IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

Michael Korn
January 15, 2024

Outline

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

Executive Summary

- Summary of methodologies
  - Collection, wrangling, explored, visualized, predicted
- Summary of all results
  - We were able to predict landing outcomes with a 94% success rate

Introduction

A SpaceX Falcon 9 rocket launches with a cost of $62m compared to $165m for other providers. Reusable rockets allow SpaceX launches to be over 60% cheaper. The goal of this research is to predict whether a particular launch will be successful. By predicting successful launches, we can ideally improve success rate to decrease launch cost. In addition, a party bidding on a launch can be informed when making a bid.

Section
1
Methodology

Methodology

Executive Summary

- Data collection methodology:
  - API, Scraping
- Perform data wrangling
  - Describe how data was processed
- 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

Methodologies

- API – Github Link
- Webscraping – Github Link

Data Collection – SpaceX API

- Data collection with SpaceX REST calls

boosterversion	launchpad	payload	cores

```
spacex_url="https://api.spacexdata.com/v4/launches/past"
response = requests.get(spacex_url)
response.content
```

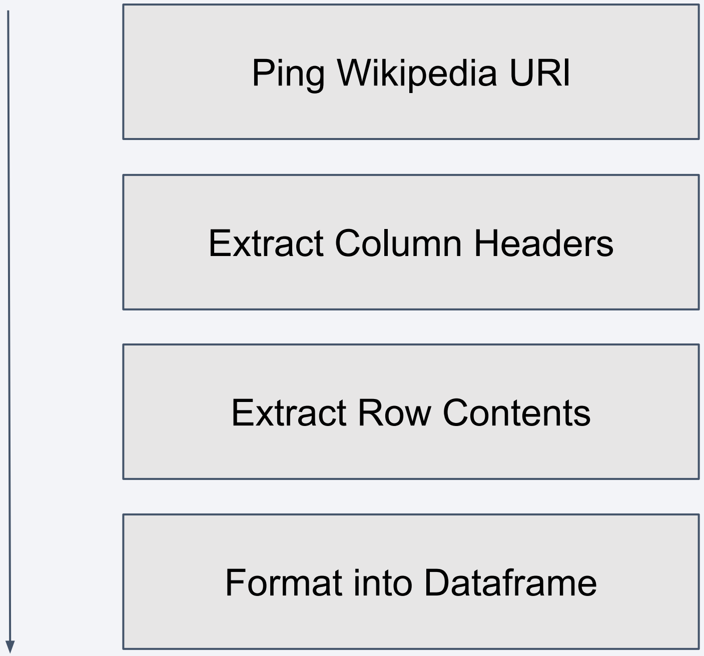
Convert to df

Clean, prepare, load

```
static_json_url='https://cf-courses-data.s3.us.cloud-object-storage.appdomain.cloud/IBM-DS0321EN-SkillsNetwork/datasets/API_call_spacex_api.json'
response = requests.get(static_json_url)
response.status_code
response = response.json()

# Lets take a subset of our dataframe keeping only the features we want and the flight number, and date_utc.
data = data[['rocket', 'payloads', 'launchpad', 'cores', 'flight_number', 'date_utc']]

# We will remove rows with multiple cores because those are falcon rockets with 2 extra rocket boosters and rows that have multiple payloads in a single rocket.
data = data[data['cores'].map(len)==1]
data = data[data['payloads'].map(len)==1]

# Since payloads and cores are lists of size 1 we will also extract the single value in the list and replace the feature.
data['cores'] = data['cores'].map(lambda x : x[0])
data['payloads'] = data['payloads'].map(lambda x : x[0])

# We also want to convert the date_utc to a datetime datatype and then extracting the date leaving the time
data['date'] = pd.to_datetime(data['date_utc']).dt.date

# Using the date we will restrict the dates of the launches
data = data[data['date'] <= datetime.date(2020, 11, 13)]
```

Data Collection - Scraping

- Web scraping process



Data Wrangling

- Data massaging was continuously necessary during collection
- After table creation, wrangling is necessary to prepare data for analysis. This includes creating policies for handling missing data.
- Elements:
    - Labels column
    - Replacing Nan with mean

Github Link

Data Wrangling – Creating Labels Column

```
for i,outcome in enumerate(landing_outcomes.keys()):
    print(i,outcome)
```
✓ 0.0s

```
0 True ASDS
1 None None
2 True RTLS
3 False ASDS
4 True Ocean
5 False Ocean
6 None ASDS
7 False RTLS
```

```
bad_outcomes=set(landing_outcomes.keys()[[1,3,5,6,7]])
bad_outcomes
```
✓ 0.0s

```
{'False ASDS', 'False Ocean', 'False RTLS', 'None ASDS', 'None None'}
```

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

df['Class']=landing_class
```
✓ 0.0s

Data Wrangling – Replacing Nan with Mean

```
# Calculate the mean value of PayloadMass column
df9_mean = data_falcon9.PayloadMass.mean()

