

# Winning Space Race with Data Science

Niveditha Pandith T S  
November 16



# Outline

---

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

# Executive Summary

---

## Summary of methodologies

- Data Collection – SpaceX API
- Data Scraping
- Data Wrangling
- EDA with Data Visualization
- EDA with SQL
- Dashboard with Plotly Dash
- Predictive Analysis (Classification)

## Summary of all results

- Overall there is a consistent improvement in the success rate of all missions over time.
- KSC LC-39 had the best success rate of all sites
- Success is best with Falcon9 booster version B5 with payloads between 2000-5500kg
- Launch site are located near railway, highway and ports, but away from cities
- The Decision Tree Classifier performed best as a predictive mode

# Introduction

---

- Project background and context
  - In the current commercial space travel environment SpaceX is the leading player. They provide a relatively inexpensive rockets compared to the conventional rockets used by NASA and other space agencies.
  - SpaceX's latest rocket is the Falcon 9. It can be reused and therefore significantly reduces the cost of operation.
  - What determines reusage is whether the first stage lands successfully
- Problems you want to find answers
  - We are a new space company embarking on a quest to provide a safe commercial space travel, with low cost, to compete with Space X
  - If we use data from the Falcon 9 project, we can generate a model that would predict whether our rocket would successfully land, and what factors go into said success.

Section 1

# Methodology

# Methodology

---

## Executive Summary

- **Data collection methodology:**
  - The data was collected with SpaceX API using GET requests.
  - Additional data for Falcon 9 was collected using Web Scraping
- **Perform data wrangling**
  - Data was cleaned and outcome labels were converted to 1 (success) and 0 (failure).
- **Perform exploratory data analysis (EDA) using visualization and SQL**
- **Perform interactive visual analytics using Folium and Plotly Dash**
- **Perform predictive analysis using classification models**
  - Data was split. Grid Search performed. Models were evaluated for best accuracy score.

# Data Collection

---

1. SpaceX launch data was collected using SpaceX API:

- Data collected included:
  - Booster versions
  - Launch sites
  - Payload data
  - Core date
- Data was filtered to include only Falcon 9 launches
- Data was then cleaned and wrangled

2. Data for historic Falcon 9 launches were collected from the web using Web Scrapping

- Data taken as HTML table and then converted to a data frame with Beautiful Soup

# Data Collection – SpaceX API

[LINK TO GITHUB](#)

- GET Request to gather data using the API

```
spacex_url="https://api.spacexdata.com/v4/launches/past"
response = requests.get(spacex_url)
```

- Decode the JSON data and turning it into a data frame

```
r1 = response.json()
data = pd.json_normalize(r1)
```

- Gathering and storing values using custom functions

```
# Call getLaunchSite
getLaunchSite(data)
```

```
# Call getPayloadData
getPayloadData(data)
```

```
# Call getCoreData
getCoreData(data)
```

```
launch_dict = {'FlightNumber': list(data['flight_number']),
'Date': list(data['date']),
'BoosterVersion':BoosterVersion,
'PayloadMass':PayloadMass,
'Orbit':Orbit,
'LaunchSite':LaunchSite,
'Outcome':Outcome,
'Flights':Flights,
'GridFins':GridFins,
'Reused':Reused,
'Legs':Legs,
'LandingPad':LandingPad,
'Block':Block,
'ReusedCount':ReusedCount,
'Serial':Serial,
'Longitude': Longitude,
'Latitude': Latitude}
```

```
data_falcon9 = launch_data[launch_data['BoosterVersion']!='Falcon 1']
```

FlightNumber	Date	BoosterVersion	PayloadMass	Orbit
1	2010-06-04	Falcon 9	NaN	LEO
2	2012-05-22	Falcon 9	525.0	LEO
3	2013-03-01	Falcon 9	677.0	ISS
4	2013-09-29	Falcon 9	500.0	PO
5	2013-12-03	Falcon 9	3170.0	GTO

# Data Collection Scraping

[LINK TO GITHUB](#)

- Request Falcon 9 Launch wiki page

```
# use requests.get() method with the provided static_url
response = requests.get(static_url).text
```

- Create BeautifulSoup Object

```
soup = BeautifulSoup(response, 'html.parser')
```

- Extract all columns from HTML tables

```
html_tables = soup.find_all('table')

column_names = []

for row in first_launch_table.find_all('th'):
    name = extract_column_from_header(row)
    if name is not None and len(name) > 0:
        column_names.append(name)
```

- Create a dataframe by parsing HTML tables

```
launch_dict= dict.fromkeys(column_names)

# Remove an irrelevant column
del launch_dict['Date and time ( )']

# Let's initial the launch_dict with each value to be an empty list
launch_dict['Flight No.']= []
launch_dict['Launch site']= []
launch_dict['Payload']= []
launch_dict['Payload mass']= []
launch_dict['Orbit']= []
launch_dict['Customer']= []
launch_dict['Launch outcome']= []
# Added some new columns
launch_dict['Version Booster']= []
launch_dict['Booster landing']= []
launch_dict['Date']= []
launch_dict['Time']= []
```

```
df= pd.DataFrame({ key:pd.Series(value) for key, value in launch_dict.items() })
```

# Data Wrangling

[LINK TO GITHUB](#)

- Calculating number of launches at each site

```
df['LaunchSite'].value_counts()
```



CCAFS SLC 40	55
KSC LC 39A	22
VAFB SLC 4E	13

- Calculating number and occurrence of each orbit

```
df['Orbit'].value_counts()
```

GTO	27
ISS	21
VLEO	14
PO	9
LEO	7
SSO	5
MEO	3
HEO	1
SO	1
ES-L1	1
GEO	1

- Mission outcome per orbit type

```
landing_outcomes = df['Outcome'].value_counts()  
print(landing_outcomes)
```



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

- Create landing outcome labels

```
for i,outcome in enumerate(landing_outcomes.keys()):  
    print(i,outcome)
```

**Bad outcomes**

0 True ASDS
1 None None
2 True RTLS
3 False ASDS
4 True Ocean
5 None ASDS
6 False Ocean
7 False RTLS

```
bad_outcomes=set(landing_outcomes.keys()[[1,3,5,6,7]])  
bad_outcomes
```



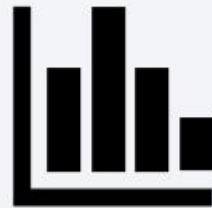
```
{'False ASDS', 'False Ocean', 'False RTLS', 'None ASDS', 'None None'}
```

# EDA with Data Visualization

- Scatter plots and bar graphs used to analyze:

- Flight Number and Launch Site
- Payload and Launch Site
- Success Rate of each Orbit Type
- Flight Number and Orbit type
- Payload and Orbit type

See results on  
slides 19-24



- Launch success yearly trend was also visualized

- Showed a consistent increase



- **Feature Engineering**: one-hot encoding

# EDA with SQL

[LINK TO GITHUB](#)

The following SQL queries were called:

1. Display the names of the unique launch sites in the space mission
2. Display 5 records where launch sites begin with the string 'CCA'
3. Display the total payload mass carried by boosters launched by NASA (CRS)
4. Display average payload mass carried by booster version F9 v1.1
5. List 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. List the names of the booster\_versions which have carried the maximum payload mass. Use a subquery
9. List the failed landing\_outcomes in drone ship, their booster versions, and launch site names for 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

See results on  
slides 26-35

# Build an Interactive Map with Folium

[LINK TO GITHUB](#)

---

- To **mark all launch sites** on a map we used:
  - **Circle** objects to highlight the location on a global map
  - **MarkerCluster** to better view multiple launches at a single location
- To **mark the success/failed** launches for each site on the map we used:
  - **MarkerCluster** were labelled **green** for success and **red** for failure
- To **calculate the distances** between a launch site to its proximities we used:
  - **MousePosition** to find the coordinates of the two locations.
  - **Line and distance function** to measure and visualise the distance.

With this visualization we were able to answer the following **questions**:

- Are launch sites in close proximity to **railways** and **highways**?  
**Yes**, to ease transportation of goods and equipment.
- Are launch sites in close proximity to **coastline**?  
**Yes**, to have port access.
- Do launch sites keep certain distance **away from cities**?  
**Yes**, to reduce the amount of noise for people living in the cities.

