IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

Augustine Pimentel
05/22/2024

Outline

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

Executive Summary

- Summary of methodologies
  - EDA With SQL
  - Interactive Visual Analytics And Dashboards
  - Predictive Analysis

- Summary of all results

Introduction

The commercial space age is here, companies are making space travel affordable for everyone. None more than Space X!! Their ideal of reusing the first stage of the rocket as cut cost and has the possible of making space flight a reality for the common man.

Therefore, if we can determine if the first stage will land, we can determine the cost of a launch and possibly make the future happen now!

Section 1

Methodology

Methodology

Data collection methodology:

- Collecting the Data via an API => SpaceX launch data that is gathered from an API, specifically the SpaceX REST API. This API will give us data about launches, including information about the rocket used, payload delivered, launch specifications, landing specifications, and landing outcome.

- Web scraping => Falcon 9 historical launch records from a Wikipedia page titled List of Falcon 9 and Falcon Heavy launches

  - https://en.wikipedia.org/wiki/List_of_Falcon_9_and_Falcon_Heavy_launches

Perform data wrangling

- Data wrangling => Using the following atributes: Flight Number, Date, Booster version, Payload mass Orbit, Launch Site, Outcome: this is the status of the first stage Flights, Grid Fins: these help with landing Reused, Legs: used in landing Landing pad, Block, Reused count, Serial, Longitude and latitude of launch.

Perform exploratory data analysis (EDA) using visualization and SQL

- Exploratory Data Analysis => Analysis using a database.

Methodology - Continued

Perform interactive visual analytics using Folium and Plotly Dash

- Visual Analytics and Dashboard Interactive
    - Folium => Interactive map
    - Plotly Dashboard

Perform predictive analysis using classification model

- Predictive Analysis => Machine learning pipeline to predict if the first stage of the Falcon 9 lands successfully.
    - Preprocessing,
    - Train test split
        - find the hyperparameters
        - Model with the best accuracy using the training data.
            - Logistic Regression,
            - Support Vector machines,
            - Decision Tree Classifier,
            - K-nearest neighbors.
    - Output => Confusion matrix.

Data Collection Charts


```
b'[{"fairings":{"reused":false,"recovery_attempt":false,"recovered":false,"ships":[
l,"media":null,"recovery":null},"flickr":{"small":[],"original":[]},"presskit":null
edia":"https://en.wikipedia.org/wiki/DemoSat"},"static_fire_date_utc":"2006-03-17T0
engine failure"}],"details":"Engine failure at 33 seconds and loss of vehicle","cre
000Z","date_unix":1143239400,"date_local":"2006-03-25T10:30:00+12:00","date_precisi
ding_type":null,"landpad":null}],"auto_update":true,"tbd":false,"launch_library_id"
4a/ZboXReNb_o.png","large":"https://images2.imgbox.com/80/a2/bkWotCIS_o.png"},"redd
be_id":"Lk4zQ2wP-Nc","article":"https://www.space.com/3590-spacex-falcon-1-rocket-f
955f709d1eb","success":false,"failures":[{"time":301,"altitude":289,"reason":"harmo
t T+7 min 30 s, Failed to reach orbit, Failed to recover first stage","crew":[],"sh
_unix":1174439400,"date_local":"2007-03-21T13:10:00+12:00","date_precision":"hour",
null,"landpad":null}],"auto_update":true,"tbd":false,"launch_library_id":null,"id":
_o.png","large":"https://images2.imgbox.com/4a/80/k1oAkY0k_o.png"},"reddit":{"campa
9p3U8860","article":"http://www.spacex.com/news/2013/02/11/falcon-1-flight-3-missio
da69955f709d1eb","success":false,"failures":[{"time":140,"altitude":35,"reason":"re
es":[],"payloads":["5eb0e4b6b6c3bb0006eeb1e3","5eb0e4b6b6c3bb0006eeb1e4"],"launchpa
ecision":"hour","upcoming":false,"cores":[{"core":"5e9e289ef3591814873b2625","fligh
y_id":null,"id":"5eb87cdbffd86e000604b32c"},{"fairings":{"reused":false,"recovery_a
g"},"reddit":{"campaign":null,"launch":null,"media":null,"recovery":null},"flickr":
ipedia":"https://en.wikipedia.org/wiki/Ratsat"},"static_fire_date_utc":"2008-09-20T
the first successful orbital launch of any privately funded and developed, liquid-p
er":4,"name":"RatSat","date_utc":"2008-09-28T23:15:00.000Z","date_unix":1222643700,
se,"landing_attempt":false,"landing_success":null,"landing_type":null,"landpad":nul
nks":{"patch":{"small":"https://images2.imgbox.com/ab/5a/Pequxd5d_o.png","large":"h
pacex.com/press/2012/12/19/spacexs-falcon-1-successfully-delivers-razaksat-satellit
s://en.wikipedia.org/wiki/RazakSAT"},"static_fire_date_utc":null,"static_fire_date_
6eeb1e6"],"launchpad":"5e9e4502f5090995de566f86","flight_number":5,"name":"RazakSat
103b2627","flight":1,"gridfins":false,"legs":false,"reused":false,"landing_attempt"
ull,"recovery_attempt":null,"recovered":null,"ships":[]},"links":{"patch":{"small":
l},"flickr":{"small":[],"original":[]},"presskit":"http://forum.nasaspaceflight.com
13/02/12/falcon-9-flight-1","wikipedia":"https://en.wikipedia.org/wiki/Dragon_Space
s":true,"failures":[],"details":null,"crew":[],"ships":[],"capsules":[],"payloads":
ate_local":"2010-06-04T14:45:00-04:00","date_precision":"hour","upcoming":false,"co
l}],"auto_update":true,"tbd":false,"launch_library_id":null,"id":"5eb87cddffd86e000
gn":null,"launch":null,"media":null,"recovery":null},"flickr":{"small":[],"original
n.wikipedia.org/wiki/SpaceX_COTS_Demo_Flight_1","wikipedia":"https://en.wikipedia.o
c","success":true,"failures":[],"details":null,"crew":[],"ships":["5ea6ed2d080df400
e":"COTS 1","date_utc":"2010-12-08T15:43:00.000Z","date_unix":1291822980,"date_loca
```


