

Winning Space Race with Data Science



Outline

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

Executive Summary

- Summary of Methodologies
 - Data Collection
 - Data Wrangling
 - EDA with Data Visualization
 - EDA with SQL
 - Building an interactive map with Folium
 - Building a dashboard with Plotly
 - Predictive analysis
- Summary of results
 - Exploratory data analysis results
 - Interactive analytics demo
 - Predictive analysis results

Introduction

- Project background and context

Our firm has predicted whether the Falcon 9 first stage will land successfully. The key sell about SpaceX is its low cost of space entry, roughly \$62 million, nearly 1/3 its competitors. By seeing if the first stage will land, we can see how much each launch will cost. Thus, by launching a competing company, we can focus on this aspect.

- Problems you want to find answers

What factors play a role in successful rocket launches?

What relationships affecting rockets can determine the success rate?

How does SpaceX achieve such great results?

Section 1

Methodology

Methodology

Executive Summary

- Data collection methodology
 - SpaceX API
 - Web Scraping from Wikipedia
- Perform data wrangling
 - One Hot Encoding for data fields for ML
- 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

- Data Collection From API

Response was retrieved from API -> Convert Response to JSON file -> Apply Custom Functions to Clean Data -> Assign List to dictionary then dataframe -> Filter dataframe and export to CSV

- Data Collection from Scraping

Get response from HTML -> Create BeautifulSoup object -> Find tables in content
-> Get Column Names -> Append Data to keys -> Convert dictionary to dataframe
-> Dataframe is sent to CSV

Data Collection – SpaceX API

Notebook

The screenshot shows a Jupyter Notebook interface with a dashed blue border around the code cells. The notebook contains three main sections:

- Fetch API**:
Code:

```
In [60]: spacex_url="https://api.spacexdata.com/v1/launches/past"
response = requests.get(spacex_url).json()

In [61]: response = requests.get(spacex_url)

Check the content of the response
```

Task 1: Request and parse the SpaceX launch data using the GET request
To make the requested JSON results more consistent, we will use the following static response object for this project.
- Convert to JSON**:
Code:

```
In [63]: static_json_url="https://cf-courses-data.s3.us.cloud-object-storage.appdomain.cloud/IBM-080912198-SkillNetwork/datasets/APIShell SpaceX API.json"

We should see that the request was successful with the 200 status response code
In [64]: response.status_code
Out[64]: 200

Now we decode the response content as a json using .json() and turn it into a Pandas dataframe using .json_normalize()
In [65]: # Use json_normalize method to convert the json result into a dataframe
response = requests.get(static_json_url).json()
data = pd.json_normalize(response)
```
- Make dict and export to CSV**:
Code:

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

Data Collection - Scraping

- [Notebook](#)

```
# use requests.get() method with the provided static_url
# Assign the response to page
page = requests.get(static_url)
page.status_code
```

200

Create a BeautifulSoup object from the HTML response

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

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

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

```
launch_dict['Flight No.']= []
launch_dict['Launch Date']= []
launch_dict['Payload']= []
launch_dict['Payload mass']= []
launch_dict['Orbit']= []
launch_dict['Customer']= []
launch_dict['Launch outcome']= []
launch_dict['Version Booster']= []
launch_dict['Booster landing']= []
launch_dict['Date']= []
launch_dict['Time']= []
```

Create categories for data

```
# Use the find_all function in the BeautifulSoup object,
# Assign the result to a list called 'html_tables'
html_tables = soup.find_all('table')
```

Starting from the third table is our target table contains the actual launch records

```
# Let's print the third table and check its content
first_launch_table = html_tables[2]
print(first_launch_table)
```

After you have filled in the parsed launch record values into launch_dict

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

We can now export it to a CSV for the next section, but to make the analysis easier

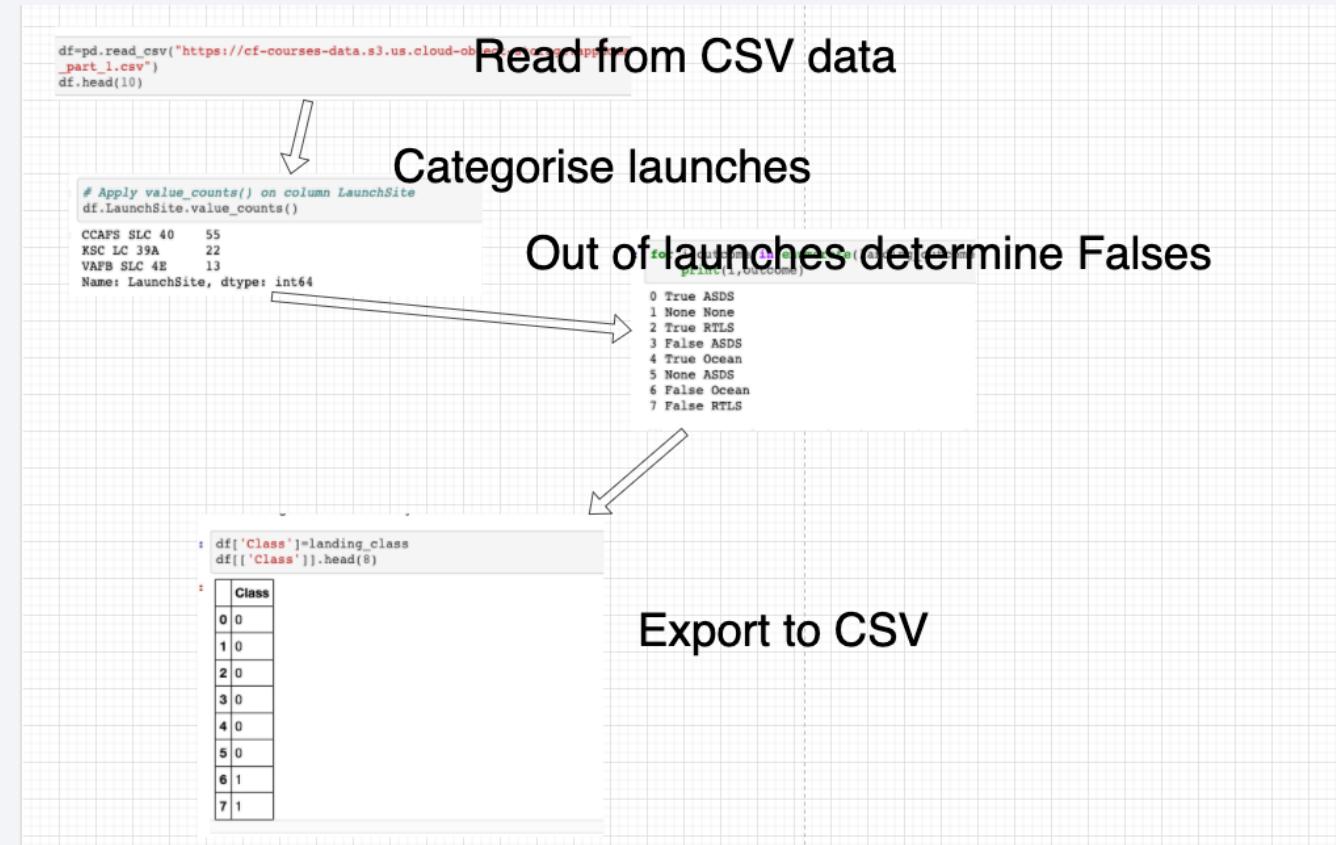
Following labs will be using a one record database to make each lab independent

