IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

Jessica Swansinger

02/13/2025

Outline

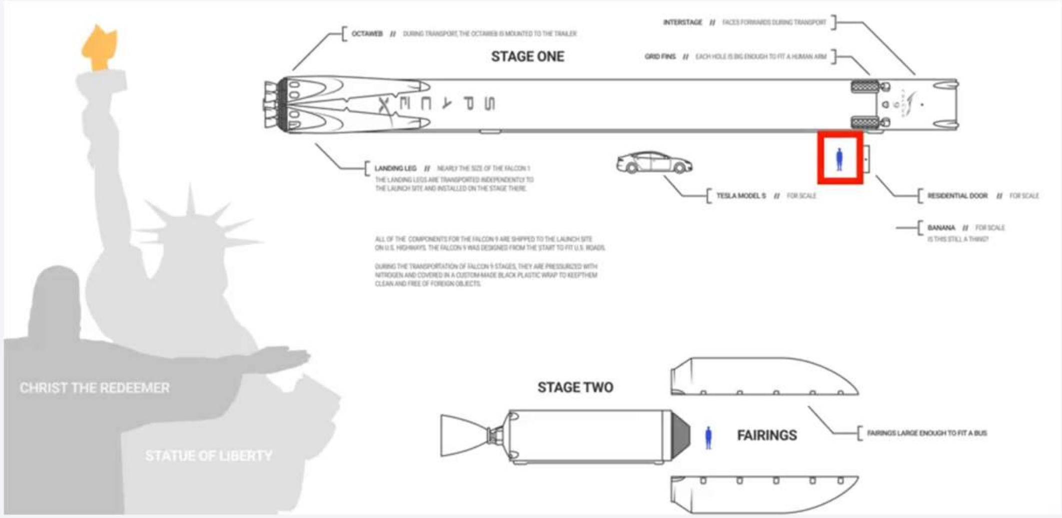
- Executive Summary
- Introduction
- Methodology
- Results
- Conclusion
- Appendix

Executive Summary

- Summary of methodologies
  - Data Collection through API & Web Scraping
  - Data Wrangling
  - Data Analysis with SQL & Data Visualization
  - Visual, Interactive Analytics with Folium
  - Machine Learning Prediction
- Summary of all results
  - Data visualization and Analytic Screenshots
  - Predictive Analysis Results for Most Accurate Model

Introduction

- Companies are looking to commercialize space by making space travel affordable
- SpaceX is one of, possibly the most successful, space travel companies
- SpaceX's rocket launches are relatively inexpensive because they reuse Stage 1



Can we use a machine learning model to predict if SpaceX will reuse the first stage?

Section 1

Methodology

Methodology

Executive Summary

- Data collection methodology:
  - SpaceX Launch data from an API, specifically REST API
  - Web scraping related Wiki pages
- Perform data wrangling
  - Wrangling data using API and Booster, Launchpad, payload and core functions
  - Data stored as lists to create a dataset
  - Data filtered/sampled to remove Falcon 1 launches
  - Replace NULL values with averages
- Perform exploratory data analysis (EDA) using visualization and SQL
- Perform interactive visual analytics using Folium and Plotly Dash
- Perform predictive analysis using classification models
  - Preprocessing to standardize the data
  - Using Train_test_split to split the data into training and testing data
  - Using hyperparameter values to determine the model with the best accuracy: Logistics Regression, Support Vector Machines, Decision Tree Classifier and K-nearest neighbors
  - Output confusion matrix

Data Collection

- Data collection done with helper functions BoosterVersion, LaunchSite, PayloadData and Core
- Requested rocket launch data from SpaceX API
- Used json_normalize to convert to a dataframe
- Extracted subset of the dataframe
- Removed multiples
- Converted date
- Restricted dataframe to date

Data Collection – SpaceX API

- Requested SpaceX API, requested JSON and normalized the content to create a dataframe

- GitHub URL: [Data-Science-Cert/jupyter-labs-spacex-data-collection-api.ipynb at Capstone · jswansin/Data-Science-Cert]()

Now let's start requesting rocket launch data from SpaceX API with the following URL:

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

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

Check the content of the response

```
print(response.content)
```

b'[{"fairings":{"reused":false,"recovery_attempt":false,"recovered":false,"ships":[]},"links":{"patch":{"small":"https://images2.imgbox.com/94/f2/NN6Ph45r_o.png","large":"https://images2.imgbox.com/5b/02/QcxHUb5V_o.png"},"reddit":{"campaign":null,"la

To make the requested JSON results more consistent, we will use the following static response object for this project:

```
static_json_url='https://cf-courses-data.s3.us.cloud-object-storage.appdomain.cloud/IBM-DS0321EN-SkillsNetwork/datasets/API_
```

We should see that the request was successfull with the 200 status response code

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

```
response.status_code
```

200

Now we decode the response content as a Json using `.json()` and turn it into a Pandas dataframe using `.json_normalize()`

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

Data Collection - Scraping

- Applied web scrapping for Falcon 9 launch records and parsed the table before converting it to a dataframe

- GitHub URL: [Data-Science-Cert/jupyter-labs-webscraping.ipynb at Capstone · jswansin/Data-Science-Cert]()

First, let's perform an HTTP GET method to request the Falcon9 Launch HTML page, as an HTTP response.

```
# use requests.get() method with the provided static_url
# assign the response to a object
html_data = requests.get(static_url)
html_data.status_code
```

```
200
```

Create a `BeautifulSoup` object from the HTML `response`

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

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

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

```
<title>List of Falcon 9 and Falcon Heavy launches - Wikipedia</title>
```