EDA Raw Data


```
<table class="wikitable plainrowheaders collapsible" style="width: 100%;">
<tbody><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> <su
</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>
<tr>
<th rowspan="2" scope="row" style="text-align:center;">1
</th>
<td>4 June 2010,<br/>18:45
</td>
<td><a href="/wiki/Falcon_9_v1.0" title="Falcon 9 v1.0">F9 v1.0</a><sup class="reference" id="cite_ref-MuskMay2012_13-0"><a href="#cite_note-Mus
k_numbers-14">[8]</a></sup>
</td>
<td><a href="/wiki/Cape_Canaveral_Space_Force_Station" title="Cape Canaveral Space Force Station">CCAFS</a>,<br/><a href="/wiki/Cape_Canaveral_S
</td>
<td><a href="/wiki/Dragon_Spacecraft_Qualification_Unit" title="Dragon Spacecraft Qualification Unit">Dragon Spacecraft Qualification Unit</a>
</td>
<td>
</td>
<td><a href="/wiki/Low_Earth_orbit" title="Low Earth orbit">LEO</a>
</td>
<td><a href="/wiki/SpaceX" title="SpaceX">SpaceX</a>
```


Web Scraping Data

Data Collection Charts

Dataframe Read:

[11]:

	FlightNumber	Date	BoosterVersion	PayloadMass	Orbit	LaunchSite	Outcome	Flights	GridFins	Reused	Legs	LandingPad	Block	ReusedCount	Serial	Longitude	Latitude
0	1	2010-06-04	Falcon 9	6104.959412	LEO	CCAFS SLC 40	None None	1	False	False	False	NaN	1.0	0	B0003	-80.577366	28.561857
1	2	2012-05-22	Falcon 9	525.000000	LEO	CCAFS SLC 40	None None	1	False	False	False	NaN	1.0	0	B0005	-80.577366	28.561857
2	3	2013-03-01	Falcon 9	677.000000	ISS	CCAFS SLC 40	None None	1	False	False	False	NaN	1.0	0	B0007	-80.577366	28.561857
3	4	2013-09-29	Falcon 9	500.000000	PO	VAFB SLC 4E	False Ocean	1	False	False	False	NaN	1.0	0	B1003	-120.610829	34.632093
4	5	2013-12-03	Falcon 9	3170.000000	GTO	CCAFS SLC 40	None None	1	False	False	False	NaN	1.0	0	B1004	-80.577366	28.561857
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
85	86	2020-09-03	Falcon 9	15400.000000	VLEO	KSC LC 39A	True ASDS	2	True	True	True	5e9e3032383ecb6bb234e7ca	5.0	2	B1060	-80.603956	28.608058
86	87	2020-10-06	Falcon 9	15400.000000	VLEO	KSC LC 39A	True ASDS	3	True	True	True	5e9e3032383ecb6bb234e7ca	5.0	2	B1058	-80.603956	28.608058
87	88	2020-10-18	Falcon 9	15400.000000	VLEO	KSC LC 39A	True ASDS	6	True	True	True	5e9e3032383ecb6bb234e7ca	5.0	5	B1051	-80.603956	28.608058
88	89	2020-10-24	Falcon 9	15400.000000	VLEO	CCAFS SLC 40	True ASDS	3	True	True	True	5e9e3033383ecbb9e534e7cc	5.0	2	B1060	-80.577366	28.561857
89	90	2020-11-05	Falcon 9	3681.000000	MEO	CCAFS SLC 40	True ASDS	1	True	False	True	5e9e3032383ecb6bb234e7ca	5.0	0	B1062	-80.577366	28.561857

90 rows × 17 columns

Data Collection – SpaceX API

- Present your data collection with SpaceX REST calls using key phrases and flowcharts

- https://github.com/AugieP57/Applied_Data_Science_Capstone.git

API Flow:

Import Libraries and Define Auxiliary Functions

define API Path

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

Get API Data

response = requests.get(spacex_url)

#Use json_normalize meethod to convert the json result into a dataframe

data = pd.json_normalize(response.json())

Data Collection - Scraping

- Present your web scraping process using key phrases and flowcharts

https://github.com/AugieP57/Applied_Data_Science_Capstone/blob/main/jupyter-labs-webscraping.ipynb

```
# Install
    !pip3 install beautifulsoup4
    !pip3 install requests

# imports
    import sys
    import requests
    from bs4 import BeautifulSoup
    import re
    import unicodedata
    import pandas as pd

# url
static_url = https://en.wikipedia.org/w/index.php?title=List_of_Falcon_9_and_Falcon_Heavy_launches&oldid=1027686922

# scrpe url
data = requests.get(static_url)
# create Beauful soup obj
soup = BeautifulSoup(data.content)
```

Data Wrangling

- How data was processed:
  - Calculate the number of launches for each site
  - Calculate the number and occurrence of each orbit
  - Create a landing Outcome column
  - Using this column => Determined overall success rate base on landing type:
    - ADSD(DROWN SHIP) = 41/47 = 0.8723
    - OCEAN = 5/7 = 0.7143
    - RTLS = 14/15 = 0.9333
- Data Wrangling URL
  - https://github.com/AugieP57/Applied_Data_Science_Capstone/blob/main/labs-jupyter-spacex-Data%20wrangling.ipynb

EDA with Data Visualization

- Following Charts Plotted:
  - Display the names of the unique launch sites in the space mission:

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

  - Display the total payload mass carried by boosters for each site

Site	Sum Payload
CCAFS LC-40	619967
VAFB SLC-4E	89730
KSC LC-39A	208837
CCAFS SLC-40	254037

  - Display average payload mass carried by booster version F9 v1.1 for each site

Sum of PAYLOAD_MASS__KG_	Booster_Version							
Launch_Site	F9_B	F9_B4	F9_B5	F9_FT	F9_FT	F9_v1.0	F9_v1.1	Grand Total
CCAFS LC-40				18364	10330	1702	66967	97363
CCAFS SLC-40	4400	22182	224205	6435				257222
KSC LC-39A		11800	146734	43098				201632
VAFB SLC-4E		25660	31592	29275	2150		1053	89730
Grand Total	**4400**	**59642**	**402531**	**97172**	**12480**	**1702**	**68020**	**645947**