# Build a Dashboard with Plotly Dash

[LINK TO GITHUB](#)

---

- Created a **Pie Chart** to represent the proportion of **successful** launches from all **launch site**.
- Created a **drop down list** to produce a separate **Pie Chart** for each **launch site** showing its **success rate**.
- Created a **range scale** for **Payload Mass** to showcase its relationship with **Booster Version**, and **Successes/Failures**.
- Launched a **Plotly Dash** that contained all the above elements.

We were able to gain **insight** that helped us answer the flowing questions:

- Which site has the largest successful launches? **KSC LC-39A**
- Which site has the highest launch success rate? **KSC LC-39A**
- Which payload range(s) has the highest launch success rate? **2000-5500kg**
- Which payload range(s) has the lowest launch success rate? **under 1500 and 5500 to 7000**
- Which F9 Booster version (v1.0, v1.1, FT, B4, B5, etc.) has the highest launch success rate? **B5**

# Predictive Analysis (Classification) [LINK TO GITHUB](#)

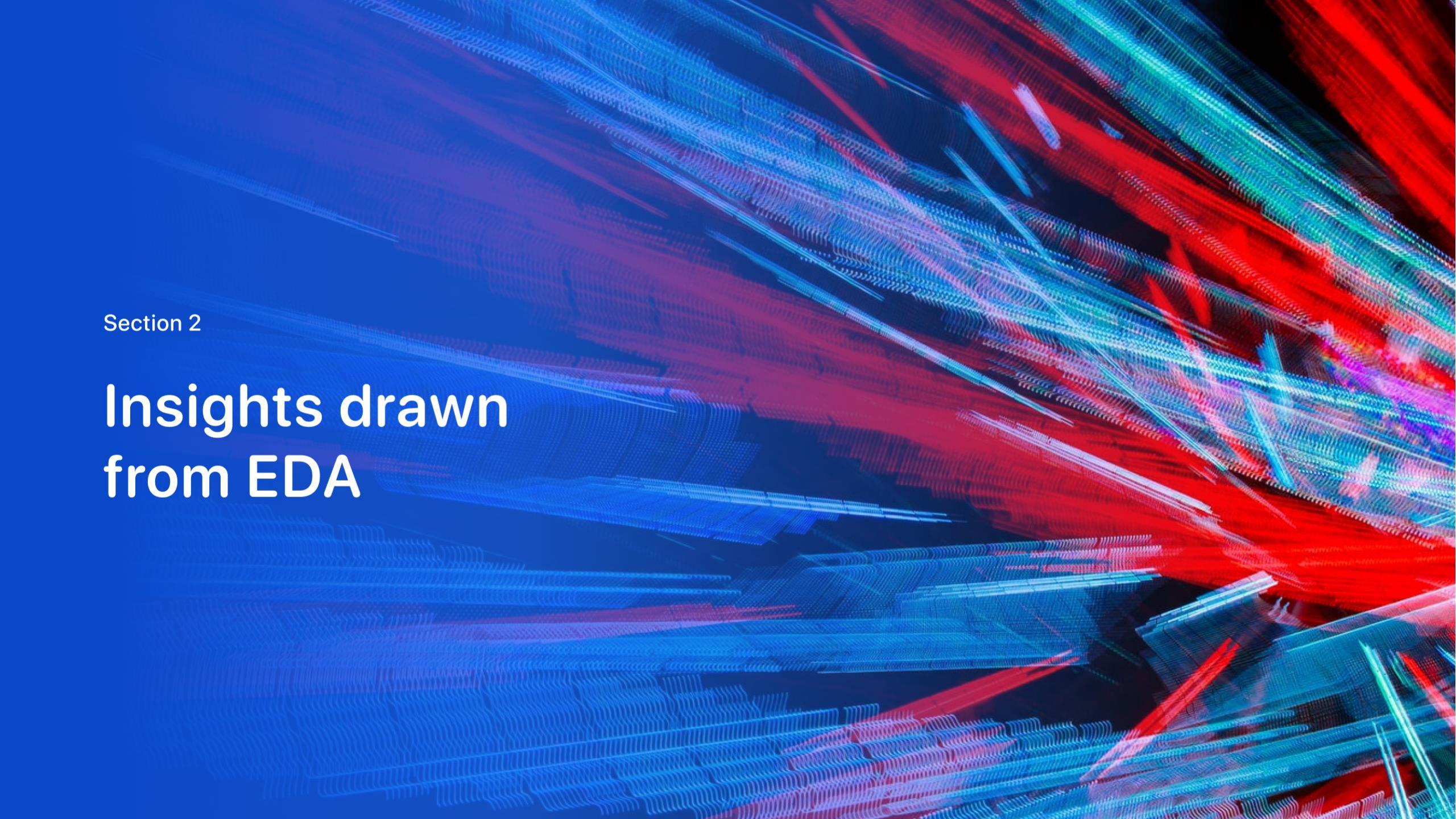
---

- To build the best model we followed the following steps:
  1. Standardize the data using **StandardScaler()**
  2. **Split the data** into **training** group and **testing** group
  3. Use **GridSearch()** to find the **best hyperparameters** for our model
  4. **Fit the model** and find **accuracy score** on the test data
  5. Create a **Confusion Matrix** to show the accuracy of the model
  6. **Steps 3-5** were each applied to the **following model types**:
    - Logistic Regression
    - Support Vector Machine (SVM)
    - **Decision Tree Classifier (was found to be the best model)**
    - K Nearest Neighbor

# Results

---

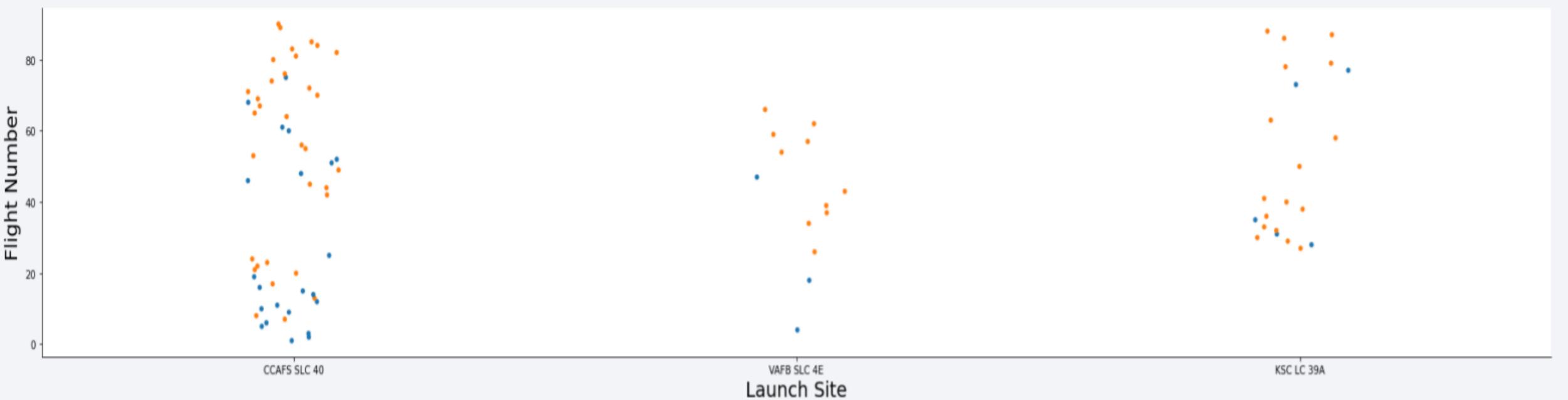
- Exploratory data analysis results
- Interactive analytics demo in screenshots
- Predictive analysis results

The background of the slide features a complex, abstract digital visualization. It consists of numerous thin, glowing lines that create a sense of depth and motion. The lines are primarily blue and red, with some green and purple highlights. They form a grid-like structure that curves and twists across the frame, resembling a three-dimensional space or a network of data points. The overall effect is futuristic and dynamic.

