IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

Gareth Ferrari
14 August 2025

Outline

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

Executive Summary

Summary of methodologies

- **Data Collection**: Gathered launch data from both the SpaceX REST API and by scraping a Wikipedia page.
- **Data Wrangling & Exploration**: Performed initial data wrangling for exploratory data analysis (EDA), which included data visualization and preparing features for our model. Also used SQL to better understand the dataset.
- **Data Visualization**: Created an interactive map with Folium to visualize launch site patterns. Built a Dashboard using Plotly to explore relationships between payloads, launch sites, and success rates.
- **Machine Learning**: Trained several machine learning models (Logistic Regression, Decision Tree, Support Vector Machine, and K-Nearest Neighbors) to predict launch success. Used Grid Search to find the optimal hyperparameters and determine the most accurate model.

Summary of results

- Success rates generally increase over time
- Launch site KSC LC-39A had the highest success rates
- The FT and B4 Booster versions appear that they may perform well with payload masses under about 5500kg, but not with heavier loads
- A Decision Tree Classifier model is the appropriate choice for predicting launch success with 94% accuracy score

Introduction

Project background and context

- The commercial space industry is booming, with companies like SpaceX leading the way.
- A key factor in SpaceX's lower costs is the reusability of its rockets' first stage.
- Our team at "Space Y" has been tasked with competing in this market and determining our own launch pricing.

Problems we wanted to find answers to

- To understand the factors that influence the reusability of a rocket's first stage.
- By analyzing public data from SpaceX, we aimed to create a machine learning model that predicts whether a rocket's first stage will be successfully recovered.
- This model will allow us to accurately estimate launch costs and inform our company's pricing strategy.

Section 1

Methodology

Methodology

Executive Summary

- Data collection methodology:
  - Data collected via the SpaceX REST API and via Web scraping a Wikipedia page
- Perform data wrangling
  - Data filtered, cleaned, formatted to prepare for analysis and modeling
- Perform exploratory data analysis (EDA) using visualization and SQL
- Perform interactive visual analytics using Folium and Plotly Dash
- Perform predictive analysis using classification models
  - Data split into training and test data. Training data used to train several models, using Grid Search to find optimal hyperparameters. Models evaluated against test data to determine

Data Collection – SpaceX API

- Data collected and processed with python to prepare it for analysis.
- Data source: SpaceX REST API
- Key python packages used for collection and processing: requests, pandas, numpy, datetime

High-level Overview of Steps

1. Request data from SpaceX API using GET requests
2. Decode response content as JSON Nand load it into a Pandas DataFrame
3. Filter data to keep only features and dates (before 13-Nov-2020) we are interested in
4. Use provided functions to parse and extract data we need (e.g. booster name from rocket, mass of payload and orbit from payload, etc.)
5. Format data for analysis
6. Filter data to keep only Falcon 9 launches
7. Clean rows with missing PayloadMass by replacing the missing data with the mean value

GitHub URL of notebook: https://github.com/gf-coursera/ibm-data-science-capstone/blob/main/gf-ibm-ds-data-collection-spacex-api.ipynb

Data Collection - Scraping

- Data collected and processed with python to prepare it for analysis.
- Data source: SpaceX Falcon 9 Wikipedia page
- Key python packages used for collection and processing: requests, pandas, BeautifulSoup, unicodedata

High-level Overview of Steps

1. Request the Falcon 9 launch Wikipedia page using GET requests
2. Load HTML response text content into a BeautifulSoup object for parsing
3. Find the table within the HTML
4. Parse table and load data into a Pandas DataFrame

GitHub URL of notebook: https://github.com/gf-coursera/ibm-data-science-capstone/blob/main/gf-ibm-ds-data-collection-webscraping.ipynb

Data Wrangling

- Data processed with python to prepare it for analysis.
- Most important objective during processing was to determine training labels, specifically the Class label based on the Outcome data. This is what we will use during subsequent modeling to predict success/failure of launches.
- Key python packages used for processing: pandas, numpy

High-level Overview of Steps

1. Load data from csv into Pandas DataFrame
2. Check for missing values
3. Inspect data types to determine which are numerical and which are categorical
4. Investigate counts of launch sites, orbits, and outcomes to understand the data better
5. Create binary Class label based on Outcome data (to facilitate further analysis)

GitHub URL of notebook: https://github.com/gf-coursera/ibm-data-science-capstone/blob/main/gf-ibm-ds-data-wrangling.ipynb

EDA with Data Visualization

- Exploratory Data Analysis and Feature Engineering
- Scatter plots to visualize relationships between various features:
  - Flight Number vs Launch Site, Payload vs Launch Site, Flight Number vs Orbit Type, Payload vs Orbit Type
- Bar chart to explore relationship between orbit type and success rate
- Line chart of yearly average success rate to show trend over time

GitHub URL of notebook: https://github.com/gf-coursera/ibm-data-science-capstone/blob/main/gf-ibm-ds-eda-data-vizualization.ipynb

EDA with SQL

Summary of the SQL queries performed

- Display the names of the unique launch sites in the space mission
- Display 5 records where launch sites begin with the string 'CCA'
- Display the total payload mass carried by boosters launched by NASA (CRS)
- Display average payload mass carried by booster version F9 v1.1
- List the date when the first successful landing outcome in ground pad was achieved.
- List the names of the boosters which have success in drone ship and have payload mass greater than 4000 but less than 6000
- List the total number of successful and failure mission outcomes
- List all the booster_versions that have carried the maximum payload mass, using a subquery with a suitable aggregate function.
- 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.
- 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.

GitHub URL of notebook: https://github.com/gf-coursera/ibm-data-science-capstone/blob/main/gf-ibm-ds-eda-sql.ipynb

Build an Interactive Map with Folium

- Built an interactive map with Folium to conduct interactive visual analytics to find relationships between various features and the success rates of the launches at different sites
- Circles were added to the map to indicate the location of the launch sites
- Colored markers were added to visually indicate success (green) and failure (red)
- The markers were grouped using marker clusters to simplify markers on the map with the same coordinates
- Lines were added to the map to mark distances to various nearby geographic features (coastline, city, railway, highway)
- Additional markers were added to provide additional textual information, such as the names of the launch sites and the distances to the geographic features

GitHub URL of notebook: https://github.com/gf-coursera/ibm-data-science-capstone/blob/main/gf-ibm-ds-interactive-map.ipynb

