IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

Amore Hui-Yi Huang, Dec. 2021

Outline

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

Executive Summary

- Summary of methodologies
  - Data collection and wrangling
  - EDA with data visualization and SQL
  - Interactive map with Folium
  - Dashboard with Plotly Dash
  - Predictive analysis (Classification)
- Summary of results
  - Exploratory data analysis results
  - Interactive analytics demo
  - Predictive model

Introduction

- **Project Purpose**
  - Predict if the Falcon 9 first stage will land successfully
- **Project background and context**
  - SpaceX advertises Falcon 9 rocket launches on its website with a cost of 62 million dollars; other providers cost upwards of 165 million dollars each, much of the savings is because SpaceX can reuse the first stage.
  - We can determine the cost of a launch through determine if the first stage will land
- **Problems you want to find answers**
  - What are the factors which influence the rocket landing successfully
  - The best classification models

01

Methodology

Methodology

- Executive Summary
  - Data collection methodology:
    - SpaceX REST API
    - Web Scraping from Wikipedia
- Perform data wrangling
- 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 Collection – SpaceX API

GitHub

1. Define functions to extract features
2. Request and parse the SpaceX launch data
3. Decode the response content as a json
4. Normalize json that convert it into dataframe
5. Data Clean and export to .csv

	static_fire_date_utc	static_fire_date_unix	net	window	rocket	success	failures	details	crew	ships	capsules	pa
0	2006-03-17T00:00:00.000Z	1.142554e+09	False	0.0	5e9d0d95eda69955f709d1eb	False	[{'time': 33, 'altitude': None, 'reason': 'merlin engine failure'}]	Engine failure at 33 seconds and loss of vehicle	[]	[]	[]	[5eb0e4b5b6c3bb0006
1	None	NaN	False	0.0	5e9d0d95eda69955f709d1eb	False	[{'time': 301, 'altitude': 289, 'reason': 'harmonic oscillation leading to premature engine shutdown'}]	Successful first stage burn and transition to second stage, maximum altitude 289 km, Premature engine shutdown at T+7 min 30 s, Failed to reach orbit, Failed to recover first stage	[]	[]	[]	[5eb0e4b6b6c3bb0006
2	None	NaN	False	0.0	5e9d0d95eda69955f709d1eb	False	[{'time': 140, 'altitude': 35, 'reason': 'residual stage-1 thrust led to collision between stage 1 and	Residual stage 1 thrust led to collision between stage 1 and stage 2	[]	[]	[]	[5eb0e4b6b6c3bb0006 5eb0e4b6b6c3bb0006

	FlightNumber	Date	BoosterVersion	PayloadMass	Orbit	LaunchSite	Outcome	Flights	GridFins	Reused	Legs	LandingPad	Block	ReusedCount
4	1	2010-06-04	Falcon 9	NaN	LEO	CCSFS SLC 40	None None	1	False	False	False	None	1.0	0
5	2	2012-05-22	Falcon 9	525.0	LEO	CCSFS SLC 40	None None	1	False	False	False	None	1.0	0
6	3	2013-03-01	Falcon 9	677.0	ISS	CCSFS SLC 40	None None	1	False	False	False	None	1.0	0
7	4	2013-09-29	Falcon 9	500.0	PO	VAFB SLC 4E	False Ocean	1	False	False	False	None	1.0	0
8	5	2013-12-03	Falcon 9	3170.0	GTO	CCSFS SLC 40	None None	1	False	False	False	None	1.0	0
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
89	86	2020-09-03	Falcon 9	15600.0	VLEO	KSC LC 39A	True ASDS	2	True	True	True	5e9e3032383ecb6bb234e7ca	5.0	8
90	87	2020-10-06	Falcon 9	15600.0	VLEO	KSC LC 39A	True ASDS	3	True	True	True	5e9e3032383ecb6bb234e7ca	5.0	8
91	88	2020-10-18	Falcon 9	15600.0	VLEO	KSC LC 39A	True ASDS	6	True	True	True	5e9e3032383ecb6bb234e7ca	5.0	9

Data Collection - Scraping

1. Define functions to extract features
2. Request the wiki page as an HTTP response
3. Extract all column/variable name from HTML table header
4. Parse the HTML table and create dataframe
5. Export dataframe to .csv

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

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

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

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

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

```
<tr>
<th scope="col">Flight No.
</th>
<th scope="col">Date and<br/>time (<a href="/wiki/Coordinated_Universal_Time" title="Coordinated Universal Time">UTC</a>)
</th>
<th scope="col"><a href="/wiki/List_of_Falcon_9_first-stage_boosters" title="List of Falcon 9 first-stage boosters">Version,<br/>Booster</a> <sup class="reference" id="cite_ref-booster_11-0"><a href="#cite_note-booster-11">[b]
</a></sup>
</th>
<th scope="col">Launch site
</th>
<th scope="col">Payload<sup class="reference" id="cite_ref-Dragon_12-0"><a href="#cite_note-Dragon-12">[c]</a></sup>
</th>
<th scope="col">Payload mass
</th>
<th scope="col">Orbit
</th>
<th scope="col">Customer
</th>
<th scope="col">Launch<br/>outcome
</th>
<th scope="col"><a href="/wiki/Falcon_9_first-stage_landing_tests" title="Falcon 9 first-stage landing tests">Booster<br/>landing</a>
</th></tr>
```

	Flight No.	Launch site	Payload mass	Orbit	Customer	Payload	Launch outcome	Version Booster	Booster landing	Date	Time
0	1	CCAFS	0	LEO	SpaceX	Dragon Spacecraft Qualification Unit	Success\n	F9 v1.0B0003.1	Failure	4 June 2010	18:45
1	2	CCAFS	0	LEO	NASA	Dragon	Success	F9 v1.0B0004.1	Failure	8 December 2010	15:43
2	3	CCAFS	525 kg	LEO	NASA	Dragon	Success	F9 v1.0B0005.1	No attempt\n	22 May 2012	07:44
3	4	CCAFS	4,700 kg	LEO	NASA	SpaceX CRS-1	Success\n	F9 v1.0B0006.1	No attempt	8 October 2012	00:35
4	5	CCAFS	4,877 kg	LEO	NASA	SpaceX CRS-2	Success\n	F9 v1.0B0007.1	No attempt\n	1 March 2013	15:1[illegible]