Section 2

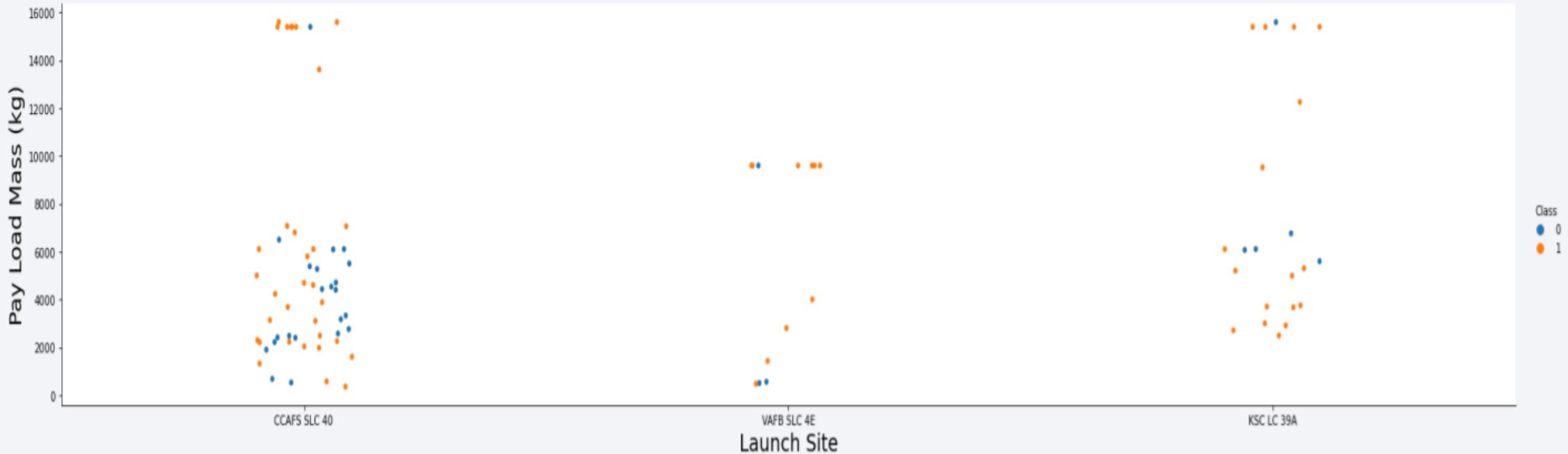
## Insights drawn from EDA

# Flight Number vs. Launch Site



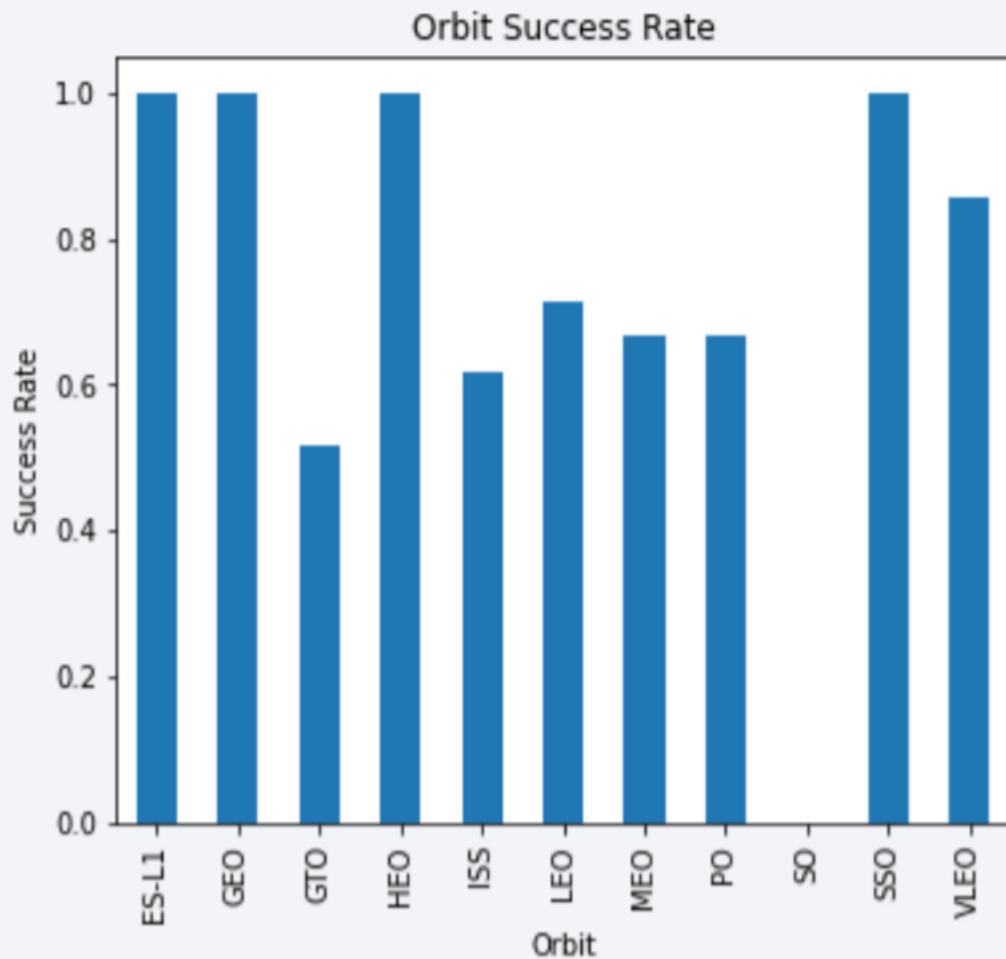
- Early flights had lower success rate than later flights, with the bulk launching from CCAFS SLC 40
- KSC LC 39A had the best overall performance

# Payload vs. Launch Site



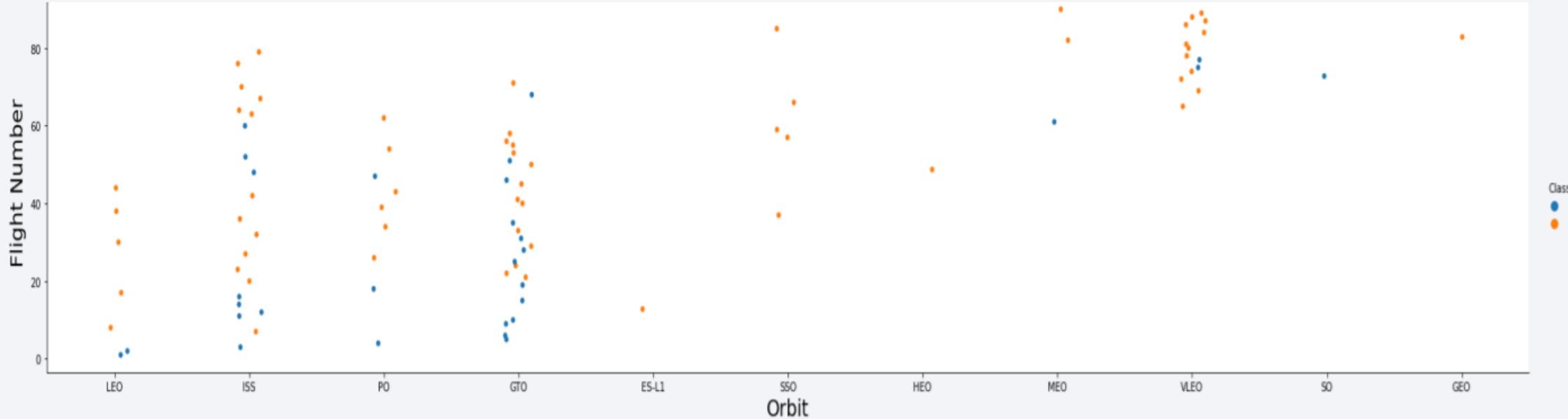
- CAFS LC-40 had the bulk of the light payloads corresponding to the early flights.
- CAFS LC-40 and VAFB SLC 4E had a better success rate as the payload mass increased.
- KSC LC-39A had a perfect success rate at lower payload. It also had an improved success rate as the weight increased beyond 8000kg.

