IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

<Name>
<Date>

Outline

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

Executive Summary

- **Summary of methodologies**
- Data collection through API
- Data Collection with Web Scrapping
- Data Wrangling
- Exploratory Data Analysis with SQL
- Exploratory Data Analysis with Data Visualization
- Interactive Visual Analytics
- Machine Learning Prediction
- **Summary of all results**
- Exploratory Data Analysis result
- Interactive Anlaytics Result
- Predictive Analysis Result

Introduction

- **Project background and context**

➢ SpaceX has revolutionized the aerospace industry by significantly reducing the cost of space missions through the reusability of its Falcon 9 first-stage boosters. A single Falcon 9 launch costs approximately $62 million, while launches by other providers cost upwards of $165 million. This cost efficiency is primarily due to SpaceX's ability to recover and reuse the first-stage boosters, making it crucial to predict whether a booster will successfully land.

- **Problems You want to find answers**

➢ Can we accurately predict whether a Falcon 9 first-stage booster will land successfully?

➢ Which factors (e.g., payload, launch site, booster type) influence landing success the most?

➢ How does the probability of a successful landing affect overall launch costs and economic feasibility?

➢ Can predictive analytics help new aerospace companies compete with SpaceX by estimating launch success rates?

➢ This project involves collecting, processing, and analyzing launch data to develop a machine learning model that can forecast landing outcomes. The insights gained will help optimize cost efficiency and competitive bidding in the commercial spaceflight industry.

Section 1

Methodology

Methodology

Executive Summary

- Data collection methodology:
  - Data was collected using SpaceX API and web scrapping from respective Wikipedia page
- Perform data wrangling
  - One hot encoding was applied to Categorical Features
- 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

How Was Data Collected ?

- To analyze and predict the Falcon 9 first-stage landing success, data was collected using multiple methods, including SpaceX API requests, JSON processing, and web scraping. The goal was to extract structured launch records and prepare them for further analysis.

Data Collection Process:

- The Data was fetched from the source using SpaceX API in JSON format.
- Using Python, JSON requests were sent, and the response data was converted into a structured Pandas DataFrame using json_normalize()

Data Cleaning & Handling Missing Values:

- The collected data was checked for any missing values or inconsistencies.
- Missing values were filled or removed to ensure data quality before analysis.

Web Scraping from Wikipedia:

- BeautifulSoup was used to scrape Falcon 9 launch records from Wikipedia.
- The extracted records, including launch dates, payload, orbit, and landing outcome, were structured into a Pandas DataFrame for further analysis.

Data Collection – SpaceX API

- We send a get request to SpaceX API to collect data, further we cleaned the data and filled in the missing values
- The link to the notebook is https://github.com/sahilp29/Capstone-Project-SpaceX/blob/88c1852bf03349615fc1bb2828d193aa31e5a643/jupyter-labs-spacex-data-collection-api.ipynb

- Get JSON data response using SpaceX API

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

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

- Use json_normalize method to convert json result to panda dataframe

```
# Use json_normalize meethod to convert the json result into a dataframe
#from pandas import json_normalise
data_json=response.json()
data=pd.json_normalize(data_json)
```

```
# Create a data from launch_dict
data=pd.DataFrame.from_dict(launch_dict)
```

- Perform data cleaning and filing in the missing values

```
# Calculate the mean value of PayloadMass column
payload_mass_mean=data_falcon9['PayloadMass'].mean()
print("Mean Payloadmass:",payload_mass_mean)
# Replace the np.nan values with its mean value
data_falcon9['PayloadMass'].fillna(payload_mass_mean,inplace=True)
print(data_falcon9['PayloadMass'].isnull().sum())
```

Data Collection - Scraping

- We applied web scrapping to webscrap falcon 9 launch records with BeautifulSoup
- The link to the notebook ishttps://github.com/sahilp29/Capstone-Project-SpaceX/blob/88c1852bf03349615fc1bb2828d193aa31e5a643/jupyter-labs-webscraping.ipynb

1. Applying get method to the static url we created, checking the status code and creating a BeautifulSoup Object and verifying the object was created by printing the page title

```
static_url = "https://en.wikipedia.org/w/index.php?title=List_of_Falcon_9_and_Falcon_Heavy_launches&oldid=1027686922"
```

```
# use requests.get() method with the provided static_url
# assign the response to a object
response=requests.get(static_url)
print("Response status code:",response.status_code)
```

```
Response status code: 200
```

```
# Use BeautifulSoup() to create a BeautifulSoup object from a response text content
from bs4 import BeautifulSoup
soup=BeautifulSoup(response.text,'html.parser')
```

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

```
# Use soup.title attribute
print("page title:",soup.title.string)
```

```
page title: List of Falcon 9 and Falcon Heavy launches - Wikipedia
```

2. Extract all column names from html table header

```
column_names = []
for header in first_launch_table.find_all('th'):
    raw_name=header.text
    print(f"Raw header:'{raw_name}'")
    name=raw_name.strip()
    name=re.sub(r'\s+',' ',name)
    #name=' '.join(name.split())
    if name is not None and len(name)>0 and not name.isdigit():
        column_names.append(name)

# Apply find_all() function with `th` element on first_launch_table
# Iterate each th element and apply the provided extract_column_from_header() to get a column name
# Append the Non-empty column name (`if name is not None and len(name) > 0`) into a list called column_names
```

Data Collection - Scraping

```
launch_dict= dict.fromkeys(column_names)

# Remove an irrelvant column
del launch_dict['Date andtime (UTC)']

# 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']=[]
```

3. We create a dictionary launch_dict[] to extract column names

```
extracted_row = 0
#Extract each table
for table_number,table in enumerate(soup.find_all('table',"wikitable plainrowheaders collapsible")):
   # get table row
    for rows in table.find_all("tr"):
        #check to see if first table heading is as number corresponding to launch a number
        if rows.th:
            if rows.th.string:
                flight_number=rows.th.string.strip()
                flag=flight_number.isdigit()
        else:
            flag=False
        #get table element
        row=rows.find_all('td')
```

4. We Parsed the table and converted it into Pandas DataFrame

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

Data Wrangling

- We performed exploratory data analysis and determined training labels
- We calculated no of launches at each site and number of occurrence of each orbit
- We created landing outcome label from outcome column and exported results to csv
- The link to notebook is https://github.com/sahilp29/Capstone-Project-SpaceX/blob/88c1852bf03349615fc1bb2828d193aa31e5a643/labs-jupyter-spacex-Data%20wrangling.ipynb

```
# Apply value_counts() on column LaunchSite
df["LaunchSite"].value_counts()
```

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

```
# Apply value_counts on Orbit column
df["Orbit"].value_counts()
```

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

```
# landing_outcomes = values on Outcome column
landing_outcomes=df["Outcome"].value_counts()
print(landing_outcomes)
```

```
Outcome
True ASDS       41
None None       19
True RTLS       14
False ASDS       6
True Ocean       5
False Ocean      2
None ASDS        2
False RTLS       1
Name: count, dtype: int64
```

