IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

Alejandro Arango
17/01/2024

Outline

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

Executive Summary

Summary of methodologies

1. **Data Collection:** Request from the SpaceX API and extraction of the Falcon 9 launch records HTML table from Wikipedia.
2. **Data Wrangling:** Exploratory data analysis and training labels selection using pandas and SQL.
3. **Data Exploration:** Data visualization with Seaborn, Matplotlib and Folium.
4. **Machine Learning Prediction:** Model building for outcome prediction.

Summary of all results

- Launch success has improved since 2013.
- KSC LC-39A has the highest success rate among landing sites.
- Orbits ES-L1, GEO, HEO, and SSO have a 100% success rate.
- Most launch sites are near the equator, close to the coast, and at a safe distance from cities and highways.
- All models had the same predictive performance on the test set (accuracy score of 0.833)

Introduction

Project background and context

SpaceX has gained worldwide attention for a series of historic milestones. It is the only private company ever to return a spacecraft from low-earth orbit, which it first accomplished in December 2010. SpaceX advertises Falcon 9 rocket launches on its website with a cost of 62 million dollars wheras other providers cost upward of 165 million dollars each, much of the savings is because Space X can reuse the first stage. Therefore, if we can determine if the first stage will land by training different machine learning models, we can determine the cost of a launch. This information can be used if an alternate company wants to bid against SpaceX for a rocket launch.

Problems you want to find answers

- What factors determine if the rocket will land successfully?
- What are the interaction amongst the features that determine the success rate of a successful landing?
- Which machine learning model will have the best predictive performance?

Section 1

Methodology

Methodology

Executive Summary

- Data collection methodology:
  - Collect the data using request from SpaceX API and web scraping from Wikipedia.
- Perform data wrangling
  - Filter the data, handle missing values and apply one hot encoding 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
  - Standardize the data and create training and test sets. Train logistic regression, support vector machine, decision tree, and k nearest neighbors models with the training set and evaluate their predictions with an accuracy score.

Data Collection

Data sets were collected using a request call for the SpaceX API and web scraping from Wikipedia.

SpaceX API

1. Import libraries and define auxiliary functions.
2. Request rocket launch data from SpaceX API.
3. Decode the response content as a Json using .json()
4. Turn the Json into a Pandas data frame using .json_normalize()
5. Clean, format and filter the data.
6. Apply auxiliary functions to create a new data frame.
7. Filter the data to only include Falcon 9 launches.
8. Deal with missing values.
9. Export data to csv file.

Web Scraping

1. Import libraries and define auxiliary functions.
2. Request the Falcon 9 Launch Wiki page with a HTTP GET.
3. Create a BeautifulSoup object from the HTML response.
4. Extract all column names from the HTML table header.
5. Create a data frame by parsing the HTML tables.
6. Export data to csv file.

Data Collection – SpaceX API

1. Request rocket launch data from SpaceX API

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

[7]: response = requests.get(spacex_url)
```

2. Use .json() and .json_normalize() to create the data frame

```
[10]: # Use json_normalize meethod to convert the json result into a dataframe
      data = pd.json_normalize(response.json())
```

3. Filter the data to only include Falcon 9 launches

```
[24]: data_falcon9 = data[data['BoosterVersion'] != 'Falcon 1']
      data_falcon9.head()
```

4. Deal with missing values

```
[69]: # Calculate the mean value of PayloadMass column
      Payload_mean = data_falcon9['PayloadMass'].mean()

      # Replace the np.nan values with its mean value
      data_falcon9.replace(np.nan,Payload_mean,inplace=True)
```

https://github.com/Jito3023/Applied-Data-Science-Capstone/blob/main/1_Data_Collection_API.ipynb

Data Collection - Scraping

1. Request the Falcon 9 Launch Wiki page with a HTTP GET.

```
[5]: # use requests.get() method with the provided static_url
     response = requests.get(static_url)
     # assign the response to a object
```

2. Create a BeautifulSoup object from the HTML response.

```
[10]: # Use BeautifulSoup() to create a BeautifulSoup object from a response text content
      soup = BeautifulSoup(response.text)
```

3. Extract all column names from the HTML table header.

```
[16]: column_names = []

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

      print(column_names)
```

4. Create a data frame by parsing the HTML tables.

```
[23]: df= pd.DataFrame({ key:pd.Series(value) for key, value in launch_dict.items() })
      df.to_csv('spacex_web_scraped.csv', index=False)
```

https://github.com/Jito3023/Applied-Data-Science-Capstone/blob/main/2_Web_Scraping.ipynb

Data Wrangling

The data wrangling process consisted of exploratory data analysis and the selection of training labels.

1. Calculate the number of launches on each site.

```
[5]: # Apply value_counts() on column LaunchSite
     df['LaunchSite'].value_counts()
```

2. Calculate the number and occurrence of each orbit.

```
[6]: # Apply value_counts on Orbit column
     df['Orbit'].value_counts()
```

3. Calculate the number and occurrence of mission outcome of the orbits.

```
[9]: # landing_outcomes = values on Outcome column
     landing_outcomes = df['Outcome'].value_counts()
     landing_outcomes
```

4. Create a landing outcome label from outcome column.