Build a Dashboard with Plotly Dash

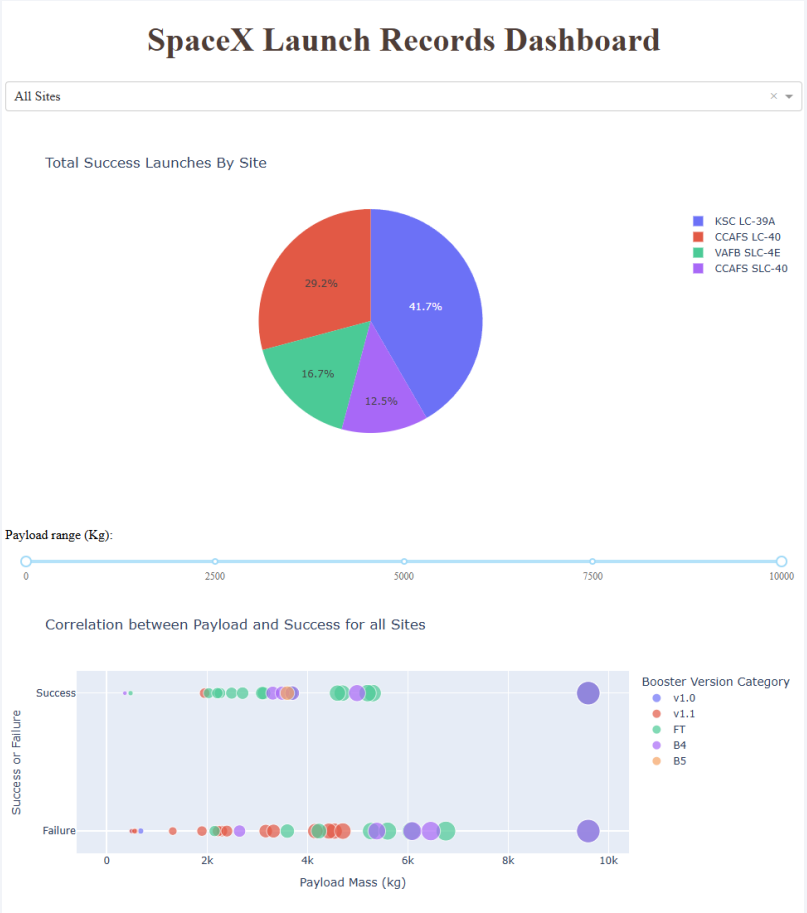
Built an interactive dashboard with Plotly Dash

Interactive controls:

- Dropdown list to enable selection of either a specific site, or of all sites
- Slider to select payload range

Charts:

- Pie chart to show the total successful launches (if all sites selected) or show success vs failed counts for the selected site
- Scatter chart to show the correlation between payload and launch success

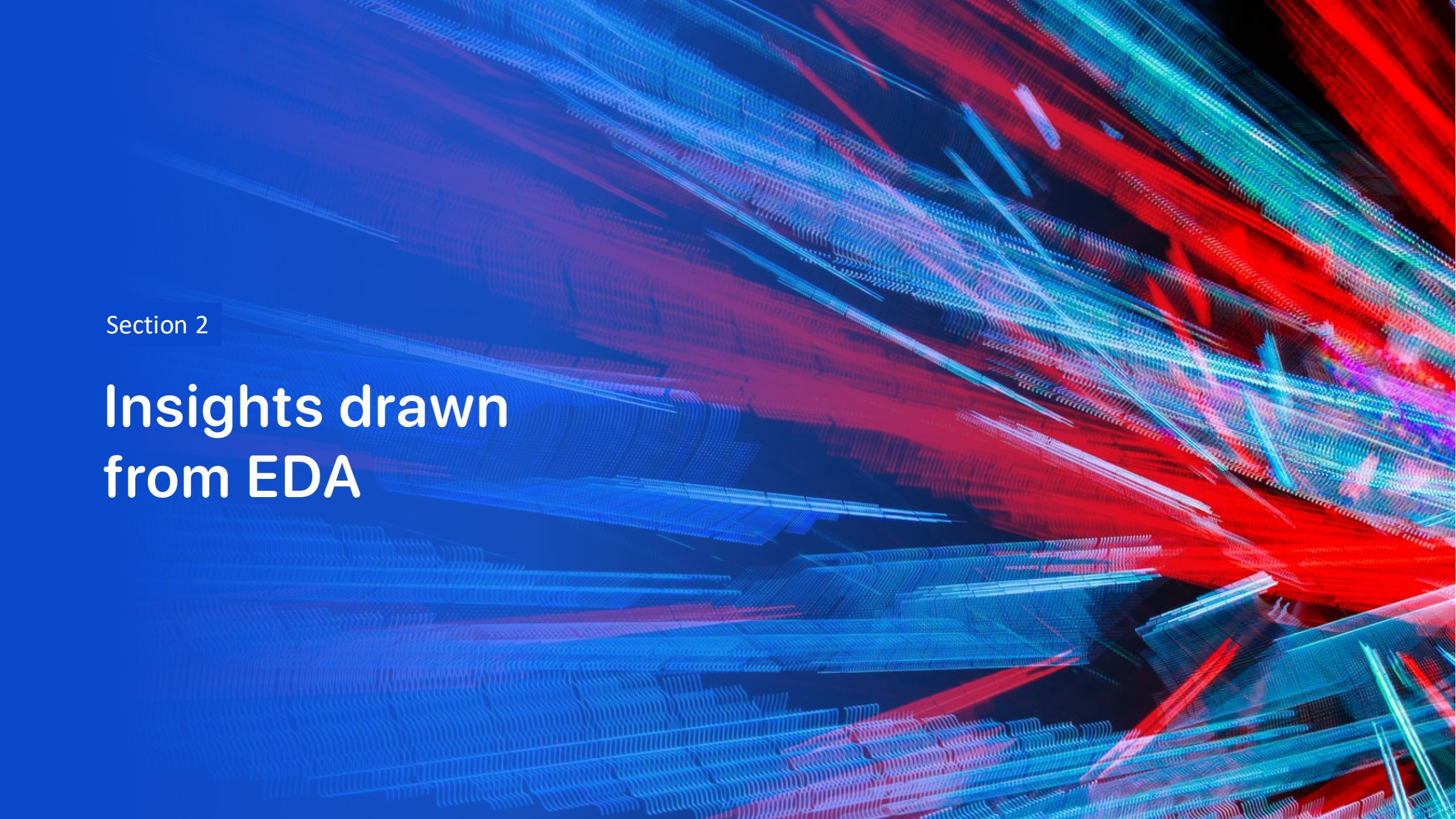


GitHub URL of notebook: https://github.com/gf-coursera/ibm-data-science-capstone/blob/main/gf-ibm-ds-dash.py

Predictive Analysis (Classification)