EDA with Data Visualization

- We explored the data by visualizing the relationship between flight number and launch site,payload and launch site,success rate of each orbit type,flight number and orbit type ,the launch success vs yearly trend
- The link to the notebook is https://github.com/sahilp29/Capstone-Project-SpaceX/blob/88c1852bf03349615fc1bb2828d193aa31e5a643/edadataviz.ipynb

EDA with SQL

- We loaded the dataset into jupyter notebook using magic sql
- We applied EDA with SQL to get insights from data.We wrote queries to find out the following:
  - The total number of unique launch sites for the mission
  - The total payload mass carried by boasters launched by NASA (CRS)
  - The average payload mass carried by booster version F9 v1.1
  - The total number of succesfull and fail mission outcomes
  - The failed landing outcomes on drone ship,their booster version and launch site for 2015
- The link to notebook is https://github.com/sahilp29/Capstone-Project-SpaceX/blob/88c1852bf03349615fc1bb2828d193aa31e5a643/jupyter-labs-eda-sql-coursera_sqllite.ipynb

Build an Interactive Map with Folium

- We have marked all launch sites and added map objects such as makrers,circles,lines to mark the success or failure of launches for each launch site on folium map
- We have assigned the feature launch outcomes(failure or success) to class 0 and 1 ie.0 for failure and 1 for success
- Using color labeled marker clusters we have successfully identified which launch sites have high success rate
- We have calculated the distance between a launch site to its proximities.We have answered some questions for instance:
  - Are launch sites near railways,highways and coastlines
  - Do launch sites keep certain distance away from cities
- The link to notebook is https://github.com/sahilp29/Capstone-Project-SpaceX/blob/67291c780e12442a287e6c3327b251d74973fdd8/lab_jupyter_launch_site_location.ipynb

Build a Dashboard with Plotly Dash

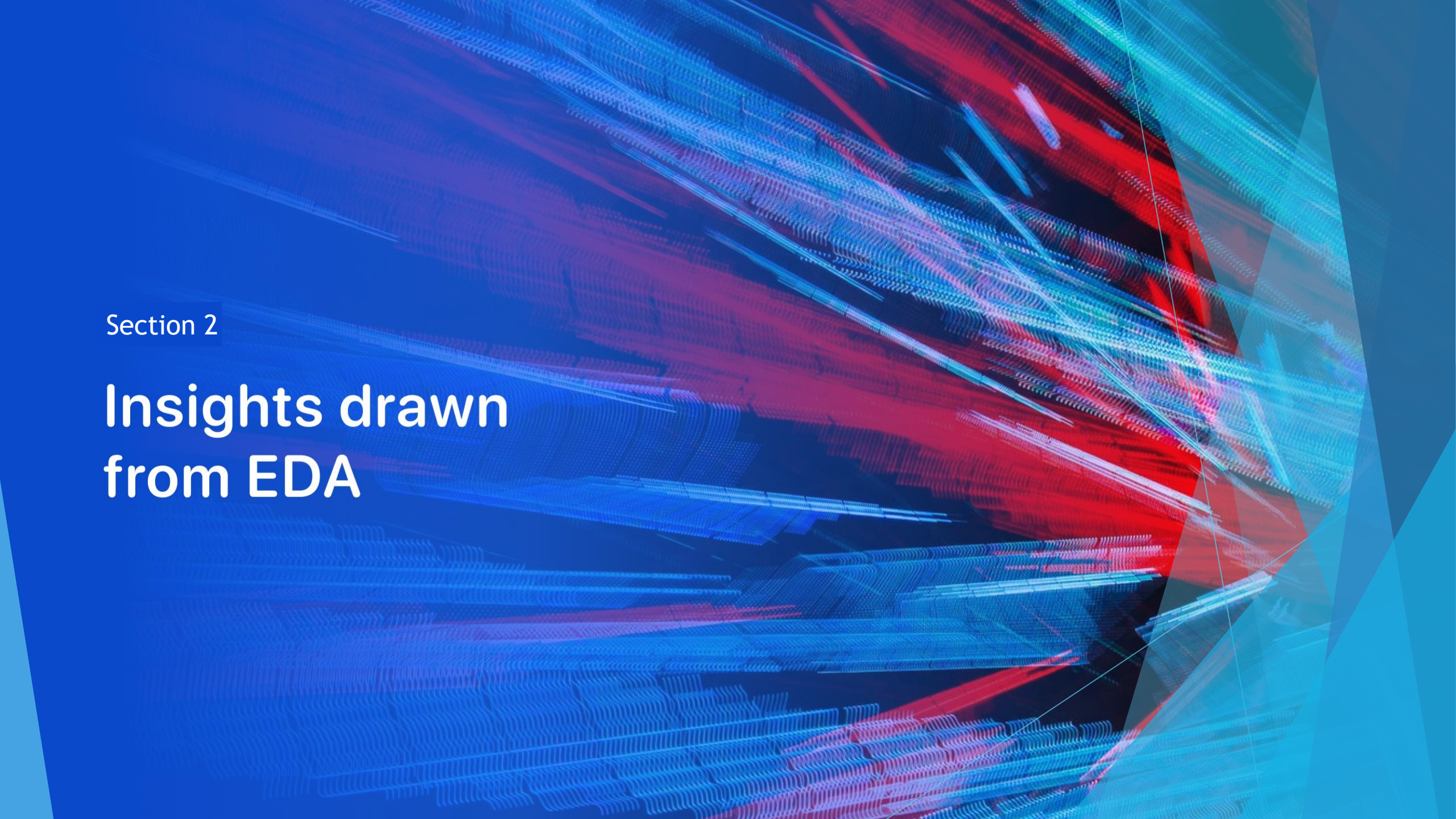
- We have built an interactive dashboard with plotly dash
- We have plotted pie charts showing the total launches by certain sites
- We have plotted scatter graph showing relationship between Outcome and payload mass(kg) for different booster version
- The link to the notebook is https://github.com/sahilp29/Capstone-Project-SpaceX/blob/67291c780e12442a287e6c3327b251d74973fdd8/spacex_dash_app.py

Predictive Analysis (Classification)