```
[13]: # landing_class = 0 if bad_outcome
      # landing_class = 1 otherwise
      landing_class = []
      for outcome in df['Outcome']:
          if outcome in bad_outcomes:
              landing_class.append(0)
          else:
              landing_class.append(1)
```

https://github.com/Jito3023/Applied-Data-Science-Capstone/blob/main/3_Data_Wrangling.ipynb

EDA with Data Visualization

Charts:

- Flight Number vs Payload Mass
- Flight Number vs Launch Site
- Launch Site vs Payload Mass
- Success Rate vs Orbit
- Flight Number vs Orbit
- Orbit vs Payload Mass
- Success Rate vs Year

https://github.com/Jito3023/Applied-Data-Science-Capstone/blob/main/5_EDA_Data_Visualization.ipynb

Analysis:

- **Scatter plots** were used for exploring the relationship between variables and determining the ones with most predictive capabilities for the machine learning models.
- **Bar charts** were used for discrete categories to understand their distribution.
- **Line charts** were used for studying the time series that describe the success rate over time.

EDA with SQL

Display:

- Names of unique launch sites.
- 5 records where launch site begins with 'CCA'.
- Total payload mass carried by boosters launched by NASA (CRS).
- Average payload mass carried by booster version F9 v1.1.

https://github.com/Jito3023/Applied-Data-Science-Capstone/blob/main/4_EDA_SQL.ipynb

List:

- Date of first successful landing on ground pad.
- Names of boosters which had success landing on drone ship and have payload mass greater than 4,000 but less than 6,000.
- Total number of successful and failed missions.
- Names of booster versions which have carried the max payload.
- Failed landing outcomes on drone ship, their booster version and launch site for the months in the year 2015.
- Count of landing outcomes between 2010-06-04 and 2017-03-20 (desc).

Build an Interactive Map with Folium

- **Circles:** A highlighted circle area on a specific coordinate to signal the location of a Falcon 9 launch site.
- **Marker:** A popup label where the name of the launch site can be displayed in a specific location.
- **Marker Cluster:** A set of markers for displaying the successful and failed launches.