GitHub

Data Wrangling

1. EDA, identify the missing value and columns type

2. Calculate the number of launches for each site

3. Calculate the number and occurrence of each orbit

4. Calculate the number occurence of mission outcome per orbit type

5. Create a landing outcome label for training label

```
FlightNumber       0.00       int64
Date               0.00      object
BoosterVersion     0.00      object
PayloadMass        0.00     float64
Orbit              0.00      object
LaunchSite         0.00      object
Outcome            0.00      object
Flights            0.00       int64
GridFins           0.00        bool
Reused             0.00        bool
Legs               0.00        bool
LandingPad        40.62      object
Block              0.00     float64
ReusedCount        0.00       int64
Serial             0.00      object
Longitude          0.00     float64
Latitude           0.00     float64
```

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

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

```
df['Orbit'].value_counts()
```

```
GTO      27
ISS      21
VLEO     14
PO        9
LEO       7
SSO       5
MEO       3
ES-L1     1
HEO       1
SO        1
GEO       1
```

```
landing_outcomes = df['Outcome'].value_counts()
landing_outcomes
```

```
True ASDS      41
None None      19
True RTLS      14
False ASDS      6
True Ocean      5
False Ocean     2
None ASDS       2
False RTLS      1
```

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

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

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

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

```
# create a list where the element is zero
landing_class = []

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

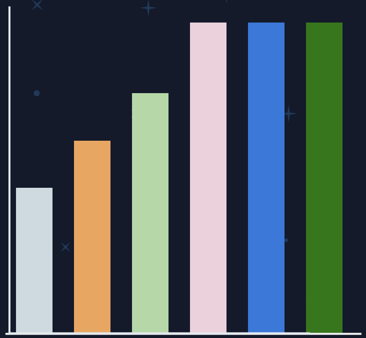
EDA with Data Visualization

GitHub

Scatter

Visualize the relationship between :

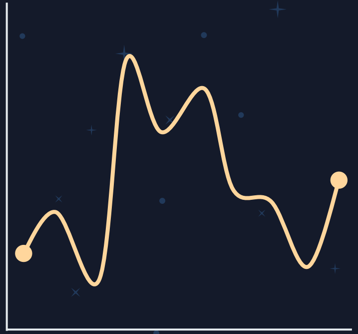
- Flight Number and Launch Site
- Payload and Launch Site
- Payload and Orbit Type



Bar

Visualize the relationship between :

- Success rate and Orbit Type
- Flight Number and Orbit Type



Line

Visualize the launch success yearly trend

EDA with SQL

- 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 the names of the booster_versions which have carried the maximum payload mass.
- List the failed landing_outcomes in drone ship, their booster versions, and launch site names for 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

Build an Interactive Map with Folium

- Mark all launch sites on a map by markers and circles.
- Use the Line to display the distances between the launch site to its proximities.
- Mark the success/failed launches for each site on the map
- Calculate the distances between a launch site to its proximities

Build a Dashboard with Plotly Dash

Pie Chart

Show the total successful launches count for all sites

Scatter

Show the correlation between payload and launch success

Predictive Analysis (Classification)

Import data and Standardize

Split data into training and test data.

Create GridSearchCV and fit model

Check accuracy for each model and tune parameters

Plot the confusion matrix

Find the best classification model to predict the outcome

Result

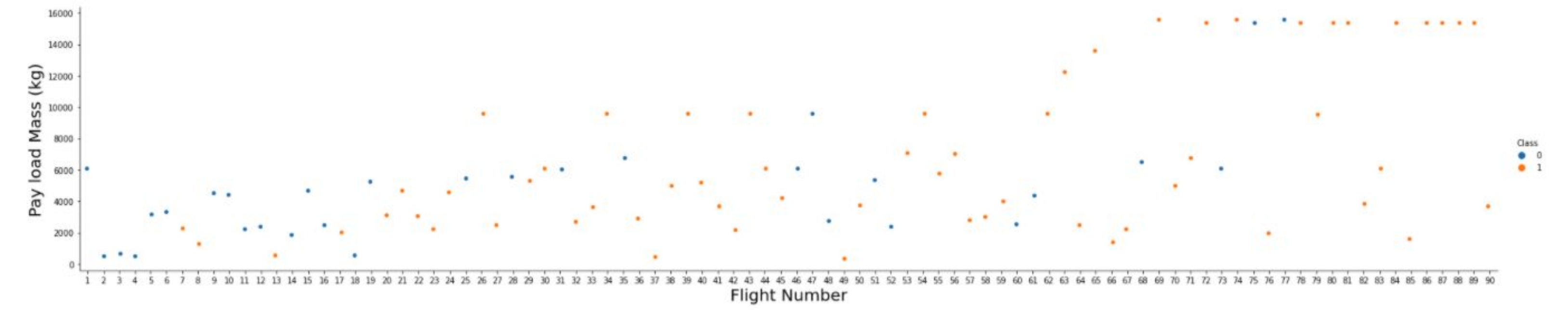
- Exploratory data analysis results