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

Export to CSV

Data Wrangling

- From the data, there were launches that were not successful and the stage did not land successfully. The data was converted where 1 indicated a successful landing and 0 did not.
 - [notebook](#)



EDA with Data Visualization

- Scatter Graphs

Flight Number vs Payload Mass

Flight Number vs Launch Site

Payload vs Launch Site

Orbit vs Flight Number

Payload vs Orbit Type

Orbit vs Payload Mass

[notebook](#)

- Bar Graph

Mean vs Orbit

- Line Graph

Success rate vs Year

EDA with SQL

- SQL Queries done for our research
- Display the names of unique launch sites
- Display records starting with KSC
- Display the total mass carried by NASA's rockets
- Displayed the average payload carried by Flacon 9
- Listed the dates for successful landings on the drone ship
- Listed the names of boosters that landed on ground pad with mass between 4k and 6k
- Listed the total number of successful and failed outcomes
- Listed the names of booster versions which carried maximum payload capacity
- Listed the records to show the month name, success on ground pad, booster version, and launch site in 2017
- Ranked the count of successful landing outcomes between 2010 and 2017

Build an Interactive Map with Folium

- We used the longitude and latitude then added circle markers around them for easier visualization for launch data.
 - Successful and Failed launches were marked as Green and Red respectively.
 - We then added in distance markers to local points in the state.
 - Add the GitHub URL of your completed interactive map with Folium map, as an external reference and peer-review purpose
-
- [notebook](#)

Build a Dashboard with Plotly Dash

- Added a pie chart to show total launches by the appointed launch site
 - Added a scatter graph to show the relationship between Outcomes and Payloads for the booster versions
-
- [dashboard](#)

Predictive Analysis (Classification)

- Building the model

Load the dataset into Numpy and Pandas

Used data transformation for splitting into training and test data

Checked to see how many test samples there were

Decided which type of ML algorithms to use based on performance

Set parameters and algorithms for GridSearch

Fitted the datasets into GridSearch and trained against our dataset

- Evaluating the model

Checked the accuracy for each model

Understood which parameters were optimal for our training

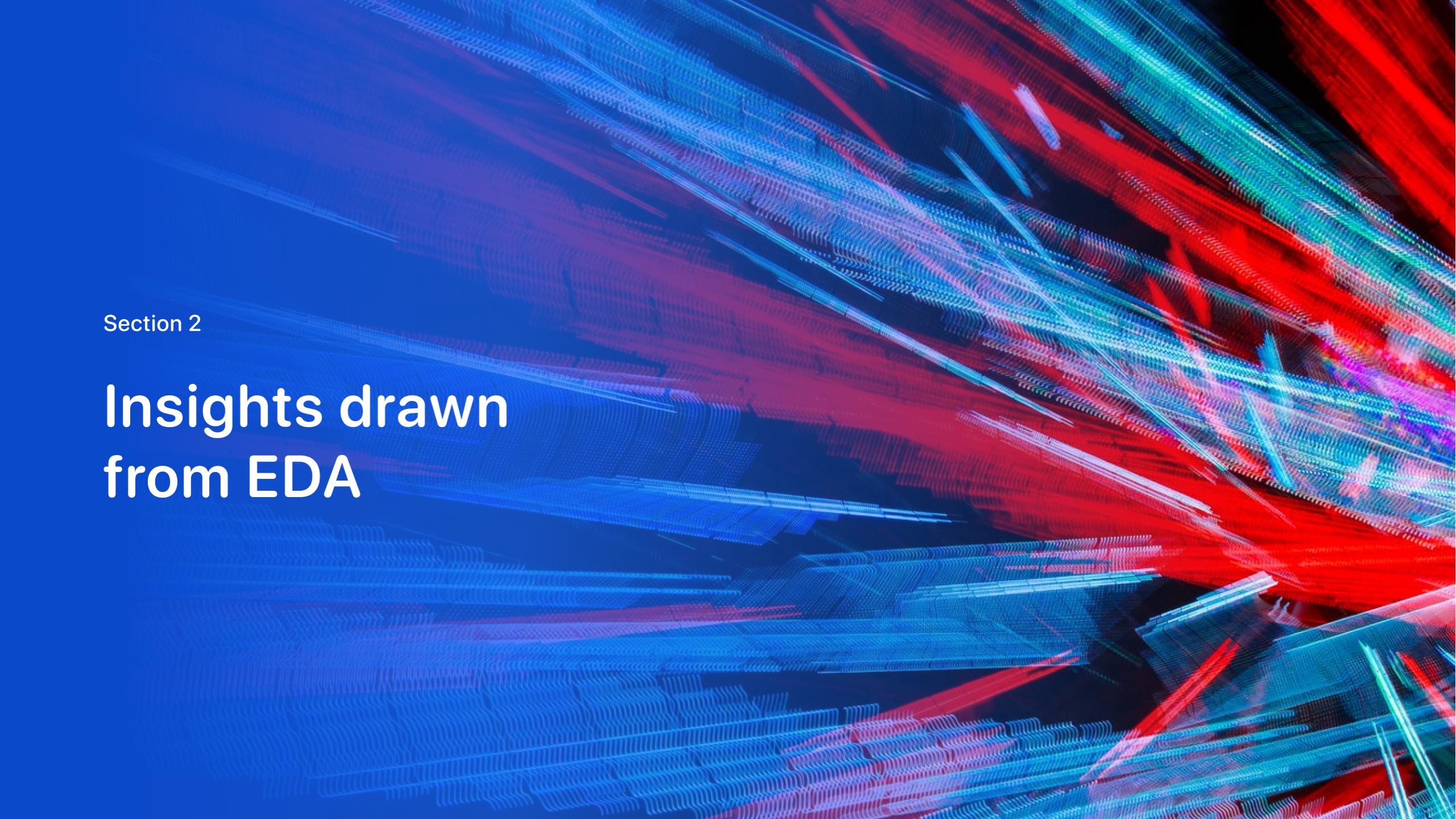
Improved the algorithms by changing the parameters

Found the best performing classification by the accuracy score

- [notebook](#)

Results

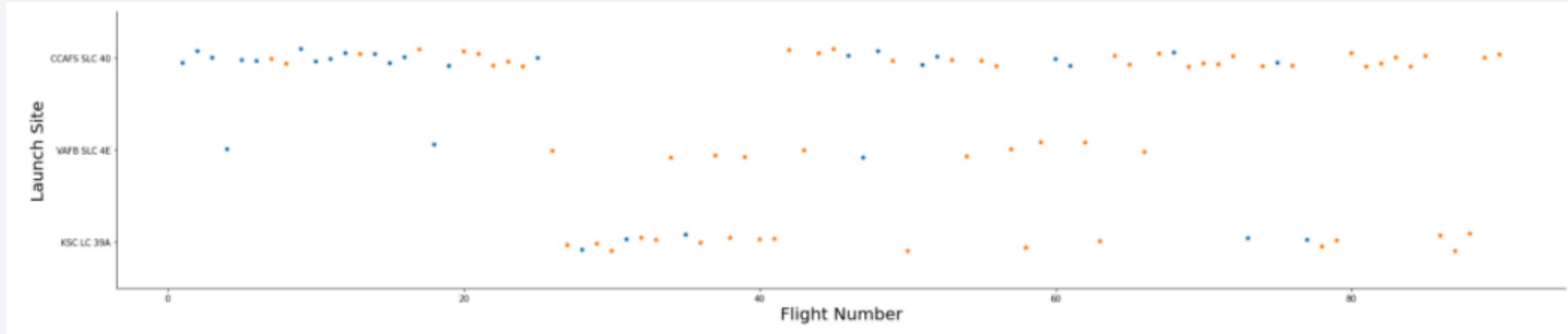
- Exploratory data analysis results
- Interactive analytics demo in screenshots
- Predictive analysis results

The background of the slide features a dynamic, abstract pattern of glowing particles. The particles are primarily blue and red, creating a sense of motion and depth. They are arranged in several parallel, slightly curved bands that radiate from the bottom right corner towards the top left. The intensity of the light varies, with some particles being brighter than others, which adds to the overall luminosity and three-dimensional feel of the design.