# Replace the np.nan values with its mean value
data_falcon9.PayloadMass.replace(np.nan, df9_mean)
```

[33] ✓ 0.0s

```
4      6123.547647
5       525.000000
6       677.000000
7       500.000000
8      3170.000000
           ...
89    15600.000000
90    15600.000000
91    15600.000000
92    15600.000000
93     3681.000000
Name: PayloadMass, Length: 90, dtype: float64
```

EDA with Data Visualization

- Flight Number vs Payload Mass
- Flight Number vs Launch Site
- Payload Mass vs Launch Site
- Success Rate vs Orbit Type
- Flight Number vs Orbit Type
- Payload Mass vs Orbit Type
- Launch Success vs Yearly Trend
- Features Engineering

[Github Link]

EDA with SQL

- Distinct launch sites
- Total payload mass carried by NASA boosters
- Average payload mass carried by booster version F9 v1.1
- Date when first successful landing outcome in ground pad was achieved
- Names of the boosters which have success in drone ship and have payload mass greater than 4000 but less than 6000
- Total number of successful and failure mission outcomes
- Names of the booster_versions which have carried the maximum payload mass
- Records which will display the month names, failure landing_outcomes in drone ship ,booster versions, launch_site for the months in year 2015
- Ranked count of landing outcomes (such as Failure (drone ship) or Success (ground pad)) between the date 2010-06-04 and 2017-03-20.)

[Github Link]()

Build an Interactive Map with Folium

- The launch success rate may depend on many factors such as payload mass, orbit type, and so on. It may also depend on the location and proximities of a launch site, i.e., the initial position of rocket trajectories. Finding an optimal location for building a launch site certainly involves many factors and hopefully we could discover some of the factors by analyzing the existing launch site locations.
    - Maps of launch sites
    - Distance between launch sites and POI

Github Link

Build a Dashboard with Plotly Dash

- Success rates by launch site