- **Polyline:** A line for denoting the distance between the launch sites and some locations of interest.

https://github.com/Jito3023/Applied-Data-Science-Capstone/blob/main/6_Interactive_Visual_Analytics_Folium.ipynb

Build a Dashboard with Plotly Dash

Interactions:

- **Dropdown list:**
  Allow the user to select all or a certain launch site for the pie chart.
- **Slider:**
  Allow the user to select a payload mass range for the scatter plot.

Plots/Charts:

- **Pie chart:**
  Allow the user to see the successful and unsuccessful launches as a percentage of the total for all the launch sites or a specific one.
- **Scatter plot:**
  Allow the user to see the correlation between payload, launch success and booster versión.

https://github.com/Jito3023/Applied-Data-Science-Capstone/blob/main/7_Interactive_Visual_Analytics_Plotly.py

Predictive Analysis (Classification)

Model building:

- Load the data using Pandas and Numpy.
- Standardize, fit and transform the data.
- Split the data into test and training sets.

Model improvement:

- Train a logistic regression, support vector machine, decision tree, and k nearest neighbors model.
- Tune the hyperparameters with GridSearchCV.

Model evaluation:

- Evaluate the models based on the accuracy metric for the test set, the best_acore_ function and the confusion matrix.

Best model:

- All the models had the same test-set accuracy, but the decision tree model outperformed the other models in the best_score_ metric.

https://github.com/Jito3023/Applied-Data-Science-Capstone/blob/main/8_Predictive_Analysis.ipynb

Results

Exploratory data analysis results

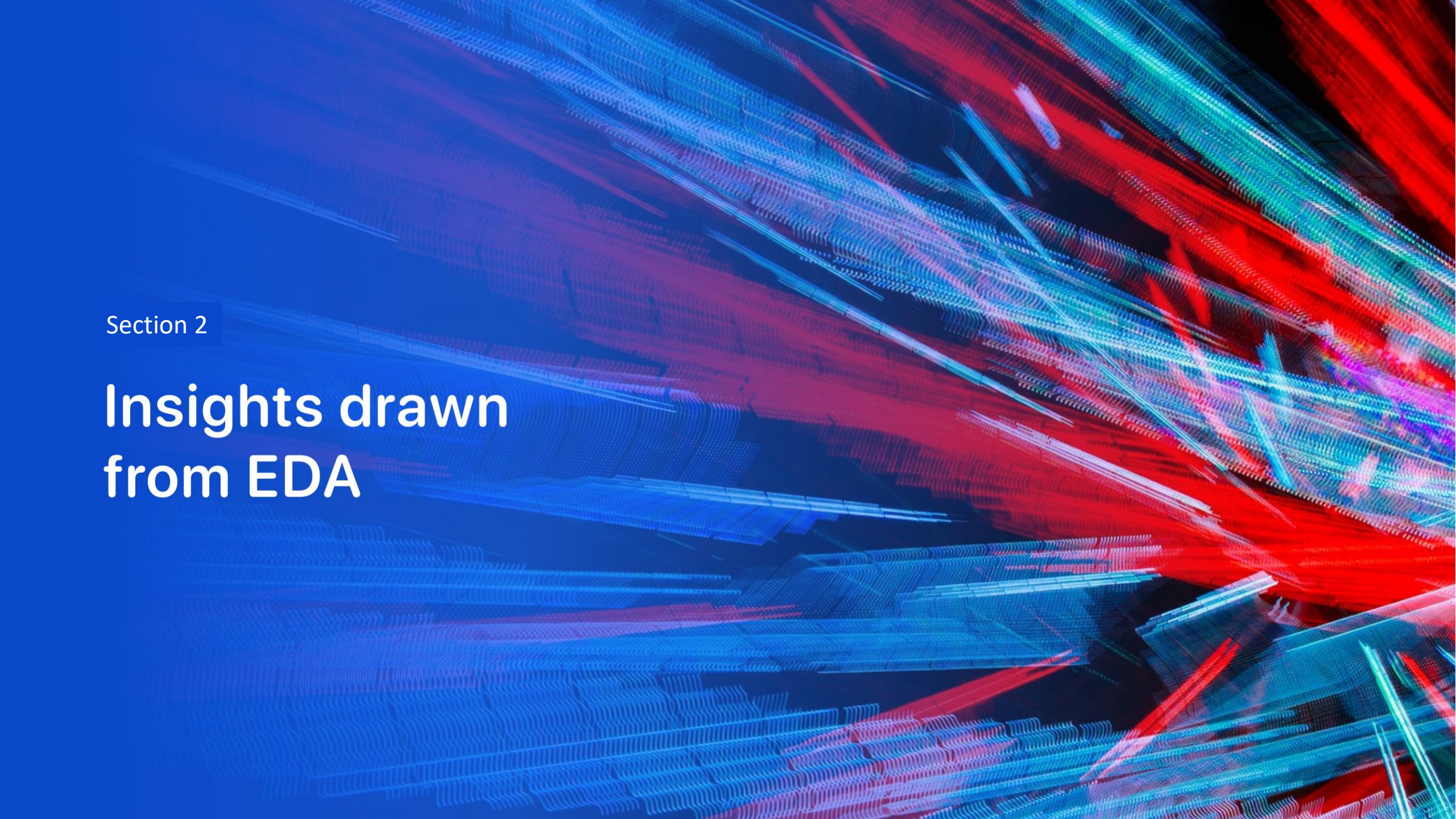
- KSC LC 39A is the launch site with the highest success rate (77.27%).
- The orbits ES-L1, GEO, HEO and SSO have 100% success rate.
- The success rate has increased over the years, going from 0% in 2013 to over 80% in 2020.

Interactive analytics

- All the launch sites are near the Equator to take advantage of the rotational speed of Earth, and near coastlines for security measures.