TASK 2: Extract all column/variable names from the HTML table header

Next, we want to collect all relevant column names from the HTML table header

Let's try to find all tables on the wiki page first. If you need to refresh your memory about `BeautifulSoup`, please check the external reference link towards the end of this lab

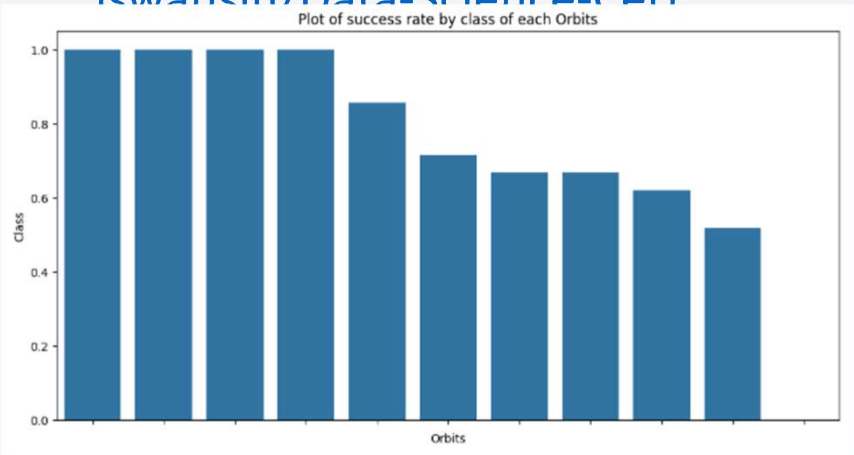
```
# Use the find_all function in the BeautifulSoup object, with element type `table`
# Assign the result to a list called `html_tables`
html_tables = soup.find_all('table')
```

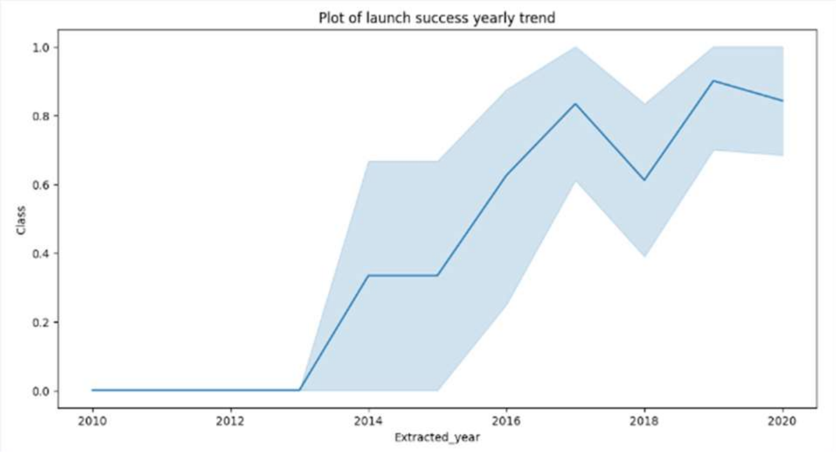
Data Wrangling

- Calculated number of launches at each site
- Calculated the number, occurrence and outcome of each orbit
- Created a label for landing outcome
- GitHub UR: Data-Science-Cert/labs-jupyter-spacex-Data wrangling.ipynb at Capstone · jswansin/Data-Science-Cert

EDA with Data Visualization

- Relationship between success rate of each orbit – valuable to end prediction goals
- Plot of launch success yearly trend – again, valuable to end prediction goals
- GitHub UR: [Data-Science-Cert/jupyter-labs-visualization.ipynb at Capstone · iswansin/Data-Science-Cert]





EDA with SQL

- The following data insights were retrieved:
  - *Displayed names of launch sites*
  - *Displayed 5 records with CCA launch sites*
  - *Displayed total payload mass for NSA (CRS) launches*
  - *Displayed average payload mass for F9 v1.1 boosters*
  - *Determined date of first successful landing on ground pad*
  - *Listed the names of successful boosters between 4000 and 6000 payload mass*
  - *Total number of successful and failed missions*
  - *Listed the names of boosters with maximum payload*
  - *Listed failed landings by booster version for 2015*
  - *Ranked the count of landing outcomes between given date range*

- GitHub URL: Data-Science-Cert/jupyter-labs-eda-sql-coursera_sqllite.ipynb at Capstone · jswansin/Data-Science-Cert

Build an Interactive Map with Folium

- Marked all launch sites on map
- Marked the success/failed launches for each site
- Calculated the distances between a launch site to its proximities (such as railways, highways or coastlines)
- GitHub UR: Data-Science-Cert/lab_jupyter_launch_site_location.ipynb at Capstone · jswansin/Data-Science-Cert

Build a Dashboard with Plotly Dash

- Interactive dashboard was built with Plotly dash including:
  - Pie charts showing the total launches by site
  - Scatter graphs showing relationship between Outcome and Payload Mass for selected booster
- GitHub URL:Data-Science-Cert/spacex_dash_app.py at Capstone · jswansin/Data-Science-Cert

Predictive Analysis (Classification)

- After creating a columns for class, the data was standardized
- Data was split into training data and test data
- Determined best hyperparameter for SVM, classification trees and logistic regression
- Found the best method based on accuracy
- GitHub URL: Data-Science-Cert/SpaceX_Machine Learning Prediction_Part_5.ipynb at Capstone · jswansin/Data-Science-Cert