- Interactive analytics demo in screenshots
- Predictive analysis results

02

Insights drawn from EDA

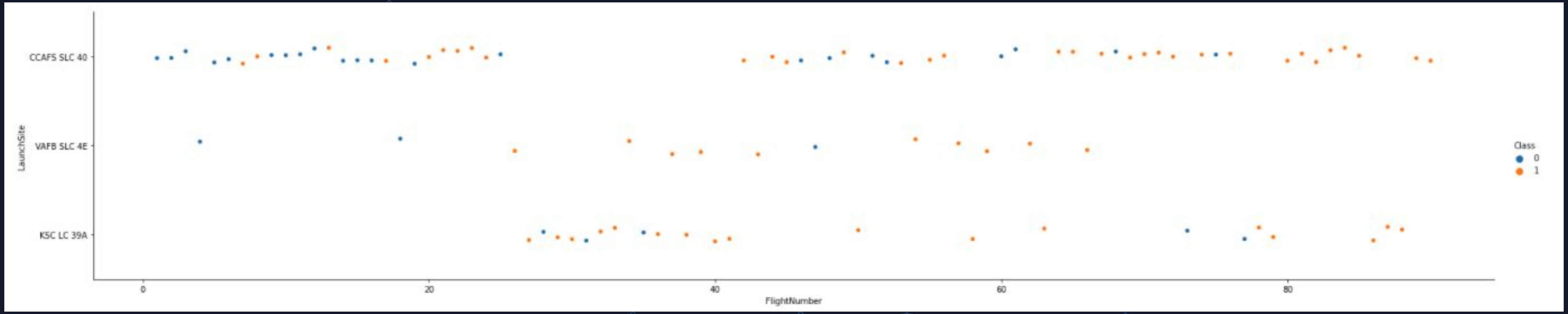
GitHub

Launch outcome with FlightNum and PayloadMass



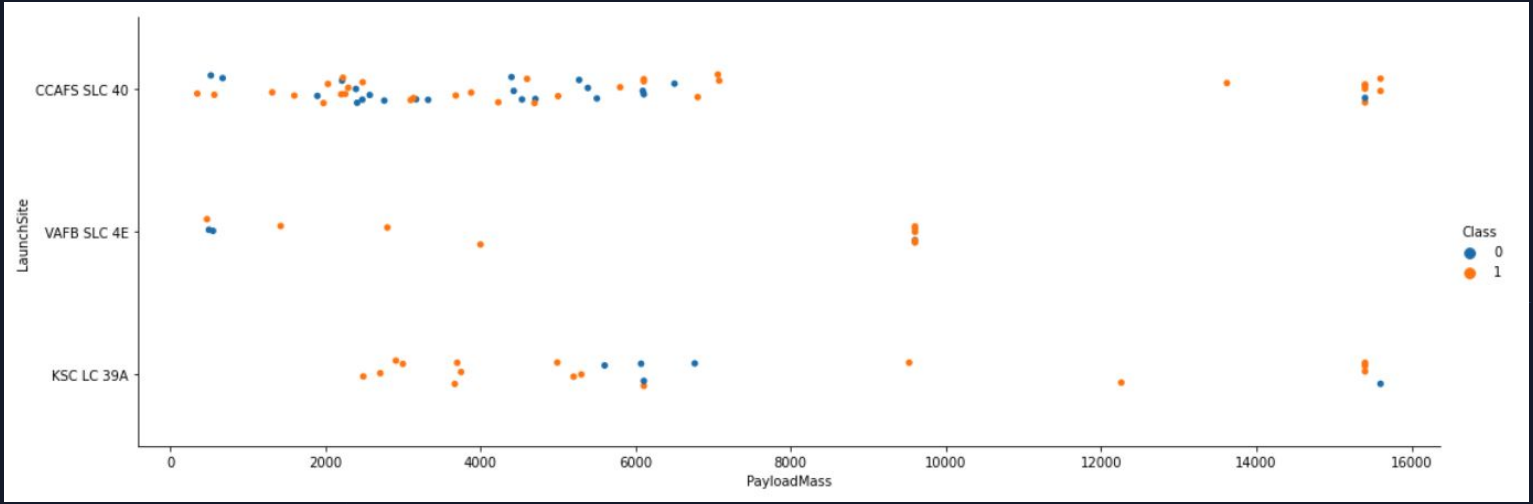
- The flight number increases, the first stage is more likely to land successfully.
- The more massive the payload, the less likely the first stage will return.

Launch outcome with FlightNum and LaunchSite



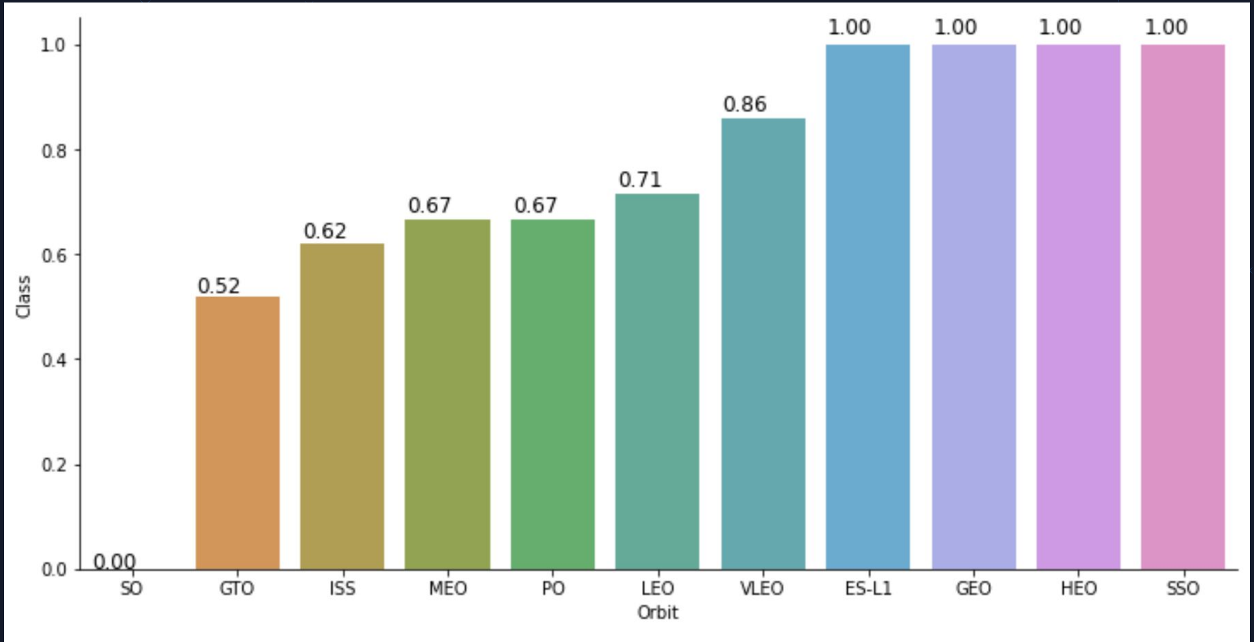
- The more amount of flights at a launch site the greater the success rate at a launch site.

Launch outcome with FlightNum and PayloadMass



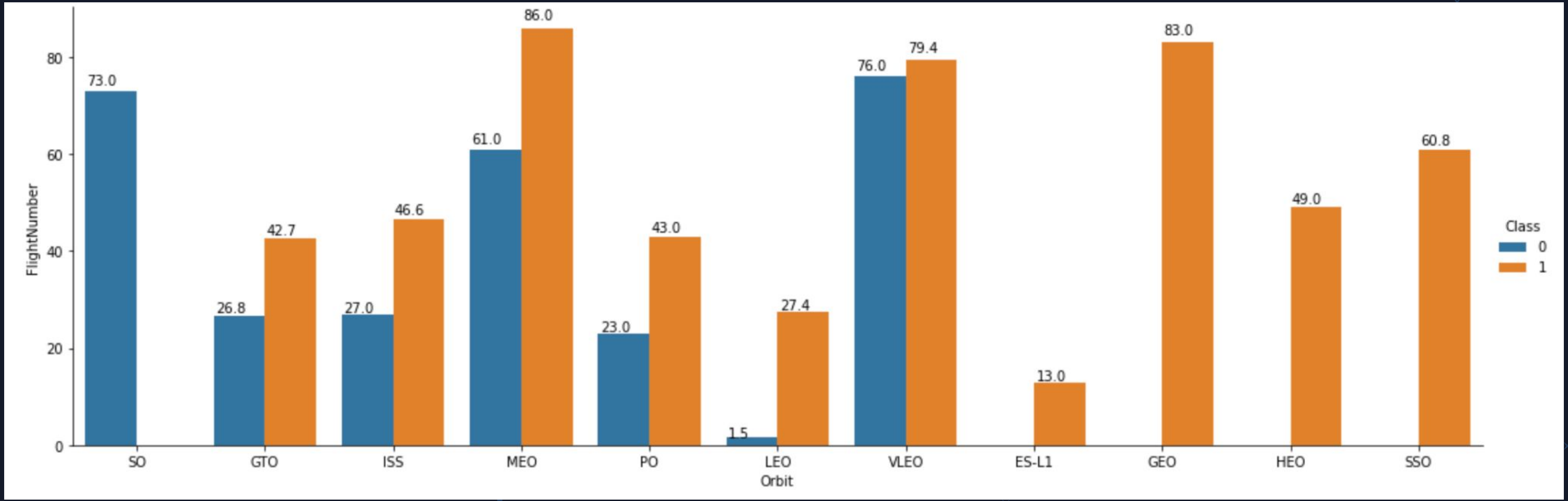
- The more amount of flights at a launch site the greater the success rate at a launch site.

The Success Rate of each Orbit Type



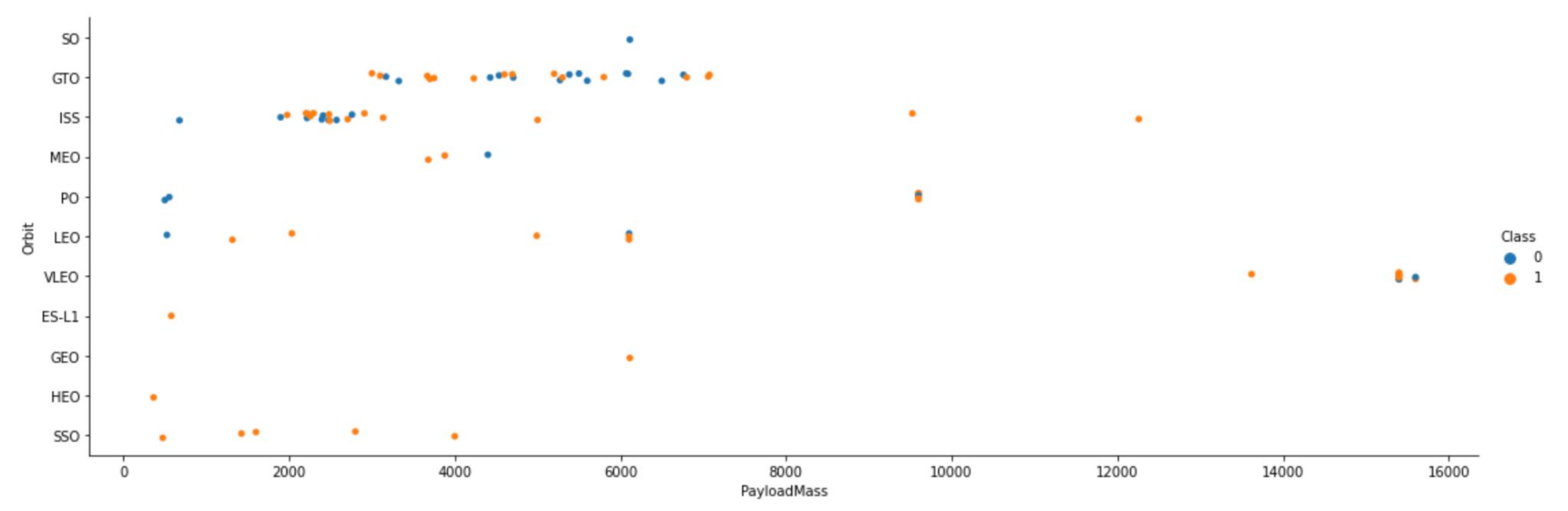
- Orbit ES-L1, GEO, HEO, SSO has the best success rate.
- The success rate of Orbit SO is zero.

The Success rate with FlightNumber and Orbit Type



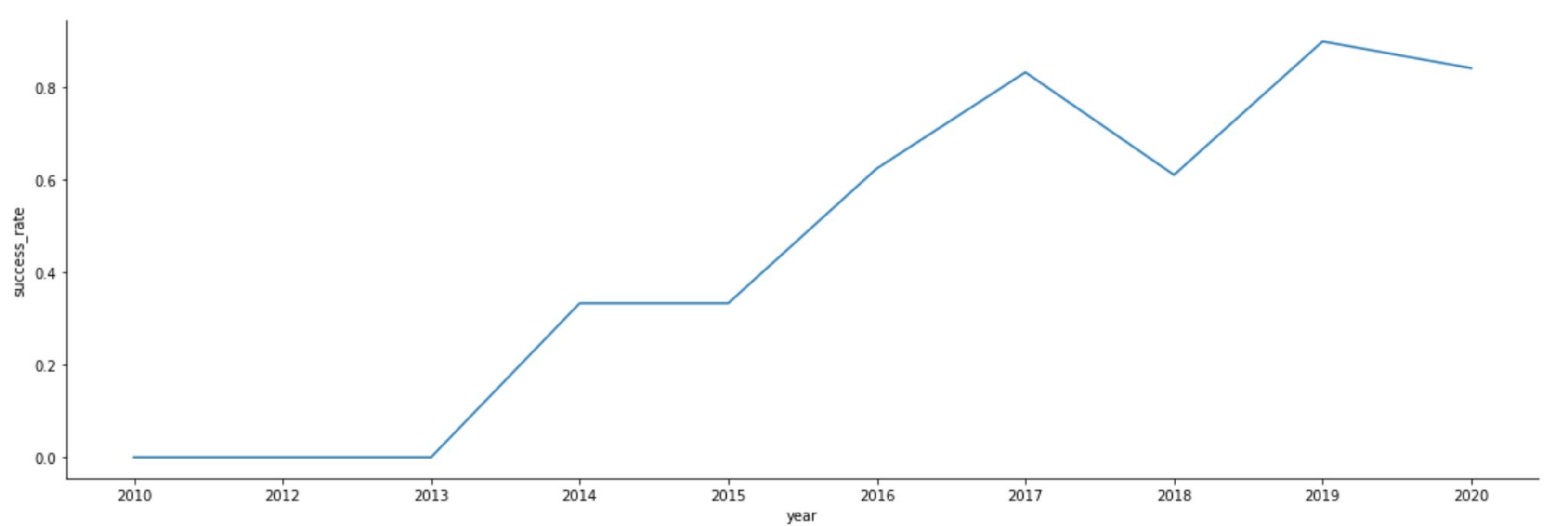
- The LEO orbit the Success appears related to the number of flights.
- No relationship between flight number when in GTO orbit.

Launch outcome with PayloadMass and Orbit Type



- The successful landing or positive landing rate are higher for Polar, LEO, and ISS may comes from the heavy payloads.
- The relationship is ambiguous for GTO cause the positive landing rate and negative landing (unsuccessful mission) are both there here.

The year trend of Launch Success Rate



- The success rate since 2013 kept increasing till 2020.
- 2017 and 2019 have the higher success rate.

4 Launch Sites in Space Mission