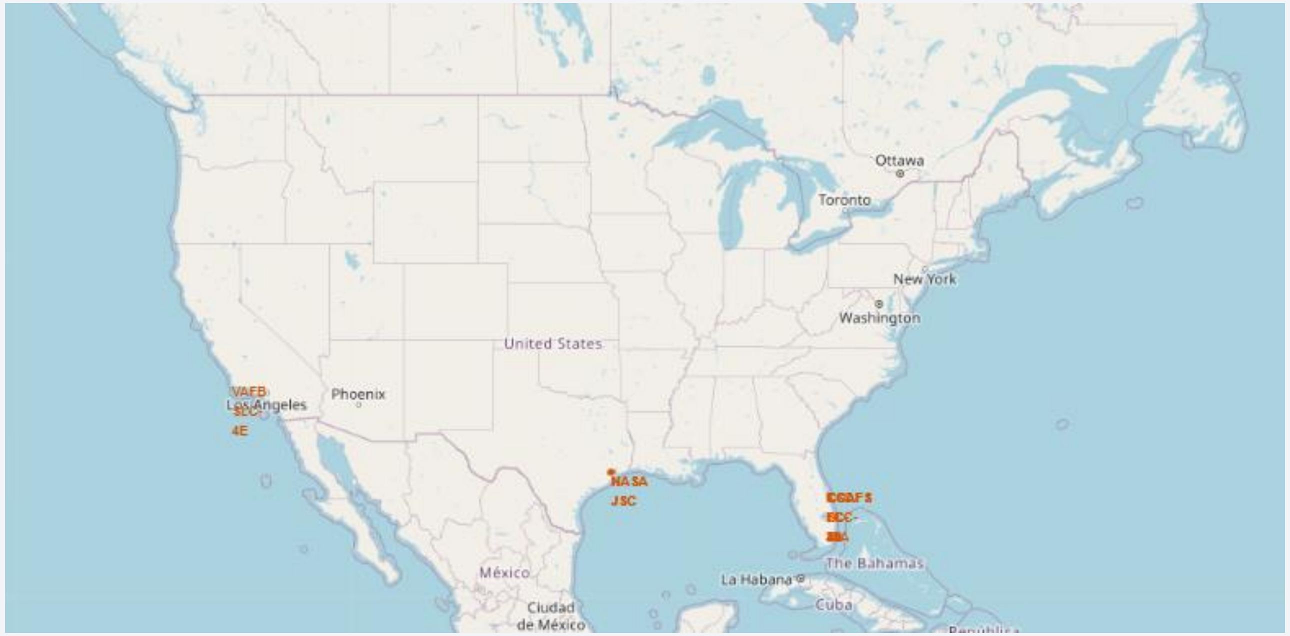
- EDA URL
  - https://github.com/AugieP57/Applied_Data_Science_Capstone/blob/main/jupyter-labs-eda-sql-coursera_sqllite.ipynb

EDA with SQL

- SQL queries:
  - %sql select distinct(Launch_Site) from SPACEXTABLE
  - %sql select count('Success (drone ship) from SPACEXTABLE where Launch_Site = "CCAFS LC-40" AND Mission_Outcome = "Succcess" and Landing_Outcome = "Success (drone ship)"
  - %sql select avg(PAYLOAD_MASS__KG_) from SPACEXTABLE where Booster_Version = 'F9 v1.1'
  - %sql select min(Date) from SPACEXTABLE where Landing_Outcome = 'Success (ground pad)'
  - %sql select distinct(Booster_Version) from SPACEXTABLE where PAYLOAD_MASS__KG_ > 4000 and PAYLOAD_MASS__KG_ < 6000
  - %sql select substr(Date, 6,2) as month, Landing_Outcome, Booster_Version, Launch_Site from SPACEXTBL where Landing_Outcome = 'Failure (drone ship)' and substr(Date,0,5)='2015' limit 5

  %sql select * from SPACEXTBL where date between '2010-06-04' and '2017-03-20' order by date desc limit 5
  - https://github.com/AugieP57/Applied_Data_Science_Capstone/blob/main/jupyter-labs-eda-sql-coursera_sqllite.ipynb

Build an Interactive Map with Folium

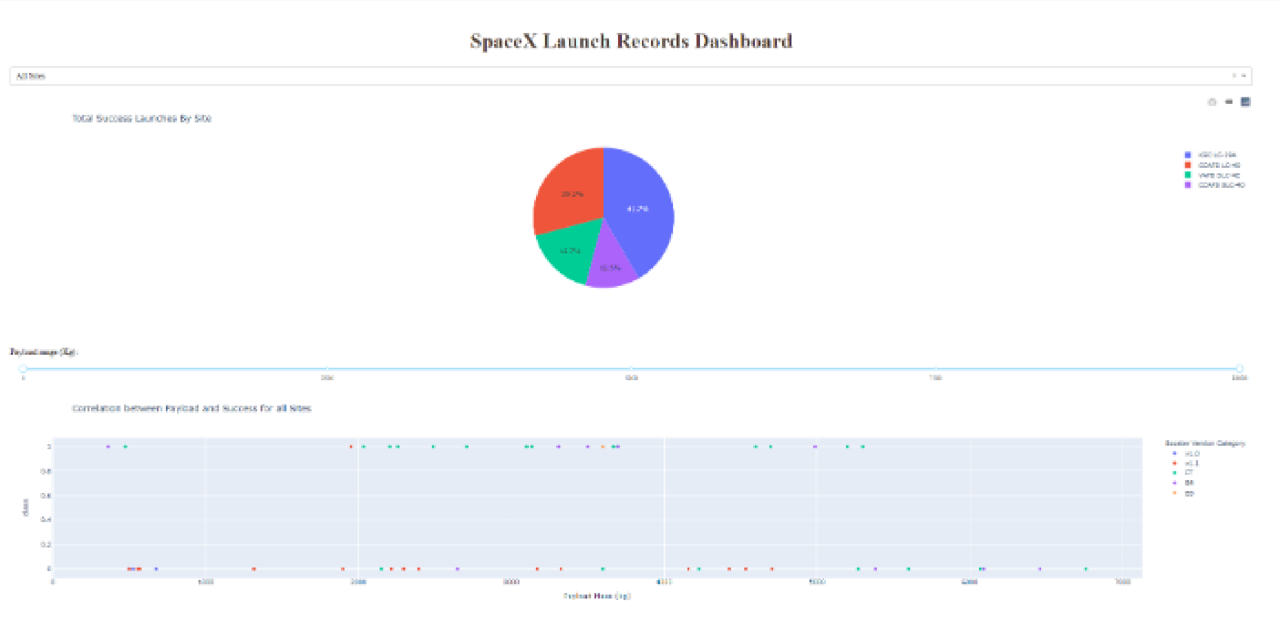


	Launch Site	Lat	Long
0	CCAFS LC-40	28.562302	-80.577356
1	CCAFS SLC-40	28.563197	-80.576820
2	KSC LC-39A	28.573255	-80.646895
3	VAFB SLC-4E	34.632834	-120.610745