# Success Rate vs. Orbit Type



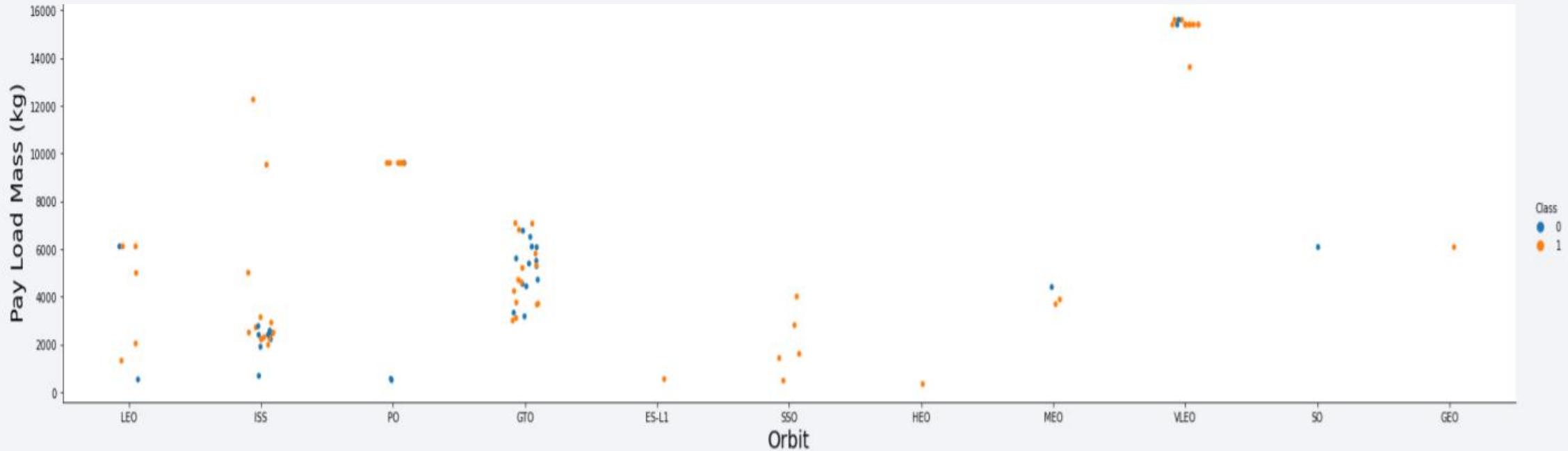
- ESL1, GEO, HEO, and SSO all had a 100% success rate.

# Flight Number vs. Orbit Type



- LEO showed improvement in success rate as the Flight Number increased.
- There seems to be no relationship between flight number in GTO orbit.
- Generally speaking as more launches were conducted the rocket was able to reach farther earth orbits.

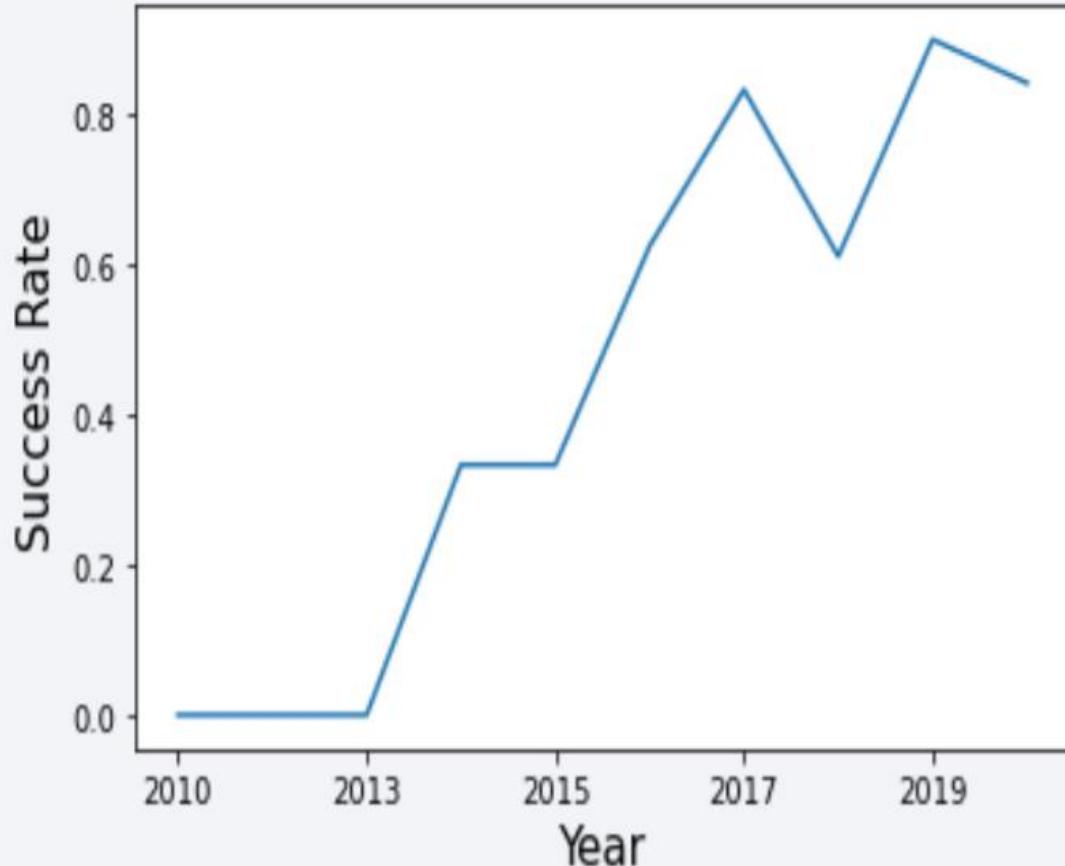
# Payload vs. Orbit Type



- PO, LEO, ISS orbits saw better success rate as the payload increased.
- GTO orbit saw lower success rates as the payload increased.

# Launch Success Yearly Trend

---



- Since 2013 there has been a steady increase in success rate approaching 100%

# All Launch Site Names

---

## SQL QUERY:

```
%sql select Unique(LAUNCH_SITE) from SPACEXTBL;
```

## SQL Explanation:

Query recall all unique launch sites. The use of DISTINCT ensures that no repetition occurs in the launch site names.

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

# Launch Site Names Begin with 'CCA'

---

## SQL QUERY:

```
%sql SELECT LAUNCH_SITE from SPACEXTBL  
where (LAUNCH_SITE) LIKE 'CCA%' LIMIT 5;
```

## SQL Explanation:

Query recall 5 launches from CCAFS LC-40 by using LIKE 'CCA%'.  
Limiting the response to 5 rows using LIMIT 5.

launch_site
CCAFS LC-40

# Total Payload Mass

---

## SQL QUERY:

```
%sql select sum(PAYLOAD_MASS__KG_) as payloadmass from SPACEXTBL;
```

payloadmass
619967

## SQL Explanation:

Using SUM() to display the total payload mass carried by boosters that were launched by NASA(CRS)

# Average Payload Mass by F9 v1.1

---

## SQL QUERY

```
%sql select avg(PAYLOAD_MASS__KG_) as payloadmass  
from SPACEXTBL;
```

payloadmass

6138

## SQL Explanation:

Using AVG() to display the average payload mass carried by booster F9 v1.1

# First Successful Ground Landing Date

---

## SQL QUERY

```
%sql select min(DATE) from SPACEXTBL;
```

1

2010-06-04

## SQL Explanation:

Using MIN(date) to display the first (earliest) successful landing.