    - Launch site is a low-hanging fruit in terms of outcome relationship. A launch site contains a large number of variables that may be unaccounted for in the dataset: from staff to weather conditions.
- Launches, outcomes by payload mass and booster version
    - As the goal is to send massive payloads to Mars, understanding success rates as payload increases and as boosters upgrade is essential.
- Success rate by booster version

Github Link

Predictive Analysis (Classification)

- Tested multiple classification models
    - Logistic regression
    - Support vector machine
    - Decision tree
    - KNN
- Method:
    - Chose parameters
    - Choose a model
    - Hyperparameter tuning
    - Fitting/tuning
    - Evaluating by viewing scores and confusion matrix
- You need present your model development process using key phrases and flowchart

Github Link

Results

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

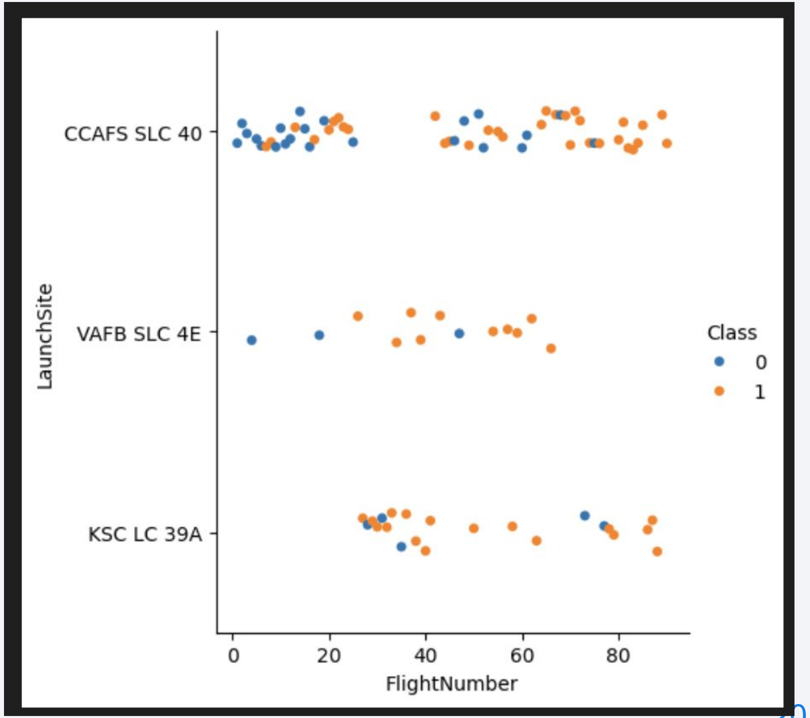
Section
2

Insights drawn from EDA

Flight Number vs. Launch Site

<u>Flight Number vs. Launch Site</u>

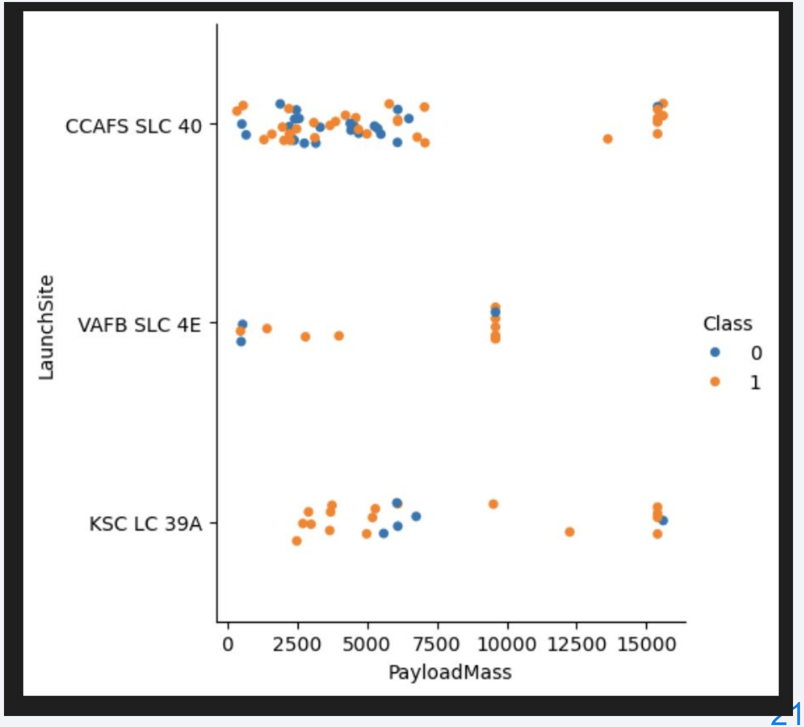
- CCAFS SLC 40 appears to have the highest volum of launches.
- VAFB seems to have slowed
- KSC seems to be a newer site



Payload vs. Launch Site

Payload vs. Launch Site

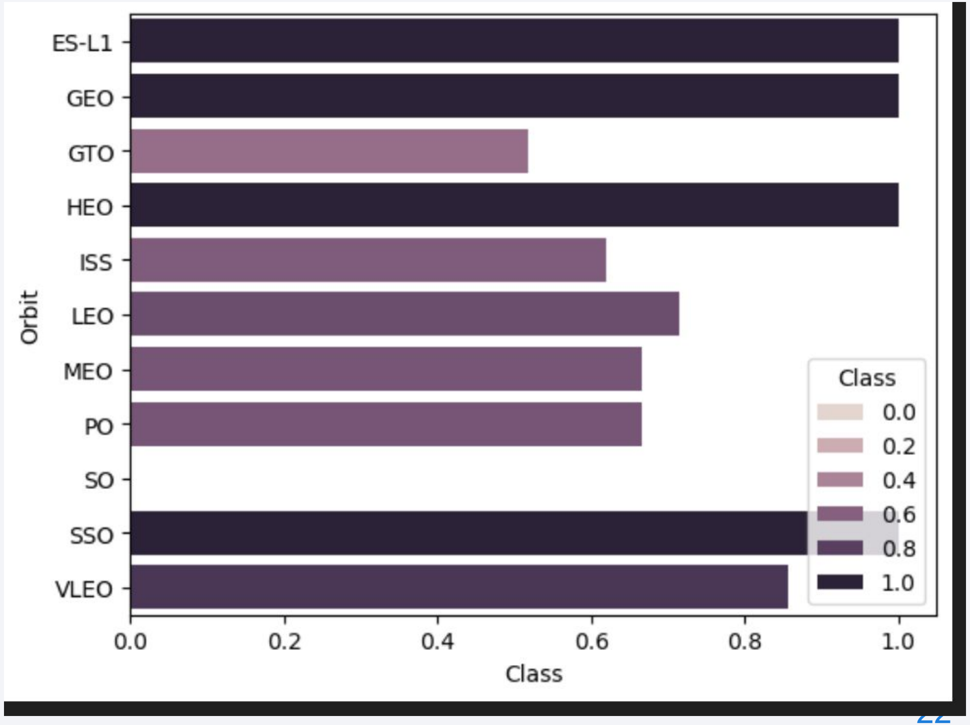
- Largest payloads are with CCAFS and KSC
- A cluster of failures appear between 5,000 and 7,500 kg payloads at KSC
- Early low payload launches at CCAFS had mixed success



Success Rate vs. Orbit Type

<u>Success rate of each orbit type</u>

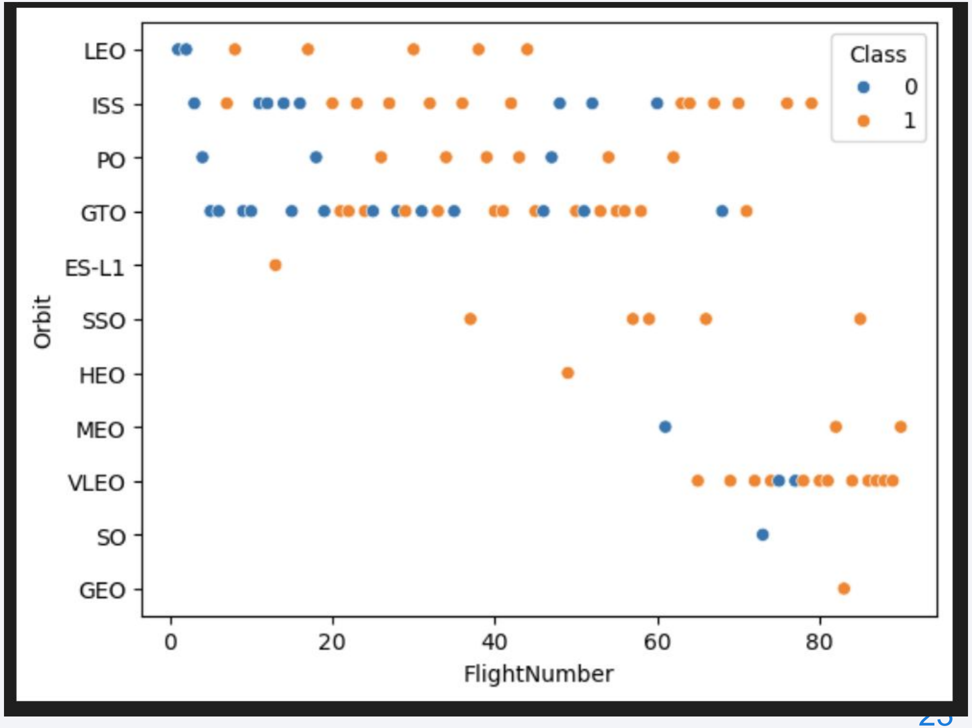
- Darker hues have higher success rates
    - ES-L1
    - GEO
    - HEO
    - SSO



Flight Number vs. Orbit Type

<u>Flight number vs. Orbit type</u>

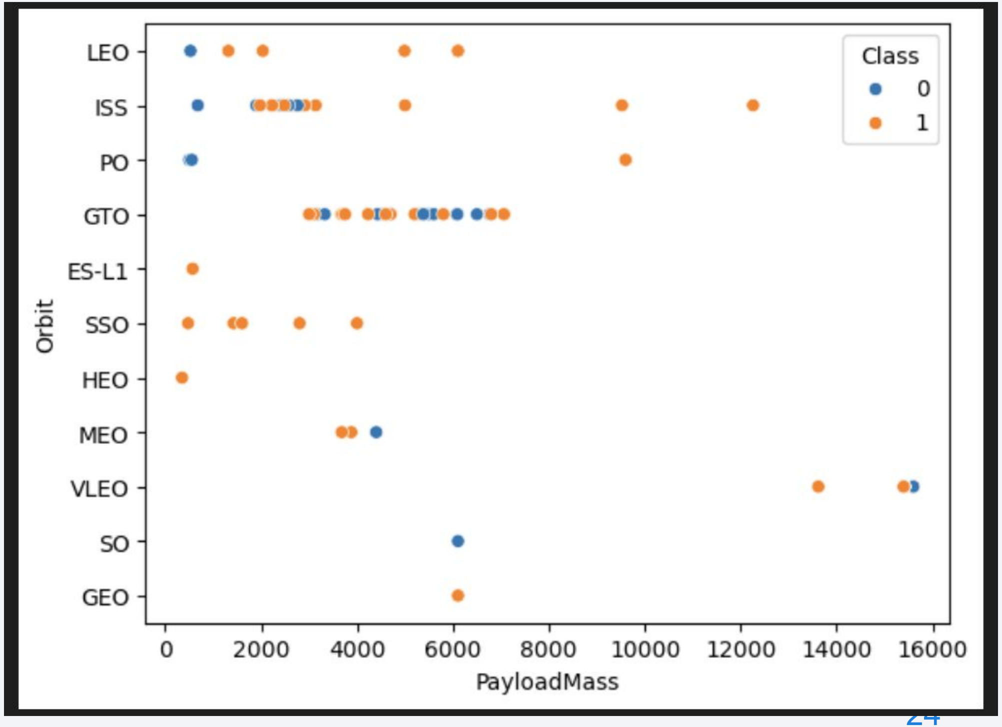
- VLEO and ISS seem to have the latest success



Payload vs. Orbit Type

<u>Payload vs. orbit type</u>

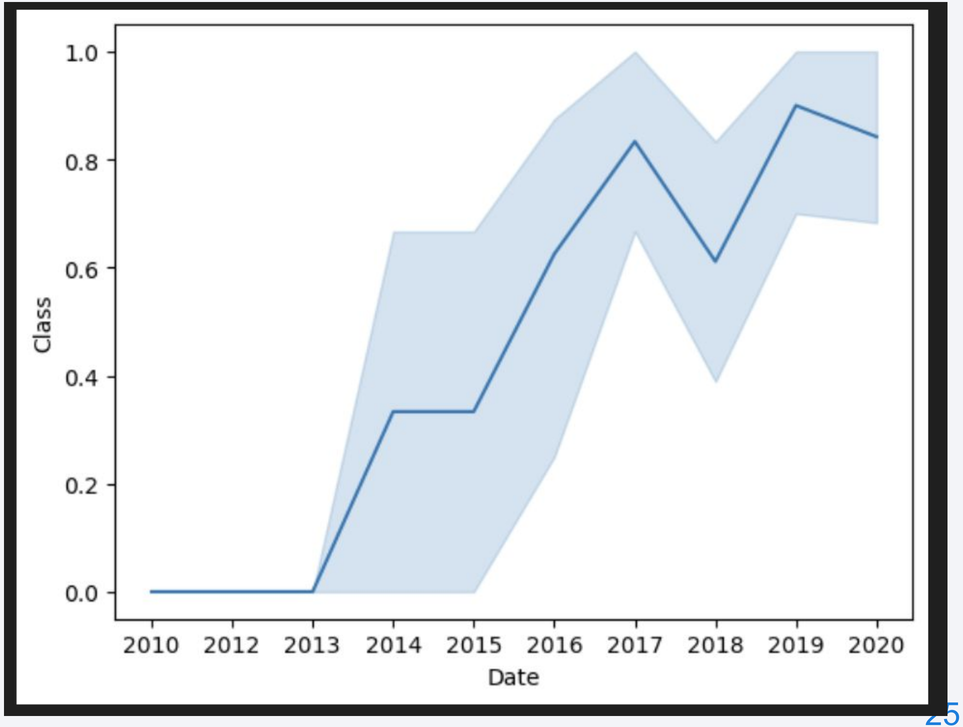
- ISS, PO, VLEO are the only orbits with successful high payload mass launches
- GTO is hard to distinguish



Launch Success Yearly Trend

<u>Yearly average success rate</u>

- The success rate has clearly improved



All Launch Site Names