- https://github.com/AugieP57/Applied_Data_Science_Capstone/blob/main/lab_jupyter_launch_site_location.ipynb

Build a Dashboard with Plotly Dash

- Add pie chart to show Successful launches by Site
- Payload and site Successful launches
- https://augustinepim-8050.theianext-0-labs-prod-misc-tools-us-east-0.proxy.cognitiveclass.ai/



Predictive Analysis (Classification)

1. Loaded rep defined functions & dataframe
   - URL1 = https://cf-courses-data.s3.us.cloud-object-storage.appdomain.cloud/IBM-DS0321EN-SkillsNetwork/datasets/dataset_part_2.csv
2. Create a NumPy array from the column `Class` in `data`, by applying the method `to_numpy()` then assign it to the variable Y,make sure the output is a Pandas series (only one bracket df['name of column']).
3. Standardize the data in X then reassign it to the variable X using the transform provided
4. Split the data into training and testing data using the function `train_test_split`. The training data is divided into validation data, a second set used for training data; then the models are trained and hyperparameters are selected using the function `GridSearchCV`.
5. Create a logistic regression object then create a GridSearchCV object `logreg_cv` with cv = 10. Fit the object to find the best parameters from the dictionary `parameters`.
6. output the `GridSearchCV` object for logistic regression. We display the best parameters using the data attribute `best_params_` and the accuracy on the validation data using the data attribute `best_score_`.
7. Calculate the accuracy on the test data using the method `score`:
8. Create a support vector machine object then create a `GridSearchCV` object `svm_cv` with cv = 10. Fit the object to find the best parameters from the dictionary `parameters`.
9. Calculate the accuracy of tree_cv on the test data using the method `score`:
10. Create a k nearest neighbors object then create a `GridSearchCV` object `knn_cv` with cv = 10. Fit the object to find the best parameters from the dictionary `parameters`.
11. Calculate the accuracy of knn_cv on the test data using the method `score`:
12. Calculate the accuracy of knn_cv on the test data using the method `score`:
13. https://github.com/AugieP57/Applied_Data_Science_Capstone/blob/main/SpaceX_Machine%20Learning%20Prediction_Part_5.ipynb

Results

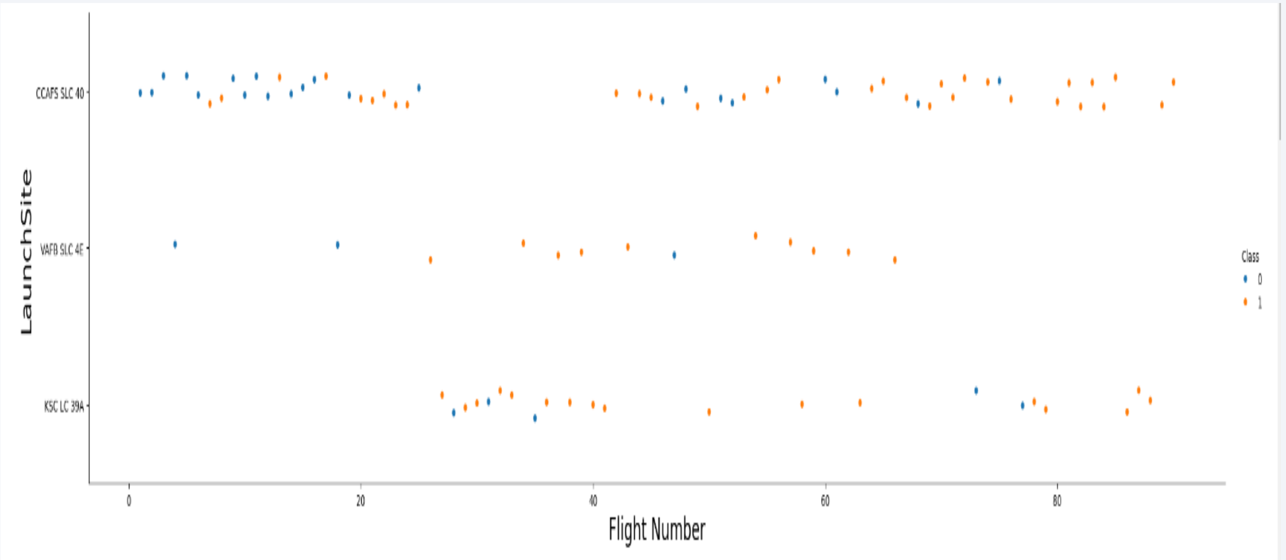
1. Based on all the data Logical regression provides the best fit
2. Higher weight payloads perform better
3. GEO as the best success rate