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

---

### SQL QUERY

```
%sql select BOOSTER_VERSION from SPACEXTBL where  
LANDING_OUTCOME='Success (drone ship)' and PAYLOAD_MASS_KG_BETWEEN  
4000 and 6000
```

### SQL Explanation:

Using BETWEEN to display successful drone ship landing  
with a payload range of 4000- 6000 kg

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

# Total Number of Successful and Failure Mission Outcomes

---

## SQL QUERY

```
%sql select count(MISSION_OUTCOME) as  
missionoutcomes from SPACEXTBL GROUP BY MISSION_OUTCOME;
```

missionoutcomes
1
99
1

## SQL Explanation:

Using COUNT to display the total successful and failed mission outcome

# Boosters Carried Maximum Payload

---

## SQL QUERY

```
%sql select BOOSTER_VERSION as boosterversion from SPACEXTBL where PAYLOAD_MASS_KG_=(select max(PAYLOAD_MASS_KG_) from SPACEXTBL);
```

## SQL Explanation:

Using a subquery with MAX() to find the boosters that have carried the maximum payload mass.

boosterversion
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

---

## SQL QUERY

```
%sql SELECT MONTH(DATE),MISSION_OUTCOME,BOOSTER_VERSION,LAUNCH_SITE  
FROM SPACEXTBL where EXTRACT(YEAR FROM DATE)='2015';
```

## SQL Explanation:

Using WHERE clause and LIKE '2015%' to find all failed drone ship landing in the year 2015.

id	mission_outcome	booster_version	launch_site
1	Success	F9 v1.1 B1012	CCAFS LC-40
2	Success	F9 v1.1 B1013	CCAFS LC-40
3	Success	F9 v1.1 B1014	CCAFS LC-40
4	Success	F9 v1.1 B1015	CCAFS LC-40
4	Success	F9 v1.1 B1016	CCAFS LC-40
6	Failure (in flight)	F9 v1.1 B1018	CCAFS LC-40
12	Success	F9 FT B1019	CCAFS LC-40

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

---

## SQL QUERY

```
%sql SELECT LANDING_OUTCOME FROM SPACEXTBL WHERE  
DATE BETWEEN '2010-06-04' AND '2017-03-20' ORDER BY DATE  
DESC;
```

## SQL Explanation:

Using ORDER BY with DESC to list and rank all landing outcomes between 2010-06-04 and 2017-03-20

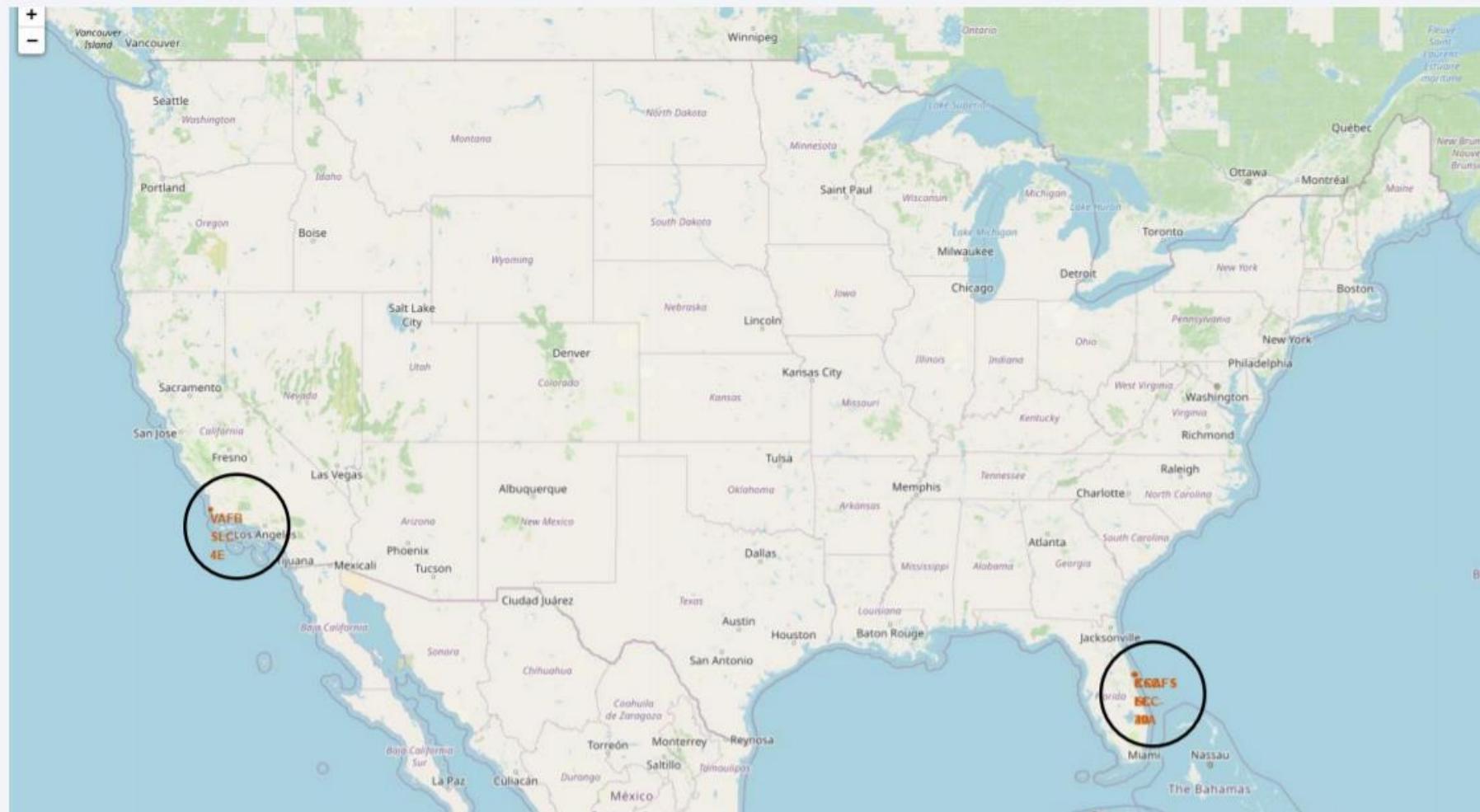
landing_outcome
No attempt
Success (ground pad)
Success (drone ship)
Success (drone ship)
Success (ground pad)
Failure (drone ship)
Success (drone ship)
Success (drone ship)
Success (drone ship)
Failure (drone ship)
Failure (drone ship)
Success (ground pad)
Precalculated (drone ship)
No attempt
Failure (drone ship)
No attempt
Controlled (ocean)
Failure (drone ship)
Uncontrolled (ocean)
No attempt
No attempt
Controlled (ocean)
Controlled (ocean)
No attempt
No attempt
Uncontrolled (ocean)
No attempt
No attempt
No attempt
Failure (parachute)
Failure (parachute)

The background of the slide is a photograph taken from space at night. It shows the curvature of the Earth's horizon against a dark blue sky. Numerous glowing yellow and white points represent city lights, concentrated in coastal and urban areas. In the upper right quadrant, there is a bright, horizontal green band, likely representing the Aurora Borealis or a similar atmospheric phenomenon.