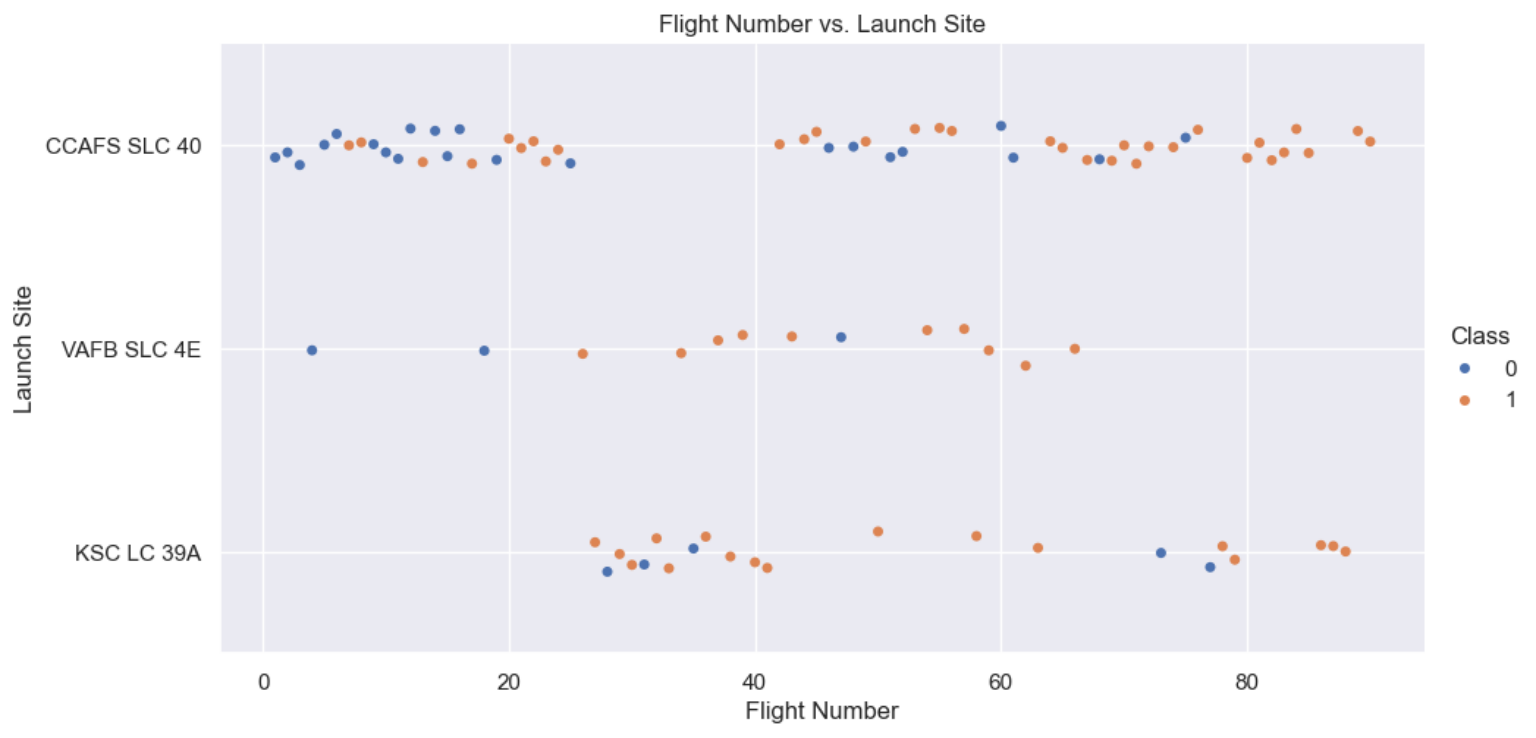
- Loaded the dependent variable (target) and the independent variables (predictive features) into datasets
- Standardized the independent variables using StandardScaler
- Split data into training and test sets (20% for the test set)
- Used GridSearchCV to fit several models to find the best of the given parameters (models tested: Logistic Regression, Support Vector Machine, Decision Tree Classifier, K-Nearest Neighbors)
- For each "best" model, calculated accuracy on the test data to determine which, if any, model performed best

GitHub URL of notebook: https://github.com/gf-coursera/ibm-data-science-capstone/blob/main/gf-ibm-ds-predictive-analysis.ipynb


Section 2
Insights drawn from EDA

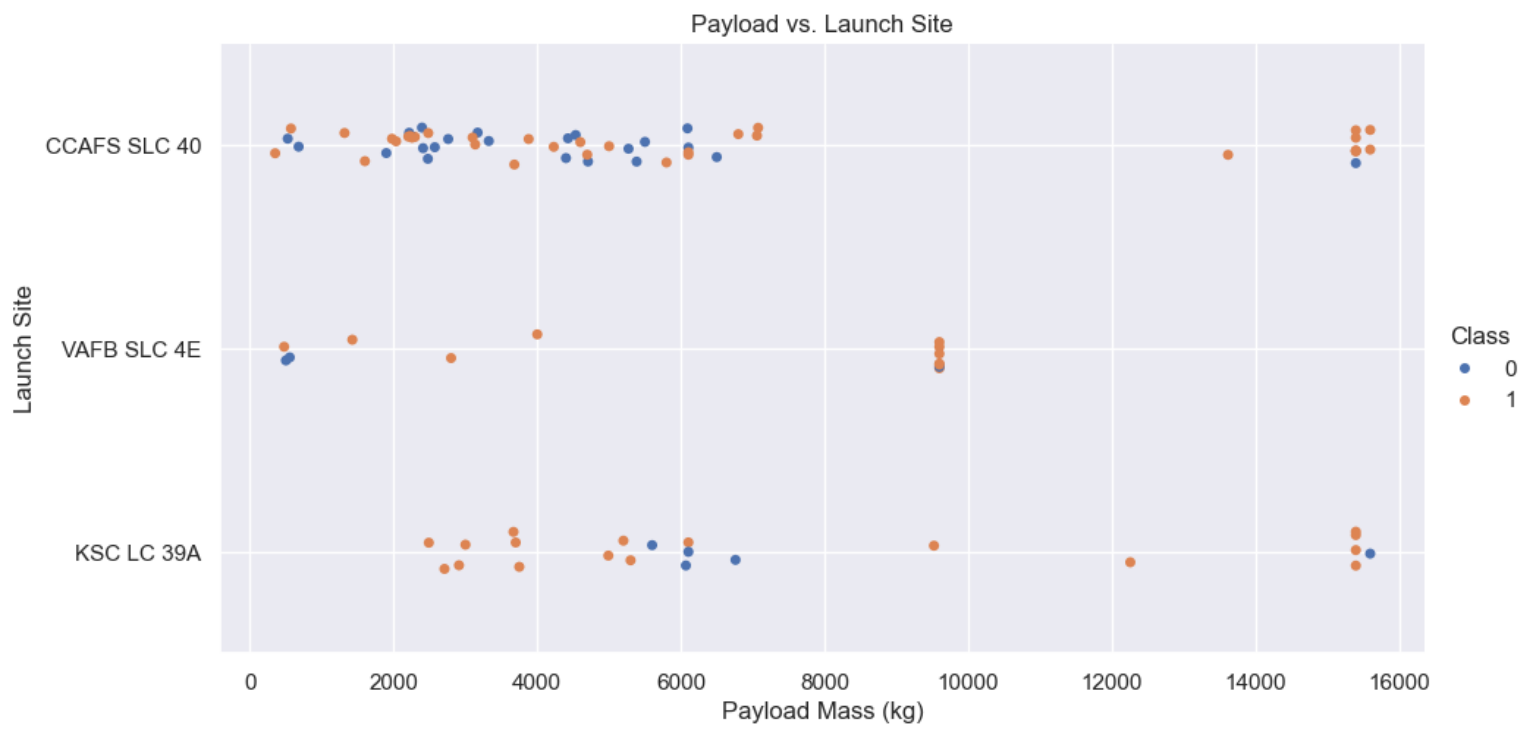
Flight Number vs. Launch Site

- It appears that the rate of success increases with flight number
- Intuitively this logically makes sense as the team learned from past launches and improved the success rate
- We can also observe that there were no early launches at site KSC LC 39A. This is important to keep in mind when comparing total success rates at different sites.