```
% SQL SELECT DISTINCT launch_site FROM SPACEXTBL;
```

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

- This dataset SPACEXTBL includes a record for each payload carried during a SpaceX mission into outer space.
- DISTINCT can come only once in a given select statement that display the name of unique launch sites.

5 records from "CCA-* " Launch Sites

```
% SQL SELECT * FROM SPACEXTBL3 WHERE launch_site LIKE 'CCA%' LIMIT 5;
```

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	07:44:00	F9 v1.0 B0005	CCAFS LC-40	Dragon demo flight C2	525	LEO (ISS)	NASA (COTS)	Success	No attempt
2012-10-08	00:35:00	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

- LIKE in a given select statement that determines if a character string matches a specified pattern.
- LIMIT 5 could limit the number of rows returned by a query, here we use 5 for get 5 records.

Total payload mass carried by NASA (CRS)

```
% SQL SELECT SUM(payload_mass__kg_) FROM SPACEXTBL3 WHERE customer LIKE '%NASA (CRS)%';
```

1
48213

- SUM () function returns the SUM of selected column.
- LIKE in a given select statement that determines if a character string matches a specified pattern.

Average payload mass carried by F9 v1.1

```
% SQL SELECT AVG(payload_mass__kg_) FROM SPACEXTBL3 WHERE booster_version LIKE '%F9 v1.1%';
```

1
2534

- AVG () function returns the average value of a column
- LIKE in a given select statement that determines if a character string matches a specified pattern.

GitHub

First Successful Ground Landing Date

```
% SQL SELECT MIN(DATE) FROM SPACEXTBL3 WHERE landing_outcome LIKE '%Success (ground pad)%';
```

1
2015-12-22

- MIN () function find the minimum value or lowest value of a column.
- LIKE in a given select statement that determines if a character string matches a specified pattern.

GitHub

Successful Drone Ship Landing with Payload between 4000 and 6000

```
% SQL SELECT booster_version FROM SPACEXTBL3
WHERE  landing_outcome LIKE '%Success (drone ship)%'
AND (payload_mass__kg_ between 4000 and 6000);
```

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

- WHERE used to filter and extract the records which fulfill the following condition.
- AND used to set more than one conditions.

Total Number of Successful and Failure Mission Outcomes

```
% SQL SELECT COUNT(*) FROM SPACEXTBL3 WHERE mission_outcome LIKE '%Success%';
```

1
100

```
% SQL SELECT COUNT(*) FROM SPACEXTBL3 WHERE mission_outcome LIKE '%Failure%';
```

1
1

- COUNT () function returns the number of rows that matches the condition.
- There are 100 successful records and 1 failure based on the SPACEXTBL3.

Boosters Carried Maximum Payload

```
% SQL SELECT booster_version, payload_mass__kg_ FROM SPACEXTBL3
WHERE payload_mass__kg_= (select MAX(payload_mass__kg_) FROM SPACEXTBL3)
```

booster_version	payload_mass__kg_
F9 B5 B1048.4	15600
F9 B5 B1049.4	15600
F9 B5 B1051.3	15600
F9 B5 B1056.4	15600
F9 B5 B1048.5	15600
F9 B5 B1051.4	15600
F9 B5 B1049.5	15600
F9 B5 B1060.2	15600
F9 B5 B1058.3	15600
F9 B5 B1051.6	15600
F9 B5 B1060.3	15600
F9 B5 B1049.7	15600

- Here we use subquery to select the max value of payload_mass__kg_ and display the booster_version and payload_mass__kg_
- The maximum payload is 15600 kg.

The 2015 failed landing records

```
% SQL SELECT booster_version, launch_site FROM SPACEXTBL3
WHERE landing_outcome LIKE '%Failure (drone ship)%' AND Year(DATE)=2015;
```

booster_version	launch_site
F9 v1.1 B1012	CCAFS LC-40
F9 v1.1 B1015	CCAFS LC-40

- There are two failure records in 2015 which the launch_site are both in CCAFS LC-40.

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