```
%sql select distinct launch_site from spacextable
```

✓ 0.0s

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

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

Date	Time (UTC)	Booster_Version	Launch_Site	Payload	P
2010-06-04	18:45:00	F9 v1.0 B0003	CCAFS LC-40	Dragon Spacecraft Qualification Unit	
2010-12-08	15:43:00	F9 v1.0 B0004	CCAFS LC-40	Dragon demo flight C1, two CubeSats, barrel of Brouere cheese	
2012-05-22	7:44:00	F9 v1.0 B0005	CCAFS LC-40	Dragon demo flight C2	
2012-10-08	0:35:00	F9 v1.0 B0006	CCAFS LC-40	SpaceX CRS-1	
2013-03-01	15:10:00	F9 v1.0 B0007	CCAFS LC-40	SpaceX CRS-2	

Total Payload Mass

```
%sql select sum(PAYLOAD_MASS__KG_) from spacextable where customer = 'NASA (CRS)'
```

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

sum(PAYLOAD_MASS__KG_)
45596

Average Payload Mass by F9 v1.1

```
%sql select avg(PAYLOAD_MASS__KG_) from spacextable where Booster_Version='F9 v1.1'
✓ 0.0s

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

avg(PAYLOAD_MASS__KG_)
2928.4

First Successful Ground Landing Date

```
%sql select min(date) from spacextable where landing_outcome = 'Success (ground pad)'
```

✓ 0.0s

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

min(date)
2015-12-22

Successful Drone Ship Landing with Payload between 4000 and 6000

```
%sql select booster_version from spacextable where landing_outcome='Success (drone ship)' and PAYLOAD_MASS__KG_ between 4000 and 6000
```

✓ 0.0s

```
* 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

- The vast majority of missions were successful even though many did not successfully land.