Section 2

Insights drawn from EDA

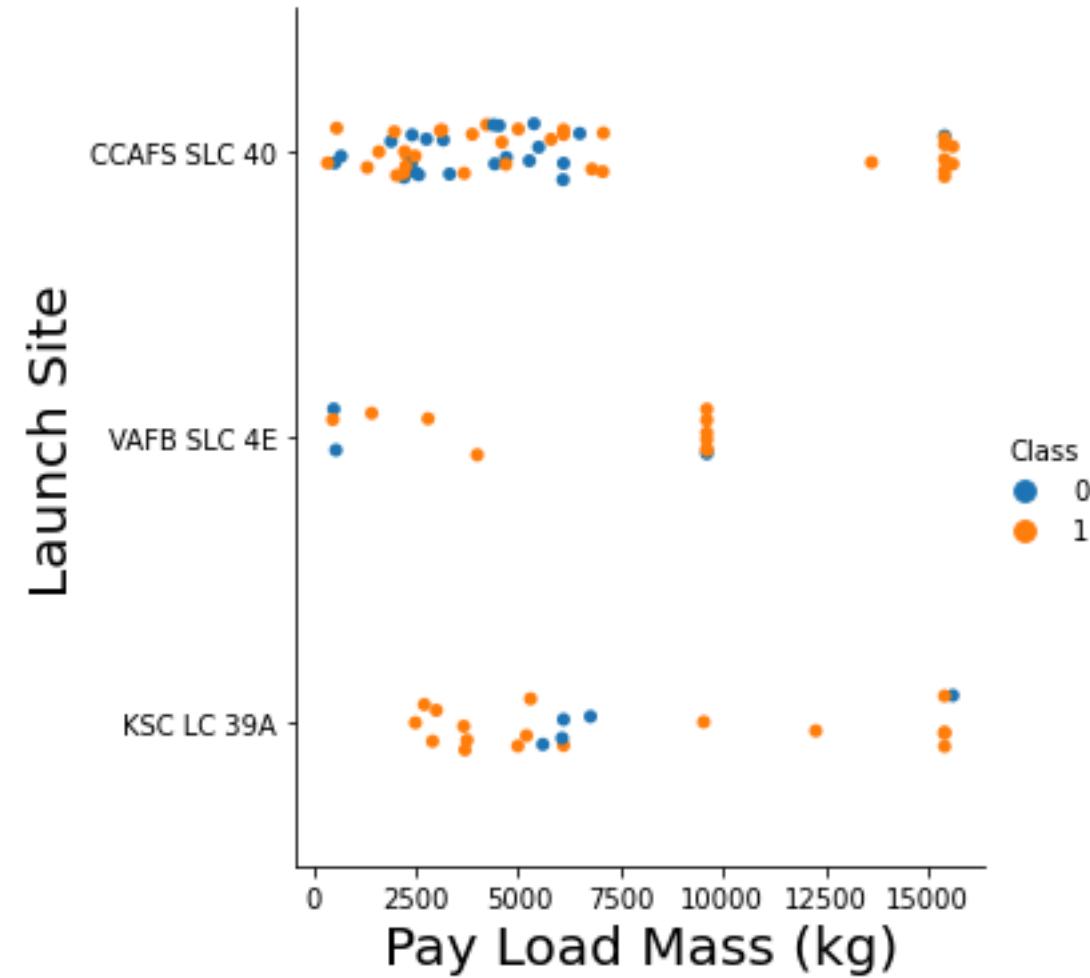
Flight Number vs. Launch Site



- More flights at the site meant higher success rate

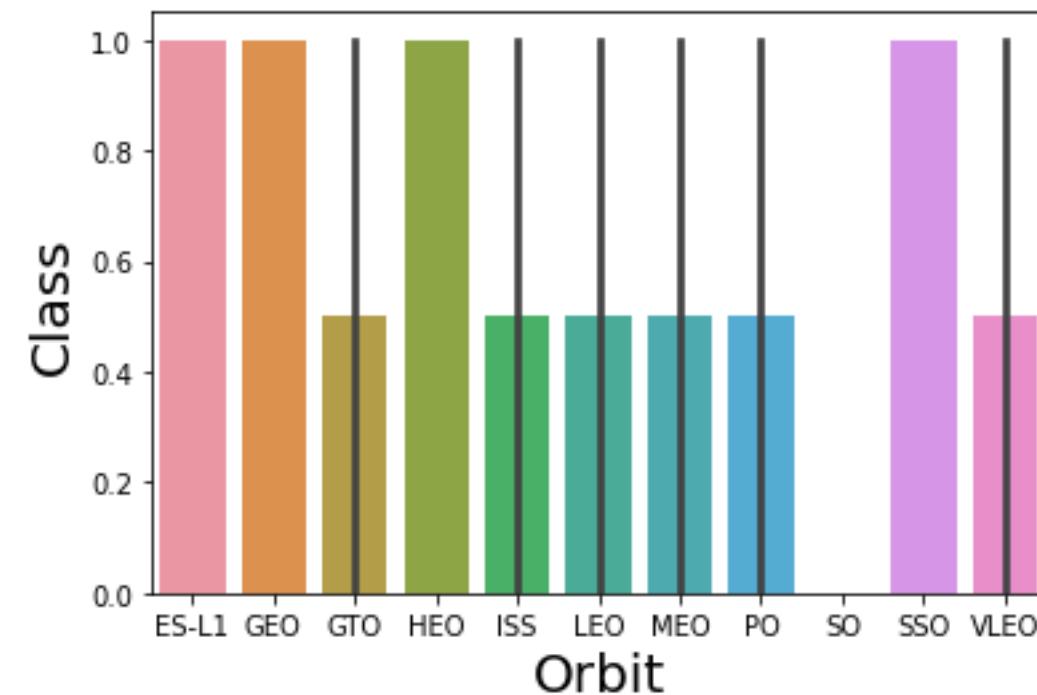
Payload vs. Launch Site

- The more payload the higher success rate.



