

# Winning Space Race with Data Science

Francisco Javier Garcia Garcia  
15/07/2024



# Outline



EXECUTIVE  
SUMMARY



INTRODUCTION



METHODOLOGY



RESULTS

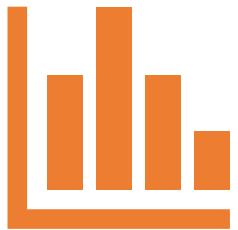


CONCLUSION



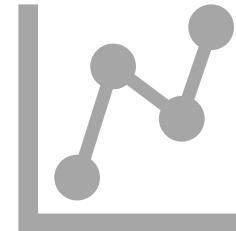
APPENDIX

# Executive Summary



## Summary of methodologies

Data collection with API and web scraping  
Exploratory data analysis using data visualization  
Exploratory data analysis using SQL  
Creating dashboards with plotly dash  
Predictive analysis



## Summary of all results

Exploratory data analysis results  
Interactive maps and dashboards  
Predictive results

# Introduction

## Project background and context

- This project was meant to predict if the Falcon 9's first stage will successfully land. SpaceX says that the cost to launch the Falcon 9 is 62 million dollars. Some of SpaceX's competitors can cost 165 million dollars to launch. The reason SpaceX has their price so cheap is because they can reuse the first stage of the launch. If we can determine if the stage will land, we can determine what the cost of the launch will be.

## Problems you want to find answers

- What characteristics determine a successful landing vs a failed landing
- What effects does each relationship of the rocket's variables have on the success/failure of each landing
- What conditions allow SpaceX achieve the best success rate

Section 1

# Methodology

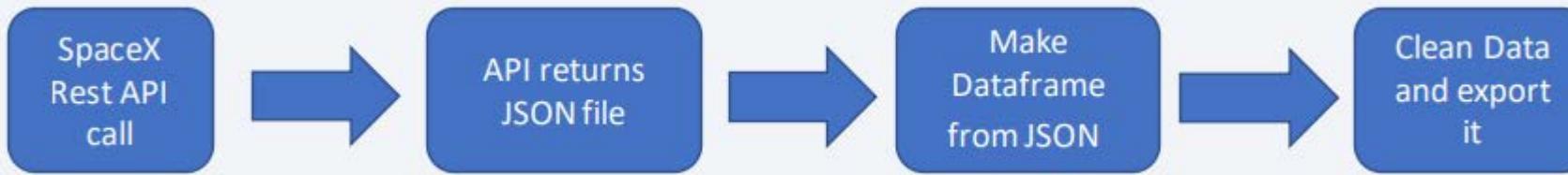
# Methodology

 Executive Summary	 Data collection methodology: SpaceX REST API Web Scraping Wikipedia	 Perform data wrangling Dropping unneeded columns One hot encoding for classification models	 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
---	---	---	---	---	---

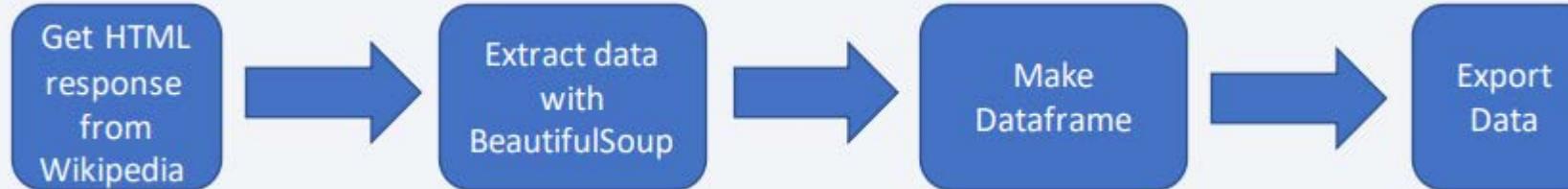
# Data Collection

---

- Describe how data sets were collected.
  - The data sets were collected from SpaceX REST API and from web scrapping Wiki
  - The API produced info on rockets, launches, and payload



- Web scraping Wiki also produced rockets, launches, and payload information



# Data Collection – SpaceX API

[GITHUB](#) link

## 1. Getting Response from API

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

## 2. Convert Response to JSON File

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

## 3. Transform data

```
getLaunchSite(data)  
getPayloadData(data)  
getCoreData(data)  
getBoosterVersion(data)
```

## 4. Create dictionary with 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}
```

## 5. Create dataframe

```
data = pd.DataFrame.from_dict(launch_dict)
```

## 6. Filter dataframe

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

## 7. Export to file

```
data_falcon9.to_csv('dataset_part_1.csv', index=False)
```

# Data Collection - Scraping

[GITHUB link](#)

## 1. Getting Response from HTML

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

## 2. Create BeautifulSoup Object

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

## 3. Find all tables

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

## 4. Get column names

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

## 5. Create dictionary

```
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']= []
```

## 6. Add data to keys

```
extracted_row = 0
#Extract each table
for table_number,table in enumerate(soup.findAll_all()):
    # get table row
    for rows in table.findAll_all("tr"):
        #check to see if first table heading is a
        if rows.th:
            if rows.th.string:
                flight_number=rows.th.string.strip()
                flag=flight_number.isdigit()

See notebook for the rest of code
```

## 7. Create dataframe from dictionary

```
df=pd.DataFrame(launch_dict)
```

## 8. Export to file

```
df.to_csv('spacex_web_scraped.csv', index=False)
```

# Data Wrangling

[GITHUB](#) link

- The dataset showed that there were several cases where the booster didn't land successfully
- I also transformed strings into categorical variables, 1 meaning success and 0 is failure

## 1. Calculate launches number for each site

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

CCAFS SLC 40	55
KSC LC 39A	22
VAFB SLC 4E	13
Name: LaunchSite, dtype: int64	

## 2. Calculate the number and occurrence of each orbit

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

GTO	27
ISS	21
VLEO	14
PO	9
LEO	7
SSO	5
MEO	3
SO	1
ES-L1	1
HEO	1
GEO	1
Name: Orbit, dtype: int64	

## 3. Calculate number and occurrence of mission outcome per orbit type

```
landing_outcomes = df['Outcome'].value_counts()  
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
Name: Outcome, dtype: int64	

## 4. Create landing outcome label from Outcome column

```
landing_class = []  
for key,value in df["Outcome"].items():  
    if value in bad_outcomes:  
        landing_class.append(0)  
    else:  
        landing_class.append(1)  
df['Class']=landing_class
```

## 5. Export to file

```
df.to_csv("dataset_part_2.csv", index=False)
```

# EDA with Data Visualization

[GITHUB](#) link

---

- Scatter Graph
    - Flight Number vs Payload Mass
    - Flight Number vs Launch Site
    - Payload vs Launch Site
    - Orbit vs Flight Number
    - Payload vs Orbit
- Scatter plots show relationships between variables
- Line Graph
    - Success Rate vs Year
- Shows data vars and their trends
- Bar Graph
    - Success Rate vs Orbit
- Shows the relationships between numeric and categorical vars

- SQL queries were used to gather and understand data from the dataset
  - Display
    - Names of the launch sites
    - First 5 records starting with “CAA”
    - Total payload carried by boosters from NASA
    - Avg payload carried by booster F9
  - List
    - Date of first successful landing outcome on ground pad
    - Names of the boosters that have success in drone ship
      - $4000 < \text{MASS} < 6000$
    - Number of successful/failure mission outcomes