Results

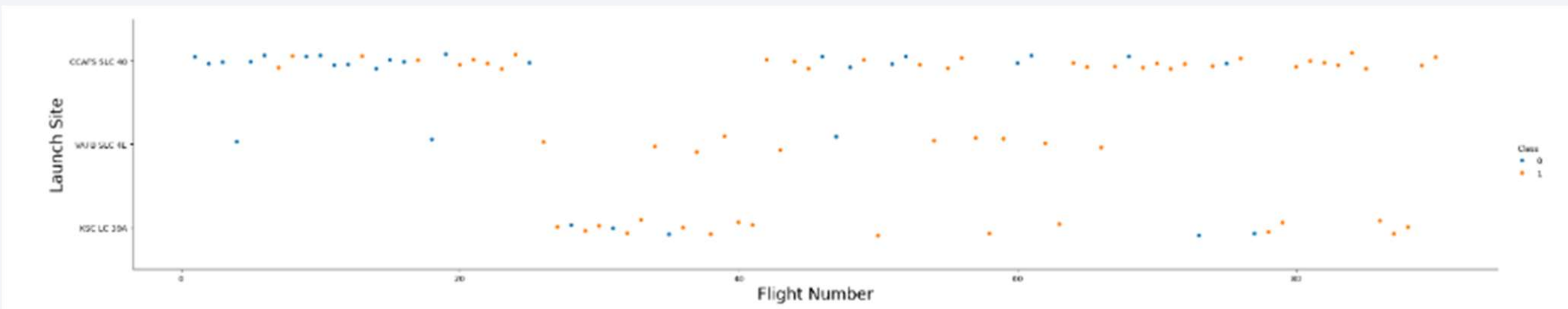
- Exploratory data analysis results
- Interactive analytics demo in screenshots
- Predictive analysis results

Section 2

Insights drawn from EDA

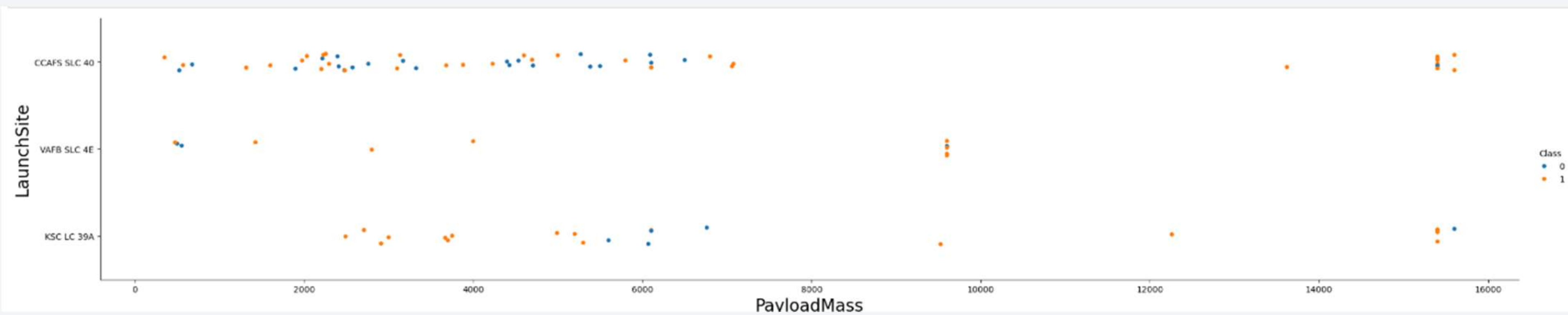
Flight Number vs. Launch Site

The more flights launched at the site is proportional to the success rate at the site.



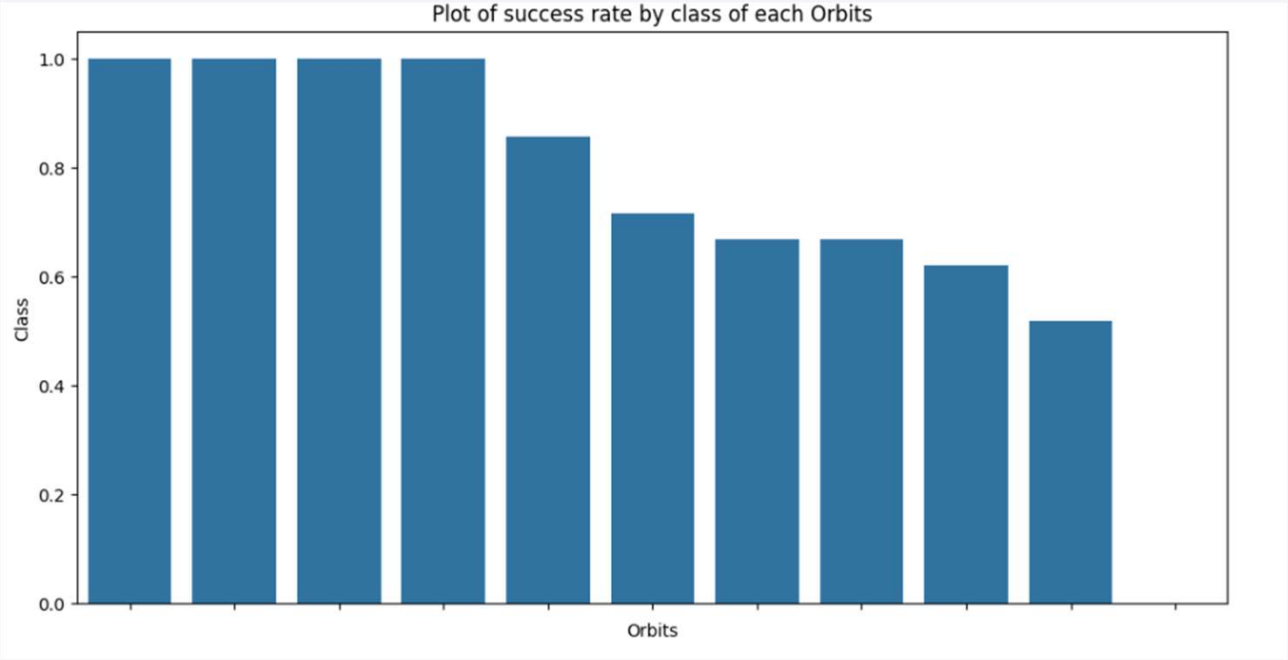
Payload vs. Launch Site

The higher the payload mass at CCAFS SLC 40, the higher the success rate.