Thank You

Section 2

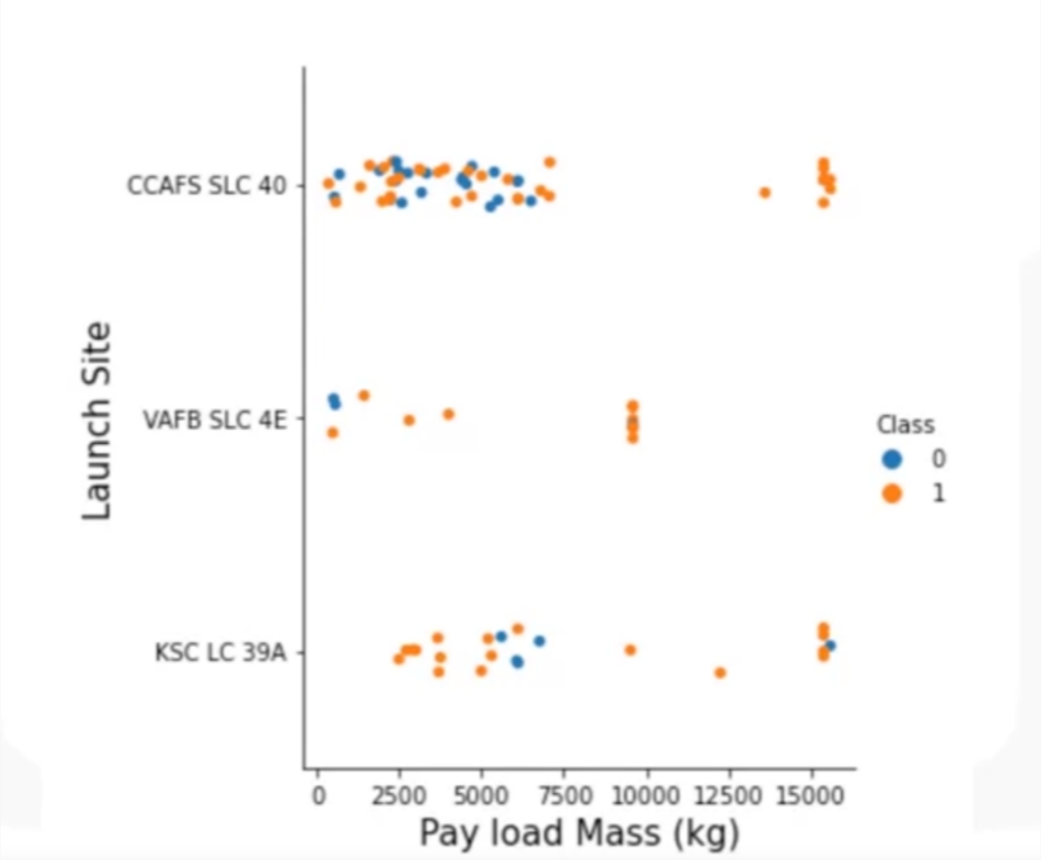
Insights drawn from EDA

Flight Number vs. Launch Site



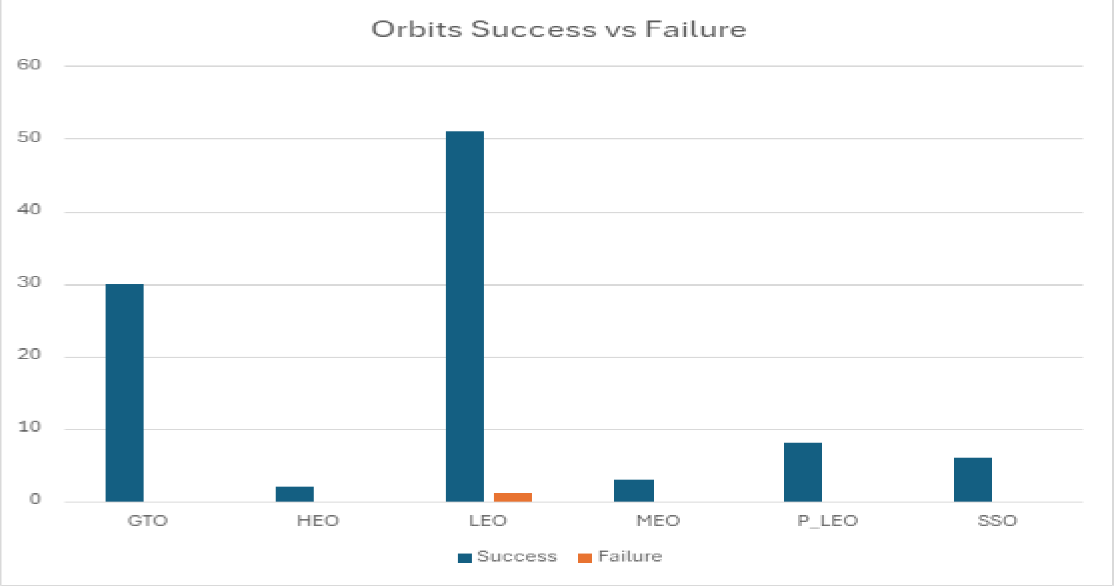
Distributions of flights => Divided pretty evenly among sites

Payload vs. Launch Site



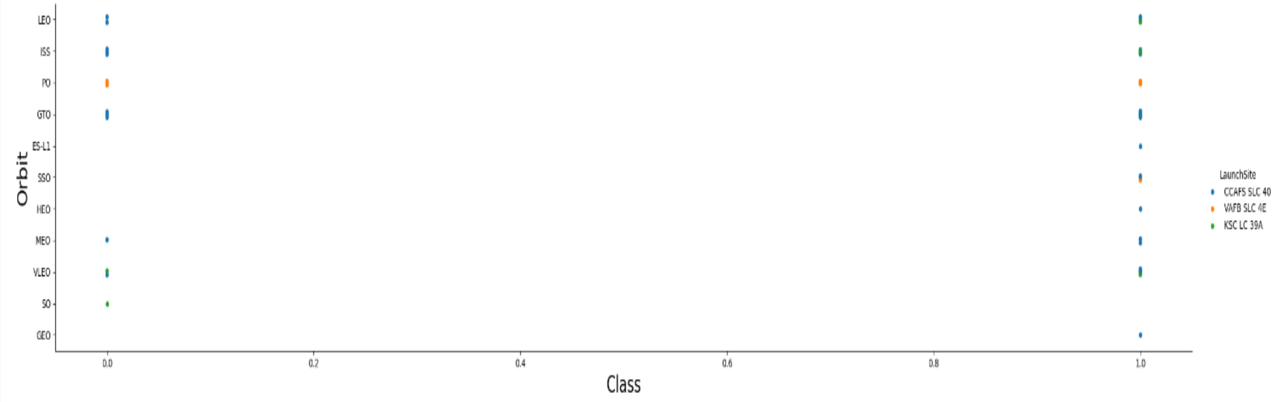
- Heavier Payloads have greater success across all Sites