# Build an Interactive Map with Folium

[GITHUB link](#)

Folium is a map that I centered around NASA Space Center

Red circle at NASA Johnson Space Center's coordinate with label showing its name

Red circles at each launch site coordinates with label showing launch site name

The grouping of points in a cluster to display multiple and different information for the same coordinates

Markers to show successful and unsuccessful landings. Green for successful landing and Red for unsuccessful landing

Markers to show distance between launch site to key and plot a line between them

# Build a Dashboard with Plotly Dash

[GITHUB](#) link

---

- My dashboard has a dropdown, pie chart, range slider, and a scatter plot
  - Dropdown allows a user to choose the launch site or all launch sites
  - Pie chart shows the total success and the total failure for the launch site chosen with the
  - Range slider allows a user to select a payload mass in a fixed range
  - Scatter chart shows the relationship between two variables, in particular Success vs Payload Mass

# Predictive Analysis (Classification)

[GITHUB](#) link

---

- Data prep
  - Load dataset, normalize data, then split the data into training and testing sets
- Model prep
  - Select proper ML models, set parameters, then train models with the training data set
- Model eval
  - Find best hyperparameters for each model type, computer accuracy for each model with the testing set, then plot the confusion matrix
- Model comp
  - Compare the models according to their accuracy and the model with the best accuracy is chosen



Results

---

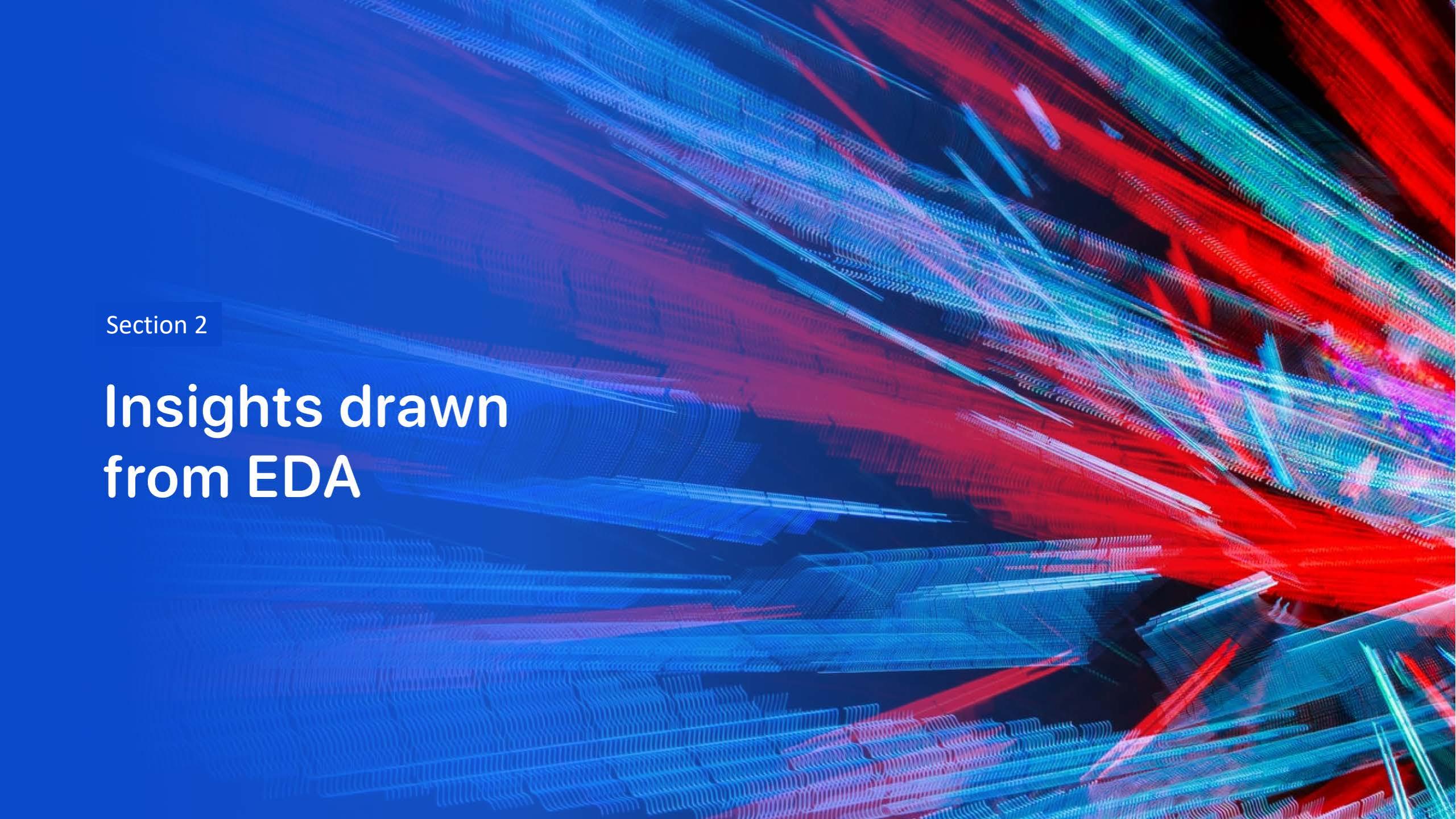
Exploratory data  
analysis results

---

Interactive analytics  
demo in screenshots

---

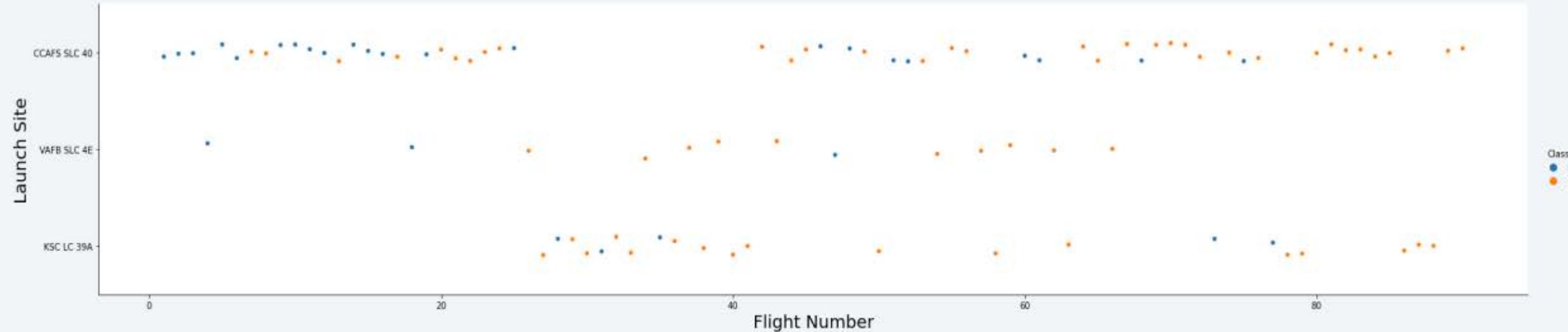
Predictive analysis  
results

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 is more dense and vibrant towards the right side of the frame, while appearing more sparse and blue-tinted on the left. The overall effect is reminiscent of a high-energy particle simulation or a futuristic circuit board.