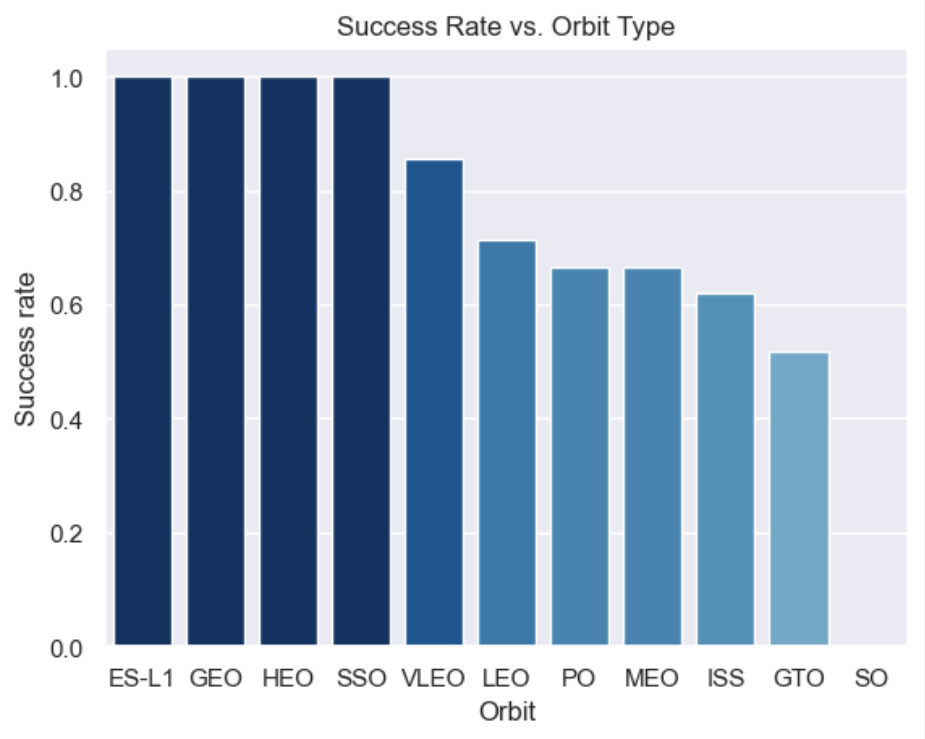
Payload vs. Launch Site

- Superficially this plot appears to show that mid-payload launches (e.g. 7K-15K kg) have the highest success rates
- However, we need to dig deeper before making that assumption. The early launches may be more likely to be lighter loads, so the higher failure rates may not be directly caused by the payload mass.



Success Rate vs. Orbit Type

- ES-L1, GEO, GEO, SSO all show perfect success rates
- SO shows no successes
- No obvious pattern based on successful orbits. They include combination of circular and elliptical, higher and lower orbits, geosynchronous and non-geosynchronous.
- Further analysis would be needed to dig deeper to determine if orbit really impacts success rates, or other factors more important.

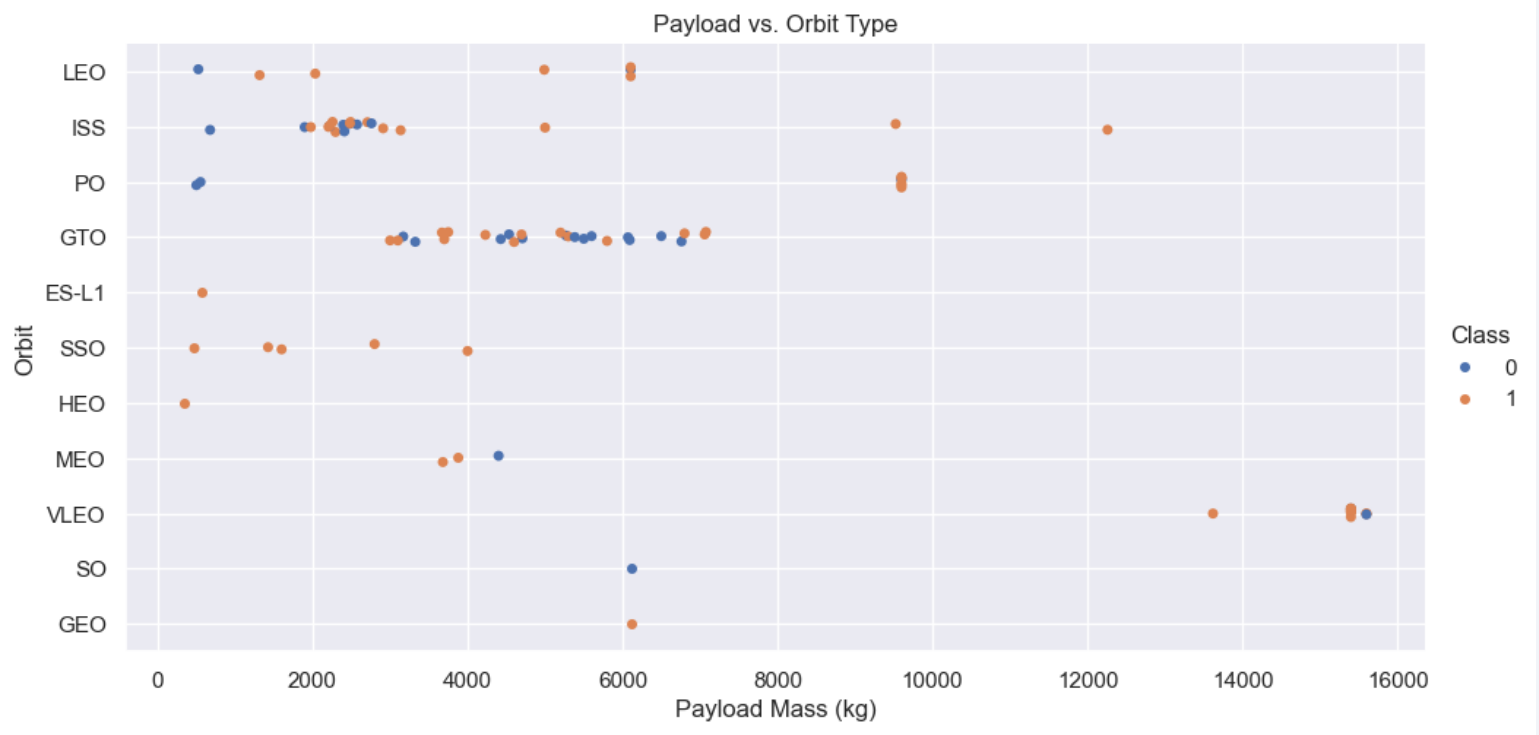


Flight Number vs. Orbit Type

- Most interesting info we can get from this chart is that not all orbits were attempted until later flights, e.g. MEO, VLEO, SO, GEO
- At first one may be tempted to interpret this plot as showing that some orbits such as ISS or GTO improved with flight number, but there isn't really enough data here to make that assumption.

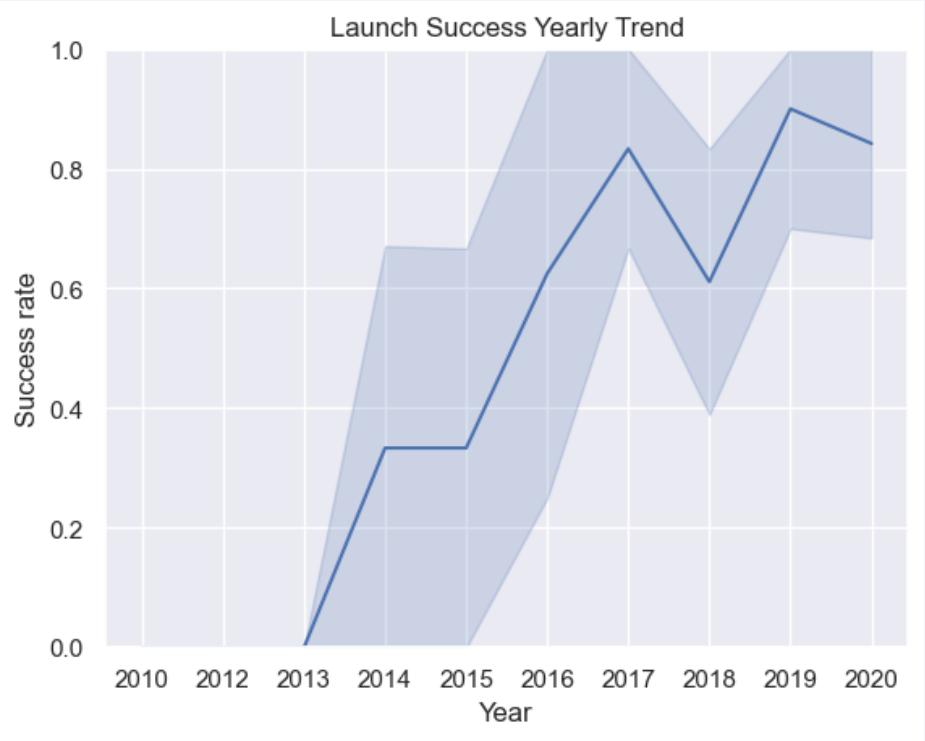
Payload vs. Orbit Type

- It appears that there may be a relationship between payload mass and success for some orbits (although not enough data points to be sure yet)
- LEO, ISS, PO have more successful launches with heavier payloads than with lighter payloads
- GTO success rate doesn't appear to be related to payload mass. The successes seem quite randomly distributed in relation to payload.