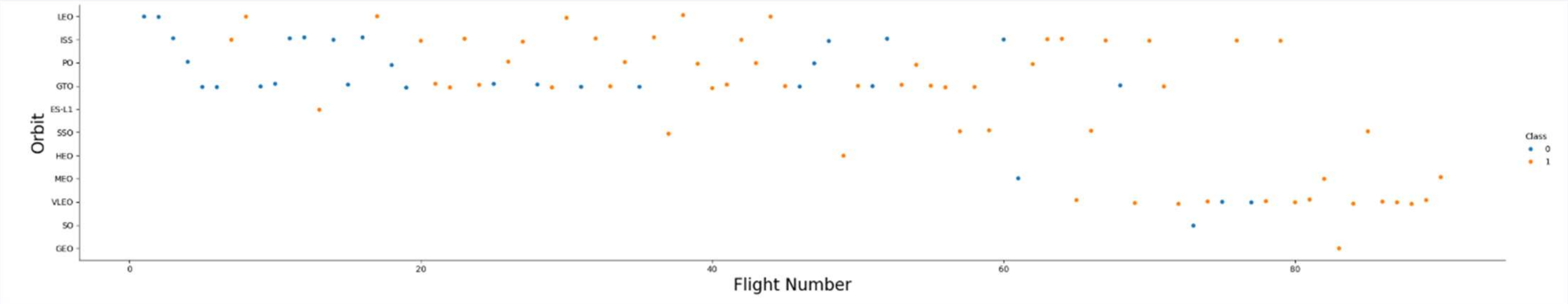
Success Rate vs. Orbit Type

Certain orbits had a higher success rate.



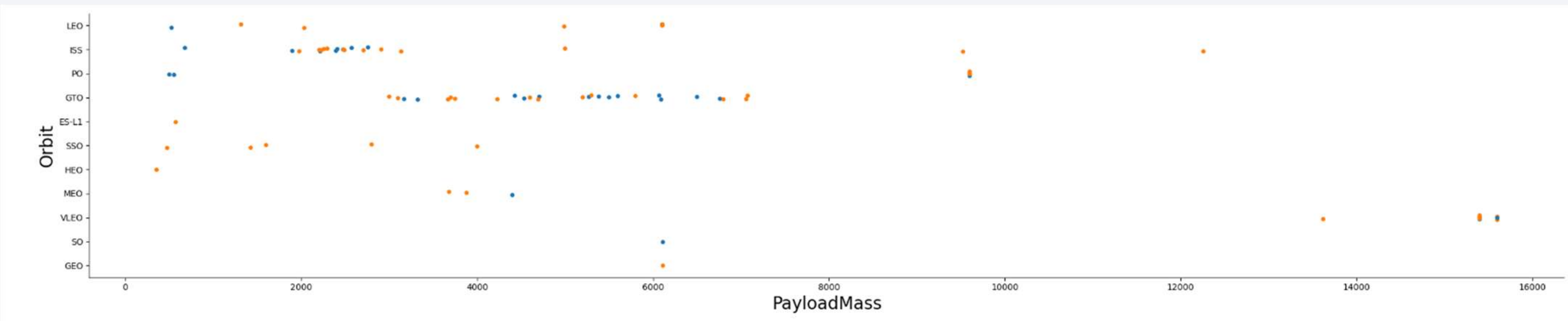
Flight Number vs. Orbit Type

In higher orbit, increasing the number of flights increases the success rate. This is not necessarily true at lower orbit.



Payload vs. Orbit Type

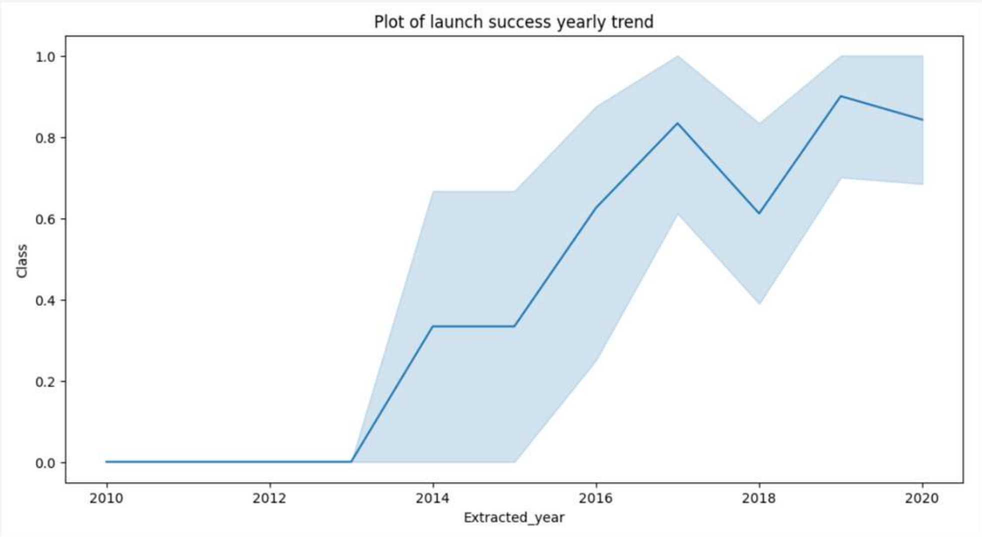
There is a higher success rate for higher payloads in PO, LEO and ISS orbits.