- Launch sites are at an optimum distance from cities, highways and railways. Preventing accidents from damaging the infrastructure and hurting civilians, but at the same time, facilitating the supply chains and support measures.

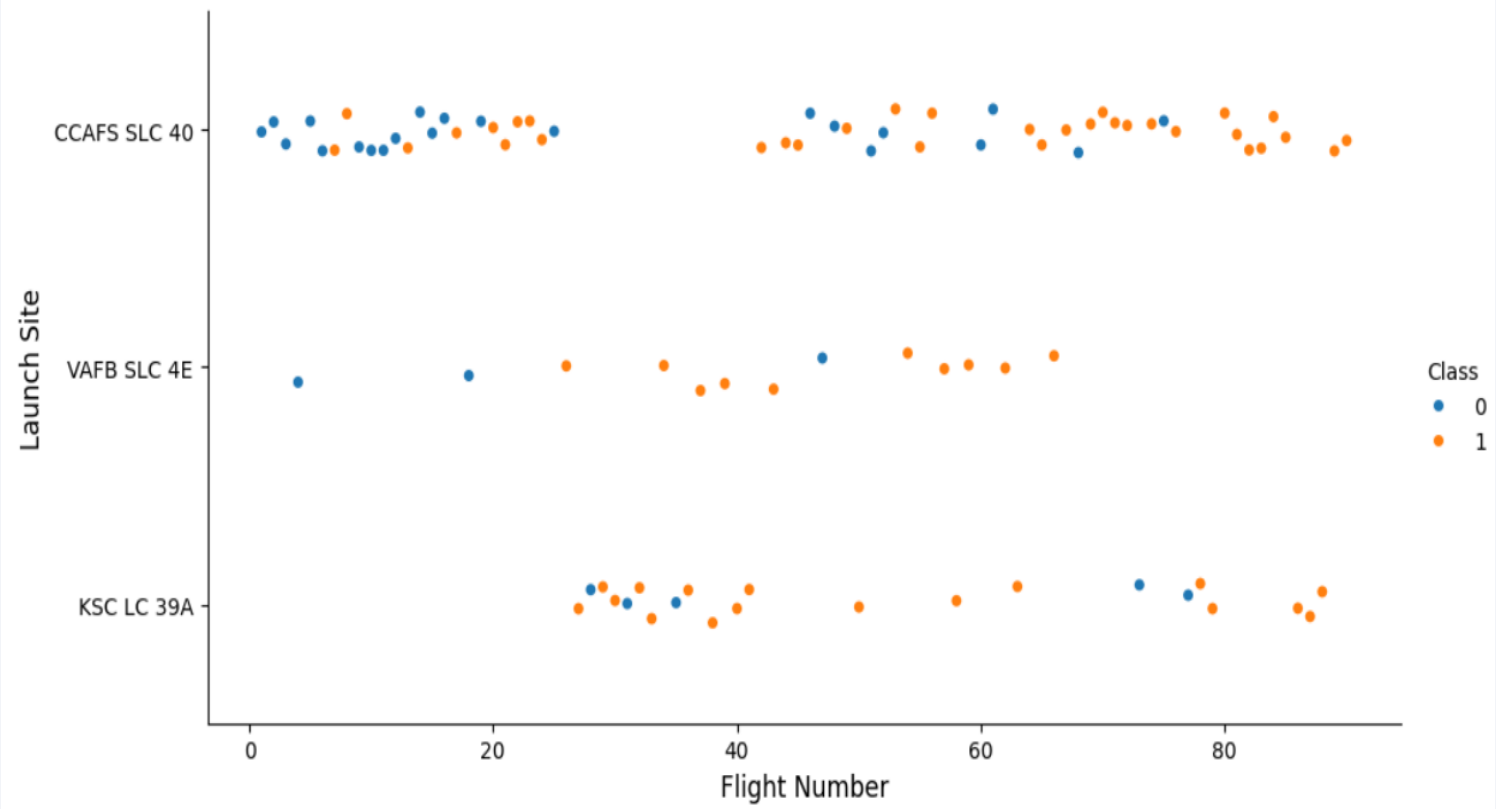
Predictive analysis results

- The Decision Tree model has the best overall accuracy.


Section 2
Insights drawn from EDA

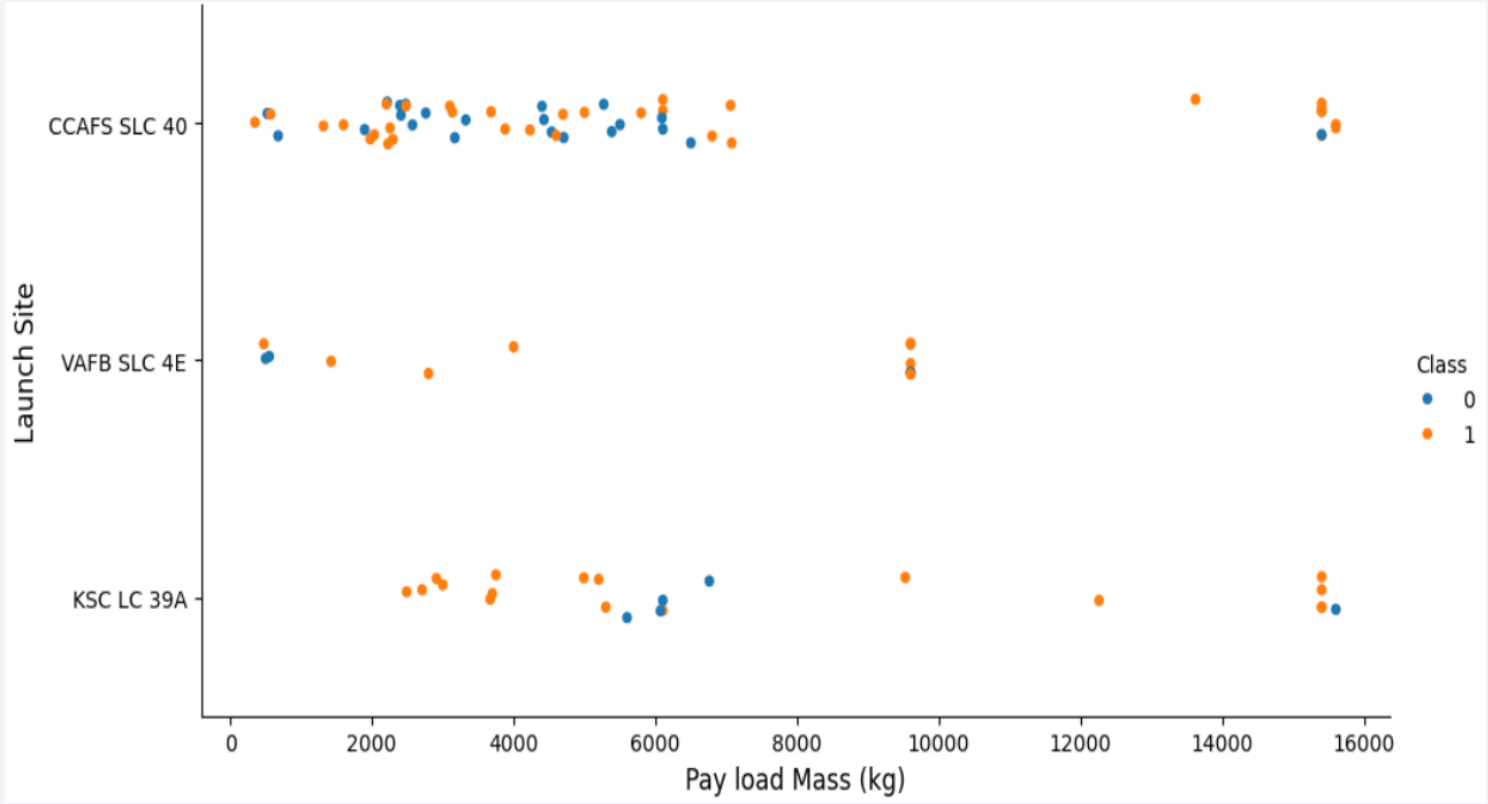
Flight Number vs. Launch Site

- **Earlier flights** had a low success rate.
- **Later flights** had a higher success rate.
- Both **KSC LC-39A** and **VAFB SLC 4E** have a success rate of **77%**, while **CCAFS LC-40** has a success rate of **66%**.
- There has been significantly **more launches** from **CCAFS LC-40** than from the other two launch sites.



Payload vs. Launch Site

- Launches from **CCAFS LC-40** have payloads mostly under **8000 kg**.
- Launches from **VAFB SLC 4E** all have payloads under **10000 kg**.
- Launches from **KSC LC-39A** with payloads under **~5500 kg** have a **100%** success rate.
- Launches with a payload mass of over **8000 kg** have a higher success rate than those under that threshold.
- Most of the launches had a payload under **8000 kg**.

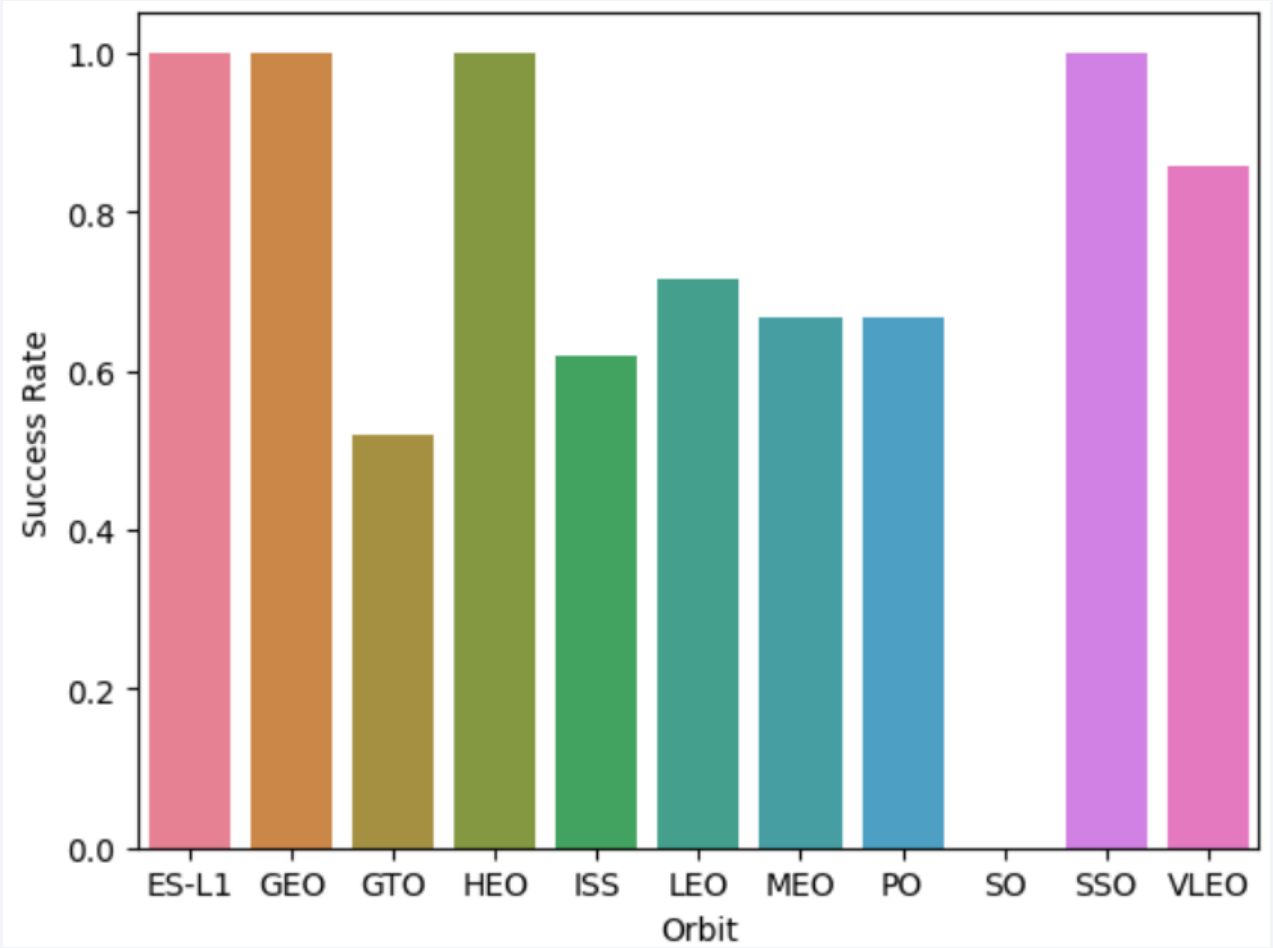


Success Rate vs. Orbit Type

100% success rate: ES-L1, GEO, HEO, SSO.

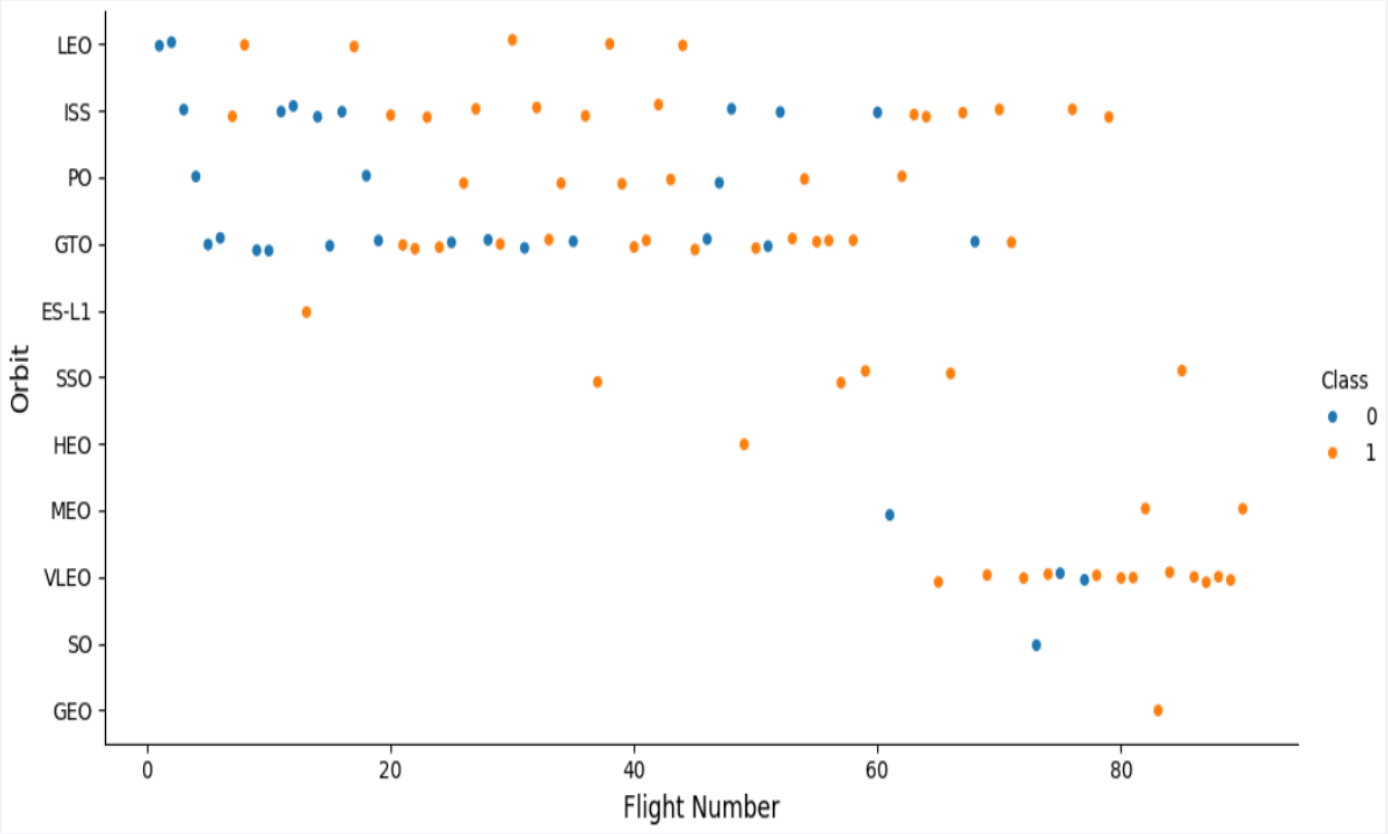
50%-90% success rate: GTO, ISS, LEO, MEO, PO, VLEO.

0% success rate: SO.



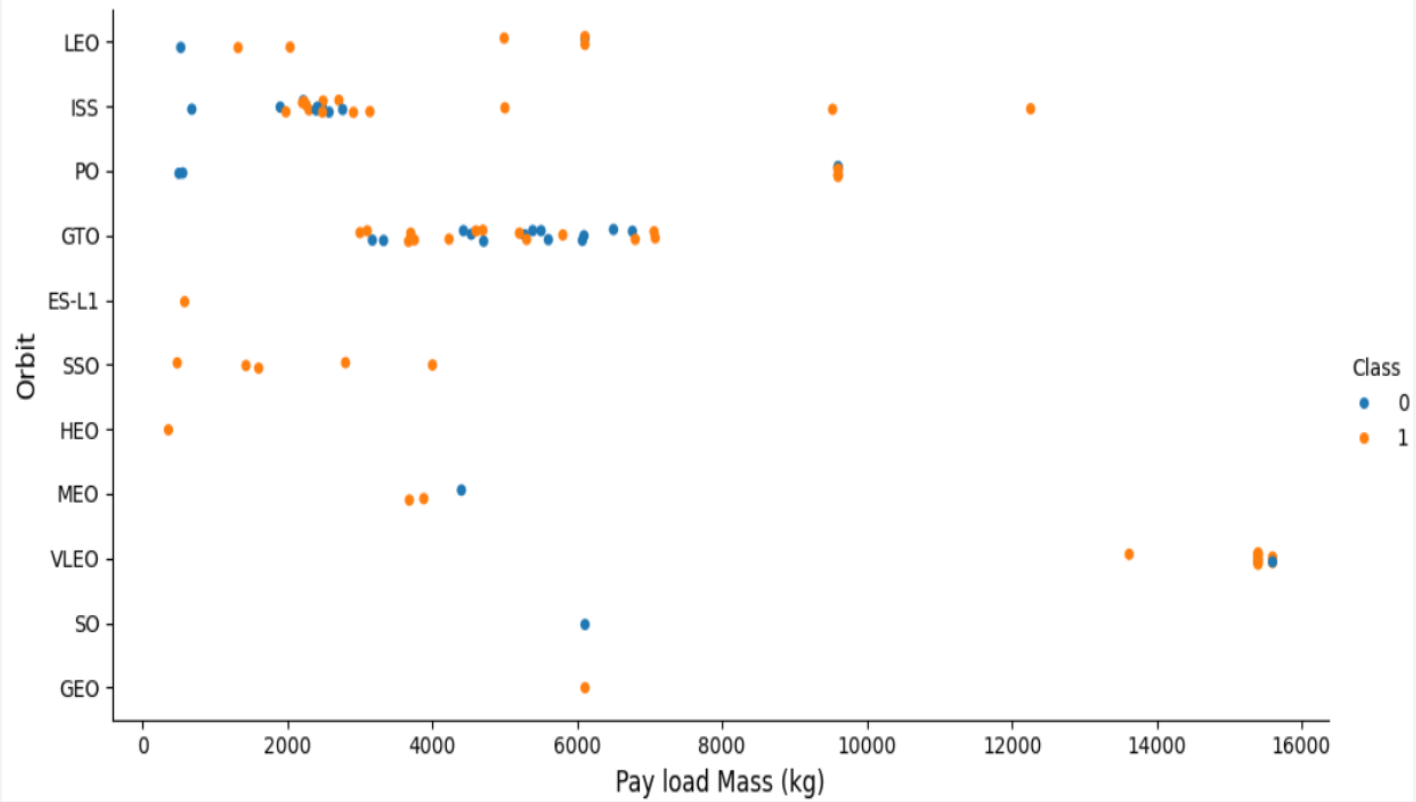
Flight Number vs. Orbit Type

- In all launch sites, the **success rate increases** with the **flight number**.
- This tendency is especially evident in the orbit **LEO**, and not that evident in the **GTO** orbit.



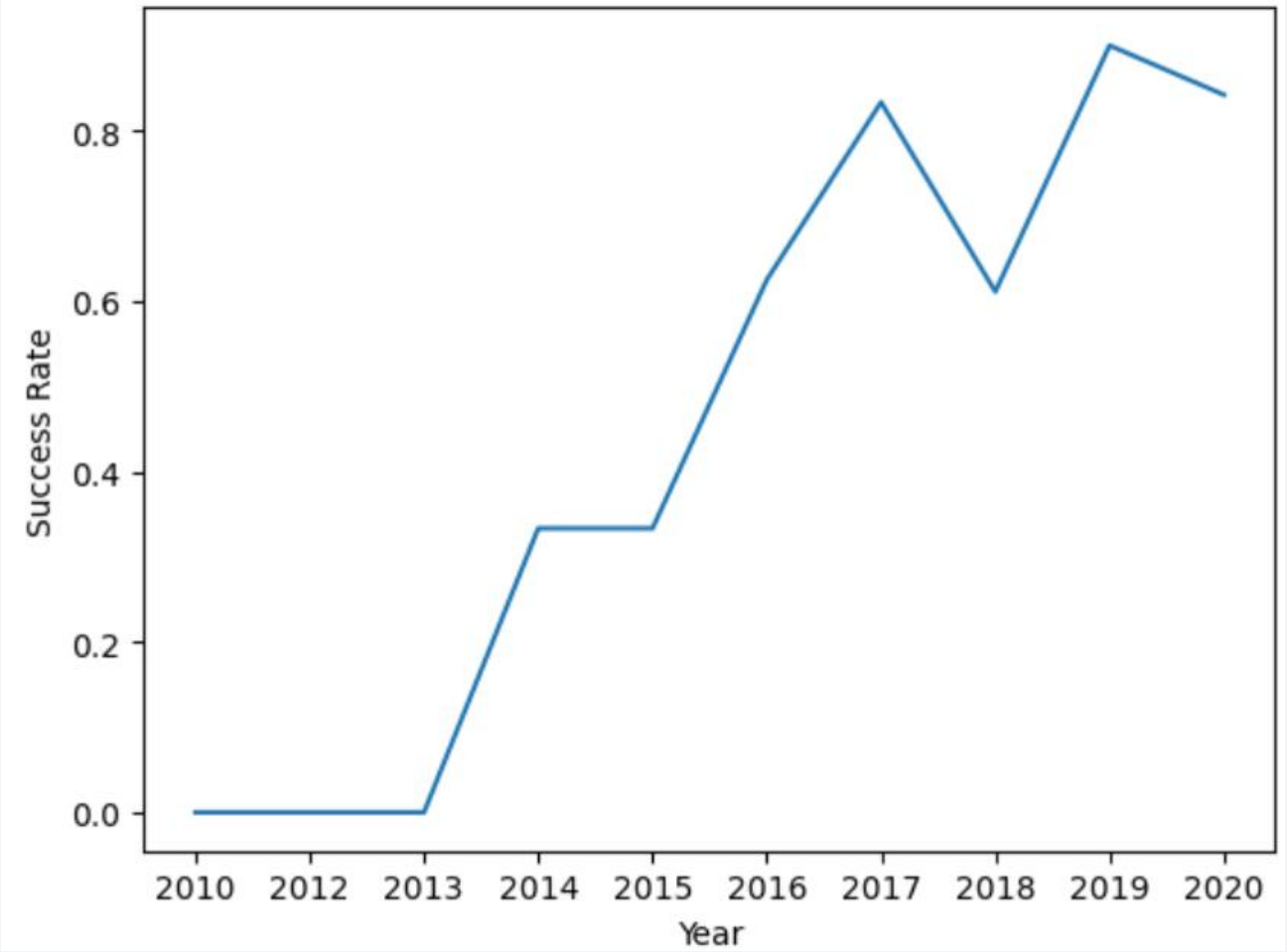
Payload vs. Orbit Type

- The orbits LEO, GTO, ES-L1, SSO, HEO, MEO, SO and GEO only include payload masses under **8000 kg.**
- The orbits LEO, ISS, PO and VLEO have better success rates for higher payload masses.



Launch Success Yearly Trend

- The success rate has **steadily increased over the years**, with a sudden decline in 2018.
- The maximum success rate of over **90%** was reached in 2019.



All Launch Site Names

The **SELECT DISTINCT** statement is used to show the unique launch sites names in the Space X data.