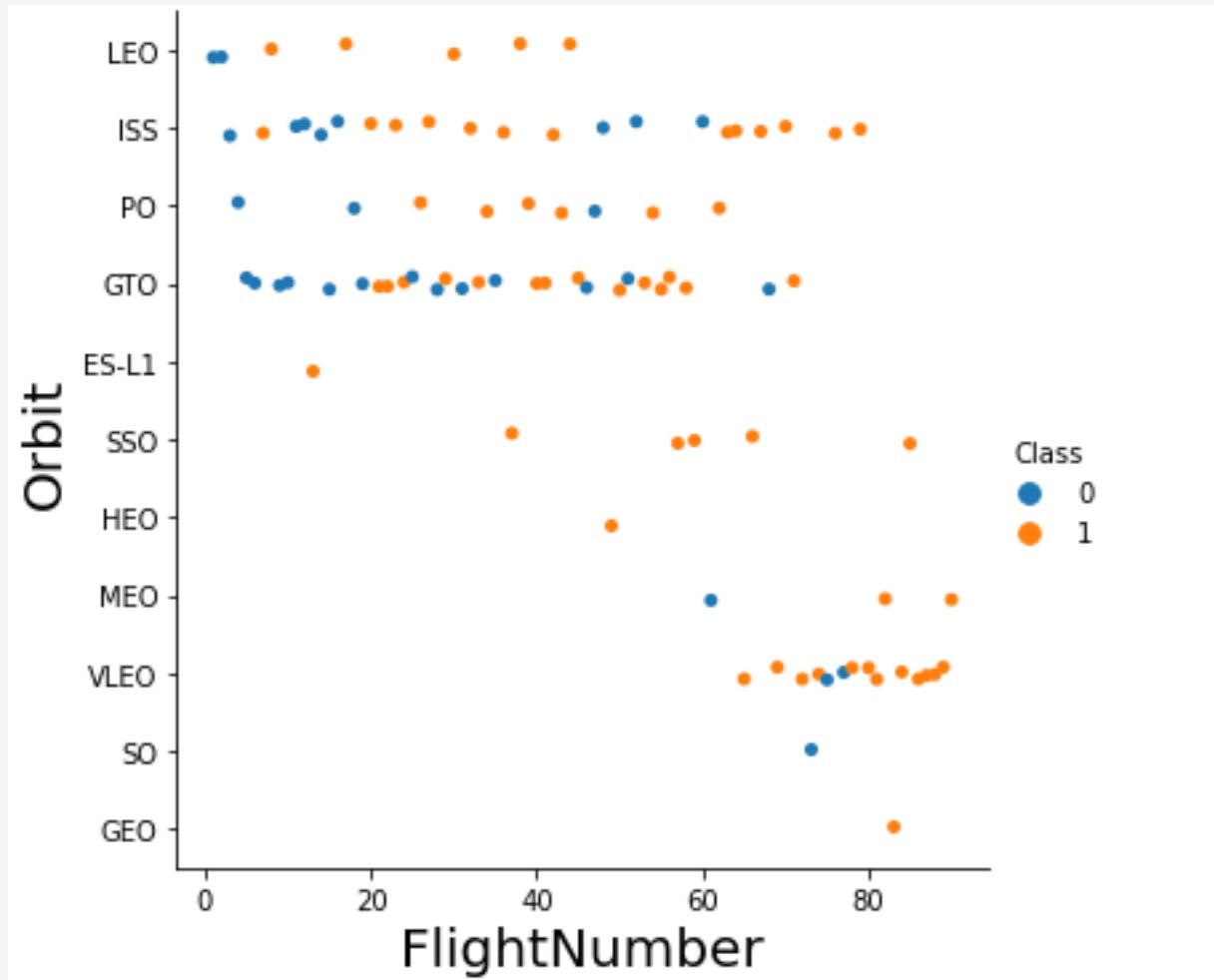
Success Rate vs. Orbit Type

- ES, GEO, HEO, and SSO had the best success rates



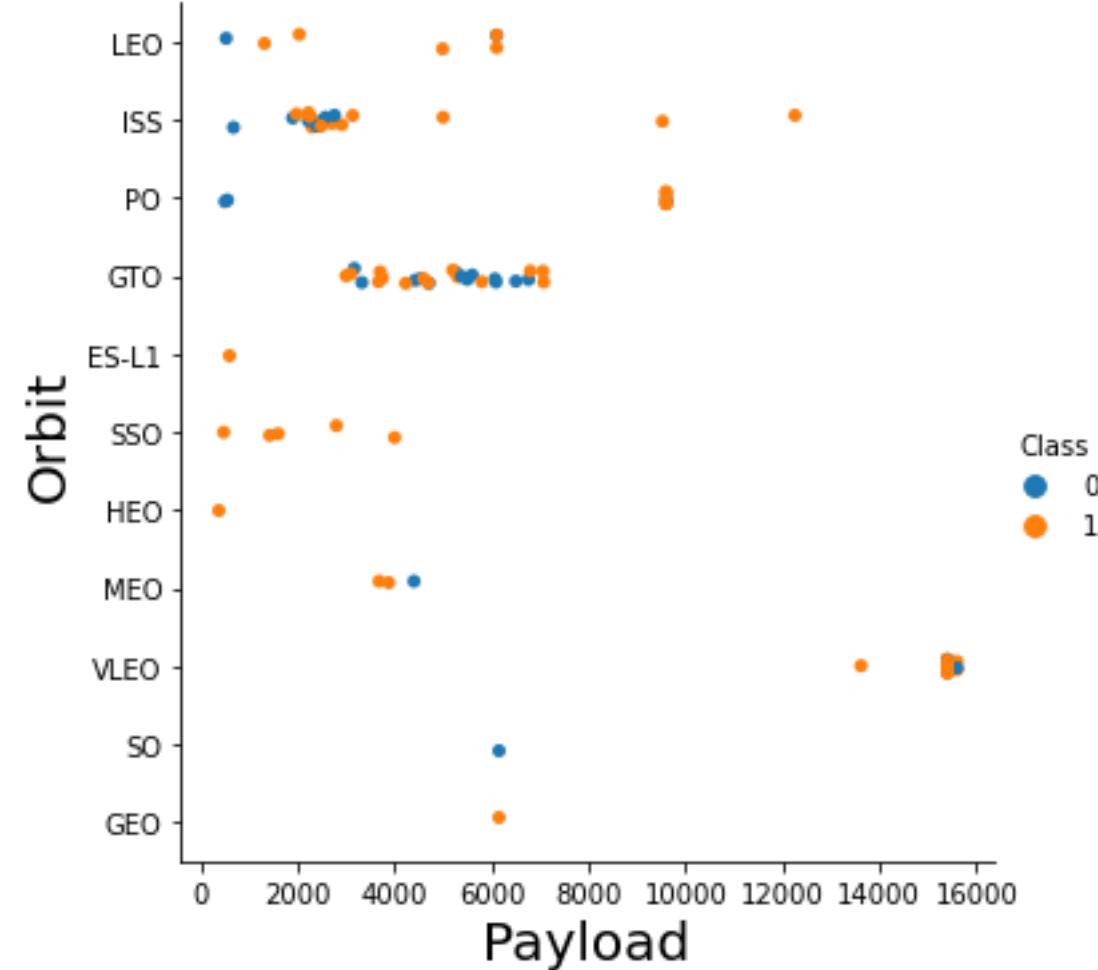
Flight Number vs. Orbit Type

- Relation between success and number of flights is visible.