Section 2

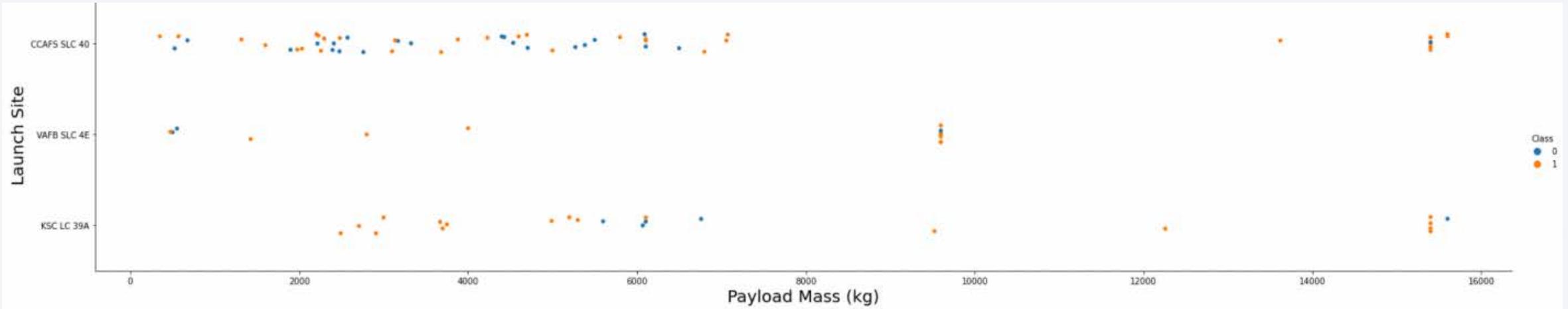
## Insights drawn from EDA

# Flight Number vs. Launch Site



- The earliest flights all failed while the latest flights all succeeded
- The CCAFS SLC 40 launch site has about a half of all launches
- VAFB SLC 4E and KSC LC 39A have higher success rates
- It can be assumed that each new launch has a higher rate of success

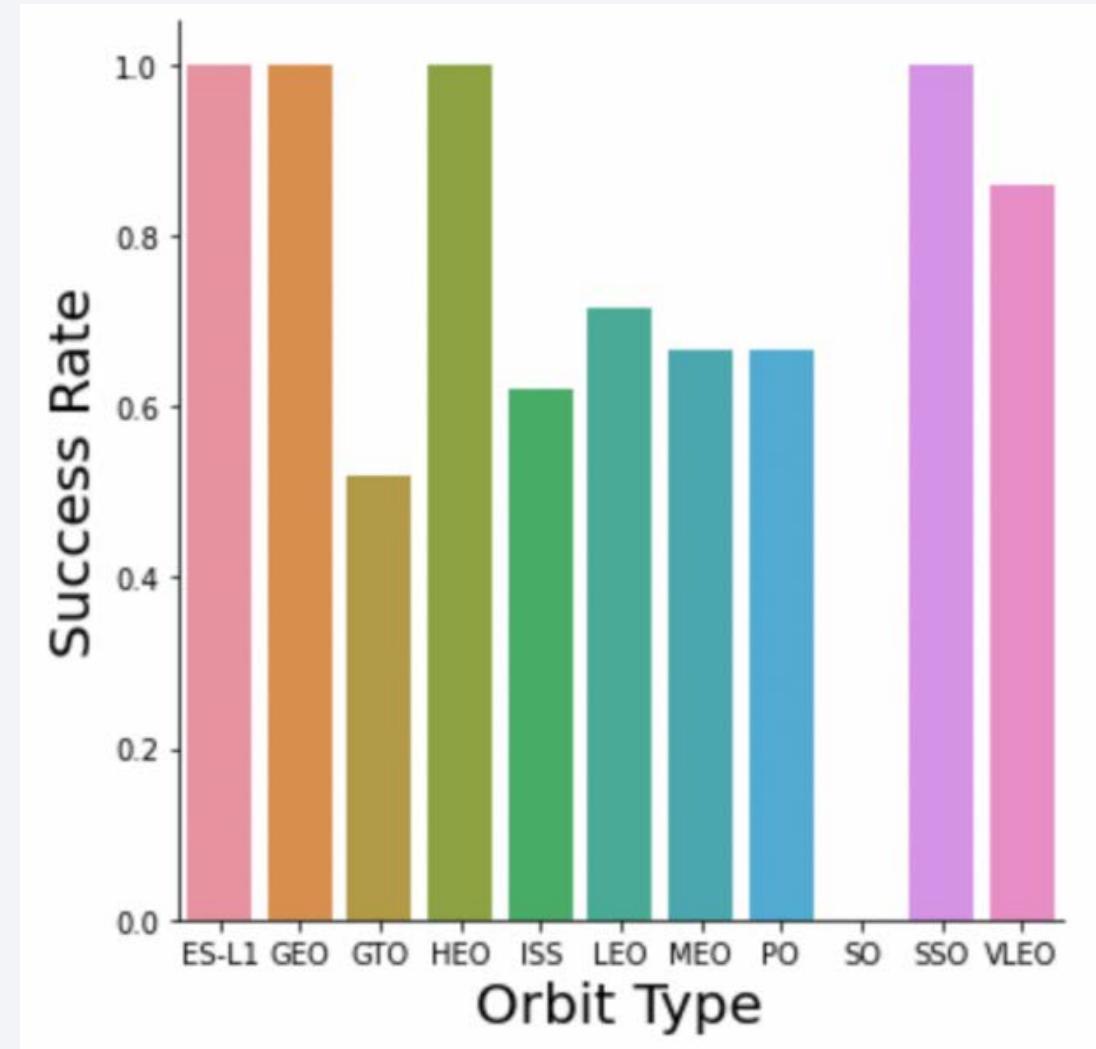
# Payload vs. Launch Site



- For every launch site the higher the payload mass, the higher the success rate
- Most of the launches with payload mass over 7000 kg were successful
- KSC LC 39A has a 100% success rate for payload mass under 5500 kg too

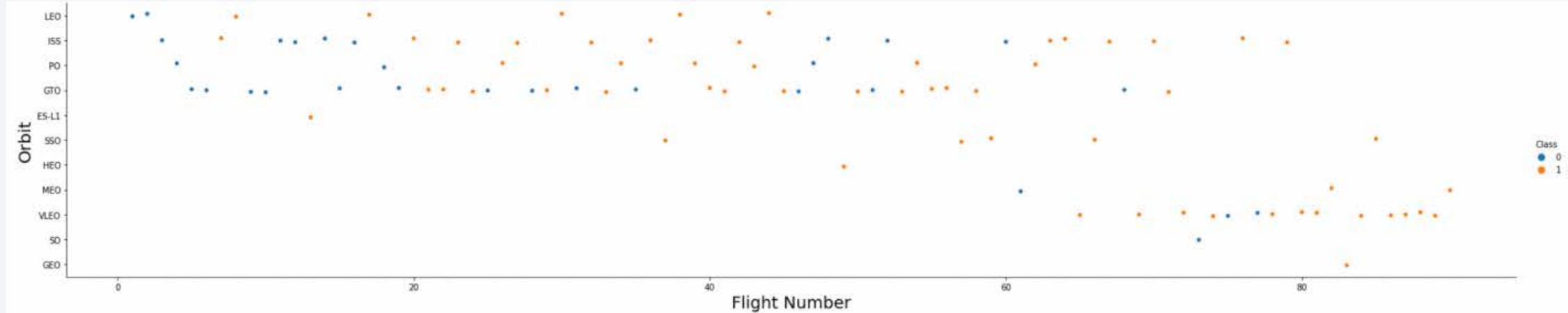
# Success Rate vs. Orbit Type

- 100% Success rate
  - ES-L1, GEO, HEO, SSO
- 50%-85% Success rate
  - GTO, ISS, LEO, MEO, PO, VLEO
- 0% Success rate
  - SO