```
[13]: %sql SELECT DISTINCT("Launch_Site")\
      from SPACEXTBL;
```

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

Done.
```

```
[13]: ,,,,,,,,,,,,,,,,,,,,
```

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

Launch Site Names Begin with 'CCA'

- The **LIKE** operator is used in the **WHERE** clause for a specified pattern in the column. In this case "CCA". The **%** represents multiple characters, that way, it filters all words that start with "CCA" and have any other characters after it.
- The **LIMIT** clause accompanied by a 5, restricts the results to the first 5 records.

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

 * sqlite:///my_data1.db

Done.
```

[14]:

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

Total Payload Mass

- The **SUM** function returns the total sum of a numeric column.
- The **WHERE** clause filters the data set so that the function is only applied to the payload mass carried by boosters launched by **NASA (CRS)**.

```
[18]: %sql SELECT SUM(PAYLOAD_MASS__KG_)\
      FROM SPACEXTBL\
      WHERE CUSTOMER == 'NASA (CRS)';
```

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

Done.
```

```
[18]: ,,,,,,,,,,
```

SUM(PAYLOAD_MASS__KG_)
45596

Average Payload Mass by F9 v1.1

- The **AVG** function returns the average or mean of a numeric column.
- The **WHERE** clause filters the data set so that the function is only applied to the payload mass carried by booster version **F9 v1.1**.

```
[20]: %sql SELECT AVG(PAYLOAD_MASS__KG_)\
      FROM SPACEXTBL\
      WHERE BOOSTER_VERSION = 'F9 v1.1';
       * sqlite:///my_data1.db
      Done.
[20]: ,,,,,,,,,,
```

AVG(PAYLOAD_MASS__KG_)
2928.4

First Successful Ground Landing Date

- The **MIN** function returns the minimum number of a numeric column. In this case, the date column.
- The **WHERE** clause filters the data set so that the function is only applied to the dates where the landing outcome was **Success (ground pad).**

```
[21]: %sql SELECT MIN(DATE)\
      FROM SPACEXTBL\
      WHERE LANDING_OUTCOME  = 'Success (ground pad)';
```

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

```
[21]: ,,,,,,,,,,
```

MIN(DATE)
2015-12-22

Successful Drone Ship Landing with Payload between 4000 and 6000

- There are two conditions for the **WHERE** clause:
    1. First, it filters the results where the landing outcome was **Success (drone ship).**
    2. Second, it filters the payload mass using the **BETWEEN** operator to only return values that have a lower limit of **4000 kg** and an upper limit of **6000 kg**.

```
[23]: %sql SELECT BOOSTER_VERSION\
      FROM SPACEXTBL\
      WHERE LANDING_OUTCOME = 'Success (drone ship)'\
      AND PAYLOAD_MASS__KG_ BETWEEN 4000 AND 6000;
```

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

```
[23]: ''''''''''''''''''
```

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

Total Number of Successful and Failure Mission Outcomes

- The **COUNT** function returns the number of rows that matches an specific criterion. In this case, the criterion is the **successful** and **failed** mission outcomes.
- To get the count of each mission outcome, the **GROUP BY** statement groups the rows that have the same values.

```
[27]: %sql SELECT MISSION_OUTCOME, COUNT(*) as total\
      FROM SPACEXTBL\
      GROUP BY MISSION_OUTCOME;
```

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

```
[27]: ''''''''''''''''''''''''
```

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

Boosters Carried Maximum Payload

- In this case, a subquery is used to initially filter the results to get the maximum payload mass from the data set. The **SELECT** statement and the **MAX** function are used for this purpose.
- After the subquery, the **WHERE** clause is used for further filtering the previous results to get the **booster version** which carried the **maximum payload.**

```
[28]: %sql SELECT BOOSTER_VERSION\
      FROM SPACEXTBL\
      WHERE PAYLOAD_MASS__KG_ IN (SELECT MAX(PAYLOAD_MASS__KG_) FROM SPACEXTBL);
       * sqlite:///my_data1.db
      Done.
[28]: ,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,
```

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

- First, the information of interest is selected with the **SELECT** statement.
- Then, the data set is filtered with both the date and the landing outcome to obtain the **failed landing outcomes in drone ship** for the year **2015.**

[31]:
```
%sql SELECT substr(Date, 6,2) AS MONTH, DATE, LANDING_OUTCOME, BOOSTER_VERSION, LAUNCH_SITE\
FROM SPACEXTBL\
WHERE substr(Date,0,5) = '2015' AND\
LANDING_OUTCOME = 'Failure (drone ship)';
```

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

Done.
```

[31]:

MONTH	Date	Landing_Outcome	Booster_Version	Launch_Site
01	2015-01-10	Failure (drone ship)	F9 v1.1 B1012	CCAFS LC-40
04	2015-04-14	Failure (drone ship)	F9 v1.1 B1015	CCAFS LC-40

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

- First, the **SELECT** statement returns the wanted information from the data set, including the count of each landing outcome by means of the **COUNT** function.
- Second, a **WHERE** clause and the **BETWEEN** operator are used for filtering the data to match the lower limit of **2010-06-04** and the upper limit of **2017-03-20.**
- Third, the data is grouped by landing outcome with the **GROUP BY** statement.
- Lastly, the **ORDER BY** keyword accompanied by the **DESC** command orders the count of each outcome in **descending** order.

```
[35]: %sql SELECT LANDING_OUTCOME, COUNT(*) as COUNT\
FROM SPACEXTBL\
WHERE DATE BETWEEN '2010-06-04' AND '2017-03-20'\
GROUP BY LANDING_OUTCOME\
ORDER BY COUNT DESC;

