
                         title='Correlation Between Payload and Success for All Sites')
    else:
        df = spacex_df[spacex_df['Launch Site']==str(launch_site)]
        fig = px.scatter(df[df['Payload Mass (kg)'].between(payload_mass[0], payload_mass[1])],
                         x="Payload Mass (kg)",
                         y="class",
                         color="Booster Version Category",
                         hover_data=['Launch Site'],
                         title='Correlation Between Payload and Success for Site {}'.format(launch_site))
    return(fig)
```



Thank you!