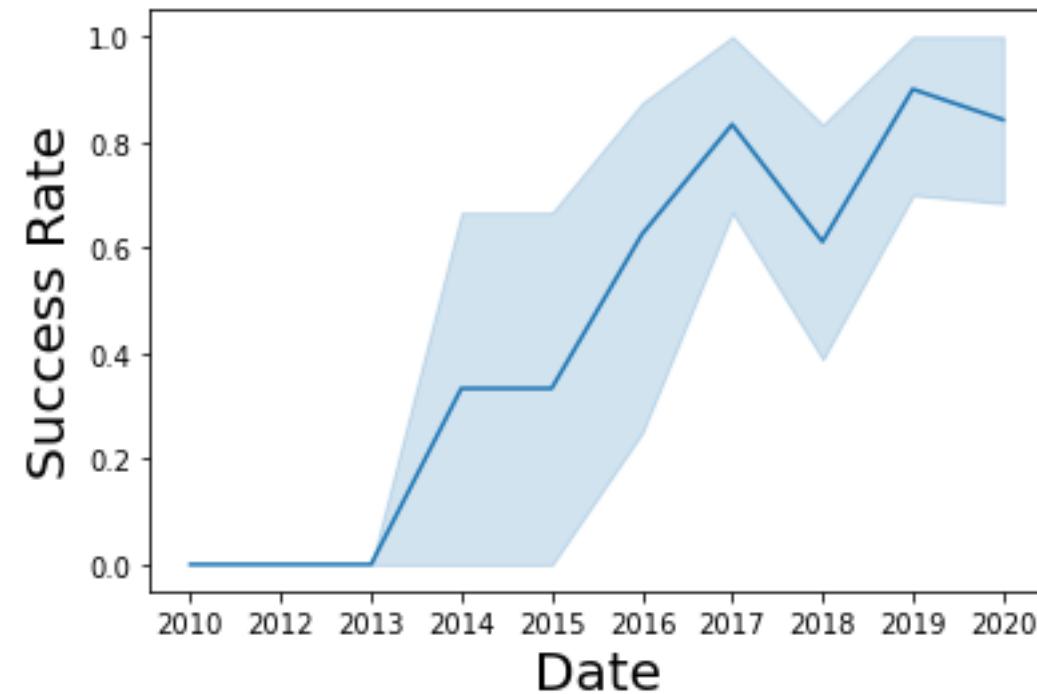
Payload vs. Orbit Type

- Heavy payloads do not do well with GTO flights but do well when shuttling to the ISS



Launch Success Yearly Trend

- Their success rate is doing very well



All Launch Site Names

- Launch sites are CCAFS LC-40
CCAFS SLC-40
KSC LC-39A
VAFB SLC-4E

Display the names of the unique launch sites in the space mission

```
: %sql select distinct(LAUNCH_SITE) from SPACEXDATASET  
* ibm_db_sa://ppb98920:***@54a2f15b-5c0f-46df-8954-7e38  
Done.
```

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

Launch Site Names Begin with 'CCA'

```
%sql select * from SPACEXDATASET where LAUNCH_SITE like 'CCA%' limit 5
```

```
* ibm_db_sa://ppb98920:***@54a2f15b-5c0f-46df-8954-7e38e612c2bd.clogj3sd0tgtu0lgde00.databases.appdomain.cloud:32733/bludb  
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 Brie 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

- Queries should have like because of name similarities between sites

Total Payload Mass

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

```
%sql select sum(PAYLOAD_MASS__KG_) from SPACEXDATASET where CUSTOMER = 'NASA (CRS)'  
* ibm_db_sa://ppb98920:***@54a2f15b-5c0f-46df-8954-7e38e612c2bd.clogj3sd0tgtu0lgde00.databases.  
Done.
```

1
45596

- SpaceX is doing lots of heavy lifting

Average Payload Mass by F9 v1.1

- V1.1 is doing well for payload capacity

Task 4

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

```
| : %sql select avg(PAYLOAD__MASS__KG_) from SPACEXDATASET where BOOSTER_VERSION = 'F9 v1.1'  
| * ibm_db_sa://ppb98920:***@54a2f15b-5c0f-46df-8954-7e38e612c2bd.clogj3sd0tgtu0lqde00.database.  
| Done.  
| : 

|      |
|------|
| 1    |
| 2928 |


```

First Successful Ground Landing Date

- Near to Christmas

QUESTION

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

Hint: Use min function

6]: %sql select min(DATE) from SPACEXDATASET where Landing__Outcome = 'Success'

* ibm_db_sa://ppb98920:***@54a2f15b-5c0f-46df-8954-7e38e612c2bd.clogj3sd0
Done.

6]:

1

2015-12-22

Successful Drone Ship Landing with Payload between 4000 and 6000

- Their success are growing

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 BOOSTER_VERSION from SPACEXDATASET where Landing_Outcome = 'Success (drone ship)' and PAYLOAD_MASS_KG_ < 6000
```

```
* ibm_db_sa://ppb98920:***@54a2f15b-5c0f-46df-8954-7e38e612c2bd.clogj3sd0tgtu0lqde00.databases.appdomain
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

- They have done many missions

List the total number of successful and failure mission outcomes

```
] : %sql select count(MISSION_OUTCOME) from SPACEXDATASET where MISSION_OUTCOME = 'Success' or MISSION_OUTCOME = 'Failure (in flight)'  
* ibm_db_sa://ppb98920:***@54a2f15b-5c0f-46df-8954-7e38e612c2bd.clogj3sd0tgtu0lgde00.databases.appdomain.cloud:32733/bludb  
Done.  
] : 

|     |
|-----|
| 1   |
| 100 |


```

Boosters Carried Maximum Payload

- The B5 boosters are quite powerful

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

```
%sql select BOOSTER_VERSION from SPACEXDATASET where PAYLOAD_MASS_KG_ = (select max(PAYLOAD_MASS_KG_) from SPACEXDATASET)  
* ibm_db_sa://ppb98920:***@54a2f15b-5c0f-46df-8954-7e38e612c2bd.clogj3sd0tgtu0lqde00.databases.appdomain.cloud:32733/bludb  
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

- They are learning from their mistakes

```
*sql SELECT MONTH(DATE),MISSION_OUTCOME,BOOSTER_VERSION,LAUNCH_SITE FROM SPACEXDATASET where EXTRACT(YEAR FROM DATE)='2015' AND MISSION_OUTCOME LIKE N'%Failure%';
* ibm_db_sa://ppb98920:***@54a2f15b-5c0f-46df-8954-7e38e612c2bd.clogj3sd0tgtu0lgde00.databases.appdomain.cloud:32733/bludb
Done.

Out[63]:
```

	mission_outcome	booster_version	launch_site
6	Failure (in flight)	F9 v1.1 B1018	CCAFS LC-40

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

```
%sql select * from SPACEXDATASET where Landing_Outcome like 'Success%' and (DATE between '2010-06-04' and '2017-03-20') order by date desc
```

```
* ibm_db_sa://ppb98920:***@54a2f15b-5c0f-46df-8954-7e38e612c2bd.clogj3sd0tgtu0lgde00.databases.appdomain.cloud:32733/bludb  
Done.
```

DATE	time_utc	booster_version	launch_site	payload	payload_mass_kg	orbit	customer	mission_outcome	landing_outcome
2017-02-19	14:39:00	F9 FT B1031.1	KSC LC-39A	SpaceX CRS-10	2490	LEO (ISS)	NASA (CRS)	Success	Success (ground pad)
2017-01-14	17:54:00	F9 FT B1029.1	VAFB SLC-4E	Iridium NEXT 1	9600	Polar LEO	Iridium Communications	Success	Success (drone ship)
2016-08-14	05:26:00	F9 FT B1026	CCAFS LC-40	JCSAT-16	4600	GTO	SKY Perfect JSAT Group	Success	Success (drone ship)
click to scroll output; double click to hide				B1025.1	CCAFS LC-40	SpaceX CRS-9	2257	LEO (ISS)	NASA (CRS)
2016-05-27	21:39:00	F9 FT B1023.1	CCAFS LC-40	Thaicom 8	3100	GTO	Thaicom	Success	Success (drone ship)
2016-05-06	05:21:00	F9 FT B1022	CCAFS LC-40	JCSAT-14	4696	GTO	SKY Perfect JSAT Group	Success	Success (drone ship)
2016-04-08	20:43:00	F9 FT B1021.1	CCAFS LC-40	SpaceX CRS-8	3136	LEO (ISS)	NASA (CRS)	Success	Success (drone ship)
2015-12-22	01:29:00	F9 FT B1019	CCAFS LC-40	OG2 Mission 2 11 Orbcomm-OG2 satellites	2034	LEO	Orbcomm	Success	Success (ground pad)

The background of the slide is a photograph taken from space at night. It shows the curvature of the Earth against a dark blue and black void of space. City lights are visible as small white dots and larger clusters of light, primarily concentrated in the lower right quadrant where the United States appears. In the upper right, there are bright green and yellow bands of the Aurora Borealis (Northern Lights) dancing across the sky.