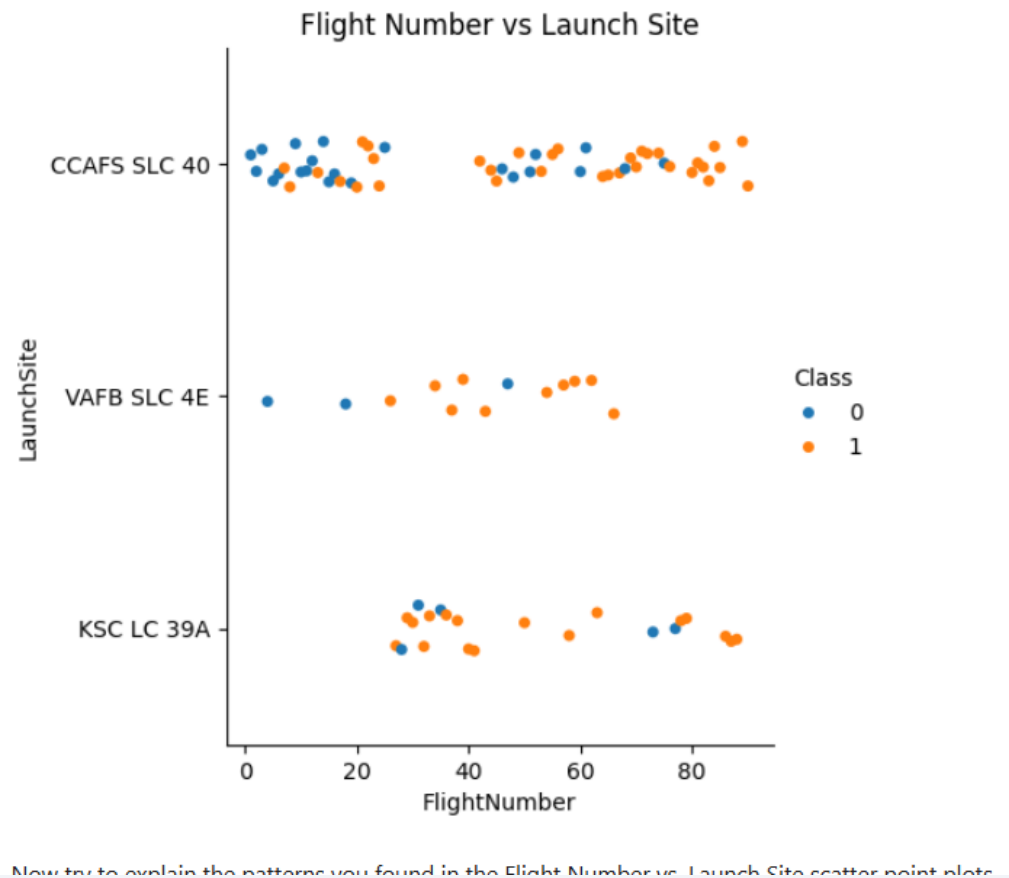
- We have loaded the data using numpy and pandas,transformed the data and split our data into testing and training
- We have built different machine learning models and tune different hyperparameters using GridSearchCV
- We have used accuracy as the metric for our model and have also plotted its confusion matrix ,improved the model using feature engineering and algorithm tuning
- We have found the best classification model
- The link to the notebook is https://github.com/sahilp29/Capstone-Project-SpaceX/blob/67291c780e12442a287e6c3327b251d74973fdd8/SpaceX_Machine%20Learning%20Prediction_Part_5.ipynb