 * sqlite:///my_data1.db

Done.
```

[35]:

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

Section 3

Launch Sites Proximities Analysis

Launch Sites

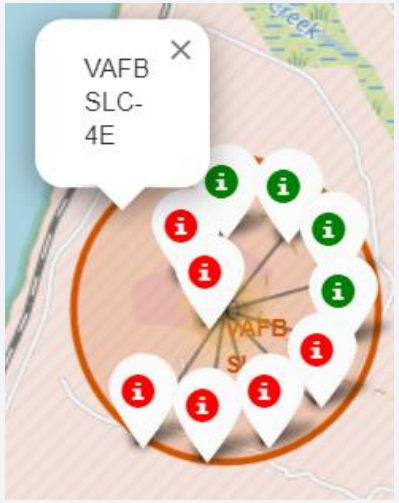
All 4 launch sites are **near the Equator.** The reason for it is to take advantage of the rotational speed of Earth, which is at a maximum of **1667 km/h** in the Equator.

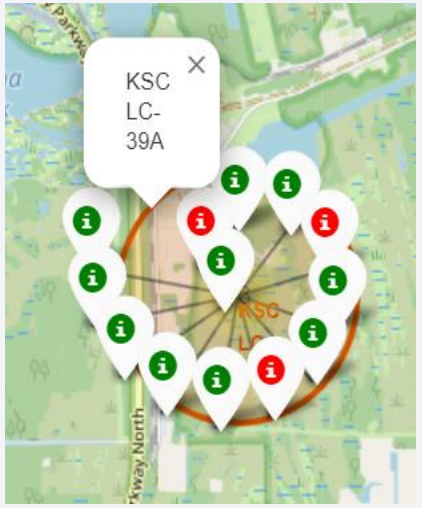
This way, the spaceships get an **additional natural boost**, reducing costs in fuel and boosters.

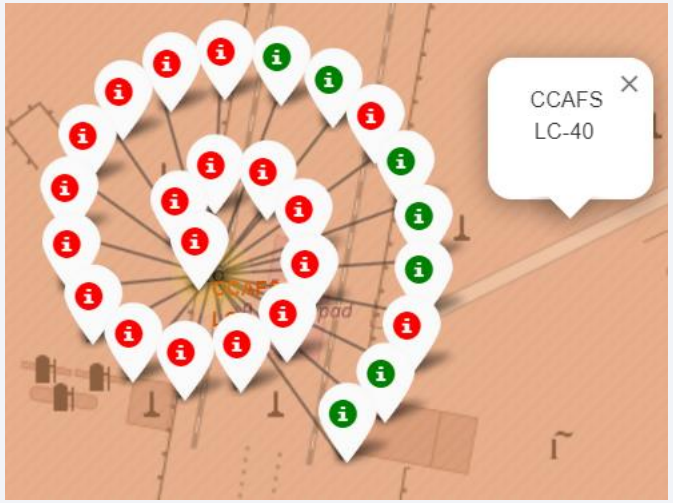
Launch Outcomes

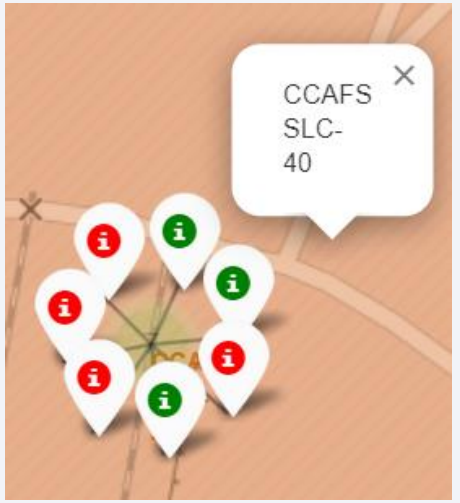
Green:
Successful
Launch

Red:
Failed
Launch





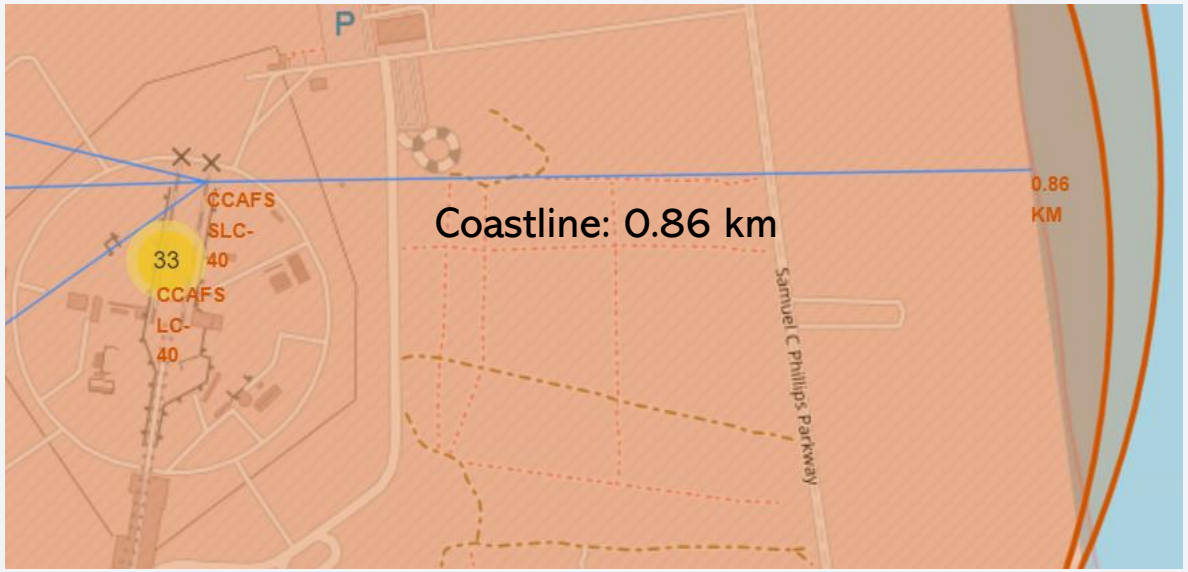


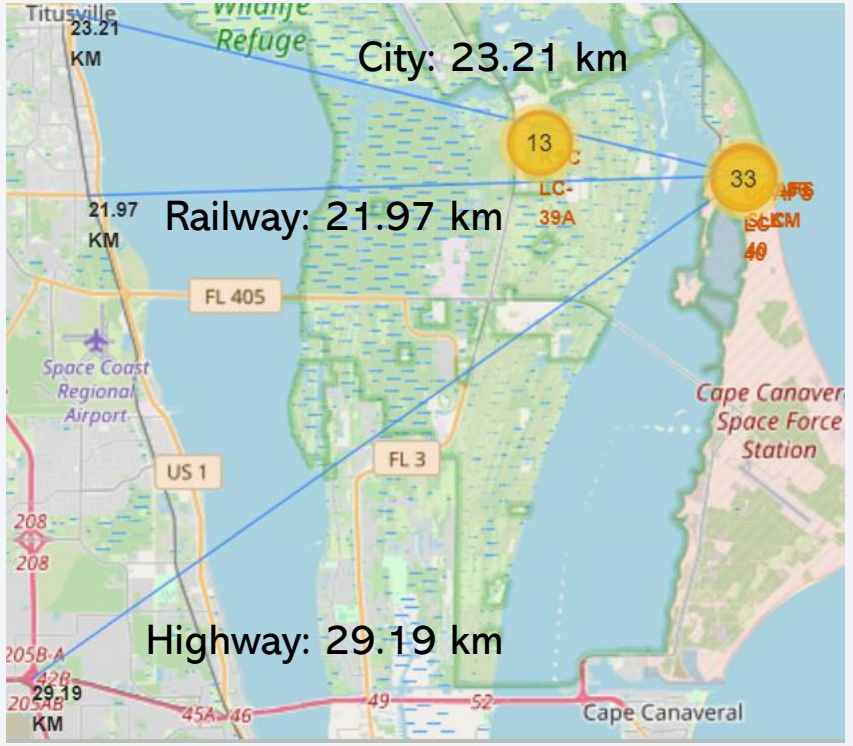


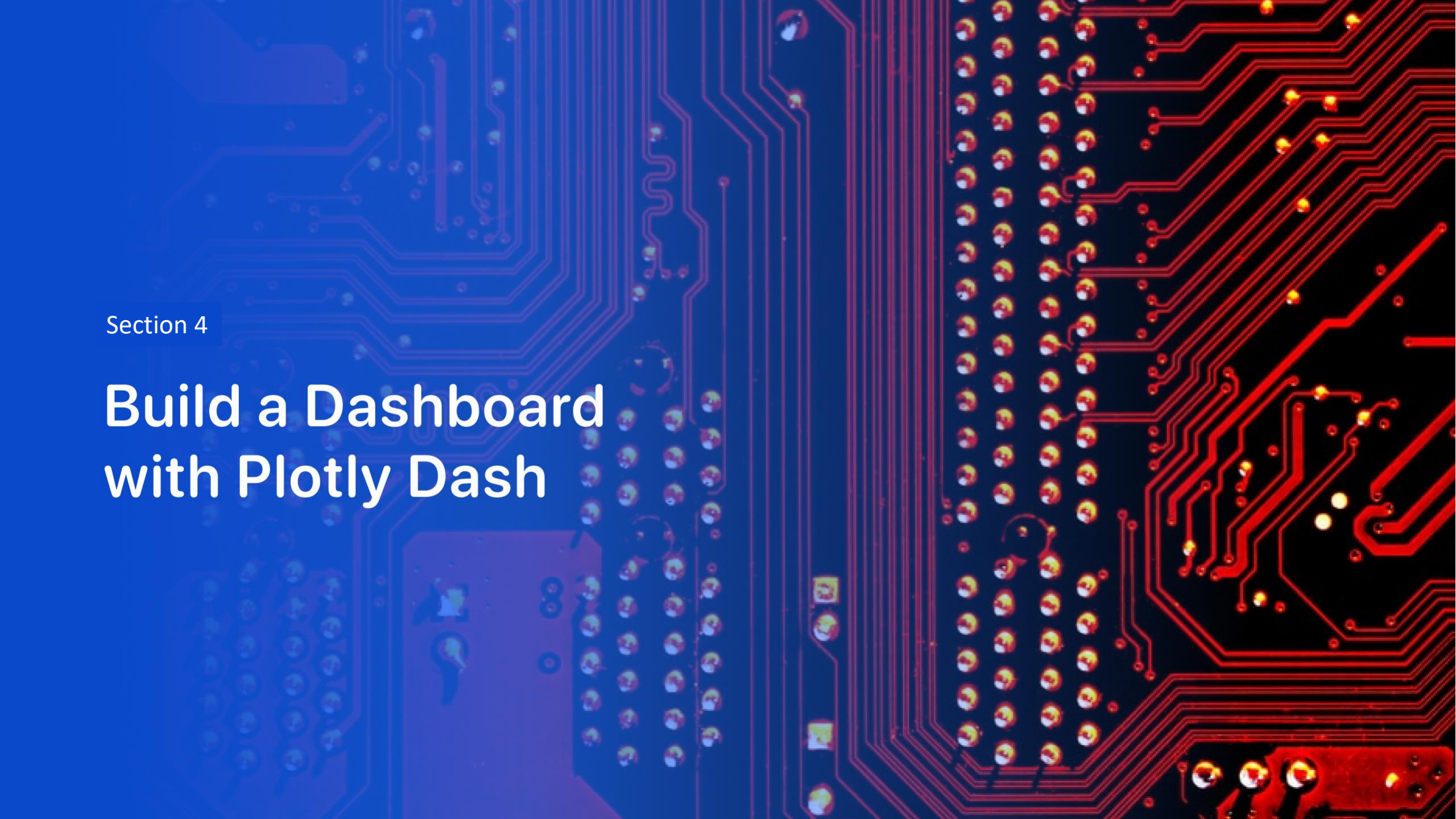
- **KSC LC-39A** has the **highest** success rate with **76.92%.**
- **CCAFS LC-40** has the **lowest** success rate with **26.92%.**

CCAF SLC-40 Proximities

- **COASTLINE:** The launch site is near the coastline for security reasons. Both to drop spent stages onto the water and prevent accidents from affecting people and public infrastructure.
- **HIGHWAYS, RAILWAYS AND CITIES:** The launch site is at a moderate distance from these locations. Both for security and logistic reasons. Not too close as to affect the vicinities, but not too far so that the supply chain is unsustainable.

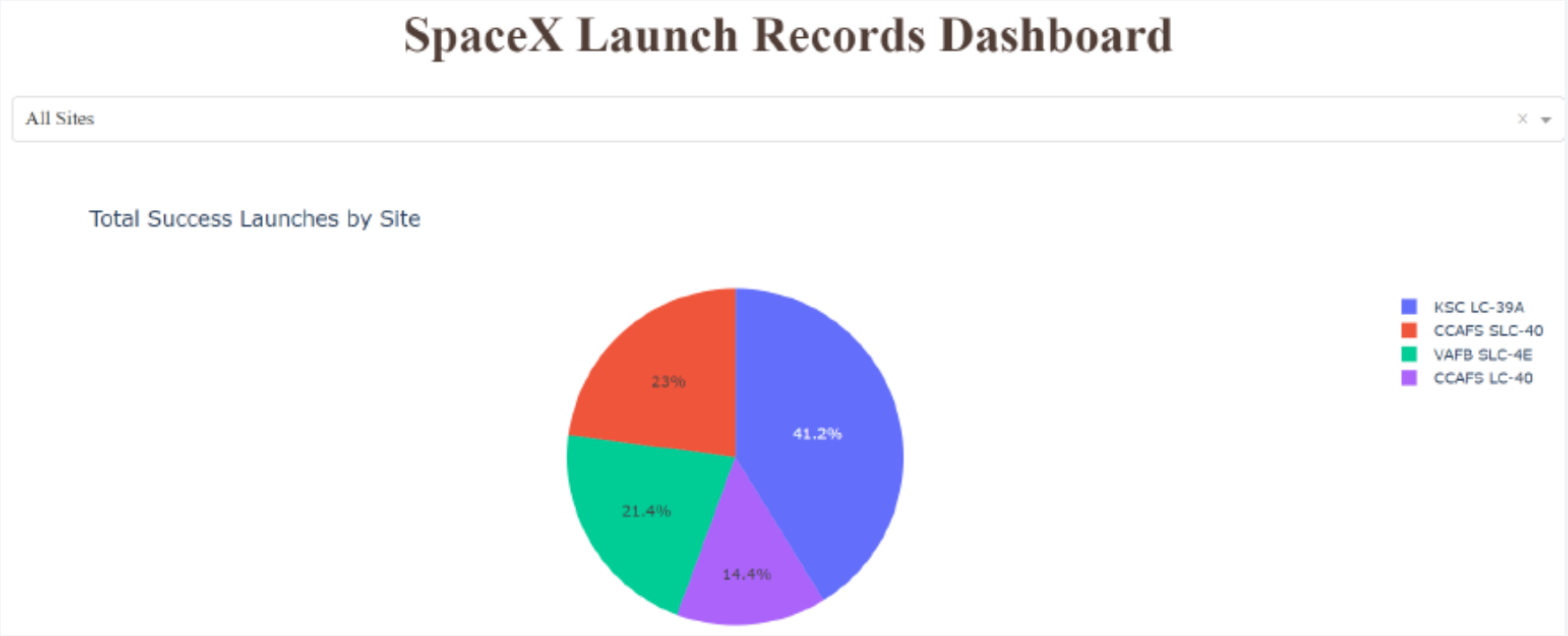