Section 4

Launch Sites Proximities Analysis

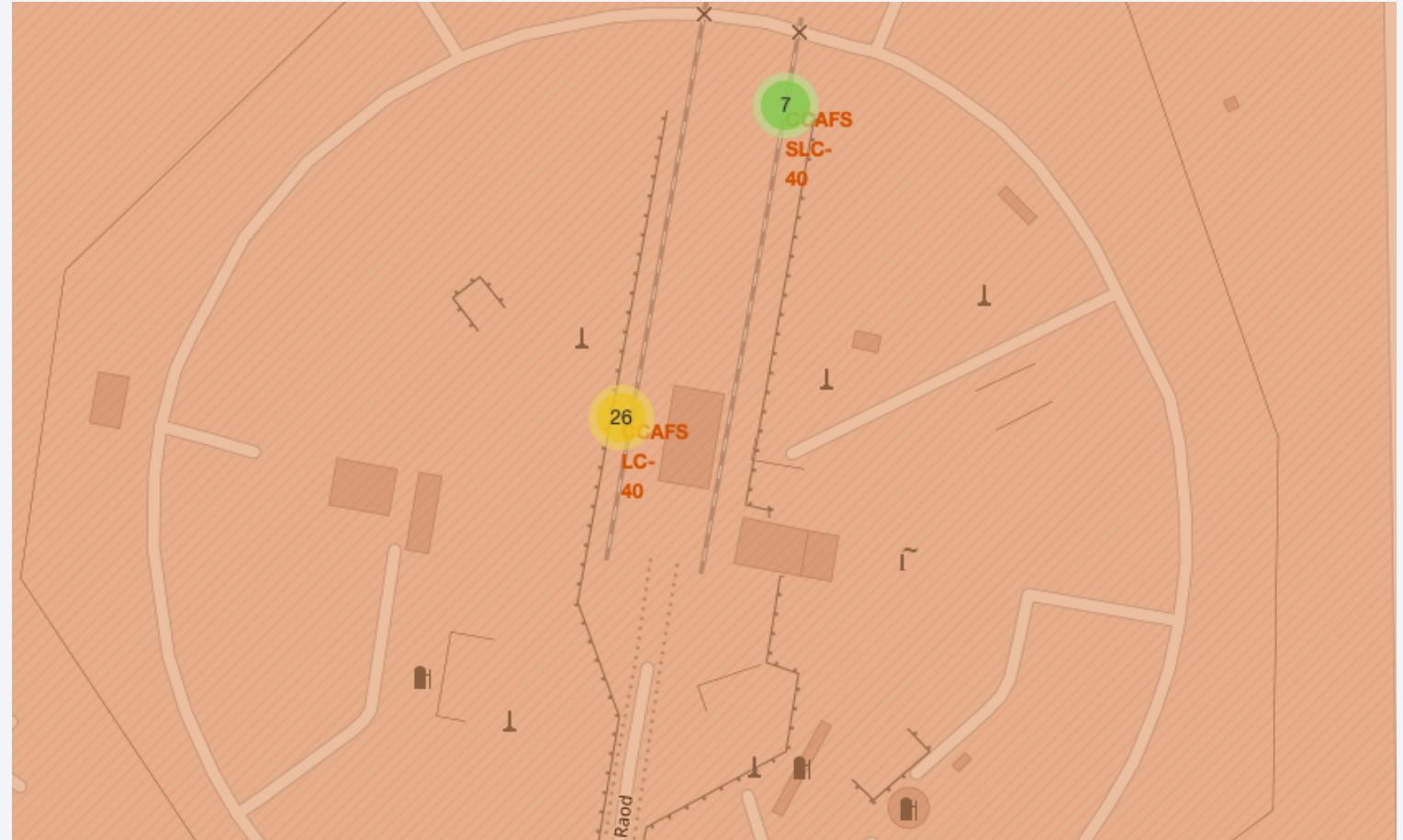
SpaceX Launch Sites

- The launch sites are in America in California and Florida
- The launch sites are in the southern area.
- Florida makes up the bulk of launches, possibly due to the sea surrounding it