Launch Success Yearly Trend

In general, success rate increases each year.

However, there were declines in 2018 and 2020.



All Launch Site Names

DISTINCT command used in sql used to extract unique launch sites.

```
%sql SELECT DISTINCT "Launch_Site" FROM SPACEXTABLE
```

```
 * 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'

WHERE, LIKE and LIMIT used to extract 5 lines starting with CCA as Launch Site.

```
%sql SELECT * FROM SPACEXTABLE 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

Total Payload Mass

Current approach inaccurate as reported total was zero.

```
%sql SELECT SUM("PAYLOAD_MASS_KG_") AS Total_Payload_Mass FROM SPACEXTABLE WHERE "Customer" LIKE 'NASA%(CRS)%'
```

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

Total_Payload_Mass
0.0

Average Payload Mass by F9 v1.1

Calculated the average payload mass carried by booster version F9 v1.1 to be 2928 kg using AVG, FROM and WHERE.

```
%sql SELECT AVG(PAYLOAD_MASS__KG_) FROM SPACEXTABLE WHERE BOOSTER_VERSION = 'F9 v1.1'

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

AVG(PAYLOAD_MASS__KG_)
2928.4

First Successful Ground Landing Date

First successful landing outcome on ground pad WAS 12/22/2015. Result was found using mid(DATE) and WHERE.

```
%sql select min(DATE) from SPACEXTABLE where Landing_Outcome = 'Success (ground pad)';
```

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

min(DATE)
2015-12-22

Successful Drone Ship Landing with Payload between 4000 and 6000

WHERE and AND commands used with ‘Success (drone ship)’ and Payload mass constraints.

```
%sql SELECT BOOSTER_VERSION from SPACEXTABLE WHERE LANDING_OUTCOME = 'Success (drone ship)' and  PAYLOAD_MASS__KG_  >4000 ar
```

```
 * 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

COUNT, WHERE and OR used to calculate number of mission outcomes.

```
%sql select count(MISSION_OUTCOME) from SPACEXTABLE where MISSION_OUTCOME = 'Success' or MISSION_OUTCOME = 'Failure (in fli
```

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

count(MISSION_OUTCOME)
99

Boosters Carried Maximum Payload

WHERE and MAX used to extract booster names that carried the maximum payload

```
%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

2015 Launch Records

WHERE and AND used to extract multiple columns.

```
%sql SELECT BOOSTER_VERSION,LAUNCH_SITE,LANDING_OUTCOME FROM SPACEXTABLE WHERE LANDING_OUTCOME = 'Failure (drone ship)' and
```

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

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

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

WHERE, AND, and ORDER used to present results.

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.