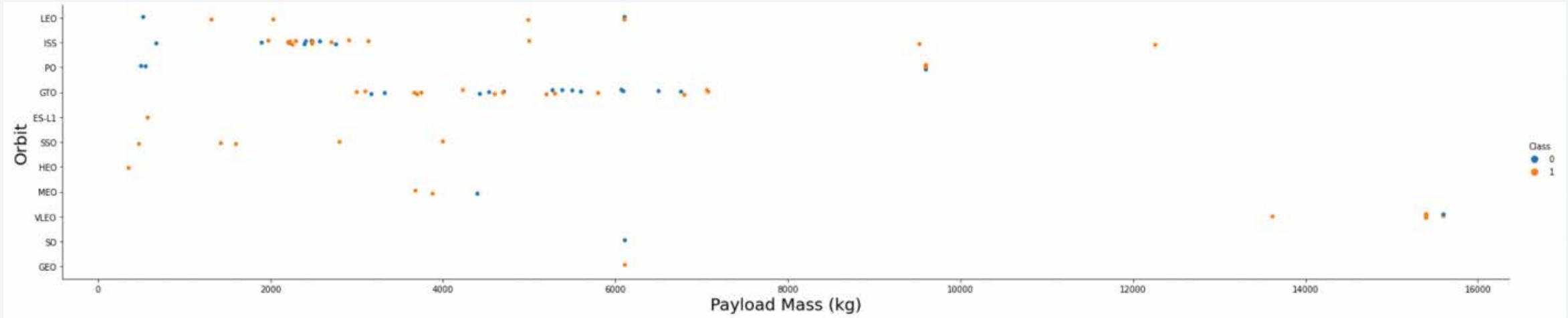
# Flight Number vs. Orbit Type

---



In the LEO orbit the Success appears related to the number of flights; however, there seems to be no relationship between flight number when in GTO orbit.

# Payload vs. Orbit Type

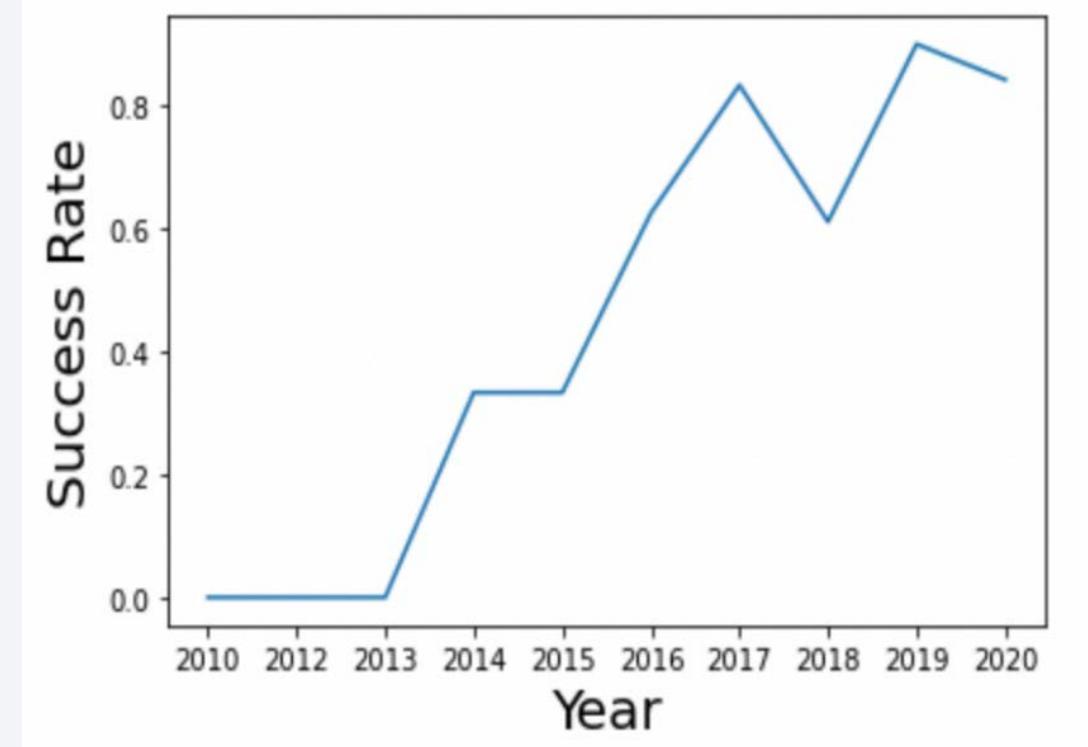


Heavy payloads have a negative influence on GTO orbits and positive on GTO and Polar LEO orbits.

# Launch Success Yearly Trend

---

The success rate increased from 2010 until 2020 with a slight dip in 2018



# All Launch Site Names

---

```
In [4]: %sql select distinct launch_site from SPACEXDATASET;  
* ibm_db_sa://wzf08322:***@0c77d6f2-5da9-48a9-81f8-86b520b87518.bs2io90108kqb1od8lcg.databases.appdomain.cloud:31198/bludb  
Done.  
Out[4]:  


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


```

Displaying the names of the unique launch sites

# Launch Site Names Begin with 'CCA'

---

In [5]: %sql select \* from SPACEXDATASET where launch\_site like 'CCA%' limit 5;

\* ibm\_db\_sa://wzf08322:\*\*\*@0c77d6f2-5da9-48a9-81f8-86b520b87518.bs2io90108kqb1od81cg.databases.appdomain.cloud:31198/bludb  
Done.

Out[5]:

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	07:44:00	F9 v1.0 B0005	CCAFS LC-40	Dragon demo flight C2	525	LEO (ISS)	NASA (COTS)	Success	No attempt
2012-10-08	00:35:00	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

Displaying 5 records where launch sites begin with 'CCA'

# Total Payload Mass

---

```
In [6]: %sql select sum(payload_mass_kg_) as total_payload_mass from SPACEXDATASET where customer = 'NASA (CRS)';

* ibm_db_sa://wzf08322:***@0c77d6f2-5da9-48a9-81f8-86b520b87518.bs2io90108kqb1od8lcg.databases.appdomain.cloud:31198/bludb
Done.

Out[6]: total_payload_mass
        45596
```

Displaying the total payload mass carried by boosters launched by NASA

# Average Payload Mass by F9 v1.1

---

```
In [7]: %sql select avg(payload_mass_kg_) as average_payload_mass from SPACEXDATASET where booster_version like '%F9 v1.1%';  
* ibm_db_sa://wzf08322:***@0c77d6f2-5da9-48a9-81f8-86b520b87518.bs2io90108kqb1od81cg.databases.appdomain.cloud:31198/bludb  
Done.
```

```
Out[7]:
```

average_payload_mass
2534

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

# First Successful Ground Landing Date

---

```
In [8]: %sql select min(date) as first_successful_landing from SPACEXDATASET where landing_outcome = 'Success (ground pad)';

* ibm_db_sa://wzf08322:***@0c77d6f2-5da9-48a9-81f8-86b520b87518.bs2io90108kqb1od81cg.databases.appdomain.cloud:31198/bludb
Done.
```