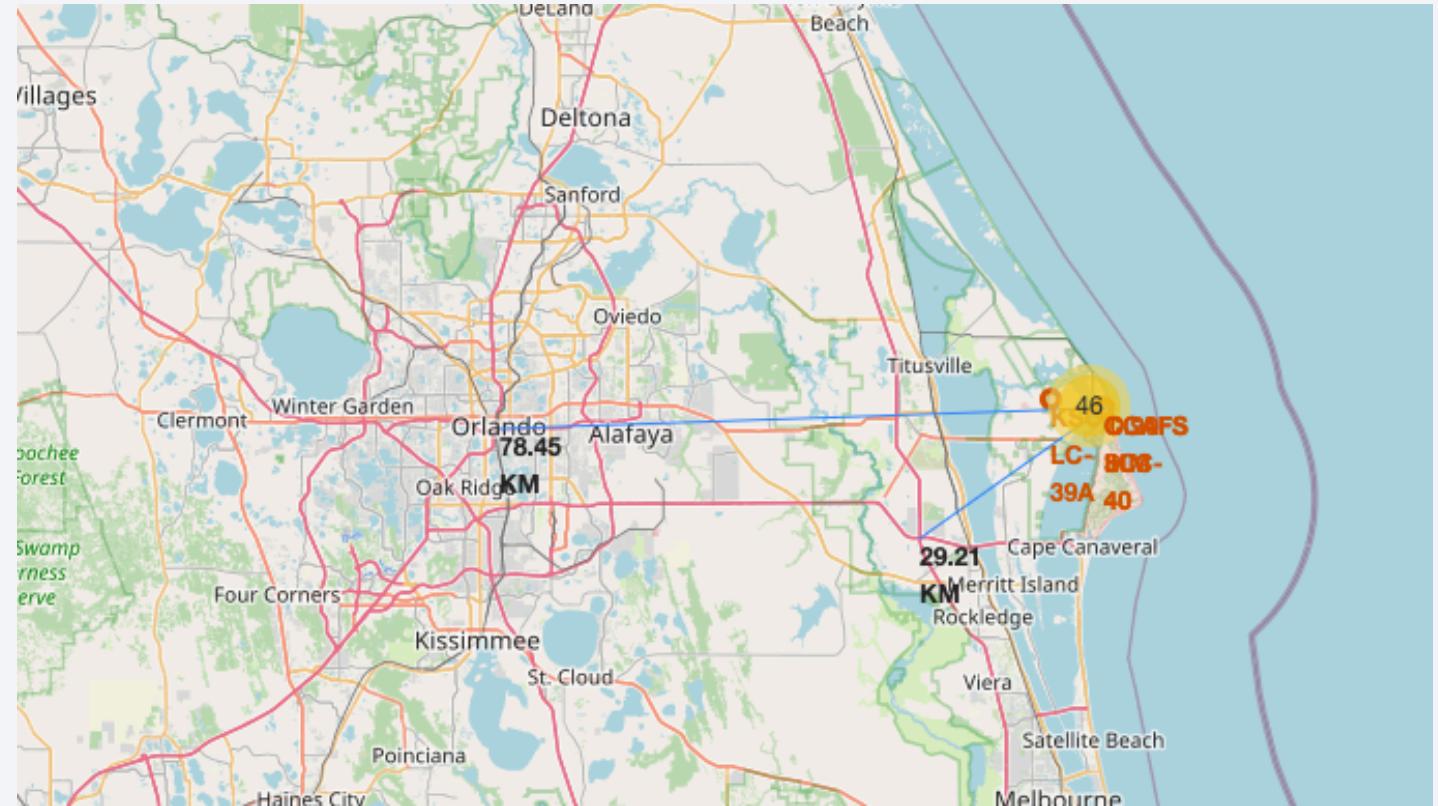
Color Labeled Launch Outcomes

- You can see that in the north, there were 7 successful launches
- For the south, expanding the view shows color while compressed shows yellow for data



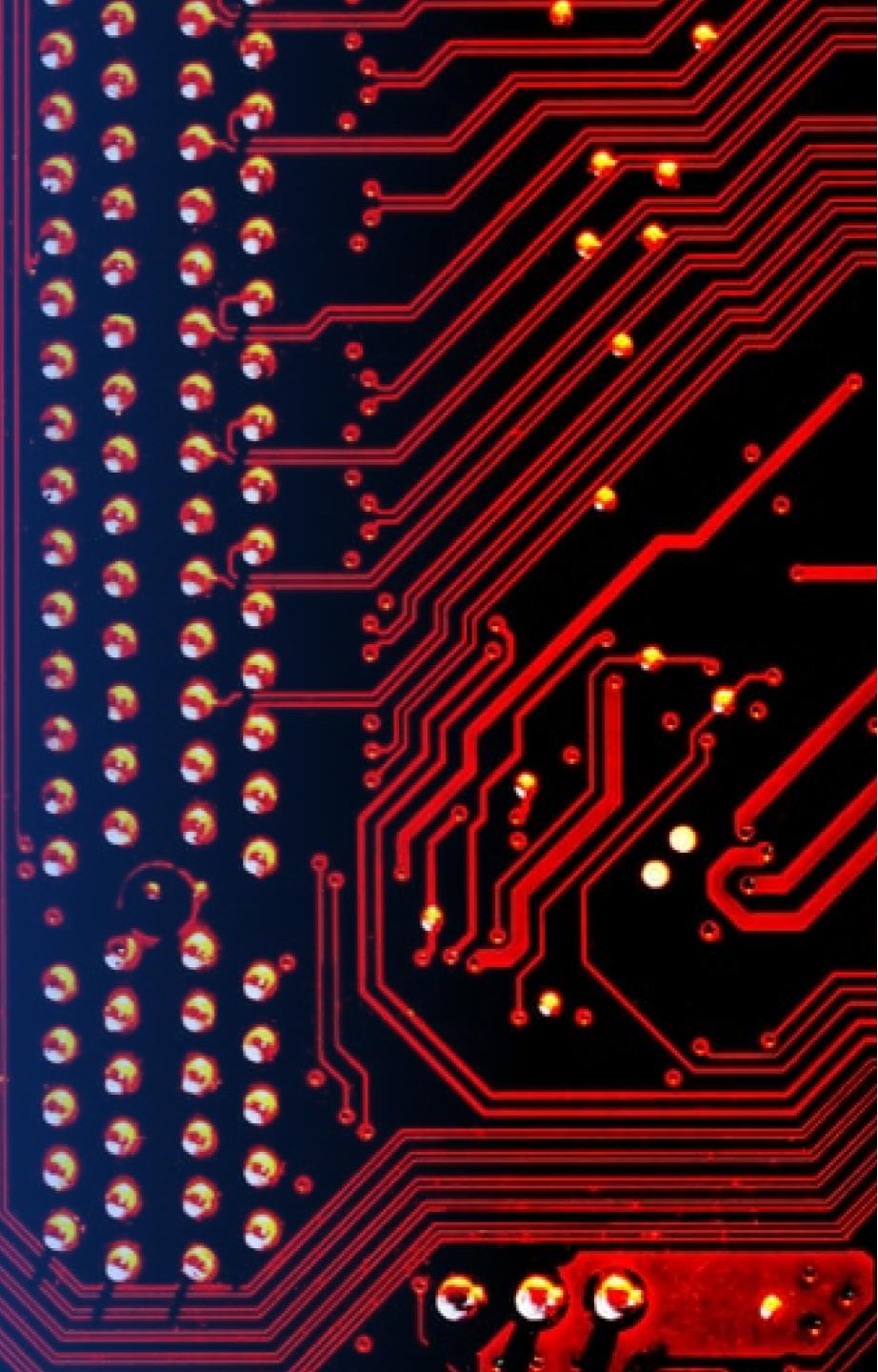
<Folium Map Screenshot 3>

- The launch sites are close to Orlando City and Cape Canveral
- There are good transportation resources.