Section 4

# Launch Sites Proximities Analysis

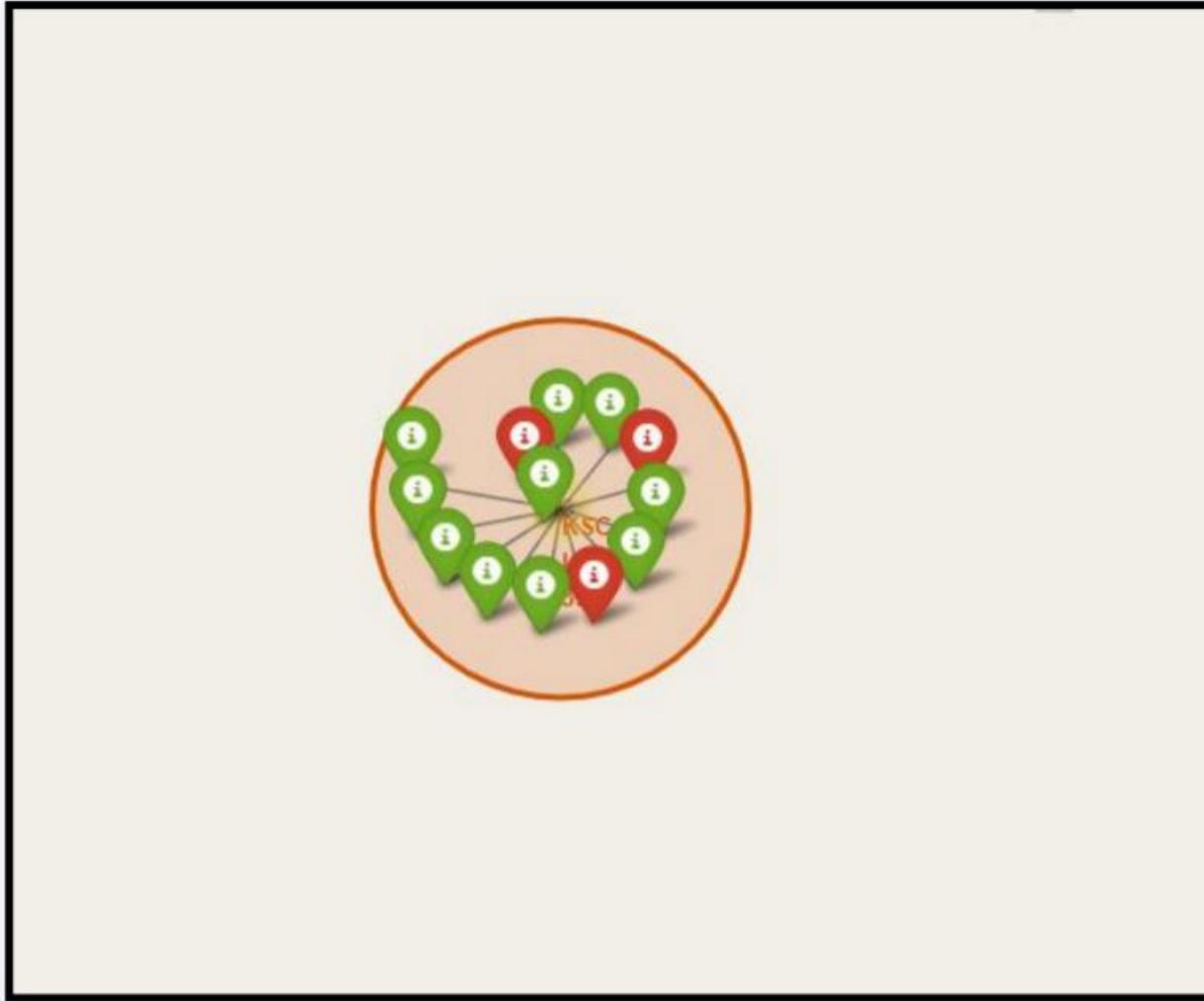
# Launch Location Locally



Launch sites are located on the **east coast** and **west coast** of the continental **United States**.

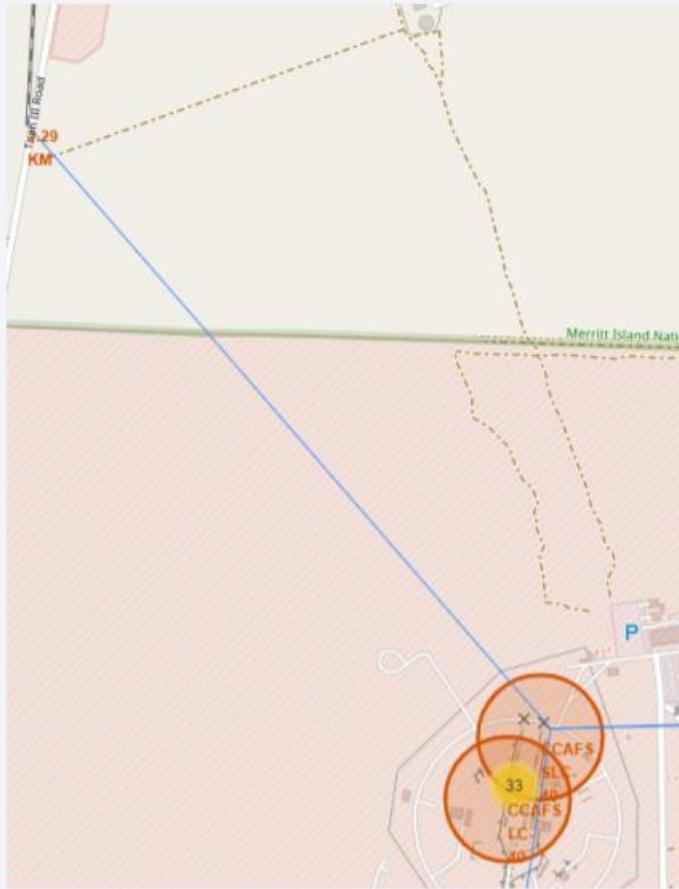
# Visualizing Launch Outcomes for KSC LC-39A

---

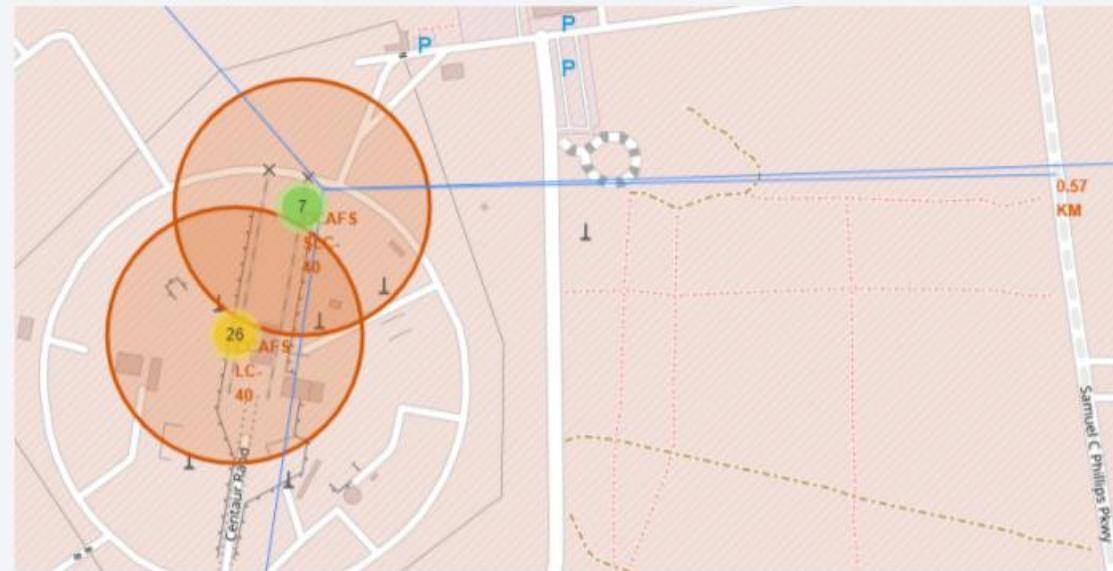


- **Folium markers** represent each **launch outcome**.
- **Successes** are labelled **green**
- **Failures** are labelled **red**