```
%sql select * from SPACEXTABLE where Landing_Outcome = 'Success (ground pad)' and (DATE between '2010-06-04' and '2017-03-2
```

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

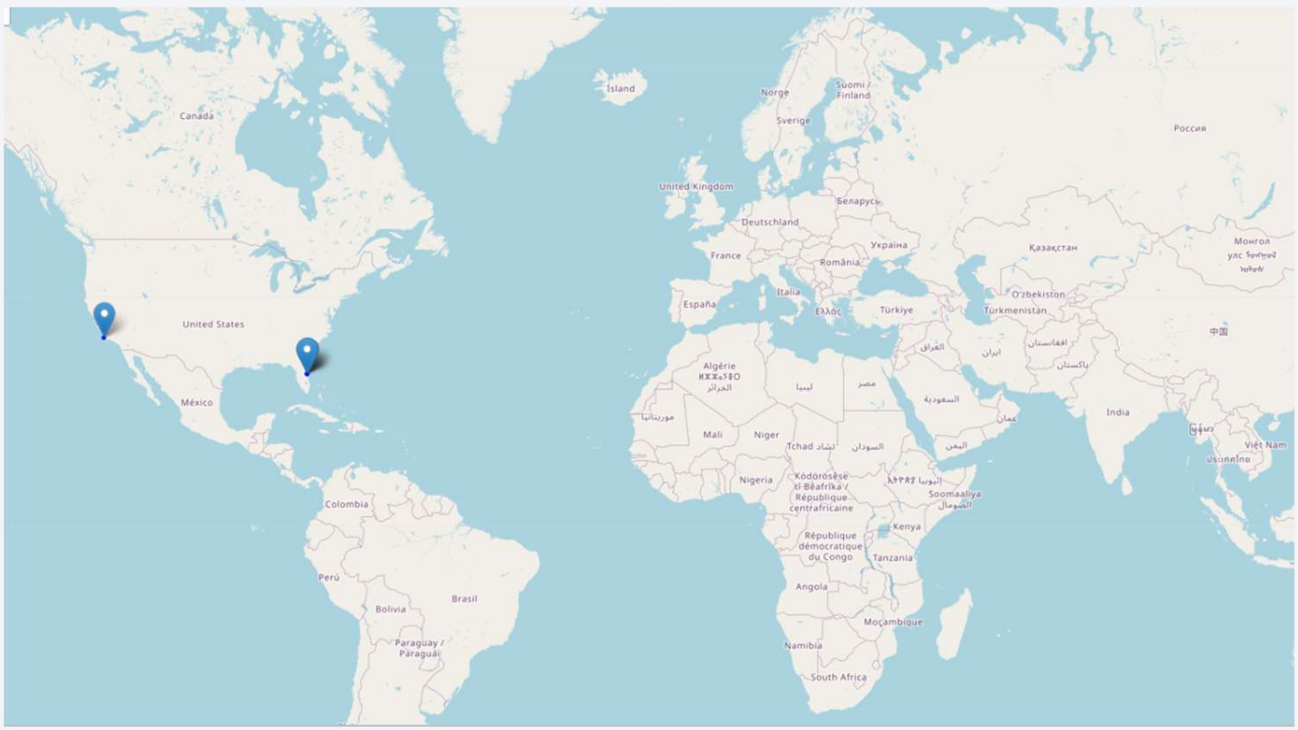
Date	Time (UTC)	Booster_Version	Launch_Site	Payload	PAYLOAD_MASS__KG_	Orbit	Customer	Mission_Outcome	Landing_Outcome
2017-02-19	14:39:00	F9 FT B1031.1	KSC LC-39A	SpaceX CRS-10	2490	LEO (ISS)	NASA (CRS)	Success	Success (ground pad)
2016-07-18	4:45:00	F9 FT B1025.1	CCAFS LC-40	SpaceX CRS-9	2257	LEO (ISS)	NASA (CRS)	Success	Success (ground pad)
2015-12-22	1:29:00	F9 FT B1019	CCAFS LC-40	OG2 Mission 2 11 Orbcomm-OG2 satellites	2034	LEO	Orbcomm	Success	Success (ground pad)

Section 3

Launch Sites Proximities Analysis

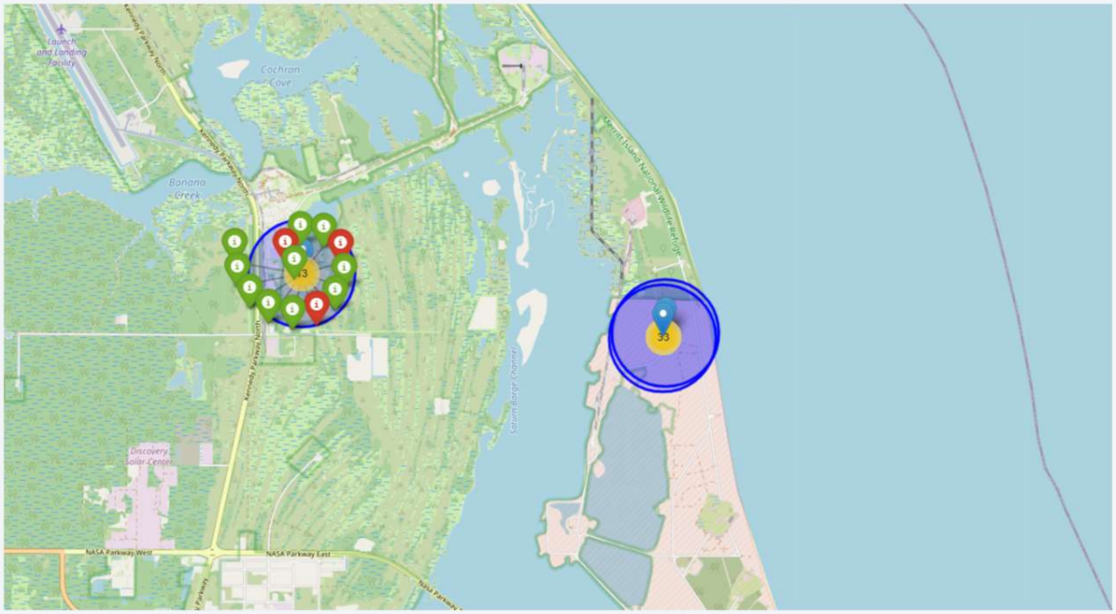