Section 5

Build a Dashboard with Plotly Dash



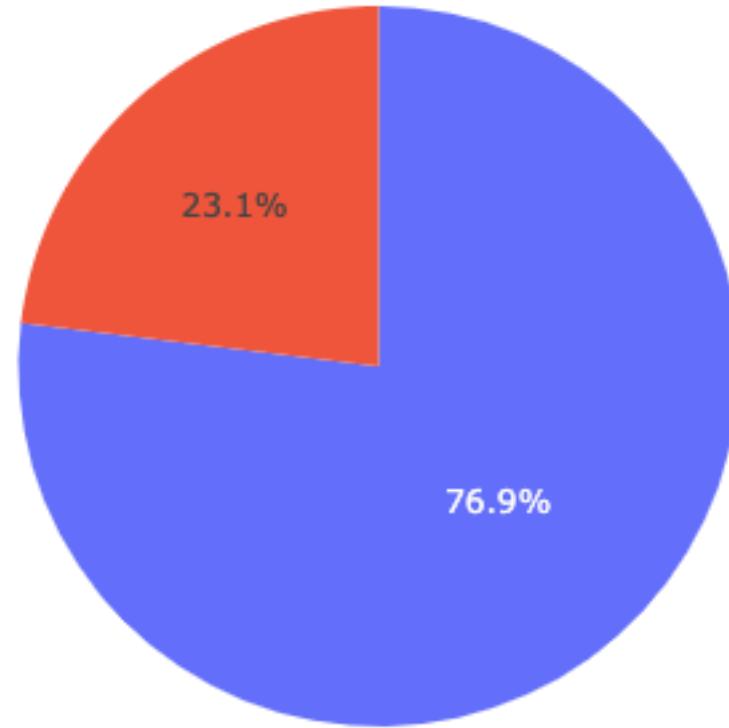
Total Launches for All Sites

- Florida is popular for launches



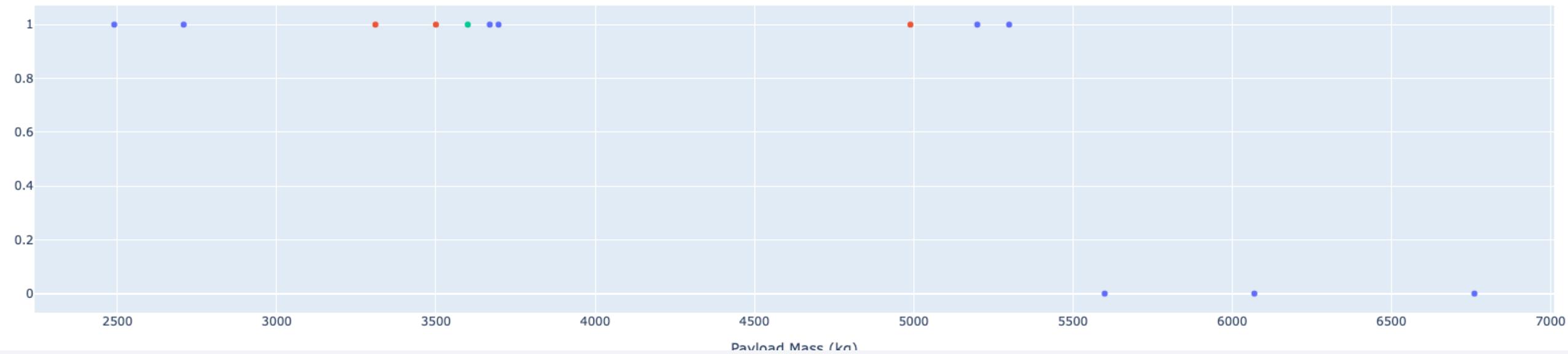
Launch Site with highest success ratio

- KSC LC-39A did the best with 76.9%



<Dashboard Screenshot 3>

- Lower weight payloads did better than higher weight



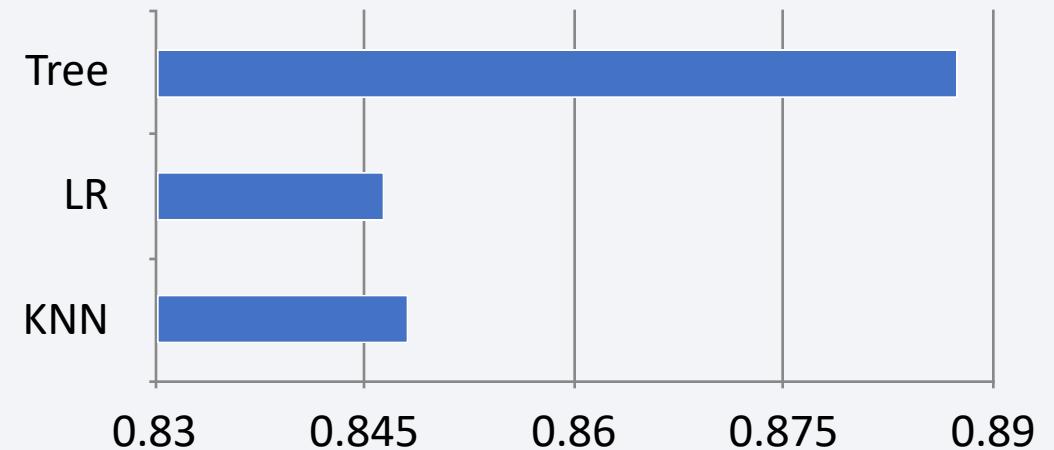
The background of the slide features a dynamic, abstract design. It consists of several thick, curved lines that transition from a bright yellow at the top right to a deep blue at the bottom left. These lines create a sense of motion and depth, resembling a tunnel or a stylized landscape. The overall effect is modern and professional.

Section 6

Predictive Analysis (Classification)

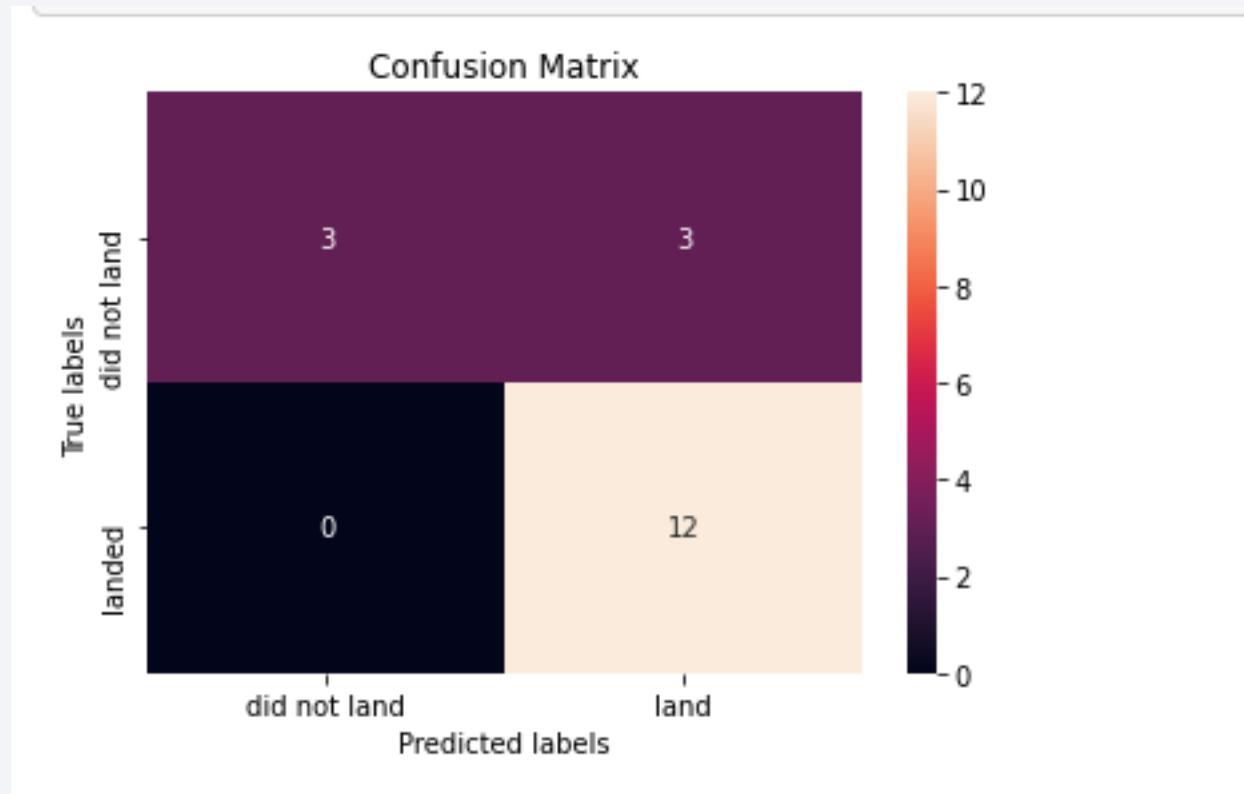
Classification Accuracy

- Our Tree model did the best at nearly 0.89 compared to LR and KNN
-



Confusion Matrix

- The tree model can distinguish between the different classes. False positives were an issue.



Conclusions

- The tree classifier was the best algorithm for our data set
- Lighter payloads did better than heavier payloads for launches
- As time passes by, SpaceX success rate increases
- KSC LC-39A had the best success rate for launches
- GEO, HEO, ES-L1 and SSO had the highest success rates

Appendix

- Pandas, Matplotlib, Algorithms guides, Google Maps, Map Analysis

Thank you!