Out[8]:

first_successful_landing
2015-12-22

Listing the date when the first successful landing outcome in ground pad was achieved

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

---

```
In [9]: %sql select booster_version from SPACEXDATASET where landing_outcome = 'Success (drone ship)' and payload_mass_kg_ between 4000 and 6000;
* ibm_db_sa://wzf08322:***@0c77d6f2-5da9-48a9-81f8-86b520b87518.bs2io90108kqb1od8lcg.databases.appdomain.cloud:31198/bludb
Done.
```

Out[9]:

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

Listing the names of the boosters which have success in drone ship and have payload mass greater than 4000 but less than 6000

# Total Number of Successful and Failure Mission Outcomes

---

```
In [10]: %sql select mission_outcome, count(*) as total_number from SPACEXDATASET group by mission_outcome;  
* ibm_db_sa://wzf08322:***@0c77d6f2-5da9-48a9-81f8-86b520b87518.bs2io90108kqb1od8lcg.databases.appdomain.cloud:31198/bludb  
Done.
```

Out[10]:

mission_outcome	total_number
Failure (in flight)	1
Success	99
Success (payload status unclear)	1

Listing the total number of successful  
and failure mission outcomes

# Boosters Carried Maximum Payload

```
In [11]: %sql select booster_version from SPACEXDATASET where payload_mass_kg_ = (select max(payload_mass_kg_) from SPACEXDATASET);
* ibm_db_sa://wzf08322:***@0c77d6f2-5da9-48a9-81f8-86b520b87518.bs2io90108kqb1od81cg.databases.appdomain.cloud:31198/bludb
Done.
```

Out[11]:

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

Listing the names of the booster versions which have carried the maximum payload mass

# 2015 Launch Records

---

```
In [12]: %%sql select monthname(date) as month, date, booster_version, launch_site, landing_outcome from SPACEXDATASET  
where landing_outcome = 'Failure (drone ship)' and year(date)=2015;  
* ibm_db_sa://wzf08322:***@0c77d6f2-5da9-48a9-81f8-86b520b87518.bs2io90108kqb1od8lcg.databases.appdomain.cloud:31198/bludb  
Done.
```

Out[12]:

MONTH	DATE	booster_version	launch_site	landing_outcome
January	2015-01-10	F9 v1.1 B1012	CCAFS LC-40	Failure (drone ship)
April	2015-04-14	F9 v1.1 B1015	CCAFS LC-40	Failure (drone ship)

Listing the failed landing outcomes in drone ship, their booster versions and launch site names for the months in year 2015

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

```
In [13]: %%sql select landing_outcome, count(*) as count_outcomes from SPACEXDATASET  
where date between '2010-06-04' and '2017-03-20'  
group by landing_outcome  
order by count_outcomes desc;  
  
* ibm_db_sa://wzf08322:****@0c77d6f2-5da9-48a9-81f8-86b520b87518.bs2io90108kqb1od8lcg.databases.appdomain.cloud:31198/bludb  
Done.
```

Out[13]:

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

Ranking the count of landing outcomes between the date 2010-06-04 and 2017-03-20 in descending order

The background of the slide is a photograph taken from space at night. It shows the curvature of the Earth against the dark void of space. City lights are visible as numerous glowing yellow and white spots, primarily concentrated in coastal and urban areas. The atmosphere appears as a thin blue layer, with darker regions indicating higher altitude or atmospheric density.

Section 3

# Launch Sites Proximities Analysis

# Launch Site Locations

---

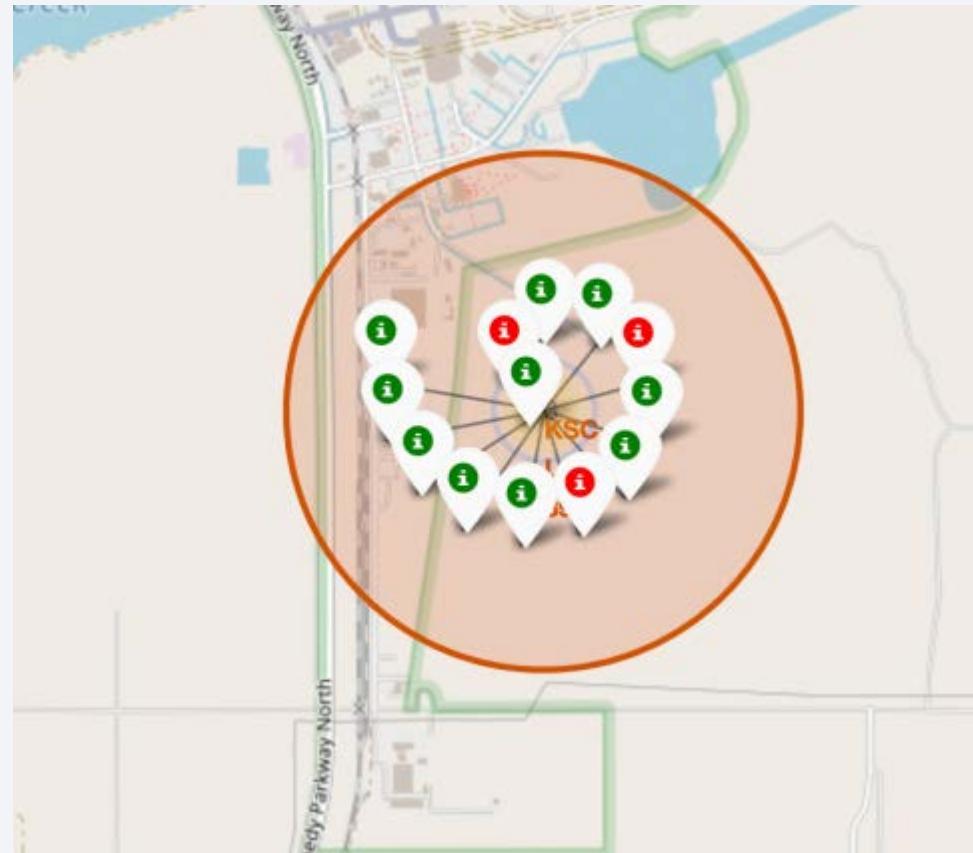
- Most of Launch sites are in proximity to the Equator line. The land is moving faster at the equator than any other place on the surface of the Earth. Anything on the surface of the Earth at the equator is already moving at 1670 km/hour. If a ship is launched from the equator it goes up into space, and it is also moving around the Earth at the same speed it was moving before launching. This is because of inertia. This speed will help the spacecraft keep up a good enough speed to stay in orbit.
- All launch sites are in very close proximity to the coast, while launching rockets towards the ocean it minimizes the risk of having any debris dropping or exploding near people.