Section 4
Build a Dashboard with Plotly Dash

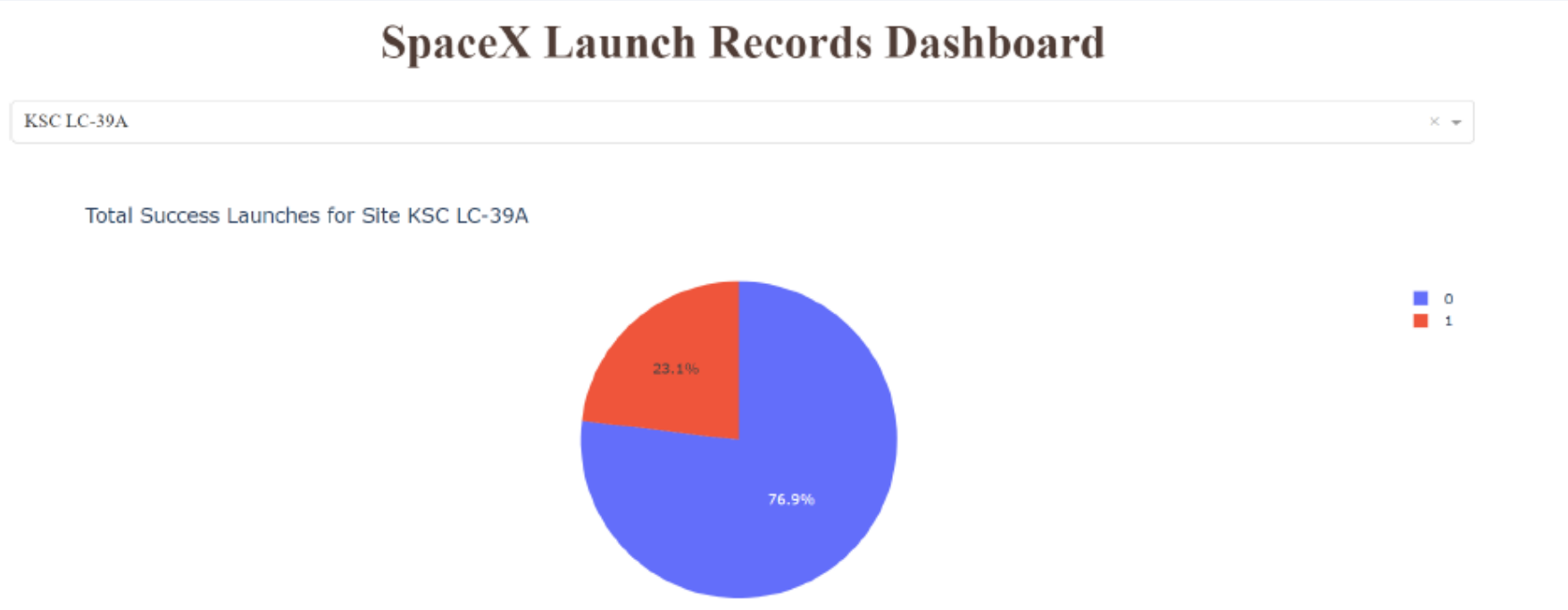
Launch Success by Launch Site

- **KSC LC-39A** has the most successful launches out of all launch sites with **41.2%** of the total launches.
- **CCAFS LC-40** has the least successful launches with **14.4%.**



KSC LC-39A Success Rate

- **KSC LC-39A** has the highest success rate, with **76.9%** (10 successful and 3 failed launches).



Payload vs Launch Outcome

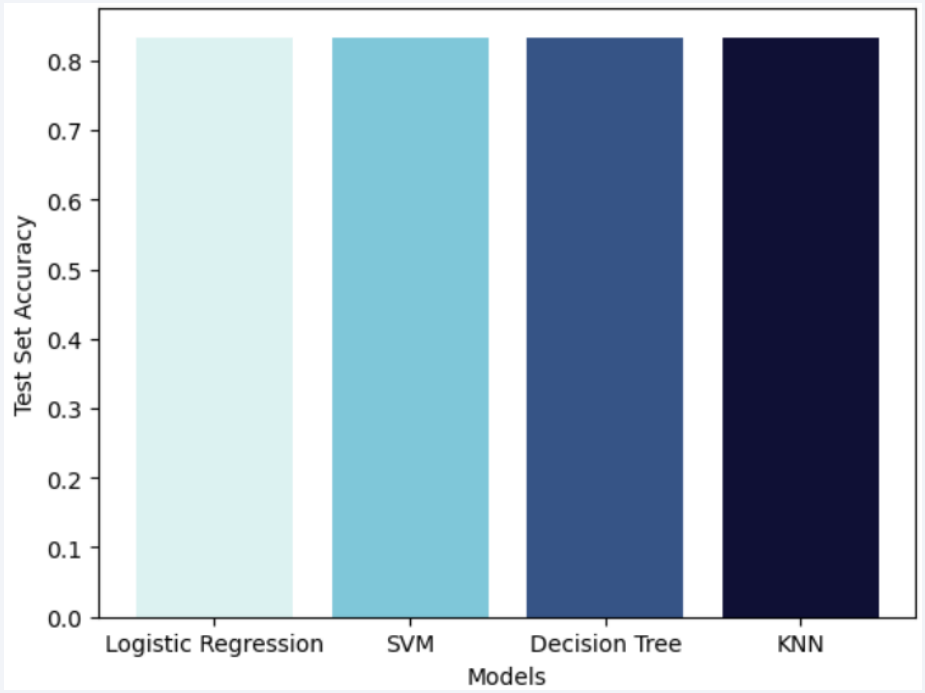
- Launches with payload masses between **2000 kg** and **4000 kg** have the highest success rate with **60%.** Payload masses between **6000 kg** and **8000 kg** have the lowest success rate with **0%.**
- Omitting the booster version **B5,** which had one out of one successful launches, the booster version with the highest success rate is the **FT** with over **63%.**

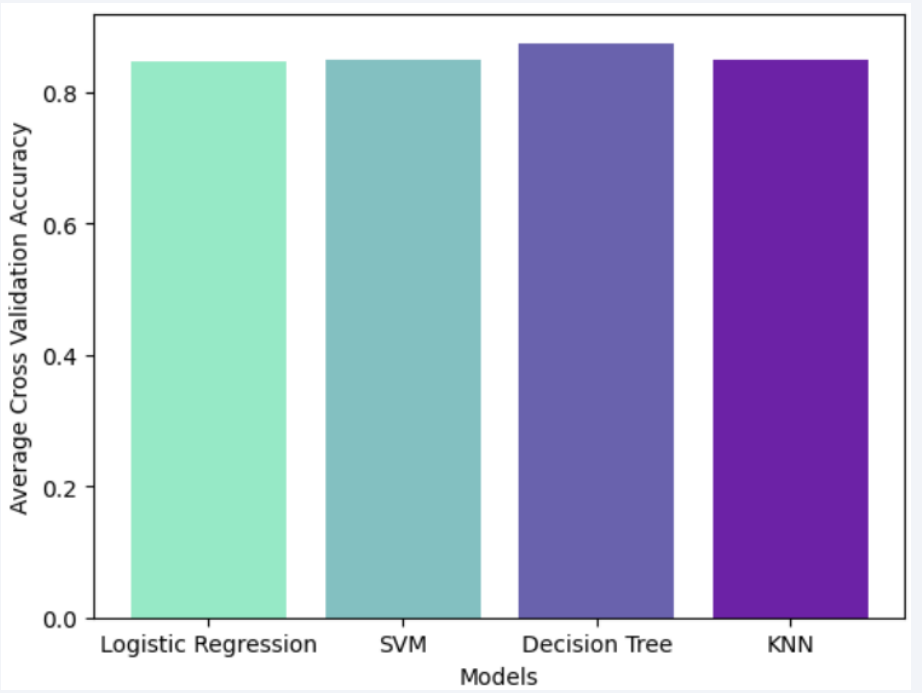
Section 5

Predictive Analysis (Classification)

Classification Accuracy

- **All** the models have the **same** test-set classification accuracy of **0.833** when evaluated with the **score()** function.
- The **Decision Tree Classifier** outperforms the other models when evaluated with the **best_score_()** function. Its average cross validation score (best_score_) was **0.875.**



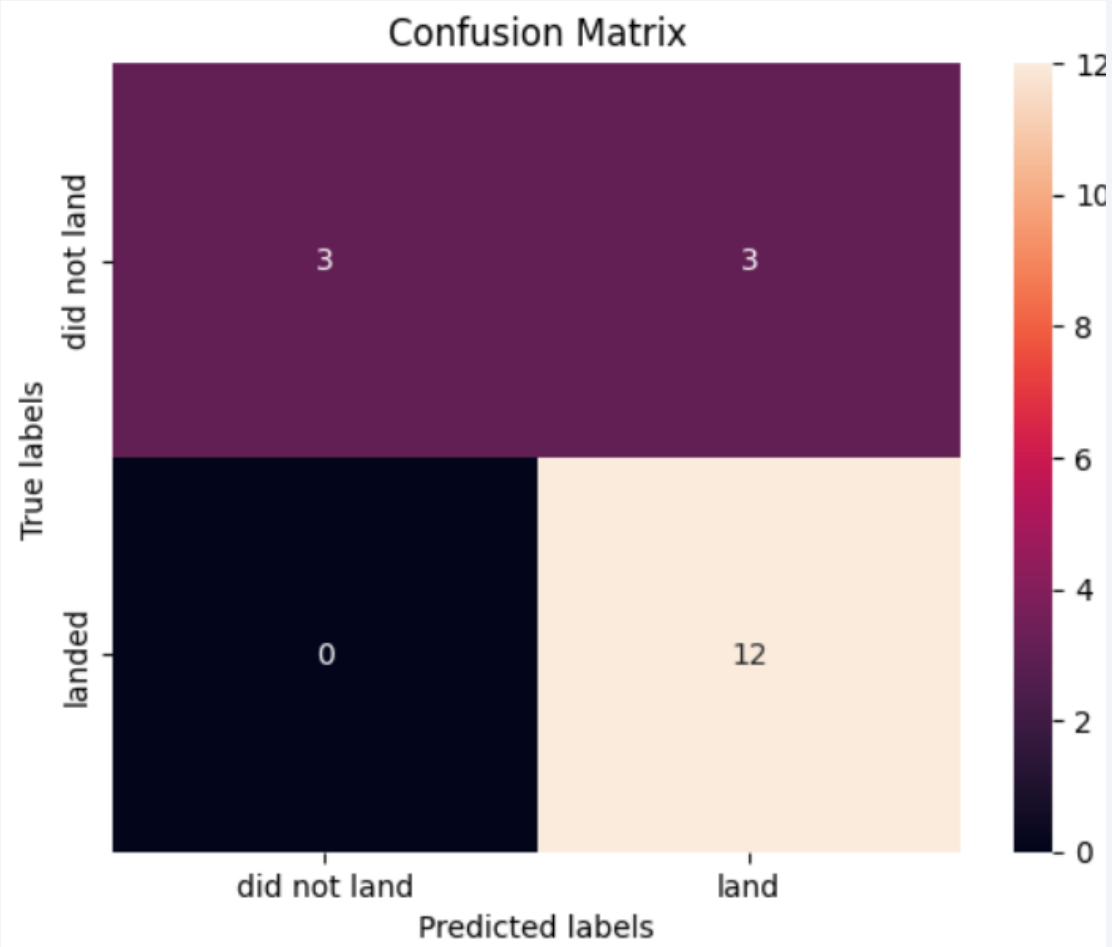


Confusion Matrix

The confusion matrices for **all** the models were **identical**, having **3 false negatives.**

- **True Positives:** 12
- **True Negatives:** 3
- **False Positives:** 0
- **False Negatives:** 3

Therefore, all models have the **same predictive performance.**



Conclusions

- The exploratory data analysis is a fundamental step in the data science process. It helps identify general insights from the data set and understand the data to a deeper end. From it, the following insights were identified:
    1. KSC LC 39A is the launch site with the highest success rate (**77%**).
    2. The orbits ES-L1, GEO, HEO and SSO have **100%** success rate.
    3. The success rate has **increased** over the years.
- Interactive analytics tools such as Folium and Dash help evaluate the data in an easier and visually insightful way, which leads to the following appreciations:
    1. Launch sites are close to coastlines and far from highways, railways and cities.
    2. The FT booster version has the highest success rate (**63%**).
    3. Launches with payload masses between 2000 kg and 4000 kg have the highest success rate (**60%**).
- All machine learning models (Logistic Regression, Support Vector Machine, Decision Tree and K Nearest Neighbors) had the **same test set accuracy (score()** function).
- The Decision Tree Classifier had a higher average cross validation accuracy (**best_score_()** function) than the other models.

Thank you!