```
% SQL SELECT landing_outcome, COUNT (landing_outcome) as count FROM SPACEXTBL3
WHERE DATE BETWEEN '2010-06-04' and '2017-03-20'
GROUP BY landing_outcome ORDER BY COUNT DESC ;
```

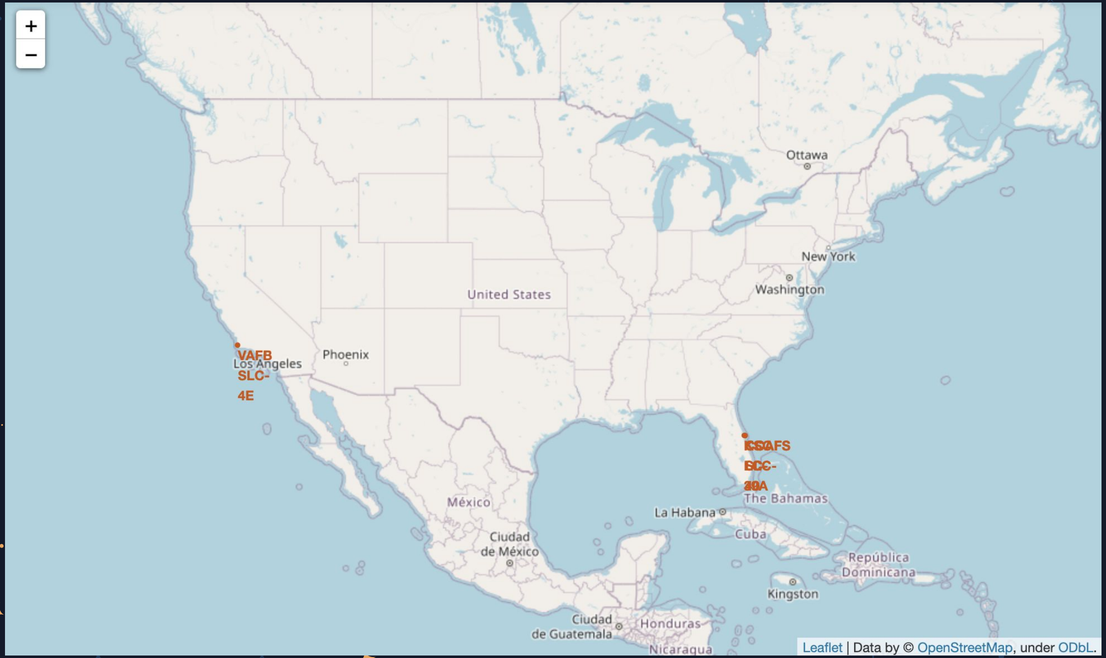
landing_outcome	COUNT
No attempt	10
Failure (drone ship)	5
Success (drone ship)	5
Controlled (ocean)	3
Success (ground pad)	3
Failure (parachute)	2
Uncontrolled (ocean)	2
Precluded (drone ship)	1

- No attempt occurs more frequently than any other outcomes.
- The failure and success in drone ship both 5 times.

03

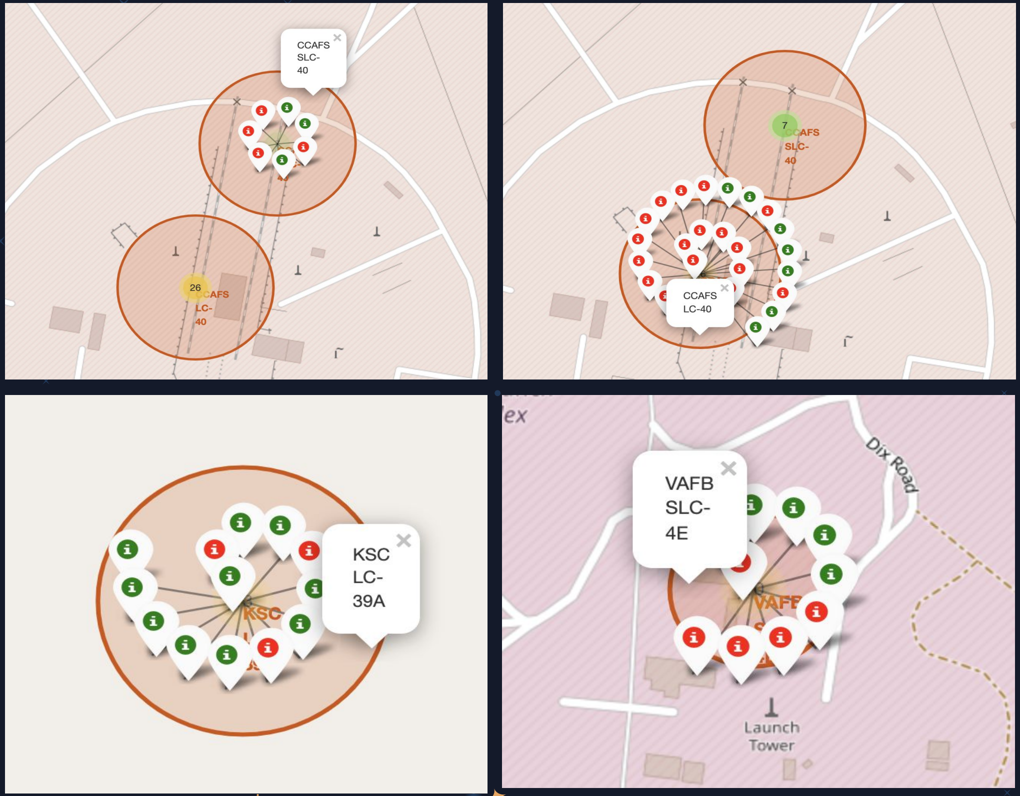
Launch Sites Proximities Analysis

4 Launch Sites Location



- 3 launch sites are in Florida and one is in California
- All launch sites are close to coastline.

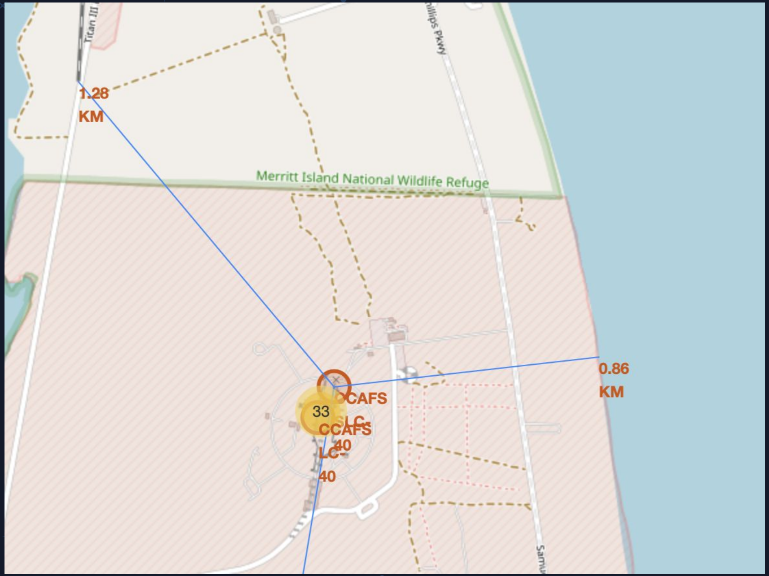
Launch Outcome for each Site



Successful Launches

Fail Launches

Distance between Launch and proximities

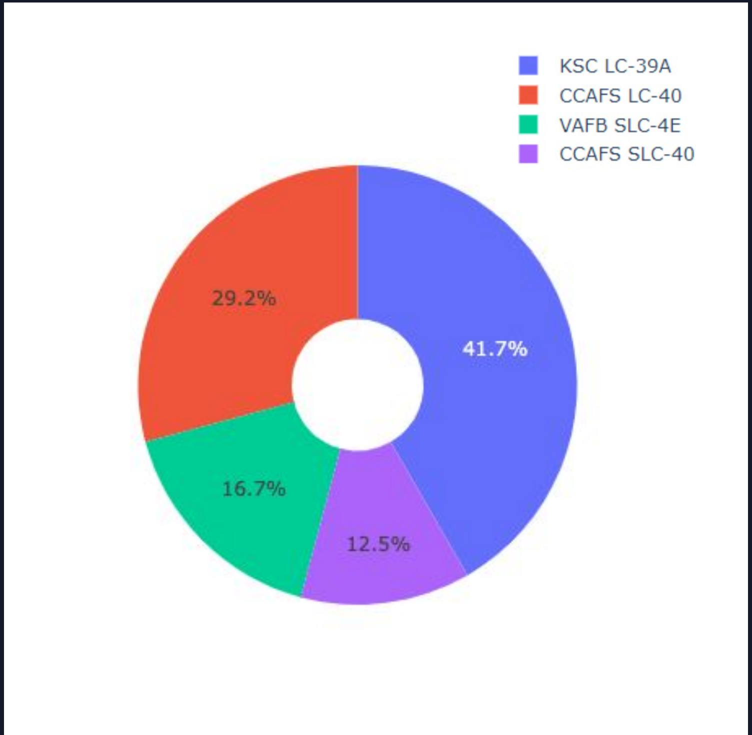


- Launch site would keep certain distance away from railways, highways, and cities, but close to coastline.