# Colored Launch Records

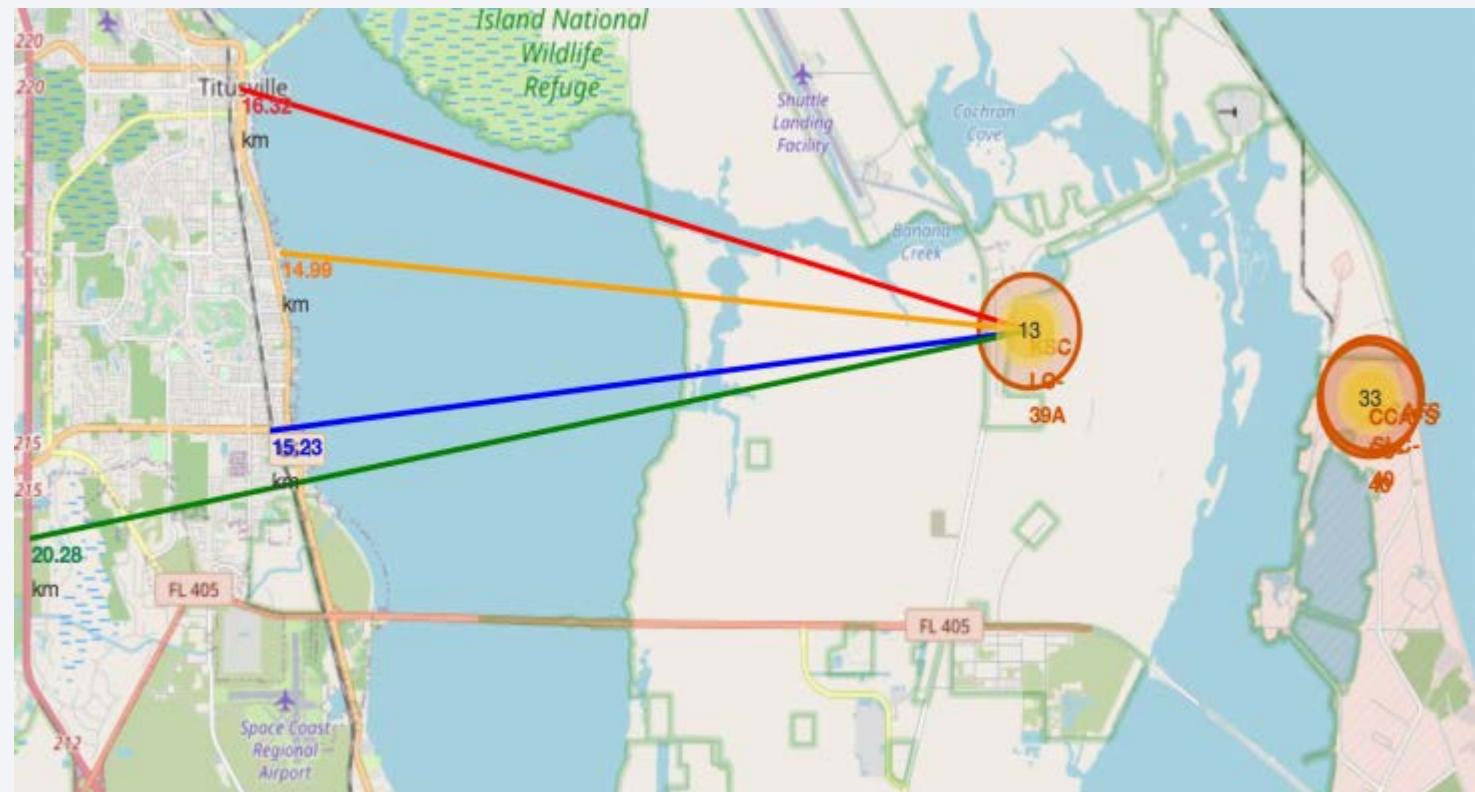
---

- From the color-labeled markers we should be able to easily identify which launch sites have relatively high success rates.
  - Green Marker = Successful Launch
  - Red Marker = Failed Launch
- Launch Site KSC LC-39A has a very high Success Rate.



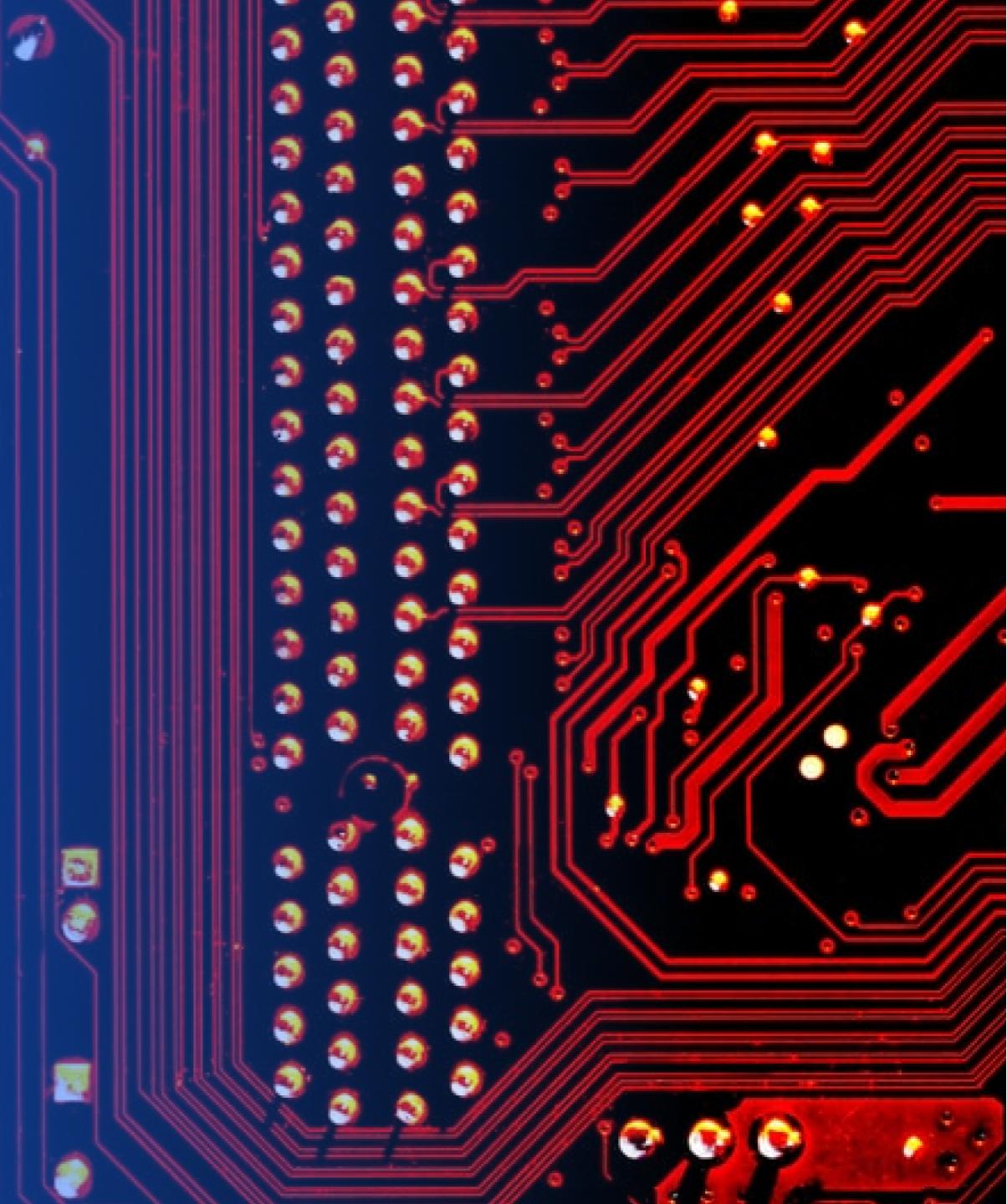
# Distance from the Launch Sites

- From the visual analysis of the launch site KSC LC-39A we can clearly see that it is:
  - relative close to railway (15.23 km)
  - relative close to highway (20.28 km)
  - relative close to coastline (14.99 km)
- Also, the launch site KSC LC-39A is relatively close to its closest city Titusville (16.32 km).
- Failed rocket with its high speed can cover distances like 15-20 km in few seconds. It could be potentially dangerous to populated areas.