Success Rate vs. Orbit Type

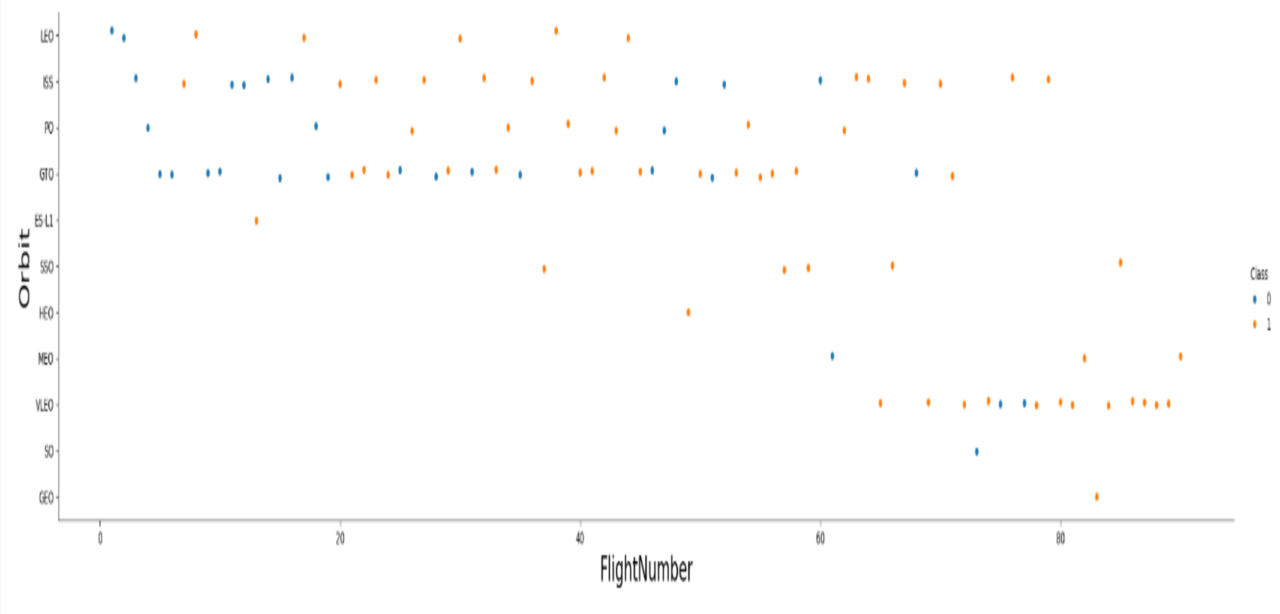


Count of Orbit	Orbit						
Mission_Outcome	GTO	HEO	LEO	MEO	Polar LEO	SSO	Grand Total
Failure			1				1
Success	30	2	51	3	8	6	100
Grand Total	**30**	**2**	**52**	**3**	**8**	**6**	**101**

- GTO, HEO, P_LEO, and SSO all have perfect success record by Mission Outcome
- LEO has 1 Failure by Mission Outcome
- Overall Not a good indicator of success – Land Outcome is the better indicator.

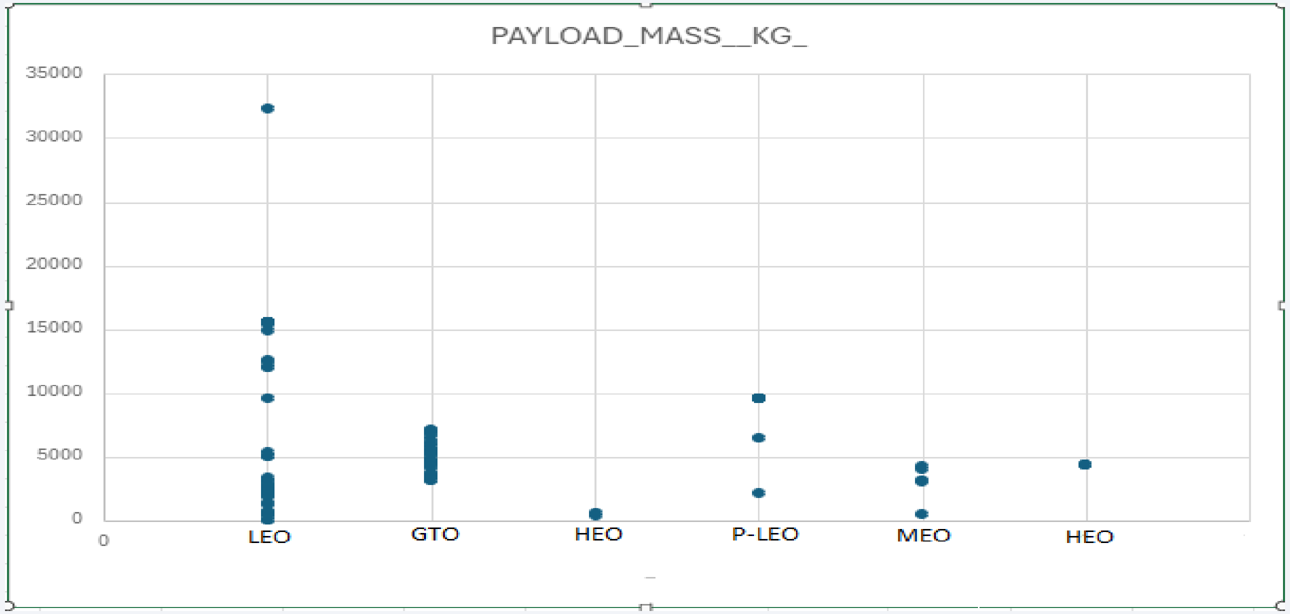


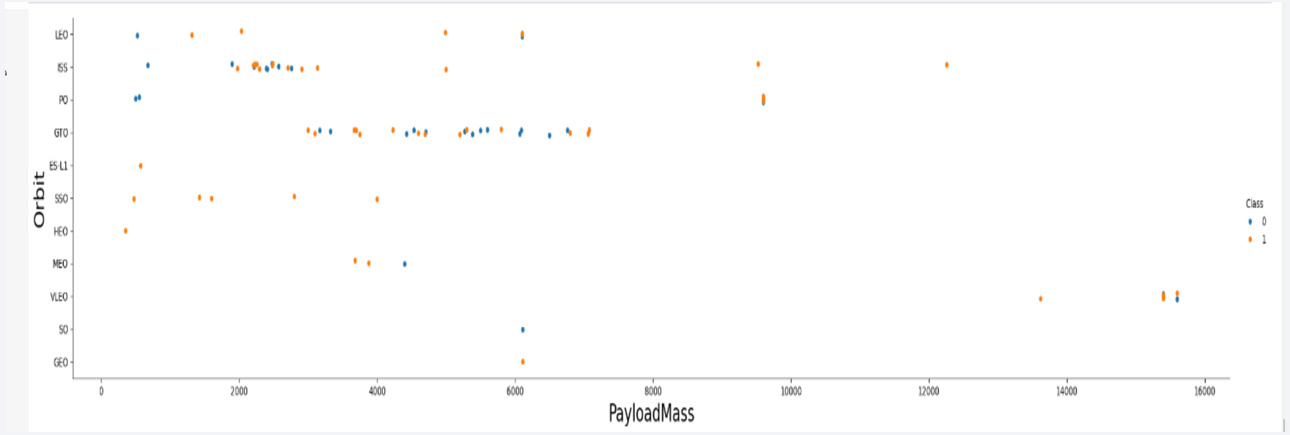
Flight Number vs. Orbit Type



- Low Orbits appear to have better success rates
- Higher orbits higher failure rates