# Distances To Features Near Launch Location



Distance to railway



Distance to highway

# Distances To Features Near Launch Location

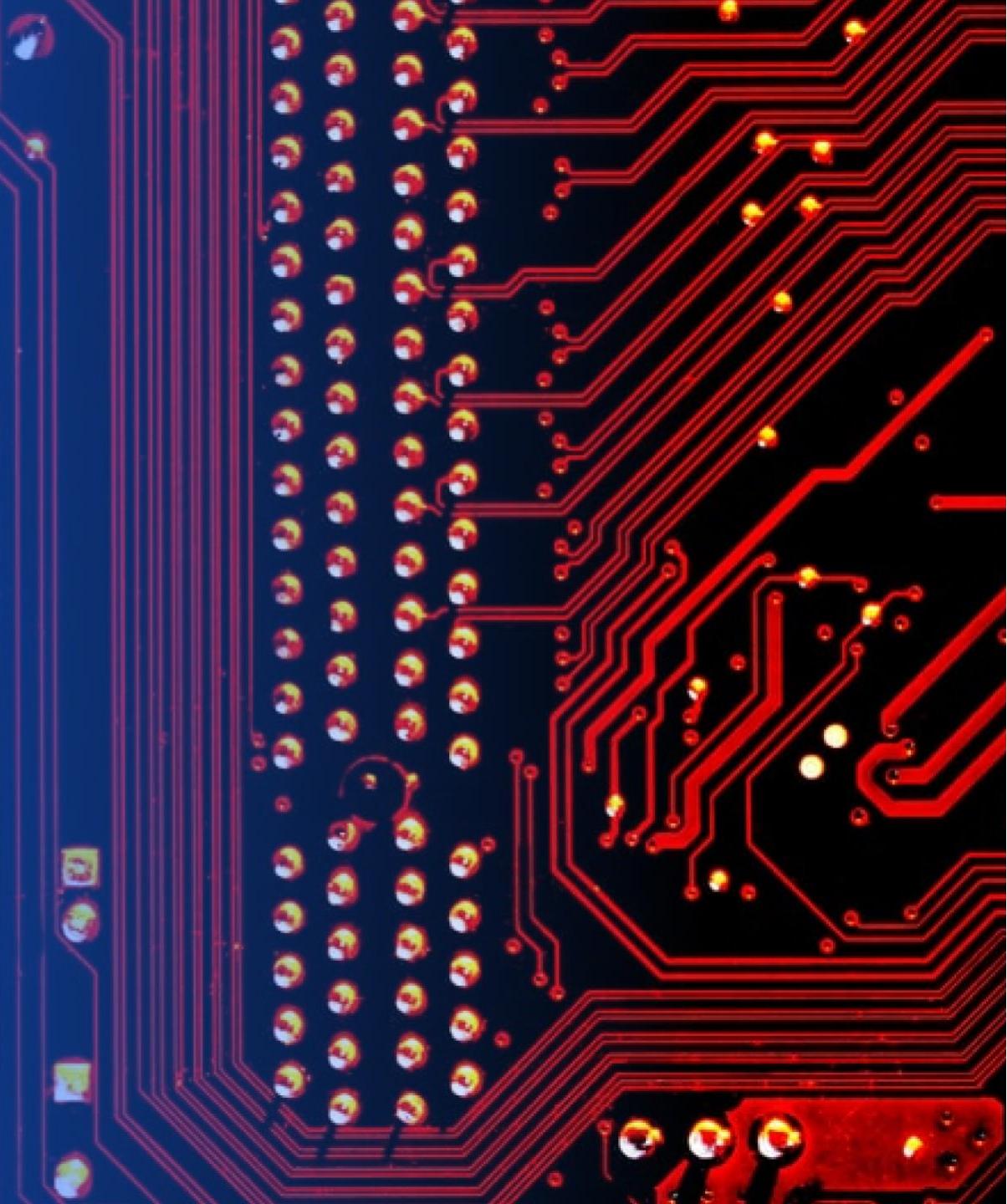
## In Summary:

Feature	Distance(KM)
Railway	1.29
Highway	0.57
Coastline	0.84
City	18.18

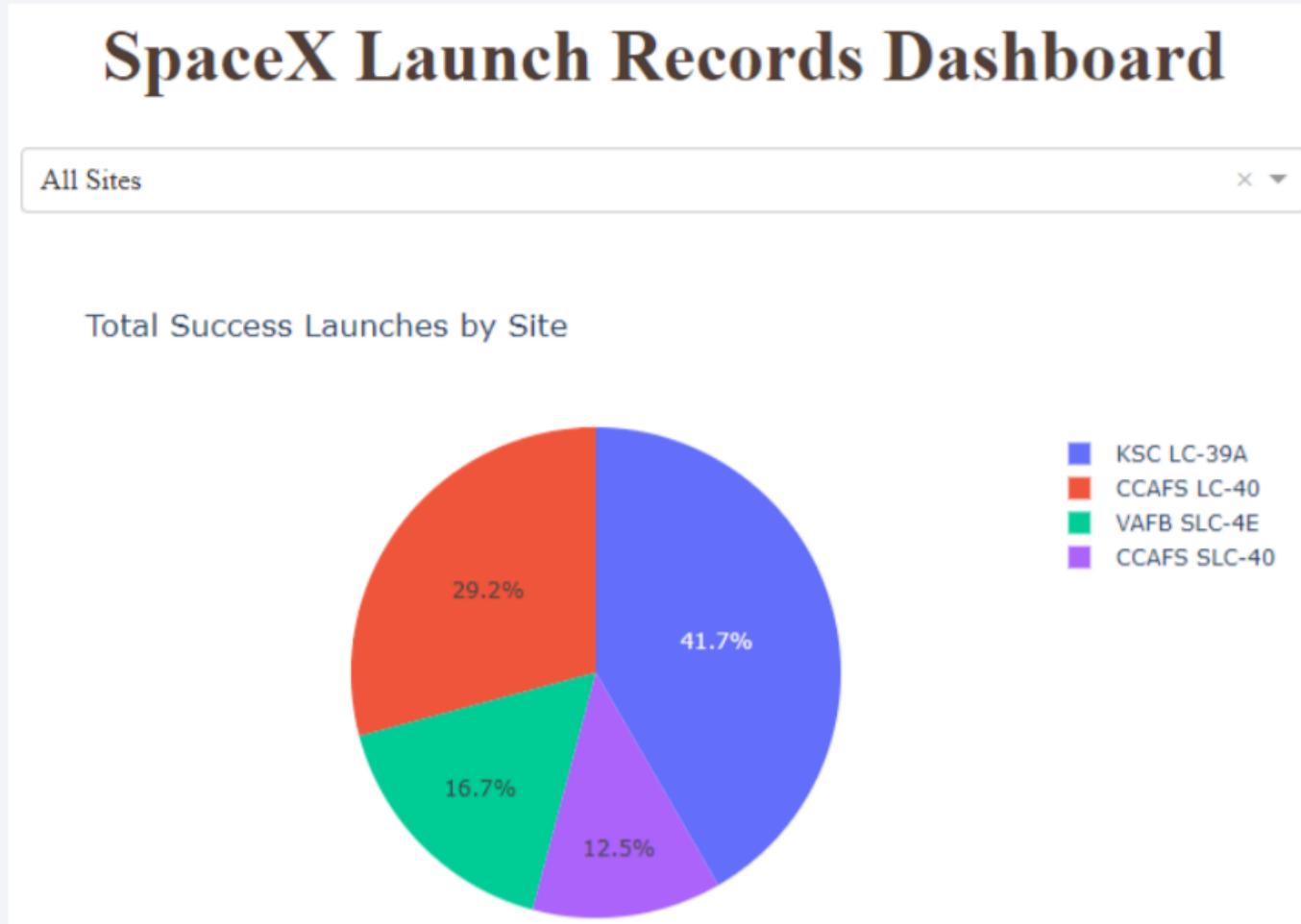
- Launch locations are selected close to railway and highways
- Launch locations are located close to coastlines
- Launch locations are located far from major cities