```
%sql select count(*) as total_number, mission_outcome from spacextable group by mission_outcome
✓ 0.0s

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

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

Boosters Carried Maximum Payload

- Several booster versions have carried the maximum payload

```
%sql select booster_version, max(PAYLOAD_MASS__KG_) from spacextable group by booster_version order by PAYLOAD_MASS__KG_ desc
```

✓ 0.0s

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

Booster_Version	max(PAYLOAD_MASS__KG_)
F9 B5 B1060.3	15600
F9 B5 B1060.2	15600
F9 B5 B1058.3	15600
F9 B5 B1056.4	15600
F9 B5 B1051.6	15600
F9 B5 B1051.4	15600
F9 B5 B1051.3	15600
F9 B5 B1049.7	15600
F9 B5 B1049.5	15600
F9 B5 B1049.4	15600
F9 B5 B1048.5	15600
F9 B5 B1048.4	15600
F9 B5 B1049.6	15440

2015 Launch Records

```
%%sql
select
    CASE strftime('%m', date)
        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'
    END AS month_name
    , Landing_Outcome
    , booster_version
    , launch_site
    , substr(Date,0,5) as year
from spacextable where landing_outcome = 'Failure (drone ship)' and substr(Date,0,5)='2015'
```

✓ 0.0s

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

month_name	Landing_Outcome	Booster_Version	Launch_Site	year
January	Failure (drone ship)	F9 v1.1 B1012	CCAFS LC-40	2015
April	Failure (drone ship)	F9 v1.1 B1015	CCAFS LC-40	2015

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

```
%%sql
select date, count(landing_outcome), landing_outcome
from spacextable
where date >= '2010-06-04' and date<='2017-03-20'
group by landing_outcome
order by count(landing_outcome) desc
```

✓ 0.0s

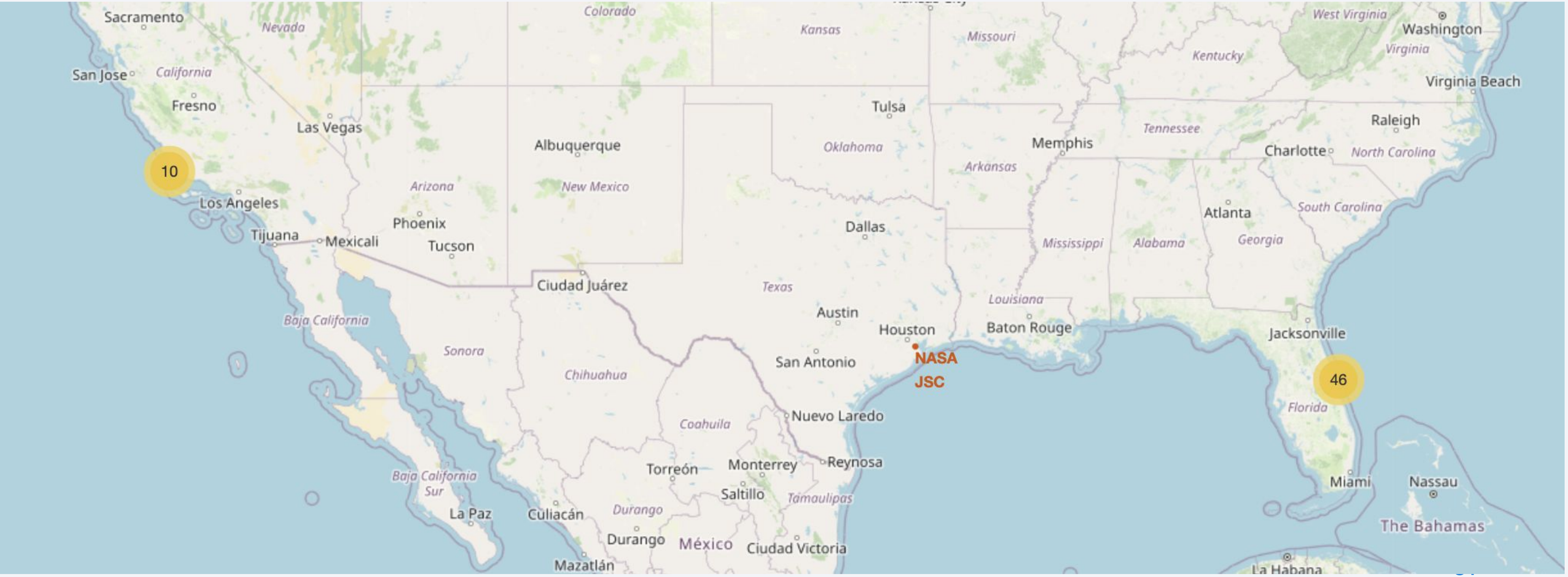
* sqlite:///my_data1.db
Done.