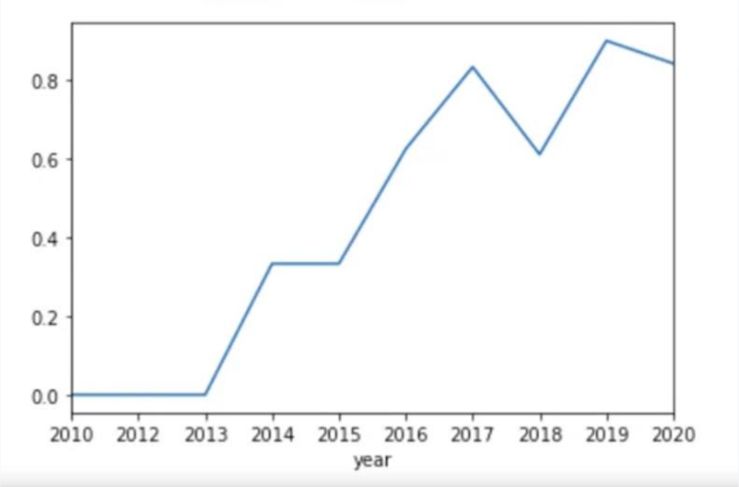
Payload vs. Orbit Type





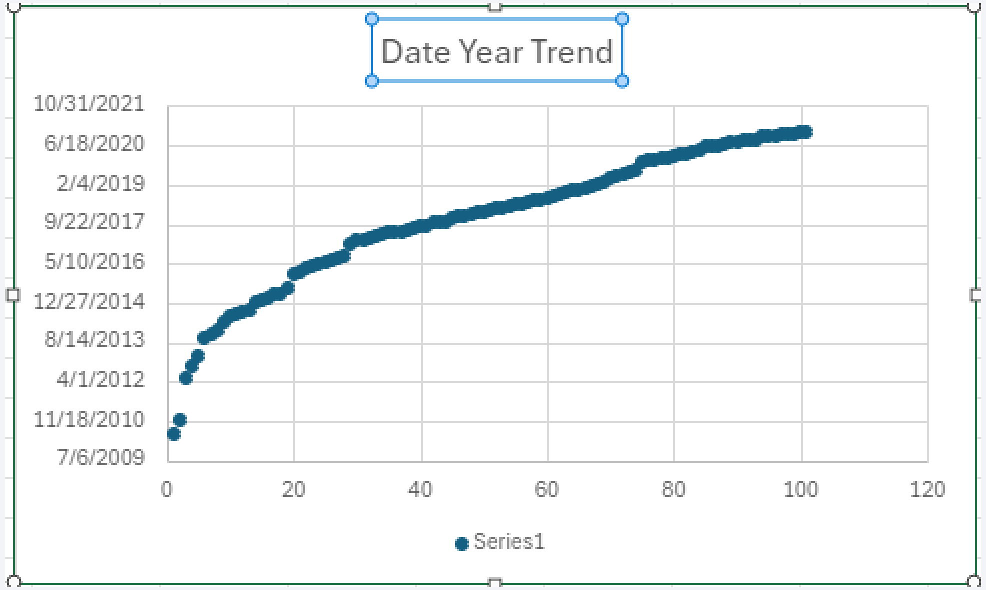
- Pay load Mass verse Orbit shows a greater total for LEO Orbit
- What is need => Payload, Orbit, and Success Rate comparison

Launch Success Yearly Trend



Yearly Success on an upward trend

Scatter Plot showing the same Trend



All Launch Site Names

- Site designations:

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

- Query
  - Select distinct (Launch_Site) from SPACEXTABLE

Launch Site Names Begin with 'CCA'

- 5 records where launch sites begin with `CCA`:

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

  - Select * From SPACEXTABLE Where Launch_Site like 'CCA%' Limit 5

Total Payload Mass

sum(PAYLOAD_MASS__KG_)
107010

- Query
  - %sql select sum(PAYLOAD_MASS__KG_) from SPACEXTABLE where Customer like "%NASA%"

Average Payload Mass by F9 v1.1

sum(PAYLOAD_MASS__KG_)
14642

- Query
  - %sql select sum(PAYLOAD_MASS__KG_) from SPACEXTABLE where Booster_Version = "F9 v1.1"

First Successful Ground Landing Date

Date
2015-12-22

- Query
  - %sql select Date from SPACEXTBL where Landing_Outcome = 'Success (ground pad)' limit 1

Successful Drone Ship Landing with Payload between 4000 and 6000

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

- Query:
  - %sql select Booster_Version from SPACEXTBL where PAYLOAD_MASS__KG_ > 4000 and PAYLOAD_MASS__KG_ < 6000 and Landing_Outcome = 'Success (drone ship)'

Total Number of Successful and Failure Mission Outcomes

- Calculate the total number of successful and failure mission outcomes
- Failure results

count(Landing_Outcome)
10

- Success results

count(Landing_Outcome)
70

- Queries
  - Failure Total = %sql select count(Landing_Outcome) from SPACEXTABLE where Landing_Outcome like "%Failure%"
  - Success Total = %sql select count(Landing_Outcome) from SPACEXTABLE where Landing_Outcome NOT like "%Failure%" and Landing_Outcome != "No attempt"

Boosters Carried Maximum Payload

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

- Query:
  - %sql select distinct (Booster_Version) from SPACEXTABLE where PAYLOAD_MASS__KG_ = (select max(PAYLOAD_MASS__KG_) from SPACEXTABLE)

2015 Launch Records

Booster_Version	Launch_Site
F9 v1.1 B1012	CCAFS LC-40
F9 v1.1 B1015	CCAFS LC-40

- Query
  - %sql select Booster_Version, Launch_Site from SPACEXTABLE where Landing_Outcome = 'Failure (drone ship)' and Date like '2015%'

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

Landing_Outcome	rank
Failure (drone ship)	1
Failure (drone ship)	1
Failure (drone ship)	1
Failure (drone ship)	1
Failure (drone ship)	1
Success (ground pad)	6
Success (ground pad)	6
Success (ground pad)	6