Results

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

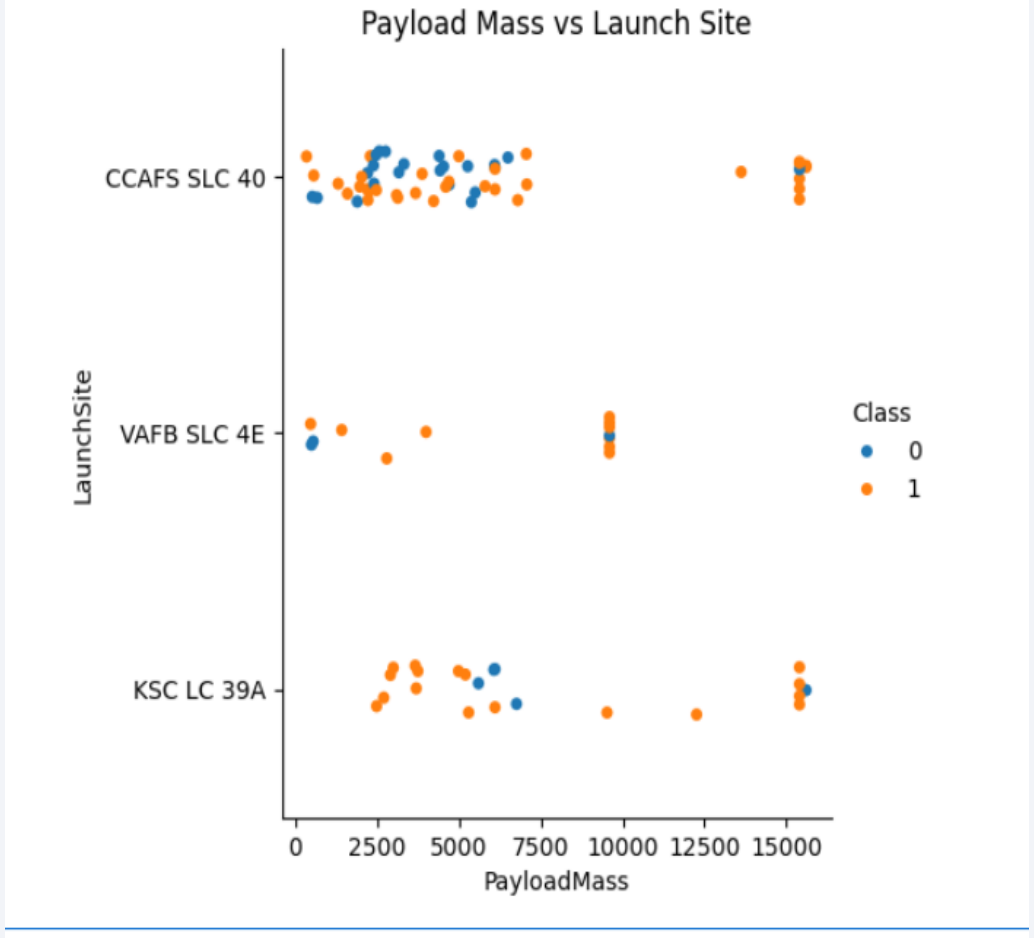

Section 2
Insights drawn from EDA

Flight Number vs. Launch Site



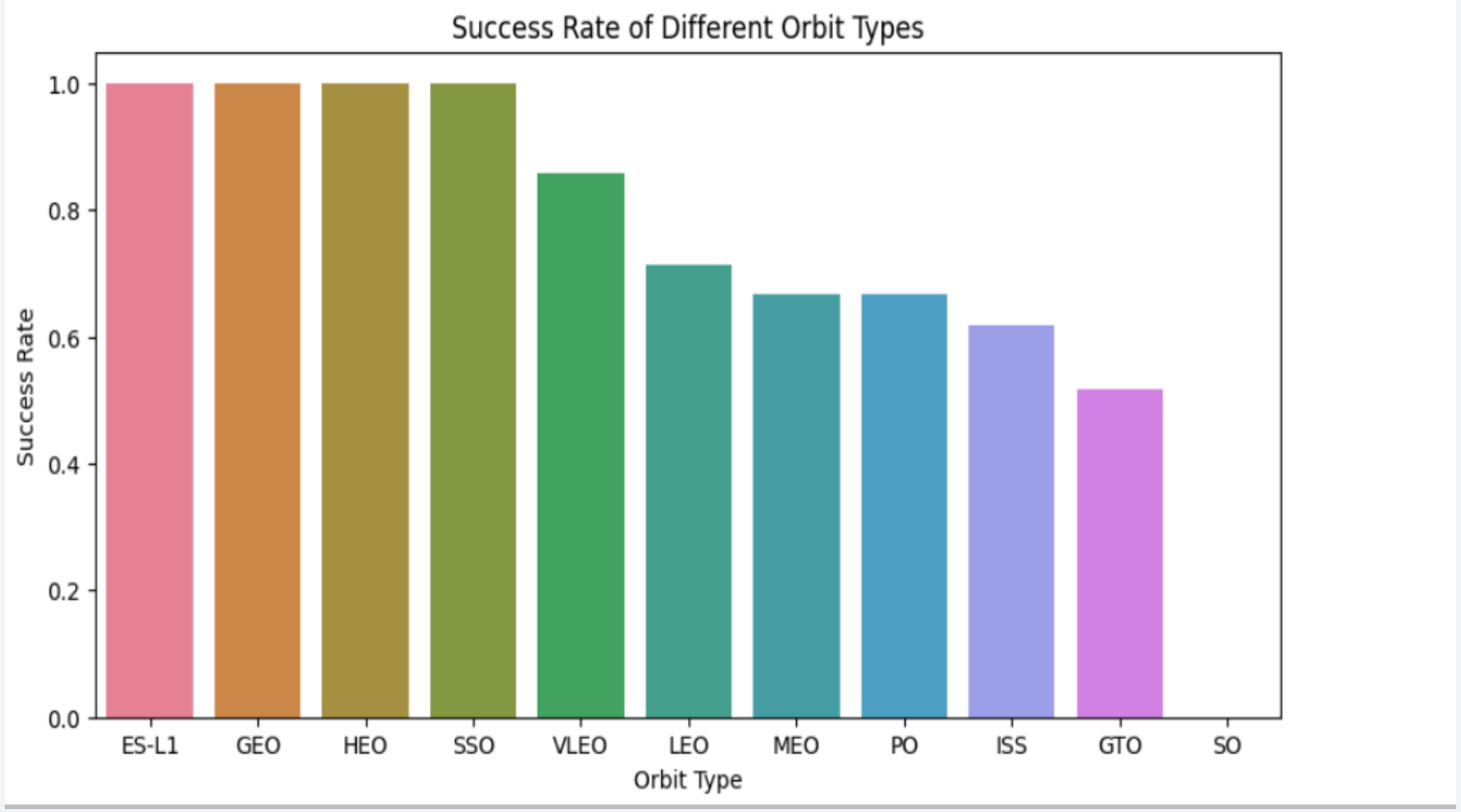
- From the plot we can concur that higher the flight number higher the success rate
- CCAFS SLC 40 has the most launches with improving success
- Initial launches had more failures but later launches saw better outcomes

Payload vs. Launch Site



- From this plot we can conclude that CCAFS SLC 40 has higher success rate with heavier payloads
- VAFB SLC 4E has fewer success compared to other sites
- Successful launches with heavy payloads indicate improved reliability of flacon 9 over time

Success Rate vs. Orbit Type

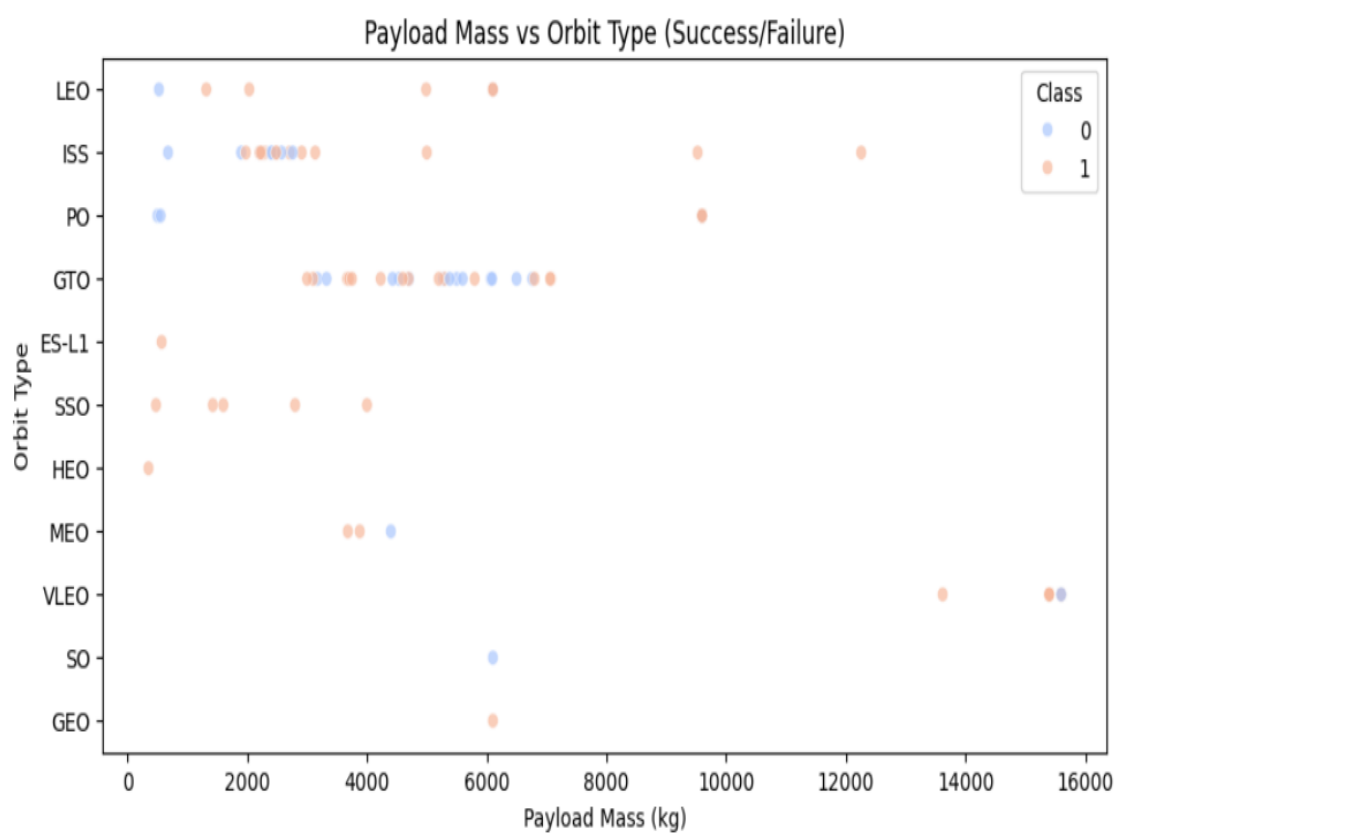


- From this bar chart we can conclude that ES-L1,GEO,HEO and SSO orbits have higher success rate