Launch Site Locations on Global Map

Launch locations are Florida and California in the USA.



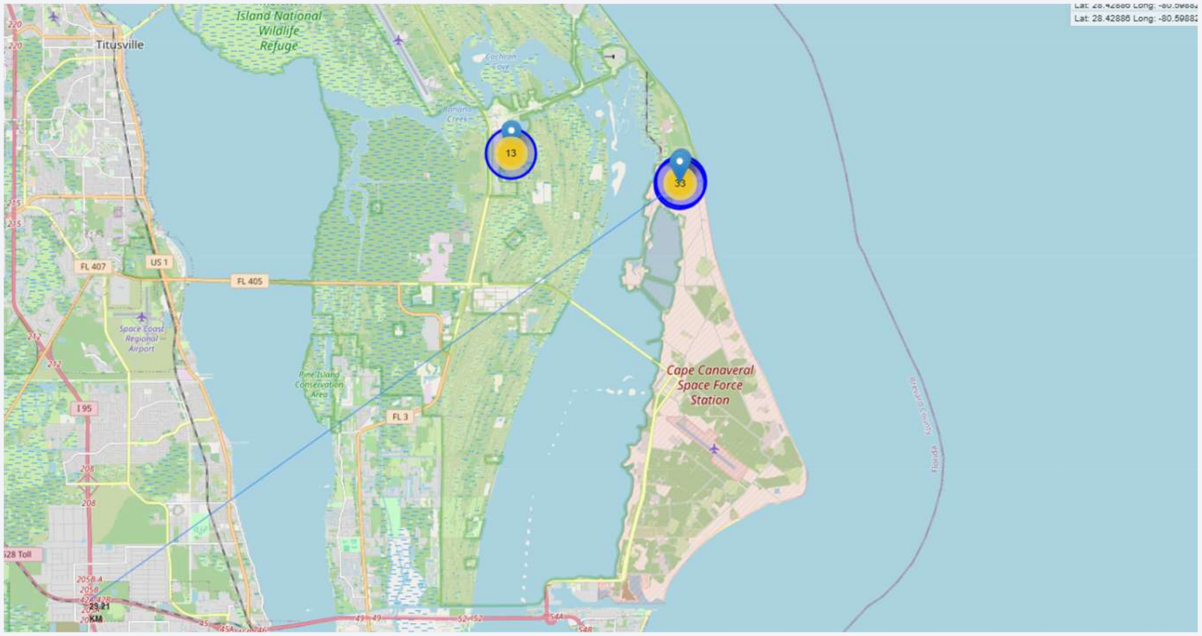
Colo-Labeled Launch Outcomes

Map indicates number of successful (green) launches and failed (red) launches.



Launch Site Proximities

Map shows selected launch site to its proximities such as railway, highway, coastline, with distance calculated and displayed.

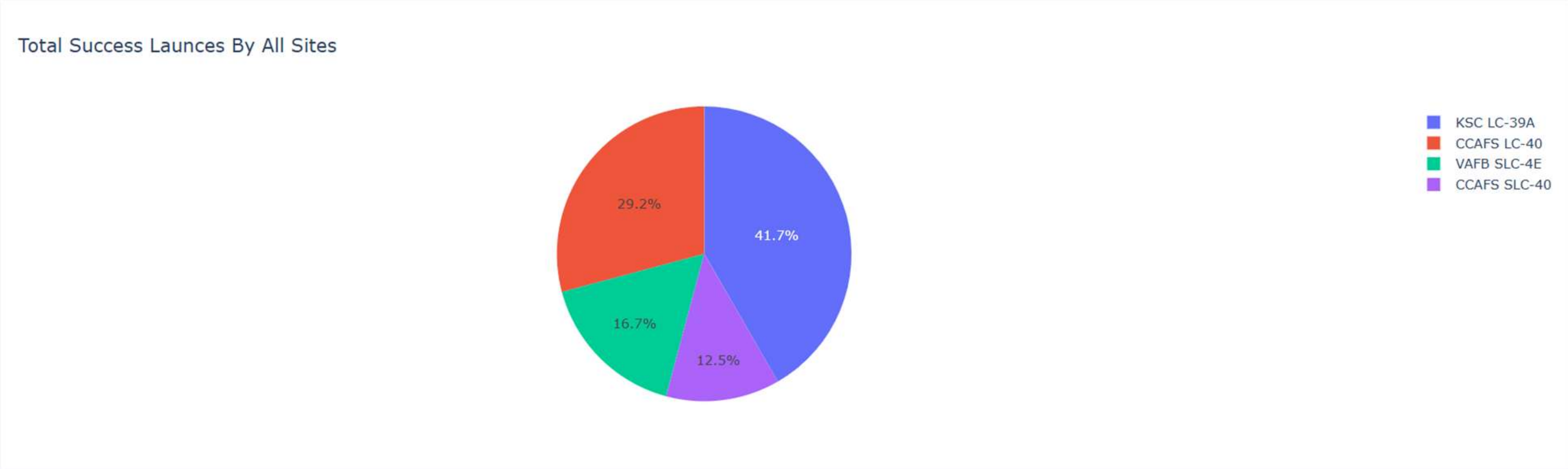


Section 4

Build a Dashboard with Plotly Dash

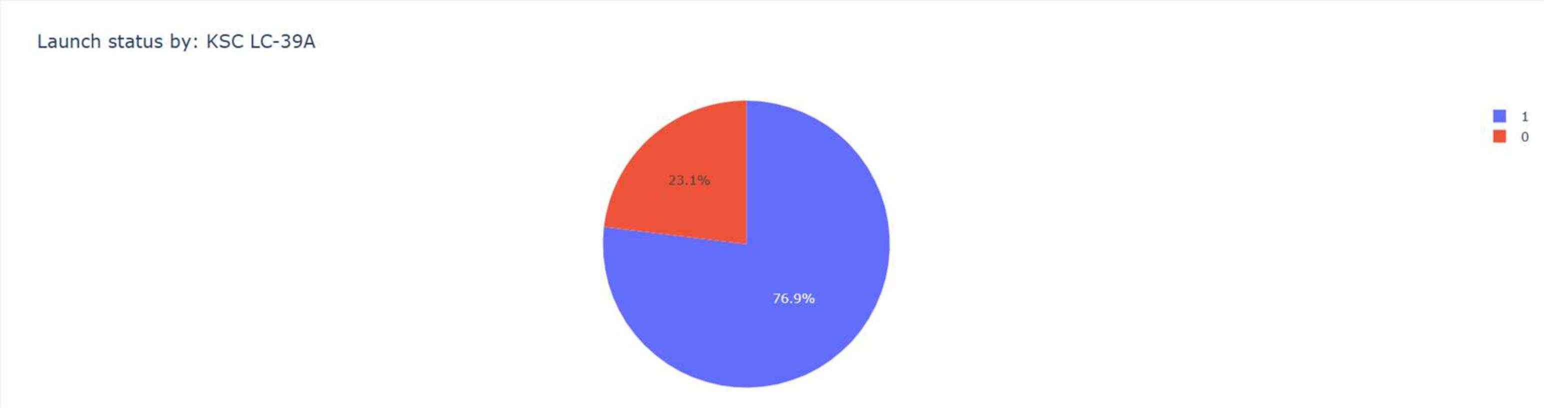
Launch Success for All Sites

KSC LC-39A had most successful launches.



Launch Success for KSC LC-39A

Launch site KSC LC-39A had a 76.9% success rate.



Payload v Launch Outcome (All Sites)

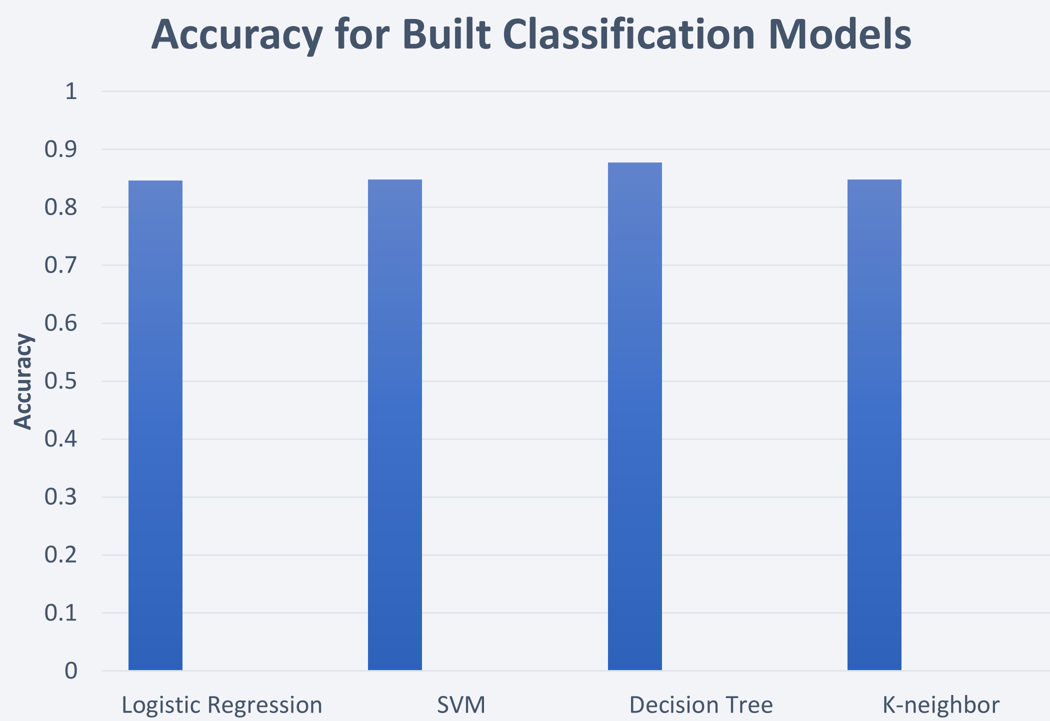