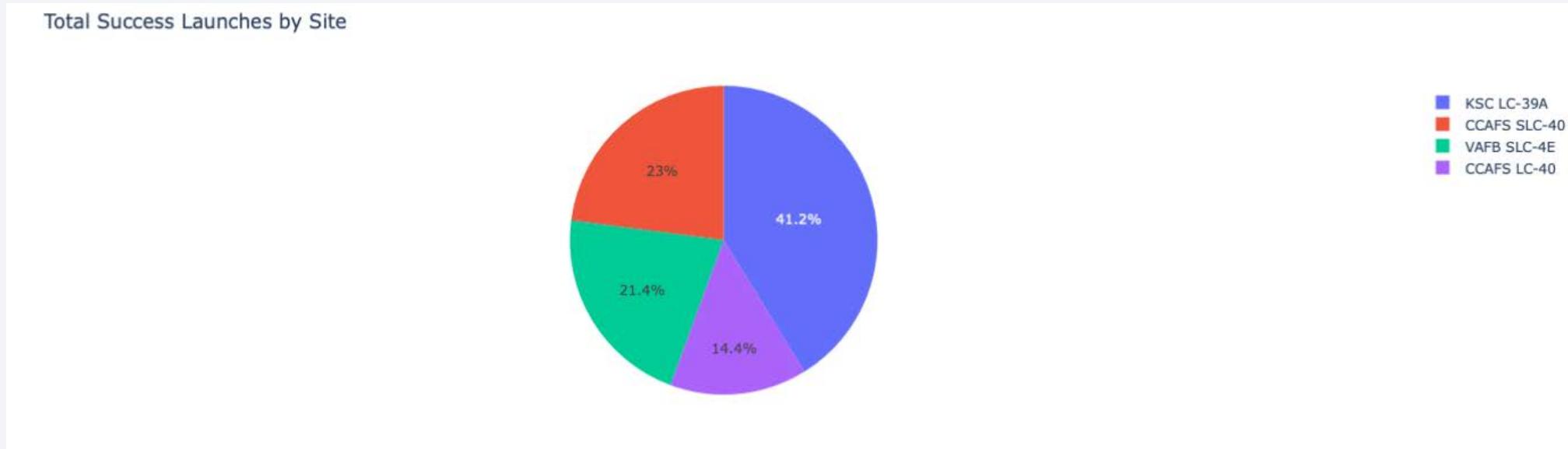
Section 4

# Build a Dashboard with Plotly Dash



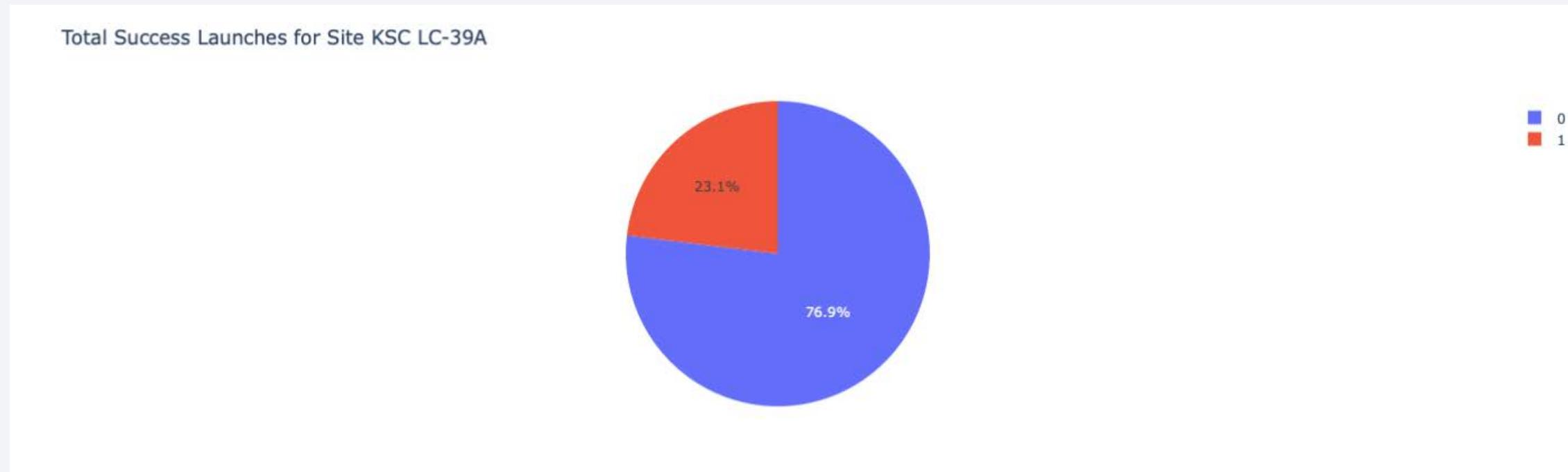
# Launch Success Count

---



The chart clearly shows that from all the sites, KSC LC-39A has the most successful launches.