Flight Number vs. Orbit Type

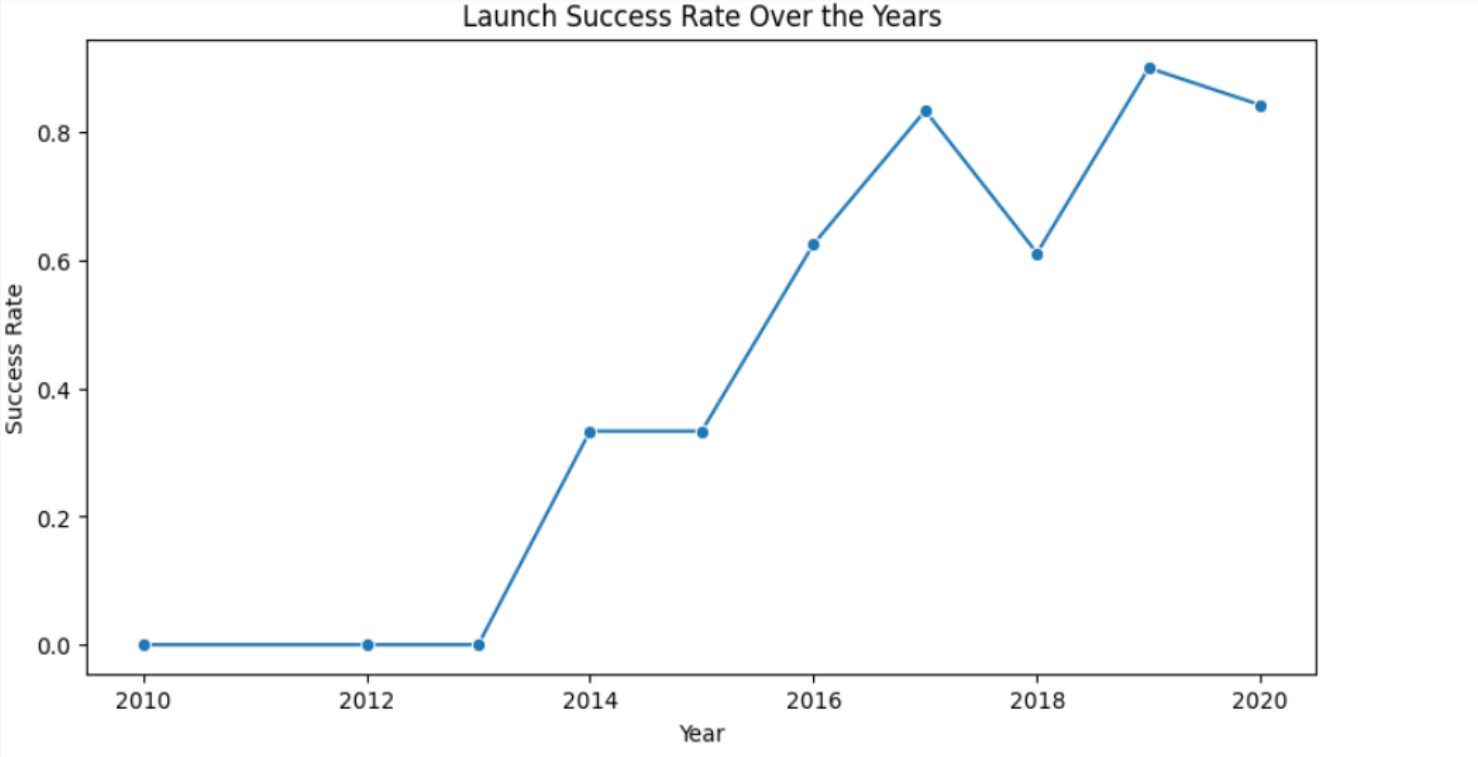
- From this scatter plot we can conclude that LEO shows a trend where success increases with more flight numbers
- GTO shows no clear relationship between flight numbers and success

Payload vs. Orbit Type



- LEO,ISS and polar orbits have higher success rate even with heavier payloads
- GTO has mixed success and failure rate across payloads

Launch Success Yearly Trend



- From the plot we can conclude that success rate has been increasing since 2013 and kept on increasing till 2020 signifying higher success rate over time

All Launch Site Names

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

```
%sql select distinct(Launch_Site) from spacextbl limit 5;
```

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

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

- We have used the keyword **DISTINCT** to show only unique launch sites from the SpaceX data

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

We have used the query above with pattern matching command ‘CCA%’ along with the like operator to display launch sites beginning with ‘CCA’

Total Payload Mass

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

```
%sql select sum(PAYLOAD_MASS__KG_),customer from spacextbl  where customer='NASA (CRS)';
```

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

sum(PAYLOAD_MASS__KG_)	Customer
45596	NASA (CRS)

- We have calculated the total payload mass carried by boosters launched by NASA (CRS) as 45596 kg using the query above

Average Payload Mass by F9 v1.1

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

```
%sql select avg(PAYLOAD_MASS__KG_) from spacextbl where Booster_Version='F9 v1.1'
```

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

avg(PAYLOAD_MASS__KG_)
2928.4

We have calculated the average payload mass as 2928.4 kg using the above query

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) from spacextbl where landing_outcome='Success (ground pad)'
```

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

min(date)
2015-12-22

- We have used the min function to understand when the first successful landing outcome in ground pad was achieved ie 22nd December 2015

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 booster_version from spacextbl where landing_outcome='Success (drone ship)' and payload_mass__kg_ > 4000 and payload_mass__kg_ <6000
```

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

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

- We used the WHERE clause along with AND condition to determine boosters which have landed successfully on drone ship while having payload greater than 4000 kg but less than 6000 kg

Total Number of Successful and Failure Mission Outcomes

List the total number of successful and failure mission outcomes

```
%sql Select mission_outcome,count(mission_outcome) from spacextbl group by TRIM(mission_outcome)
```

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

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

- We have used group by function to list the total number of successful and failure outcomes

Boosters Carried Maximum Payload

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

```
%sql select booster_version from spacextbl where payload_mass__kg_ =(select max(payload_mass__kg_)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

- We determined the booster versions that have carried max payload using a subquery using the WHERE clause and MAX() function

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) AS Month,booster_version,launch_site,landing_outcome from spacextbl where substr(Date,0,5)= '2015' and landing_outcome like 'failure%'
```

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

Month	Booster_Version	Launch_Site	Landing_Outcome
01	F9 v1.1 B1012	CCAFS LC-40	Failure (drone ship)
04	F9 v1.1 B1015	CCAFS LC-40	Failure (drone ship)

- We have used the WHERE clause,LIKE,AND and substr to display the failure while landing on drone ship for the boosters for the given timeframe

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.

```
select count(*) as outcome_count,landing_outcome  from spacextbl where date between '2010-06-04' and '2017-03-20'group by landing_outcome order by outcome_count desc
```

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

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

- We have used the between condition to filter the dates along with where clause and We have grouped by landing oucome and ordered it in descending order