Launch Success Yearly Trend

- Despite two dips in success rates (2018 and 2020) the overall trend is that success rate increased between 2013 and 2020.
- Logically this makes sense, the team should be learning from earlier launches so we expect that for the company to be viable, the success rates should generally improve over time.



All Launch Site Names

- There are four distinct Launch Sites in our dataset

```
%%sql
SELECT DISTINCT(Launch_Site)
FROM SPACEXTABLE;
```

✓ 0.0s Python

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

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

Launch Site Names Begin with 'CCA'

```
%%sql
SELECT *
FROM SPACEXTABLE
WHERE Launch_Site LIKE "CCA%"
LIMIT 5;
```

✓ 0.0s Python

```
* 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 find 5 records where launch sites begin with `CCA` by filtering with a WHERE clause and limiting the result set with LIMIT

Total Payload Mass

- The total payload carried by boosters from NASA is 45596kg

```
%%sql
SELECT SUM(PAYLOAD_MASS__KG_) AS Total_PayloadMass_kg
FROM SPACEXTABLE
WHERE Customer = "NASA (CRS)";
```

✓ 0.0s Python

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

Total_PayloadMass_kg
45596

Average Payload Mass by F9 v1.1

- The average payload mass carried by booster version F9 v1.1 is 2928.4kg

```
%%sql
SELECT AVG(PAYLOAD_MASS__KG_) AS Avg_PayloadMass_kg
FROM SPACEXTABLE
WHERE Booster_Version = "F9 v1.1";
```

✓ 0.0s Python

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

Avg_PayloadMass_kg
2928.4

First Successful Ground Landing Date

- The the first successful landing outcome on ground pad was on the 22nd December 2015

```
%%sql
SELECT MIN(Date) AS FirstSucessfulGroundPadLandingDate
FROM SPACEXTABLE
WHERE Landing_Outcome = "Success (ground pad)";
```

✓ 0.0s Python

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

FirstSucessfulGroundPadLandingDate
2015-12-22

Successful Drone Ship Landing with Payload between 4000 and 6000

- There were four boosters successfully landed on drone ship and had payload mass greater than 4000 but less than 6000

```
%%sql
SELECT DISTINCT(Booster_Version)
FROM SPACEXTABLE
WHERE Landing_Outcome = "Success (drone ship)"
    AND PAYLOAD_MASS__KG_ > 4000
    AND PAYLOAD_MASS__KG_ < 6000;
```

✓ 0.0s Python

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

- There were 100 Successful mission outcomes and 1 Failure in the dataset.
- NOTE: Values in Mission_Outcome are not just Success and Failure (e.g. "Success", "Success " (with a trailing space), "Success (payload status unclear)"), so need group together to get total counts.

```
%%sql
SELECT
    CASE
        WHEN Mission_Outcome LIKE "Success%" THEN "Success"
        WHEN Mission_Outcome LIKE "Failure%" THEN "Failure"
        ELSE "Unknown"
    END AS Mission_Outcome,
    COUNT(*)
FROM SPACEXTABLE
GROUP BY
    CASE
        WHEN Mission_Outcome LIKE "Success%" THEN "Success"
        WHEN Mission_Outcome LIKE "Failure%" THEN "Failure"
        ELSE "Unknown"
    END;
```

✓ 0.0s Python

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

Mission_Outcome	COUNT(*)
Failure	1
Success	100

Boosters Carried Maximum Payload

- There are 12 boosters which have carried the maximum payload mass of 15600kg

```
%%sql
SELECT DISTINCT(Booster_Version)
FROM SPACEXTABLE
WHERE PAYLOAD_MASS__KG_ = (
    SELECT MAX(PAYLOAD_MASS__KG_) FROM SPACEXTABLE
    );
```

✓ 0.0s Python

```
 * 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

2015 Launch Records

- There are only two failed landing outcomes in drone ship during 2015, one in January and one in April

```
%%sql
SELECT
    CASE substr(Date, 6,2)
            WHEN '01' THEN 'January'
            WHEN '02' THEN 'February'
            WHEN '03' THEN 'March'
            WHEN '04' THEN 'April'
            WHEN '05' THEN 'May'
            WHEN '06' THEN 'June'
            WHEN '07' THEN 'July'
            WHEN '08' THEN 'August'
            WHEN '09' THEN 'September'
            WHEN '10' THEN 'October'
            WHEN '11' THEN 'November'
            WHEN '12' THEN 'December'
            ELSE 'Unknown'
        END AS month_name,
    Landing_Outcome,
    Booster_Version,
    Launch_Site
FROM SPACEXTABLE
WHERE substr(Date,0,5)='2015'
    AND Landing_Outcome = "Failure (drone ship)";
```

✓ 0.0s Python

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

month_name	Landing_Outcome	Booster_Version	Launch_Site
January	Failure (drone ship)	F9 v1.1 B1012	CCAFS LC-40
April	Failure (drone ship)	F9 v1.1 B1015	CCAFS LC-40

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

- The most common outcome was "No attempt" which isn't very meaningful if we are interested in successes and failures
- The second and third rankings tell us that attempting to land on a drone ship only had a 50% success rate (5 each for success and failure)
- There were three successful landings on a ground pad and no failures listed in this dataset. This could indicate that ground landings are easier than other locations, but the sample size is too small to be confident. It could be an area for further examination.

```
%%sql
SELECT Landing_Outcome, COUNT(*)
FROM SPACEXTABLE
WHERE Date BETWEEN '2010-06-04' AND '2017-03-20'
GROUP BY Landing_Outcome
ORDER BY COUNT(*) DESC;
```