Date	count(landing_outcome)	Landing_Outcome
2012-05-22	10	No attempt
2016-04-08	5	Success (drone ship)
2015-01-10	5	Failure (drone ship)
2015-12-22	3	Success (ground pad)
2014-04-18	3	Controlled (ocean)
2013-09-29	2	Uncontrolled (ocean)
2010-06-04	2	Failure (parachute)
2015-06-28	1	Precluded (drone ship)

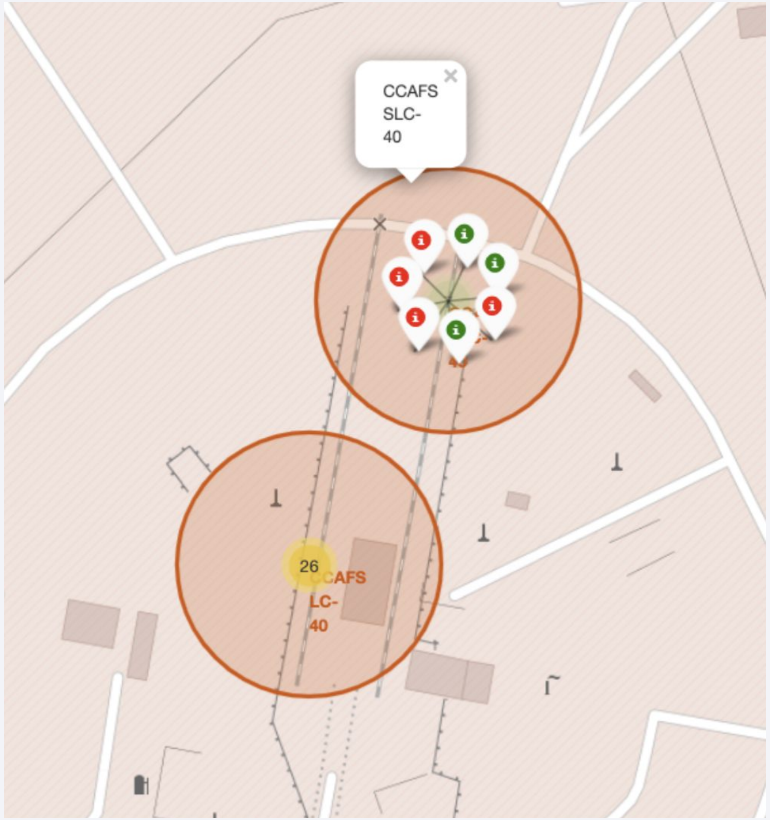
Section 3

Launch Sites Proximities Analysis

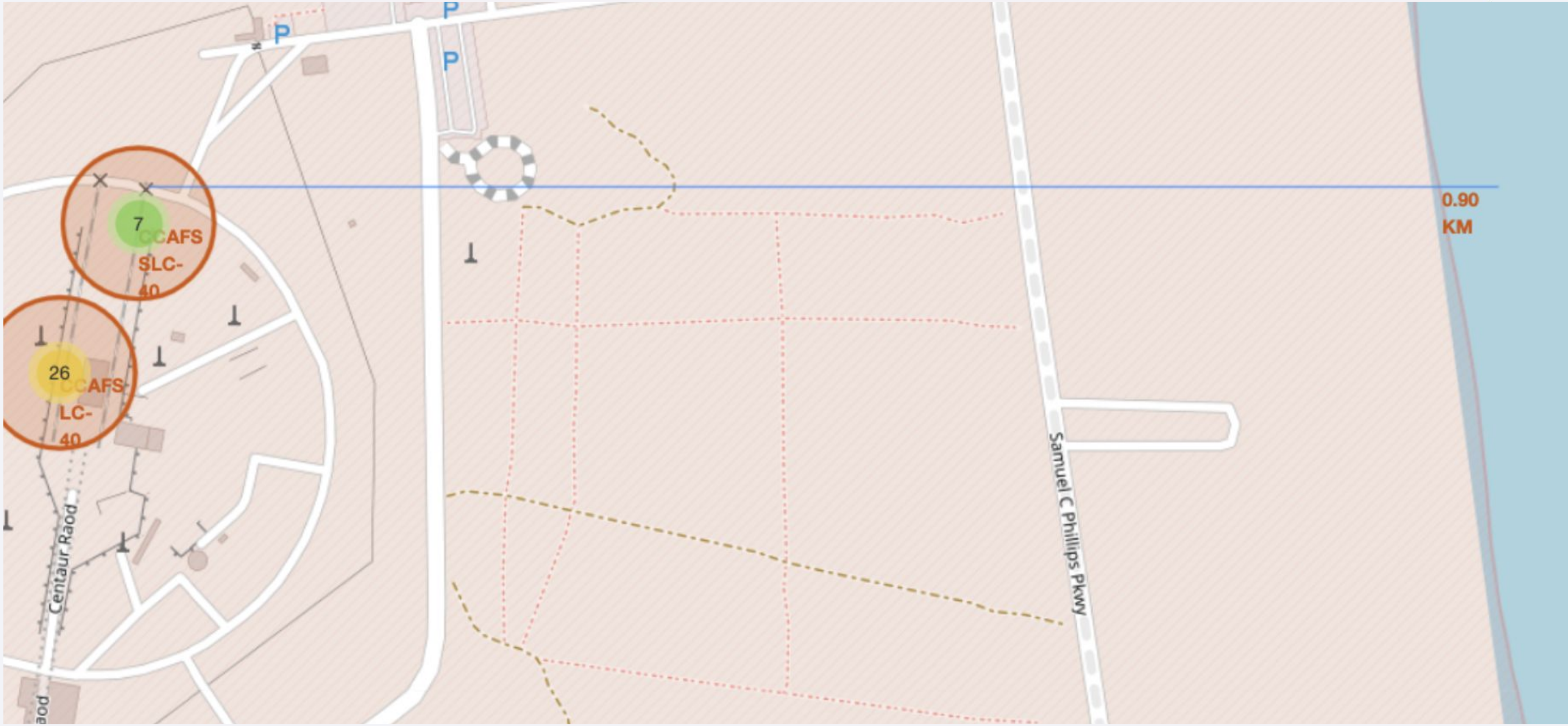
Map of Launch Sites



Color Coded Success Rate of Launch Sites



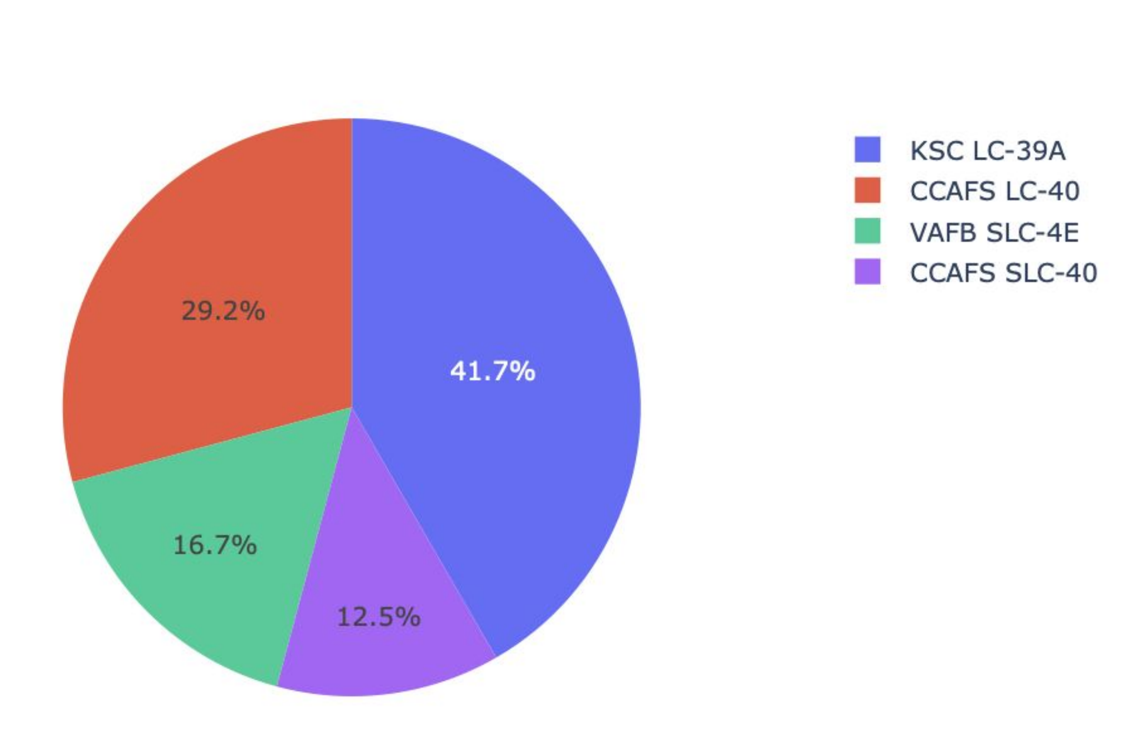
Launch Site Distance from POI



Section
4

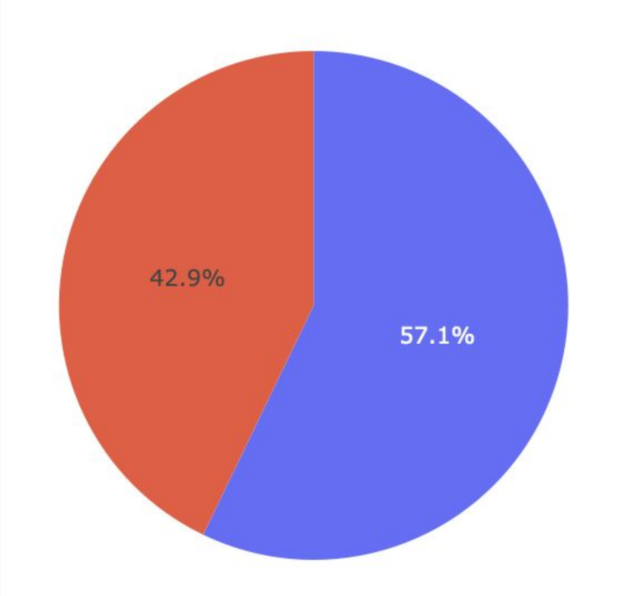