Section 3

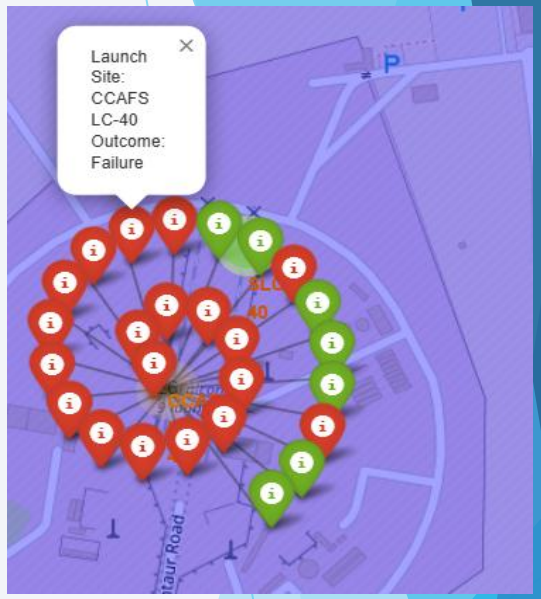
Launch Sites Proximities Analysis

SpaceX launch sites on global map

- We can see that the SpaceX launch sites are in the United States of America coasts Florida and California

Launch Sites with color labels

California launch site



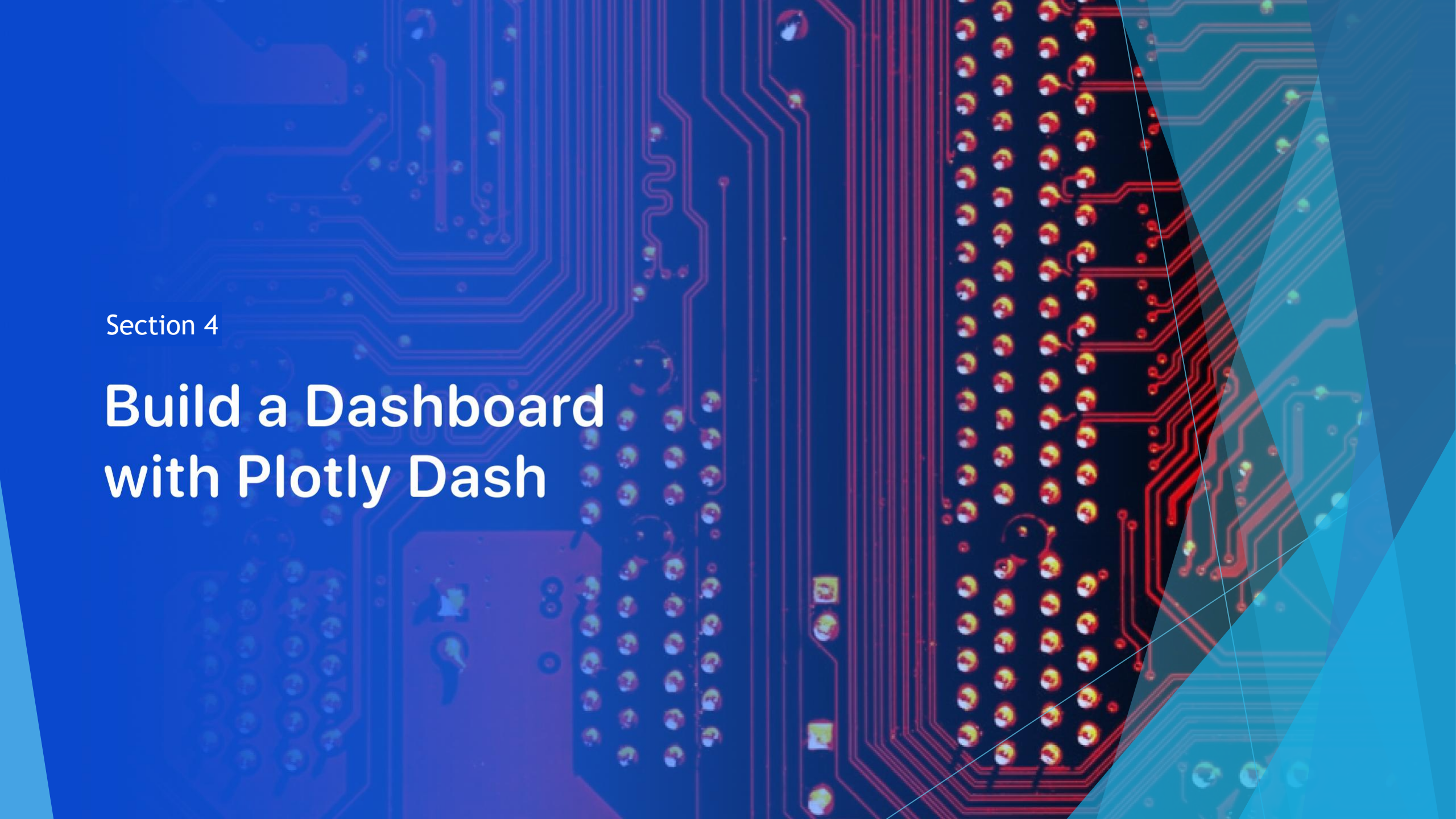
Florida launch sites

- Green marker shows successful launches while Red marker shows failures

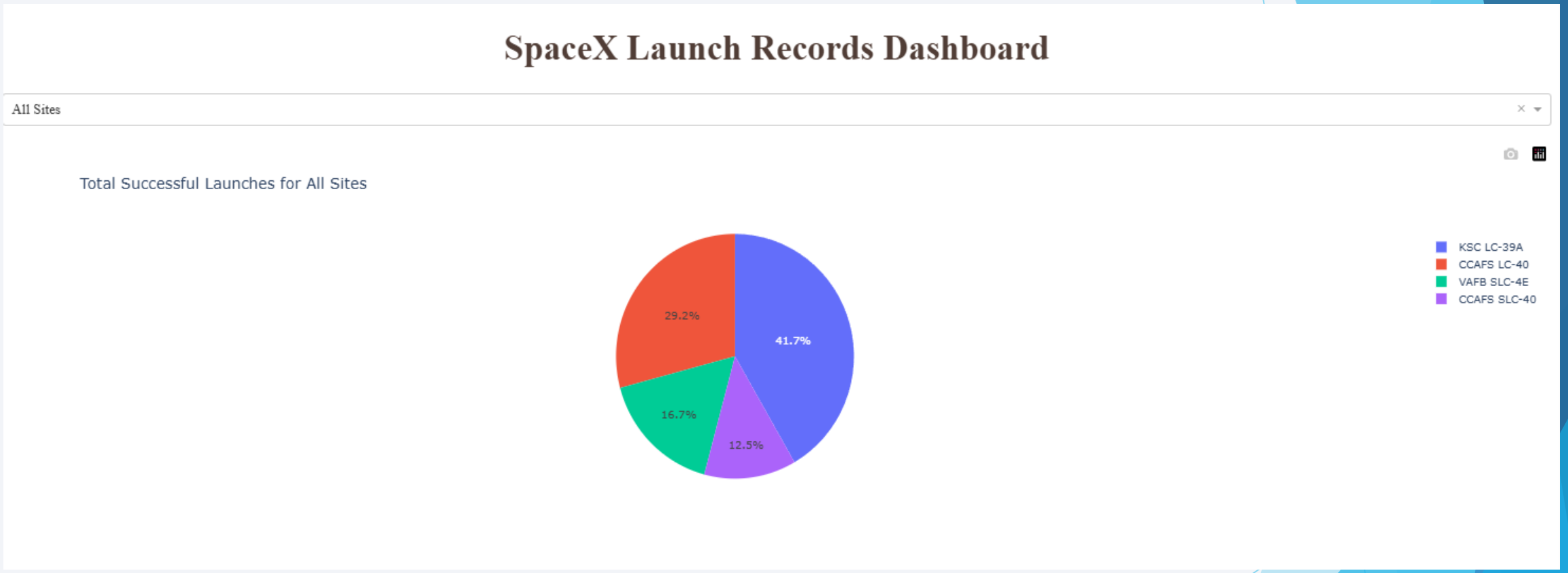
Launch Site Distance to Landmarks

After you plot distance lines to the proximities, you can answer the following questions easily:

- Are launch sites in close proximity to railways? Yes