✓ 0.0s Python

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

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

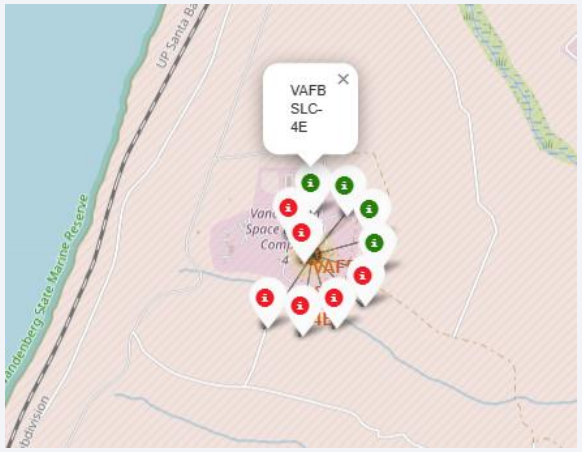
Launch Sites Proximities Analysis

Launch site locations

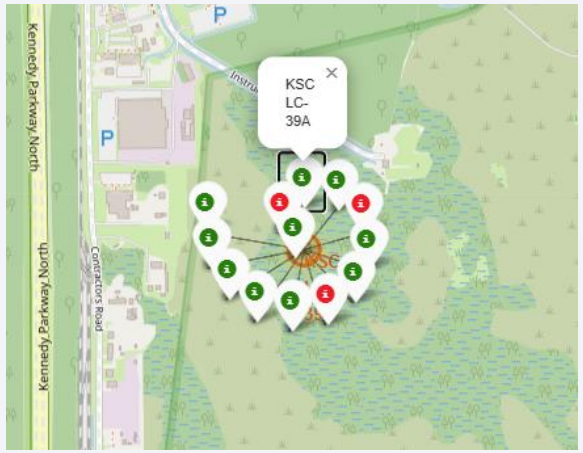
- All launch sites are within the continental USA
- All sites are located in the south and near the coasts

Launch outcome markers at each site

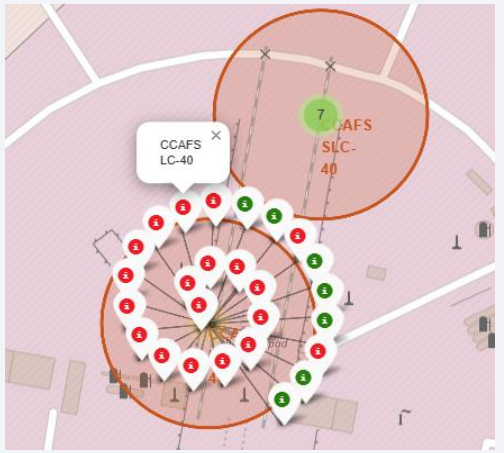
VAFB SLC-4E



KSC LC-39A



CCAFS LC-40



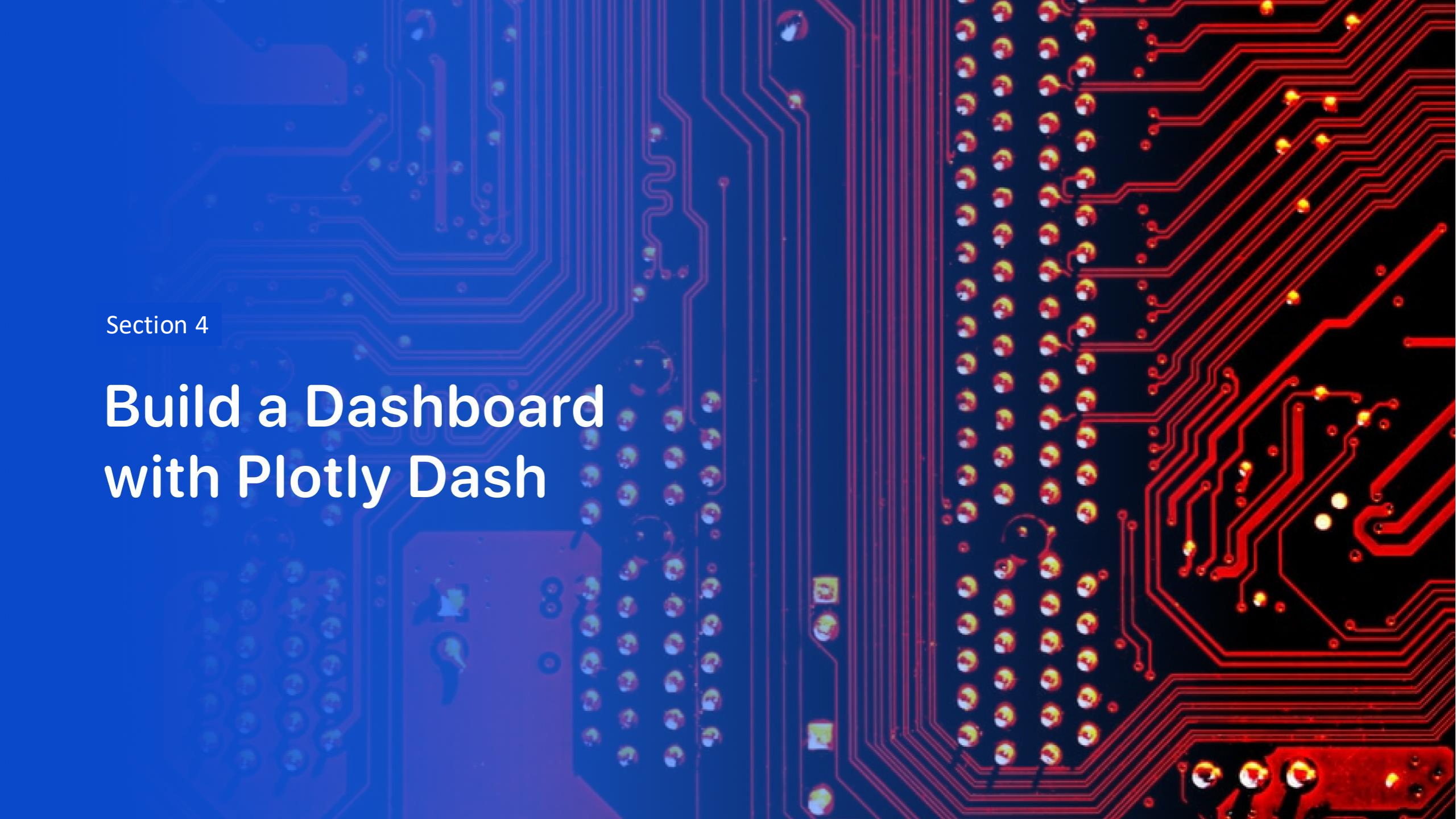
CCAFS SLC-40

Successes are marked in green and Failures in red

These plots provide a lot of information very quickly. We see for example that KSC LC-39A has the highest proportion of green (success) markers and that CCAFS LC-40 has the most total number of launches