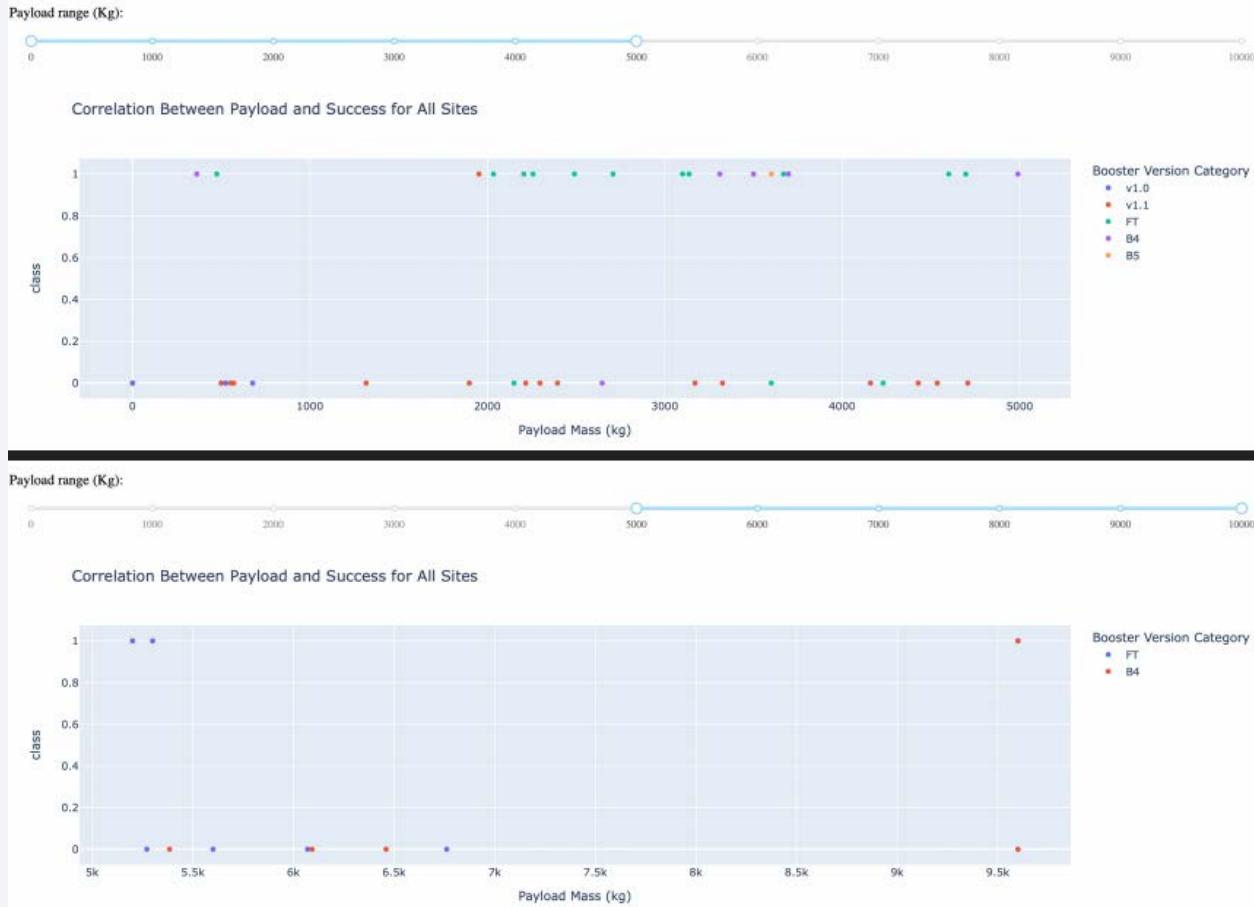
# Launch Site Success Ratio



KSC LC-39A has the highest launch success rate (76.9%) with 10 successful and only 3 failed landings.

# Payload Mass vs Launch Outcome

The charts show that payloads between 2000 and 5500 kg have the highest success rate.



Section 5

# Predictive Analysis (Classification)

# Classification Accuracy

---

- We cannot confirm which method preforms best based on the test set.
- Same Test Set scores may be due to the small test sample size. Therefore, we tested all methods based on the whole Dataset.
- The scores of the whole Dataset confirm that the best model is the Decision Tree Model. This model has not only higher scores, but also the highest accuracy.

Scores and Accuracy of the Test Set

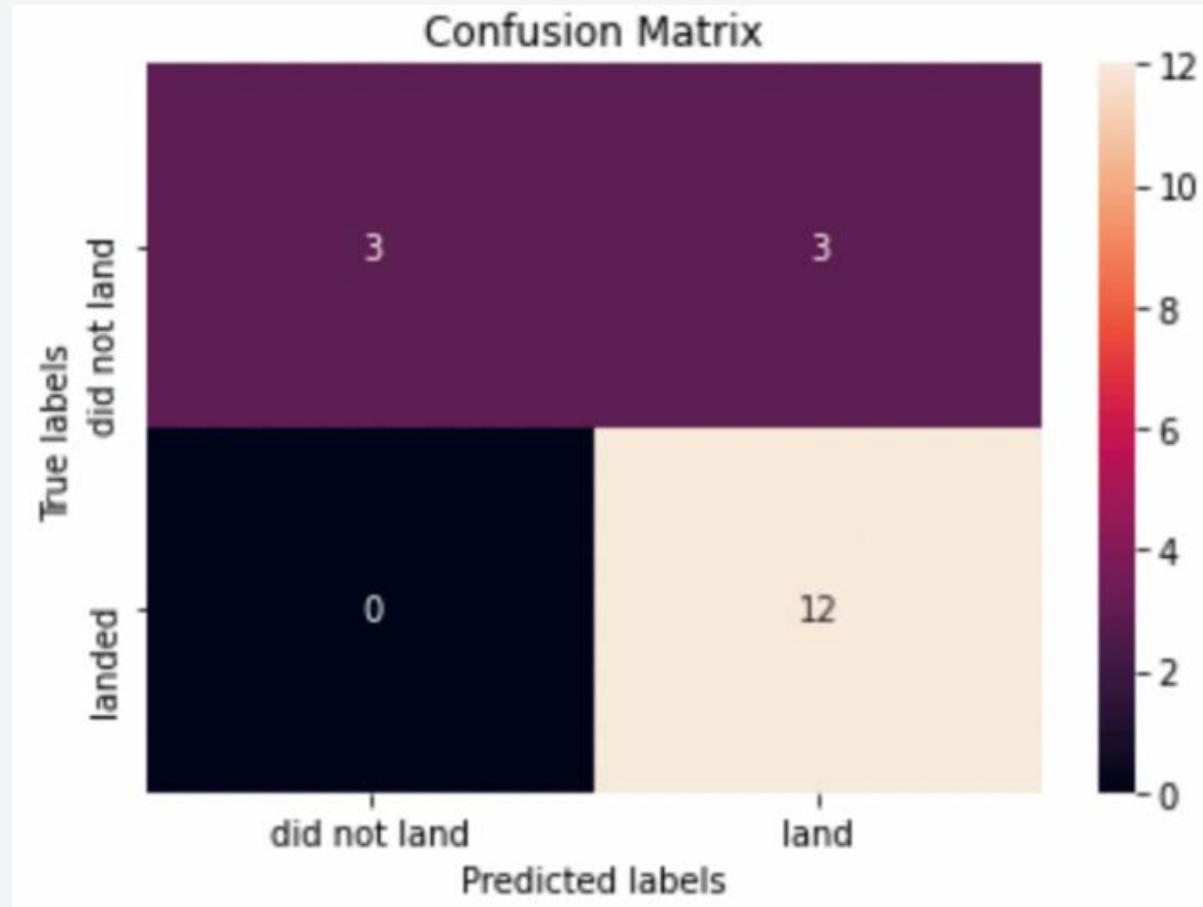
	<b>LogReg</b>	<b>SVM</b>	<b>Tree</b>	<b>KNN</b>
<b>Jaccard_Score</b>	0.800000	0.800000	0.800000	0.800000
<b>F1_Score</b>	0.888889	0.888889	0.888889	0.888889
<b>Accuracy</b>	0.833333	0.833333	0.833333	0.833333

	<b>LogReg</b>	<b>SVM</b>	<b>Tree</b>	<b>KNN</b>
<b>Jaccard_Score</b>	0.833333	0.845070	0.882353	0.819444
<b>F1_Score</b>	0.909091	0.916031	0.937500	0.900763
<b>Accuracy</b>	0.866667	0.877778	0.911111	0.855556

# Confusion Matrix

Examining the confusion matrix, we see that logistic regression can distinguish between the different classes. We see that the major problem is false positives.

		Predicted Values	
		Negative	Positive
Actual Values	Negative	TN	FP
	Positive	FN	TP



# Conclusions

---

- Decision Tree Model is the best algorithm for this dataset.
- Launches with a low payload mass show better results than launches with a larger payload mass.
- Most of launch sites are in proximity to the Equator line and all the sites are in very close proximity to the coast.
- The success rate of launches increases over the years.
- KSC LC-39A has the highest success rate of the launches from all the sites.
- Orbits ES-L1, GEO, HEO and SSO have 100% success rate.

# Appendix

---

[GITHUB](#) link

Thank you!