- Are launch sites in close proximity to highways? Yes
- Are launch sites in close proximity to coastline? Yes
- Do launch sites keep certain distance away from cities? Yes
- Launch sites are located near railways,highways and coastlines for logistical and safety reasons but away from cities to minimise risk in case of launch failure and explosion

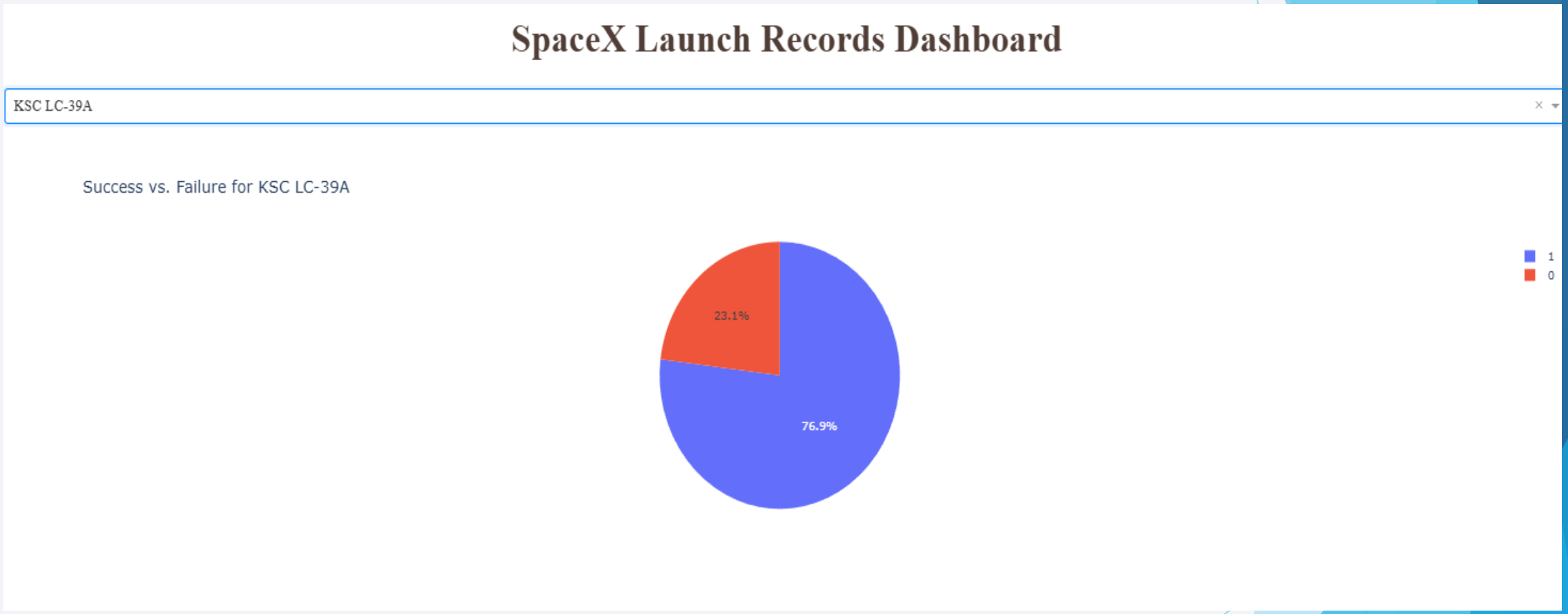

Section 4
Build a Dashboard with Plotly Dash

Pie chart showing the success percentage achieved by each launch site



We can see that KSC-LC-39A had the most successful launches

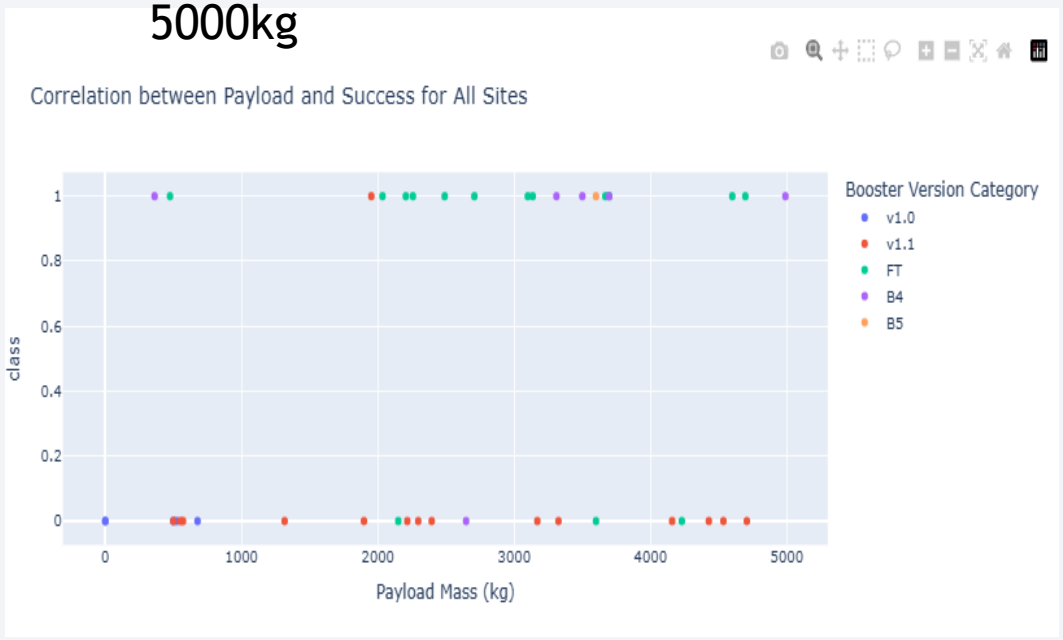
Pie chart showing the launch site with highest launch success ratio