Scatter plot for all sites, with different payload selected in the range slider. 4000 kg an above selected for image below.

Section 5

Predictive Analysis (Classification)

Classification Accuracy

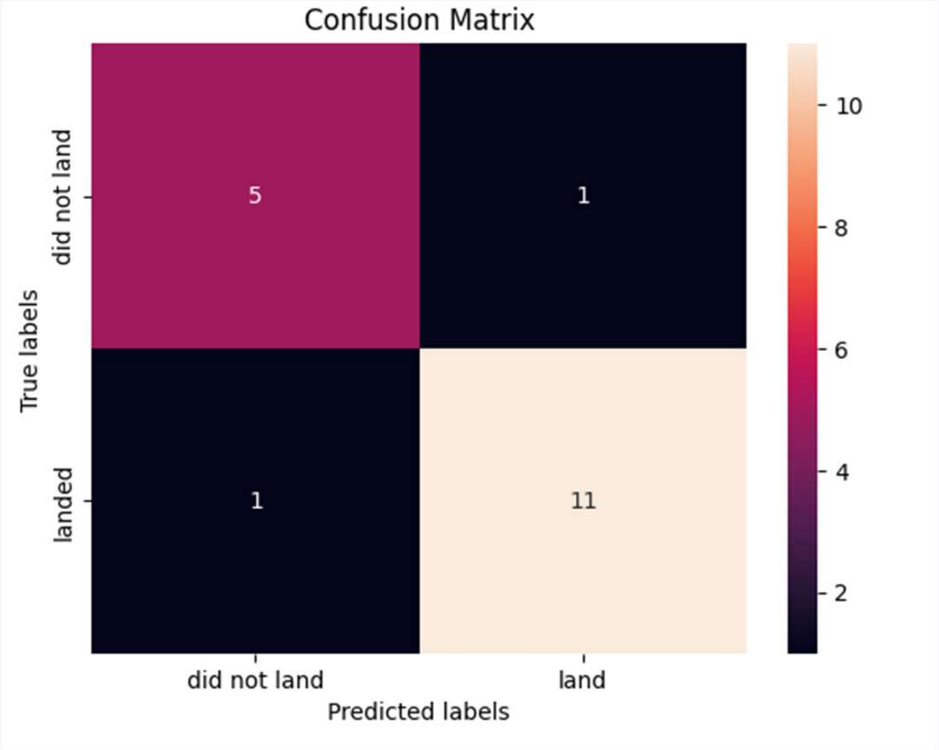
Decision tree model shows highest accuracy when compared to other models.



Confusion Matrix

Confusion tree matrix for decision tree model. It correctly predicted a landing when it landed 11 times and a failure when it did not land 5 times.

This is higher than any of the other 3 models.



Conclusions

- More flights launched at the site is proportional to the success rate at the site.
- In general, success rate increases each year.
- KSC LC-39A launch site had most successful launches.
- Decision tree classifier is the most accurate model for this study.

Thank you!