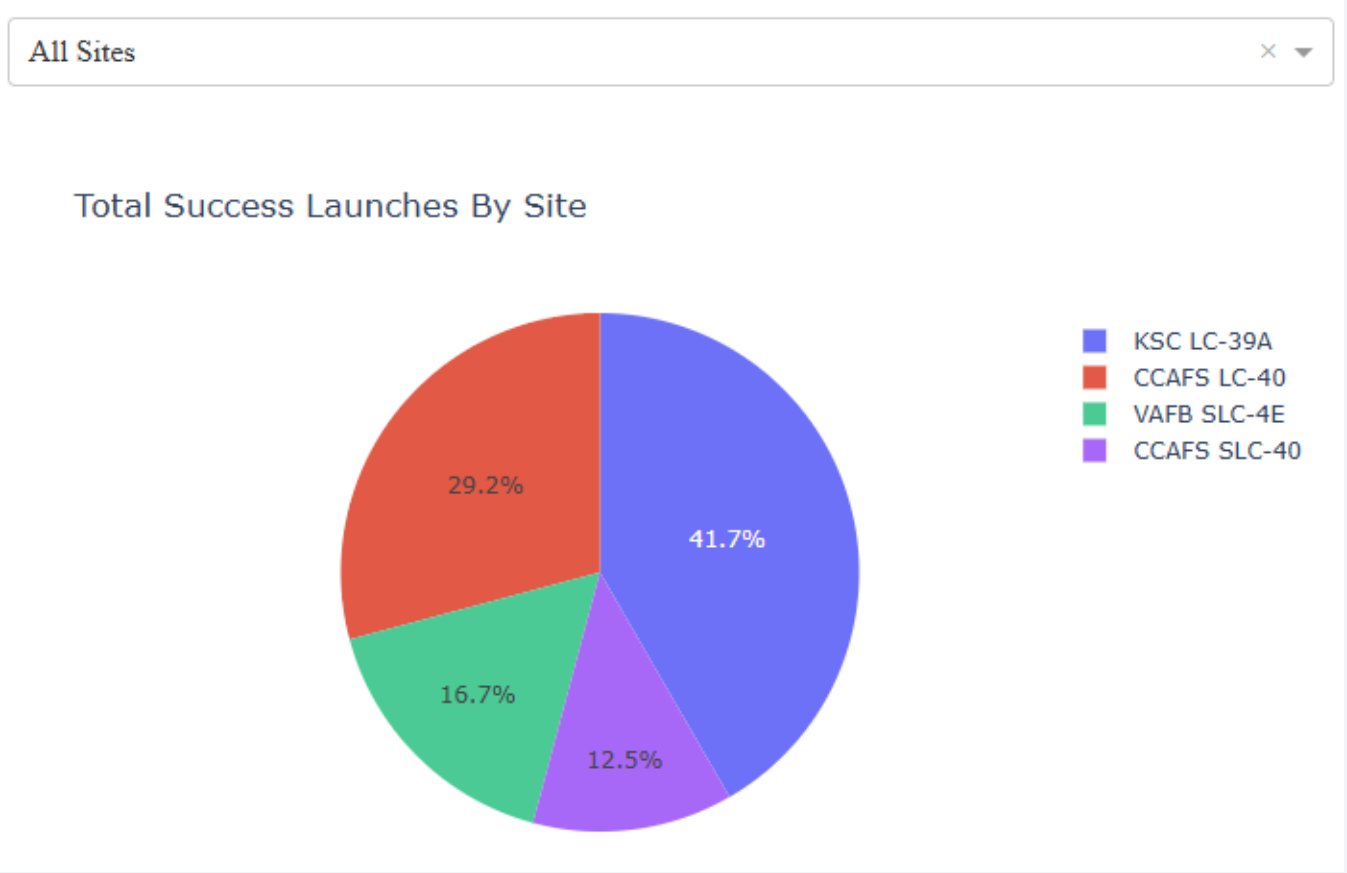
Proximities to key geographic entities

- Distances of key geographic entities from launch site CCAFS SLC-40
- Features we want close for safety
  - **Coastline: 0.86km**
- Features we don't want too close
  - **Railway: 7.53km**
  - **Highway: 0.59km**
  - **City (Cape Canaveral): 17.31km**
- The close proximity to the coastline is reassuring, however the proximity to other key geographic entities is surprisingly close.


Section 4
Build a Dashboard with Plotly Dash

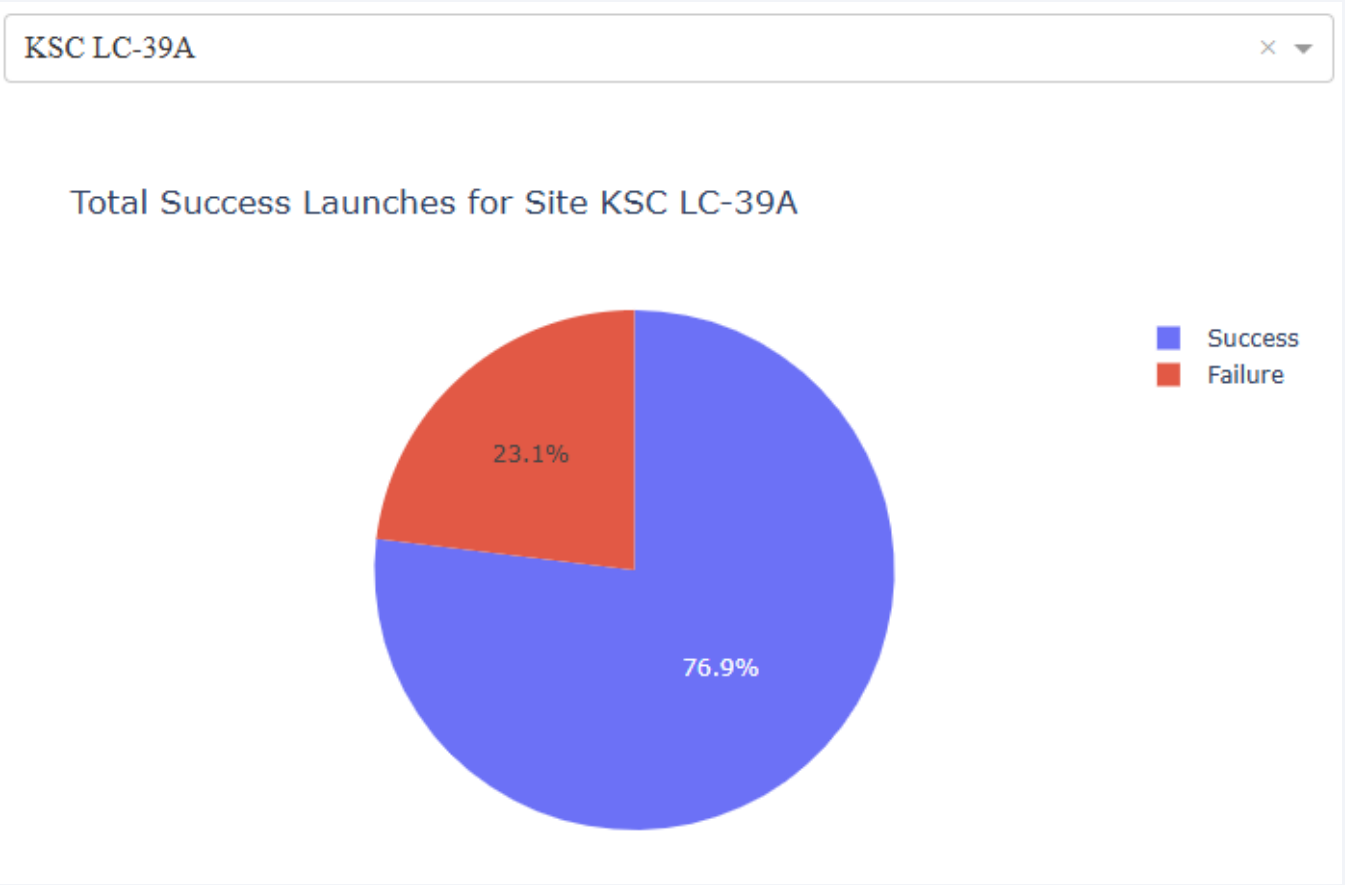
Total successes by site

- It is clear to see that the site with the most successful launches is KSC LC-39A



Site with highest success ratio

- Focusing in on the site with the most successful launches (KSC LC-39A) we can see launches from this site had a success rate of 76.9%



Payload vs Launch Outcome filtered by Payload

Full Payload Mass Range

Mid-Payload Mass Range Only

- Filtering out the smallest and largest payloads, the first thing we can see if that there were no launches by Booster v1.0 in this range
- Booster v1.1 is the least successful
- Boosters FT and B4 both have a mixture of of successes and failures, with more successes at lower payload masses. Both appear more likely to fail with heavier payloads.
- B5 doesn't have enough data points to draw any conclusions