Build a Dashboard with Plotly Dash

Success Rate for All Launch Sites

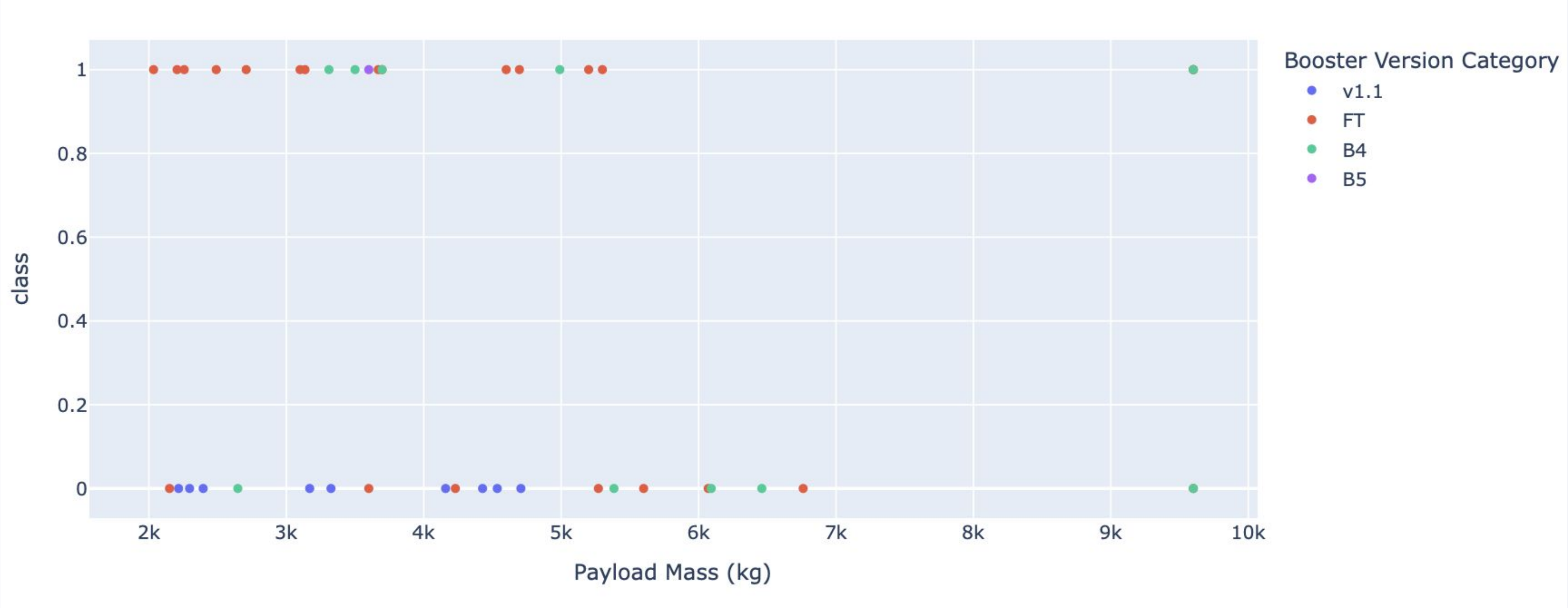


Highest Success Rate

- CCAFS SLC-40 @ 42.9% Success



Payload Mass vs Outcomes

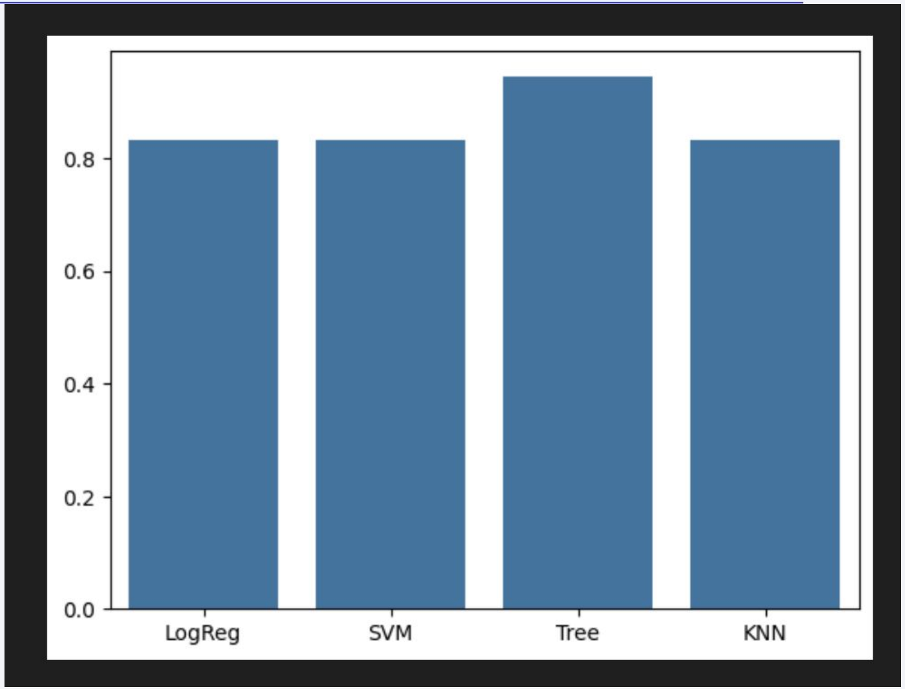


Section
5

Predictive Analysis (Classification)

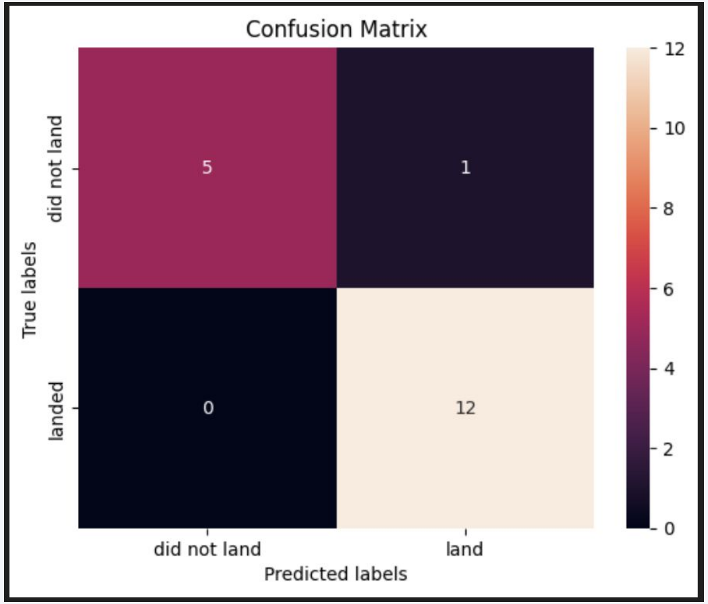
Classification Accuracy

The Decision Tree Classifier had the highest score, scoring .94 compared to .83.



Confusion Matrix

- The decision tree classifier correctly predicted 12 launches to land and 5 launches to not land.
- The model incorrectly predicted 1 launch to land that ultimately did not land.
- These results are fairly optimistic, but would like to see performance as more data becomes available.



Conclusions

- Many variables contribute to a successful launch and landing. The best predictions were able to predict 94% of landing outcomes successfully. This information is practical in pricing and bidding, designing successful launch/landing conditions, and improving unsuccessful launch/landing conditions.

Appendix

- Include any relevant assets like Python code snippets, SQL queries, charts, Notebook outputs, or data sets that you may have created during this project

Thank you!