04

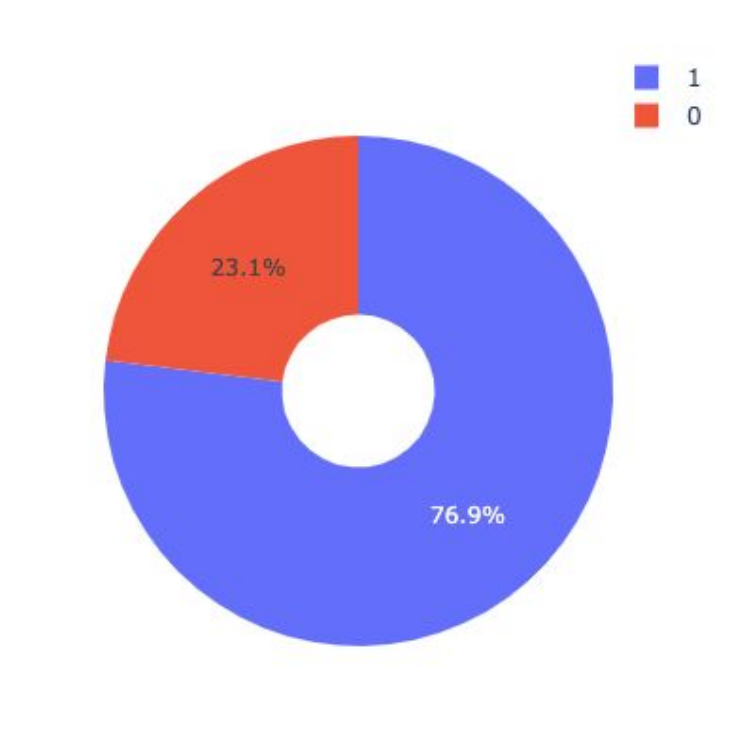
Build a Dashboard with Plotly Dash

Launch success count for all sites



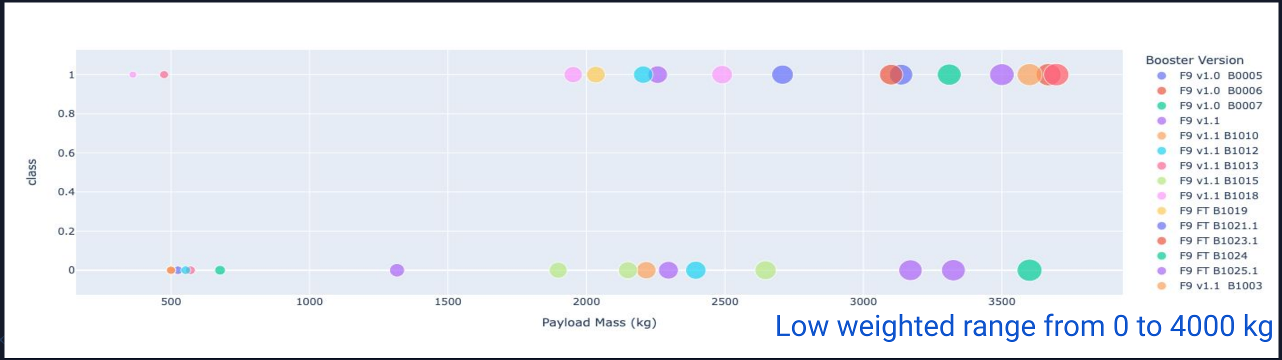
- KSC LC-39A has 41.7% launch success, which is highest.
- CCAFS SLC-40 only has 12.5%.

KSC LC-39A success ratio

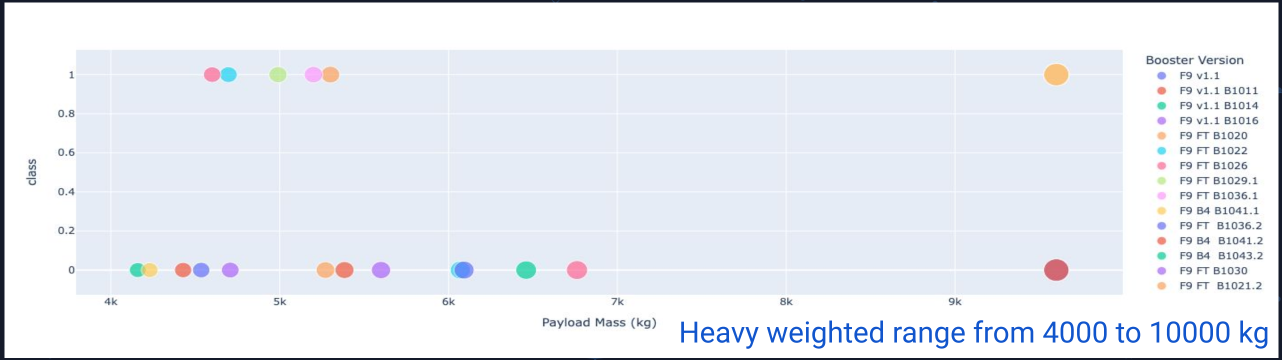


- Class = 1 means success
- KSC LC-39A has 76.9% success ratio

Payload vs. Launch Outcome scatter plot with diff range



Low weighted range from 0 to 4000 kg



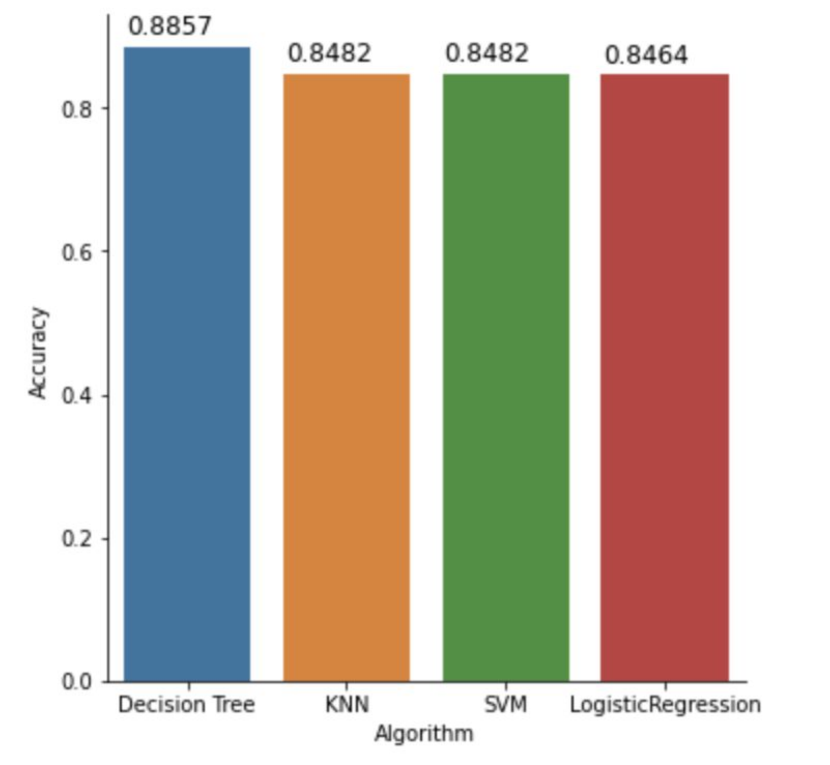
Heavy weighted range from 4000 to 10000 kg

- In low range group, the payload mass has more success outcome.
- Compare to low range, heavy weighted range does not have the obvious relation when the payload mass with more than 7000 kg.

05

Predictive Analysis (Classification)

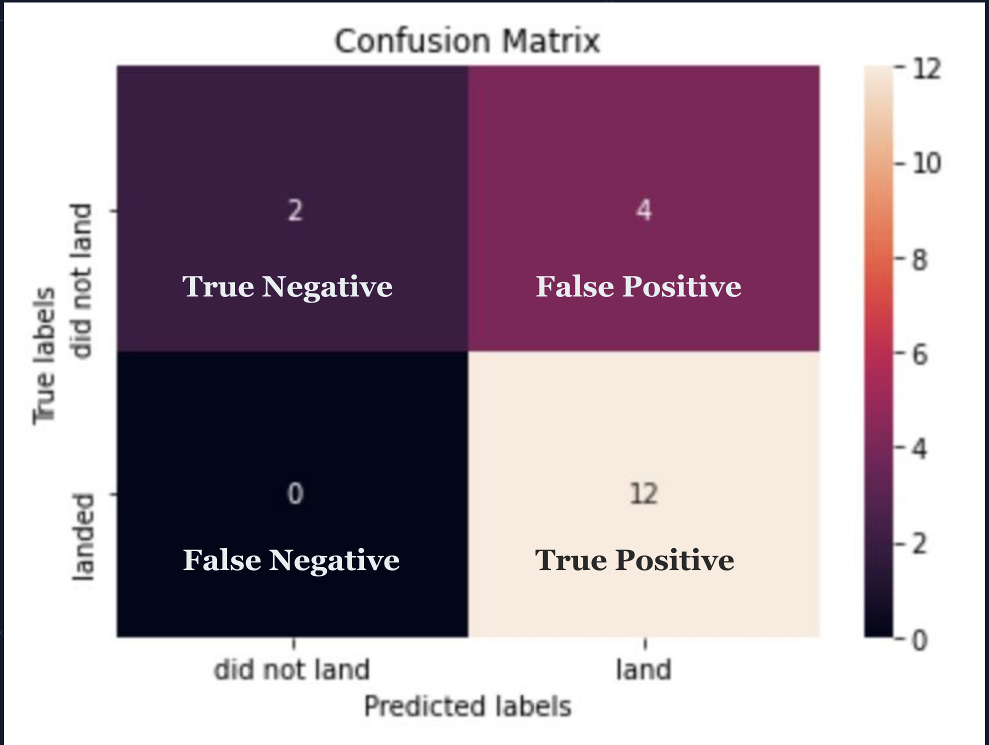
Classification Accuracy



- Decision Tree model has the highest classification accuracy.

- Best Parameters:
  - criterion: gini
  - max_depth: 10
  - max_features: sqrt
  - min_samples_leaf: 1
  - min_samples_split: 2
  - splitter: random

Confusion Matrix



- The best decision tree model predicted 12 records would landing and 2 records would not landing correctly.
- The false positive means 4 predictions did not fit the truth.

Conclusions

- The more massive the payload, the less likely the first stage will return.
- The more amount of flights at a launch site the greater the success rate.
- Orbit ES-L1, GEO, HEO, SSO has the best success rate.
- The success rate since 2013 kept increasing till 2020.
- KSC LC-39A has 41.7% launch success, which is highest.
- Decision Tree model has the highest classification accuracy, 0.8857.

GitHub

Appendix

- Python code
- Dataset
- SQL
- Dashboard
- Charts

Thanks!

Any suggestions would be greatly appreciated!

CREDITS: This presentation template was created by **Slidesgo**, including icons by **Flaticon,** infographics & images by **Freepik**

Please keep this slide for attribution