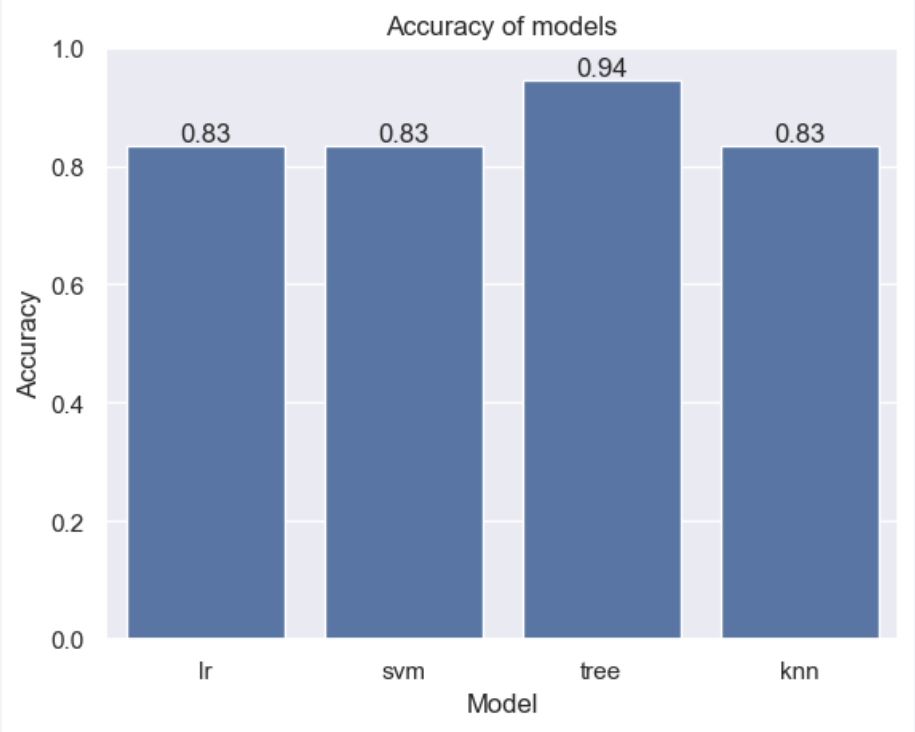
Section 5

Predictive Analysis (Classification)

Classification Accuracy

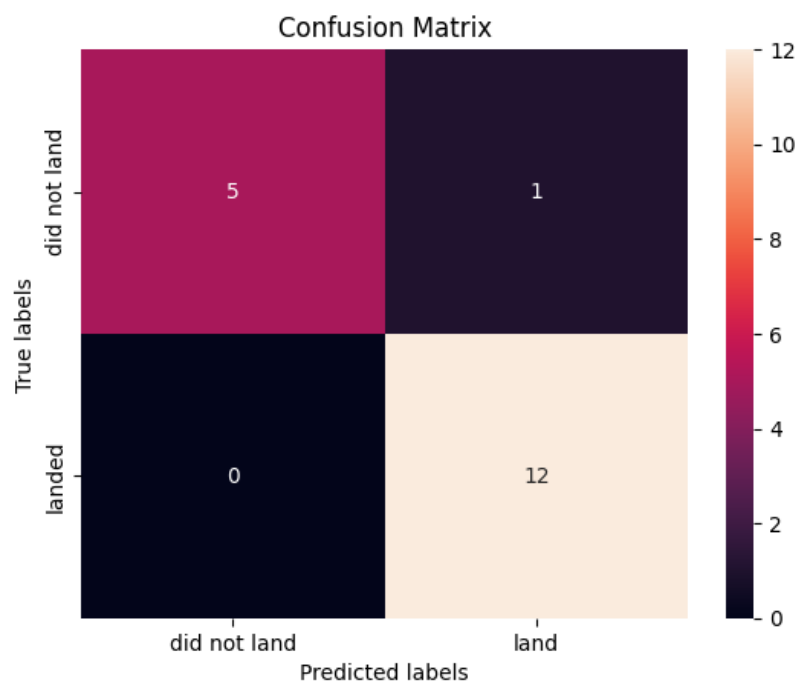
NOTE: For the purposes of this presentation, the Decision Tree will be treated as-if it is the best performing model. I am doing this because I'm guessing this is what IBM wants, however this is not really a valid assumption. The Decision Tree model is very sensitive to the random seed in this case. Very different accuracy scores (including much worse) can be achieved with different seeds. The real answer is that based on the data and analysis conducted in this project, it is not possible to determine which model is really the best. More analysis would be needed before picking the best model, but that is outside the scope of this project.

- The Decision Tree model gives the highest accuracy on test data with a score of 94% compared to 83% for the other models (Logistic Regression, SVM, KNN)

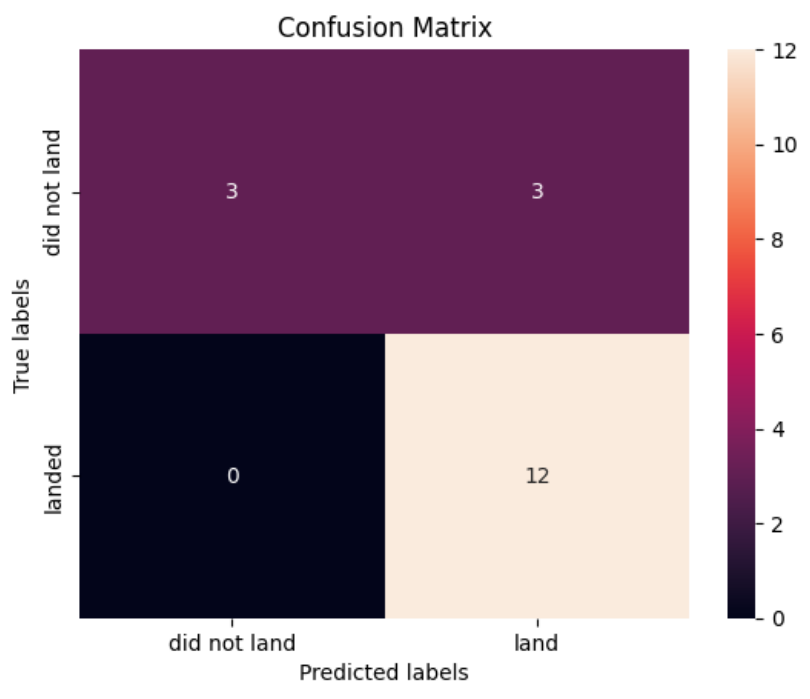


Confusion Matrix

Decision Tree



Logistic Regression, SVM, KNN



- The confusion matrix for the Decision Tree shows only a single misclassification. It incorrectly predicted one launch as a landing that actually wasn't (Type I error – False Positive).
- This is a much better result than the other models which incorrectly predicted 3 launches as landings that actually did not land.

Conclusions

- Success rates generally increase over time
- Site KSC LC-39A had the highest success rates
- A Decision Tree Classifier model is the appropriate choice for predicting launch success with a 94% accuracy score
  (NOTE: Not really – see caveat in the earlier slide, but for the purposes of this presentation we will pretend this is true)
- The FT and B4 Booster versions appear that they may perform well with payload masses under about 5500kg, but not with heavier loads.
- There is enough to indicate that there may be a relationship between Orbit and success rate,
  but insufficient evidence to be confident yet. This could be a good area for further investigation with more data and analysis.

Thank you!