Section 5

# Build a Dashboard with Plotly Dash



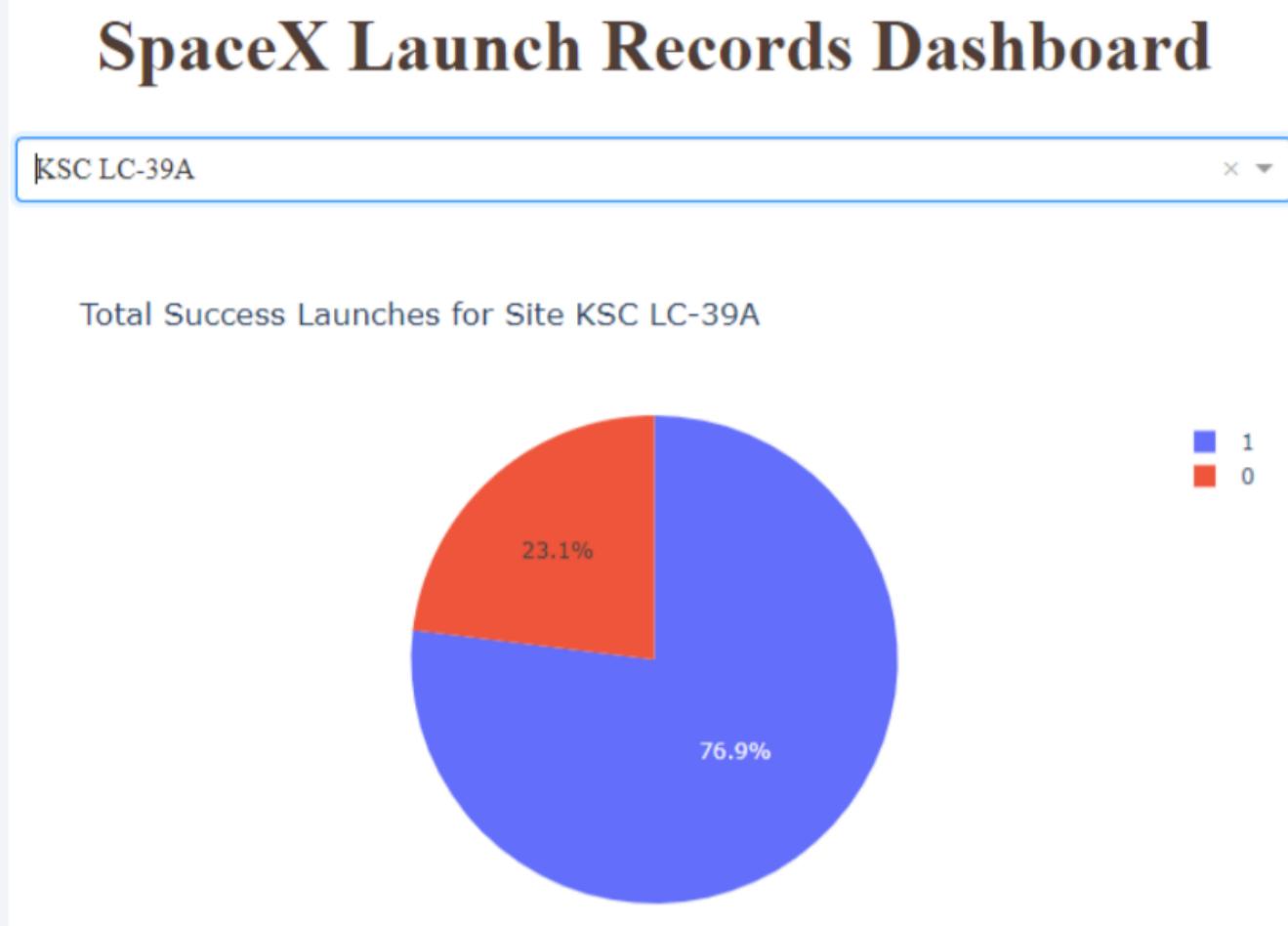
# Dashboard – Launch success count



- **Kennedy Space Center Launch Complex 39 (KSC LC-39A)** had the **most success count** at **41.7%** of total launches.
- **Vandenberg Space Launch Complex 4 (VAFB SLC-4E)** had the **lowest success count** at **12.5%** of total launches.

# Dashboard-

---



- **Kennedy Space Center Launch Complex 39 (KSC LC-39A) had the highest success rate at 76.9% of total launches.**

# Payload Vs Launch Outcome For All Sites



- **Best success rate** occurs at weight range of **2000-5500kg**
- The **booster version** with the **highest success rate** is **B5**

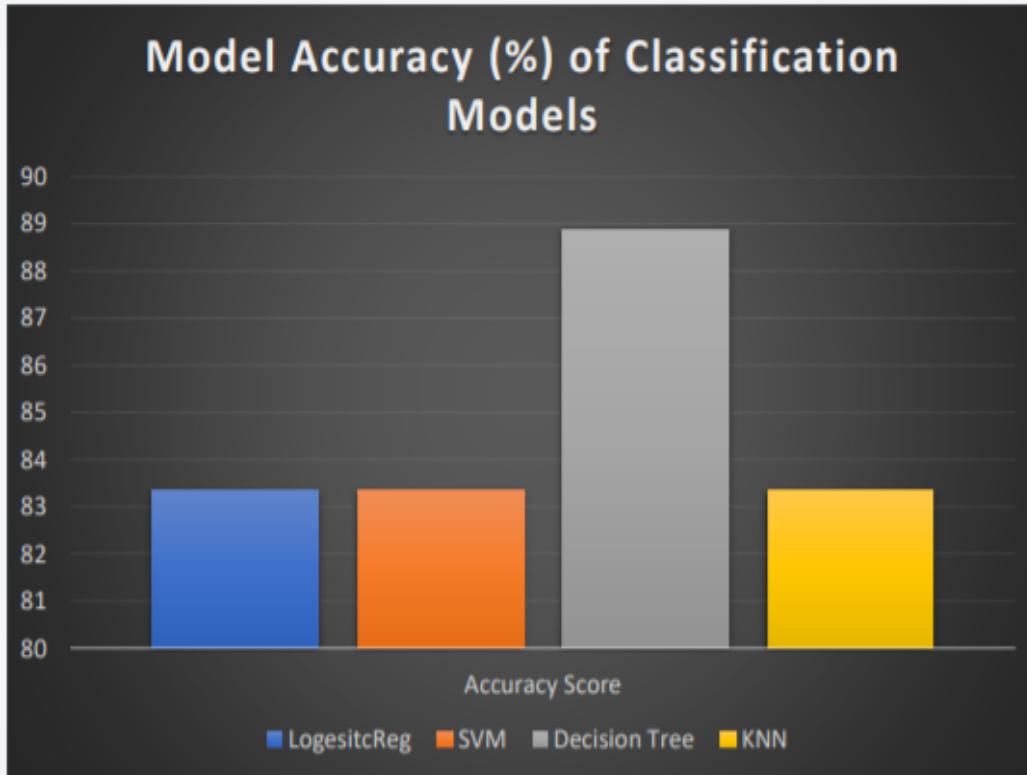
The background of the slide features a dynamic, abstract design. It consists of several curved, overlapping bands of color. A prominent band on the left is a bright blue, while another on the right is a warm yellow. These colors transition into lighter, more diffused tones towards the edges of the frame. The overall effect is one of motion and depth.

Section 6

# Predictive Analysis (Classification)

# Classification Accuracy

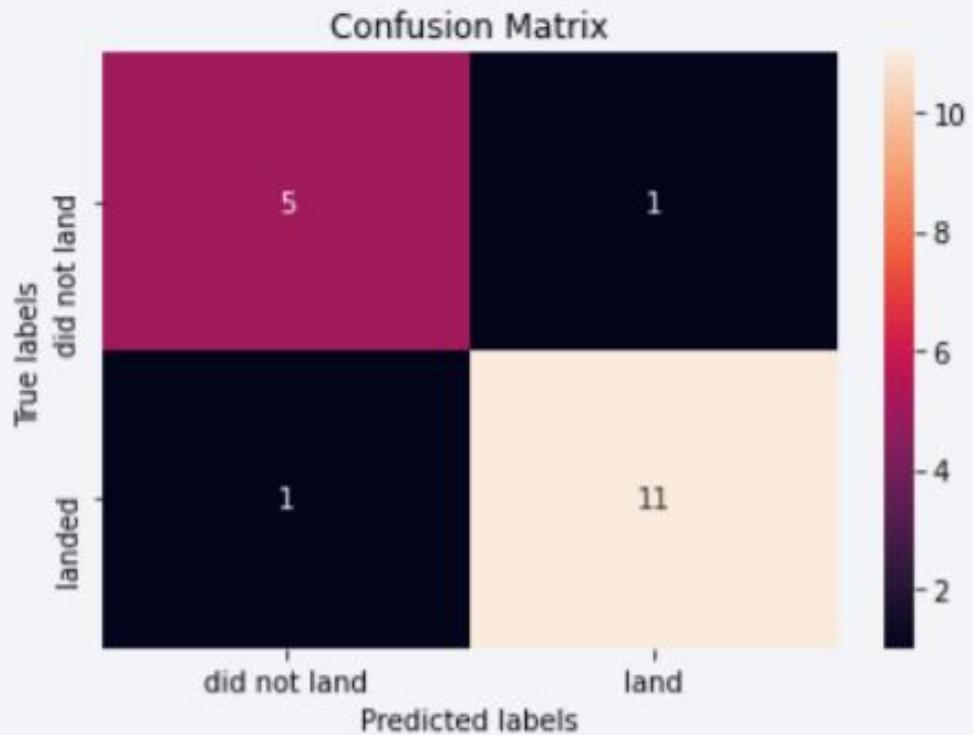
---



- The **Decision Tree** classification model had the **highest accuracy** at **88.888%**.

# Confusion Matrix

---



- The Confusion Matrix clearly demonstrates the high accuracy of the Decision Tree Model.
- Only a single case each of false positive and false negative.

# Conclusions

---

- Overall, there is a consistent improvement in the success rate of all missions over time.
- KSC LC-39 had the best success rate of all sites
- Success is best with Falcon9 booster version B5 with payloads between 2000-5500kg
- Launch site are located near railway, highway and ports, but away from cities
- The Decision Tree Classifier performed best as a predictive model.

Thank you!