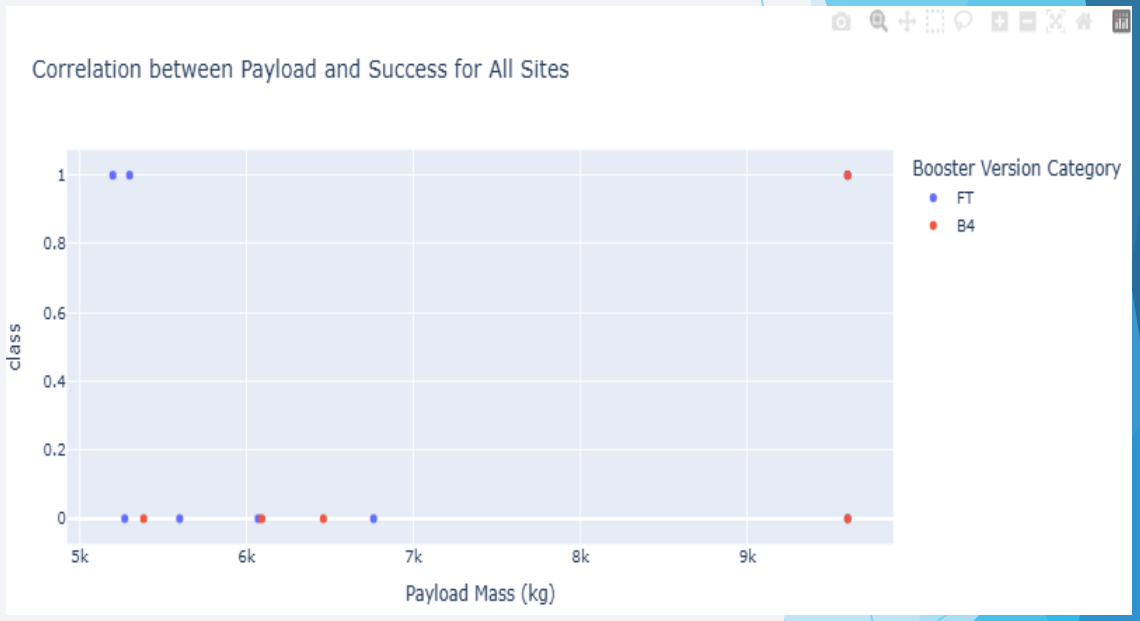
- KSC LC-39A achieved a 76.9% success rate while getting a failure rate of 23.1%

Scatter plot of payload vs Launch Outcome for all sites with different payload slected in range slider

Low weight payload 0 to 5000kg



Heavy weight payload 5000 to 10000kg



- We can see the success rate for low weight payloads is higher than heavy weight payloads

Section 5

Predictive Analysis (Classification)

Classification Accuracy

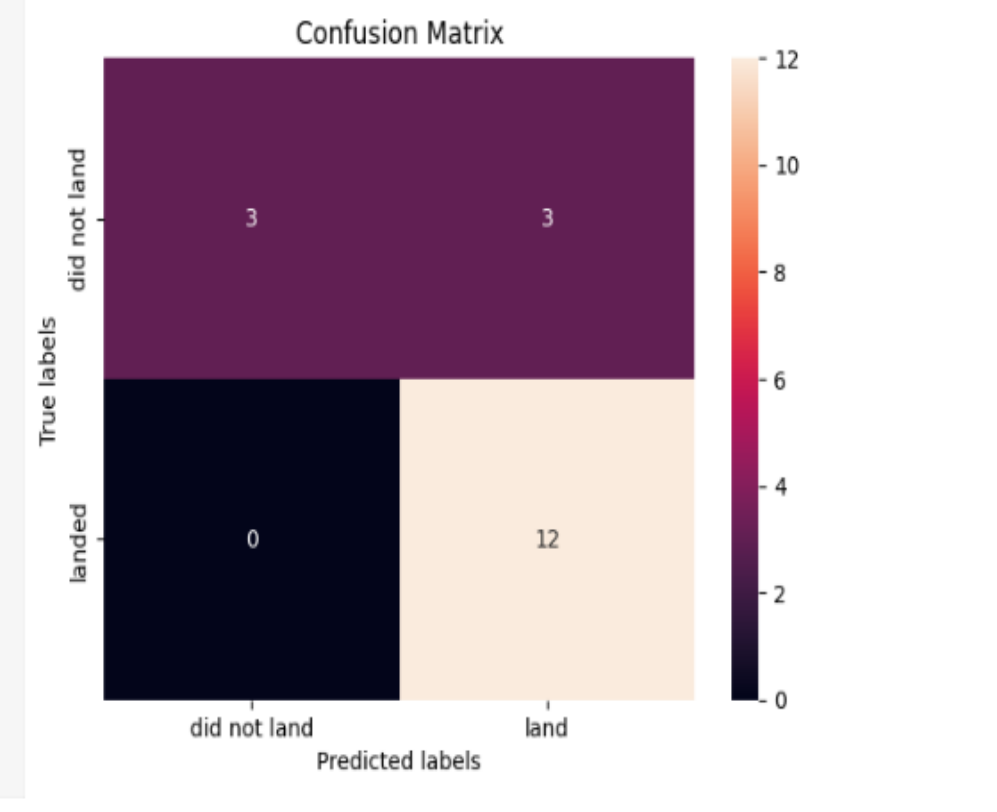
Find the method performs best:

```
# Print best cross-validation scores from GridSearchCV
print("Best Logistic Regression Score:", logreg_cv.best_score_)
print("Best SVM Score:", svm_cv.best_score_)
print("Best Decision Tree Score:", tree_cv.best_score_)
print("Best KNN Score:", knn_cv.best_score_)

# Find the model with the highest cross-validation score
best_model = max([
    ('Logistic Regression', logreg_cv.best_score_),
    ('SVM', svm_cv.best_score_),
    ('Decision Tree', tree_cv.best_score_),
    ('KNN', knn_cv.best_score_)
], key=lambda x: x[1])

print("Best performing model based on validation score:", best_model[0], "with score:", best_model[1])
```

```
Best Logistic Regression Score: 0.8464285714285713
Best SVM Score: 0.8482142857142856
Best Decision Tree Score: 0.8625
Best KNN Score: 0.8482142857142858
Best performing model based on validation score: Decision Tree with score: 0.8625
```

- The Decision tree classifier is the best model with the highest validation score of 88.9%

Confusion Matrix



- The confusion matrix for the decision tree classifier shows that classifier can distinguish between different classes.The major problem is false positives ie unsuccessful landing marked as successive landing by mistake
- This confusion matrix has balanced false positives and true negatives which means that this is not significantly skewed towards one class
- As the model has few misclassifications it is effective

Conclusions

- We can conclude that
  - The larger the flight number at launch site the greater the success rate
  - Launch success rate started to increase steadily from 2013 to 2020
  - Orbits ES-L1,GEO,HEO,SSO,VLEO had the most success rate
  - KSC LC-39A had the most successful launches of any site
  - The Decision tree classifier is the best machine learning algorithm for this task

Thank you!