- Query
  - %sql select Landing_Outcome, rank() OVER (ORDER BY Landing_Outcome) as rank from SPACEXTABLE where Landing_Outcome = 'Failure (drone ship)' or Landing_Outcome = 'Success (ground pad)' and Date between '2010-06-04' and '2017-03-20'

Section 3

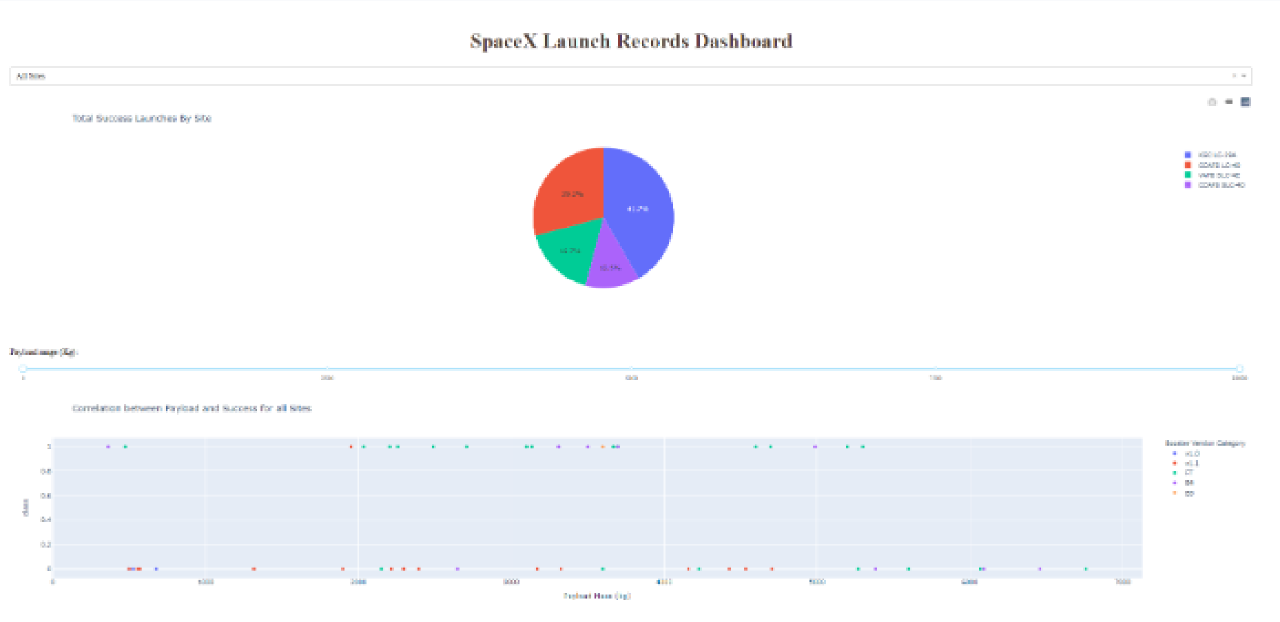
Build a Dashboard with Plotly Dash

<Dashboard Screenshot 1>

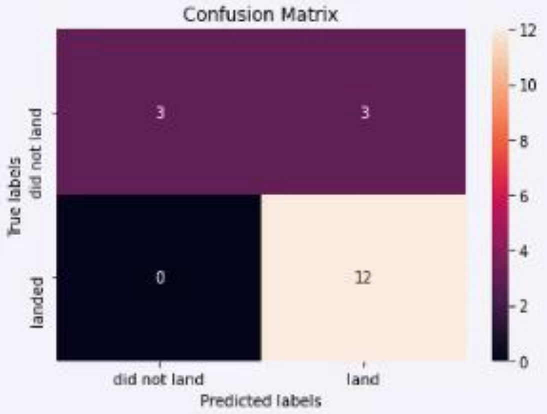
- Replace <Dashboard screenshot 1> title

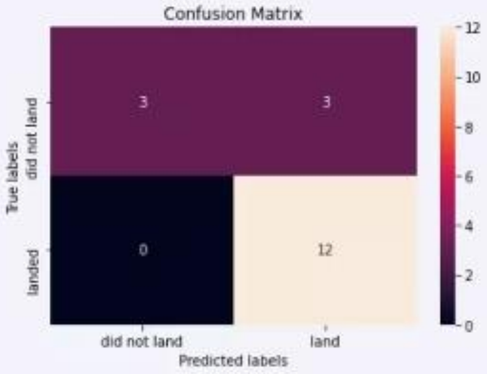
- Show the screenshot of launch success c

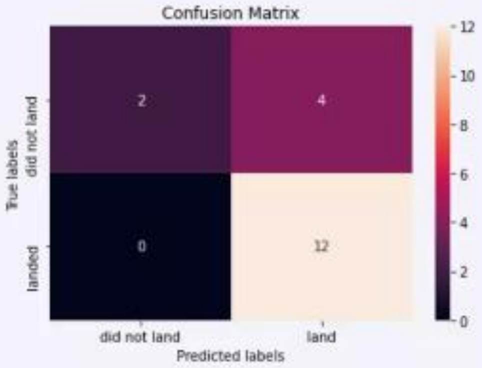
- Explain the important elements and findi

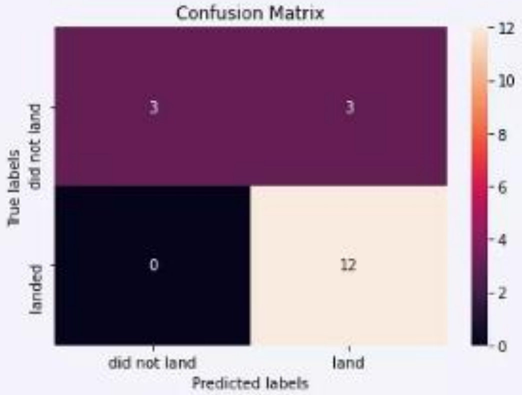


Confusion Matrix









The confusion matrix provides a breakdown of a classification model's performance by categorizing predictions into four categories: true positives (correctly predicted positive instances), true negatives (correctly predicted negative instances), false positives (incorrectly predicted positive instances), and false negatives (incorrectly predicted negative instances).

Conclusions

1. Based on all the data Logical regression provides the best fit
2. Higher weight payloads perform better
3. GEO as the best success rate

Model Accuracy

Thank you!