IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

<Name>
<Date>

Outline

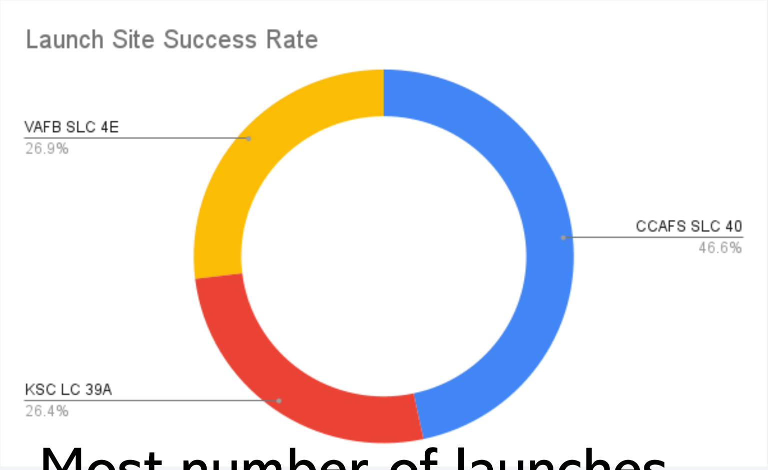
- Executive Summary
- Introduction
- Methodology
- Results
- Conclusion
- Appendix

Types Of Orbits

- Geostationary orbit (GEO)
- Low Earth orbit (LEO)
- Medium Earth orbit (MEO)
- Polar orbit(PO) and Sun-synchronous orbit (SSO)
- Transfer orbits and geostationary transfer orbit (GTO)
- Lagrange points (L-points)
- International Space Station Orbit (ISS)
- Very Low Earth Orbit (VLEO)

Executive Summary

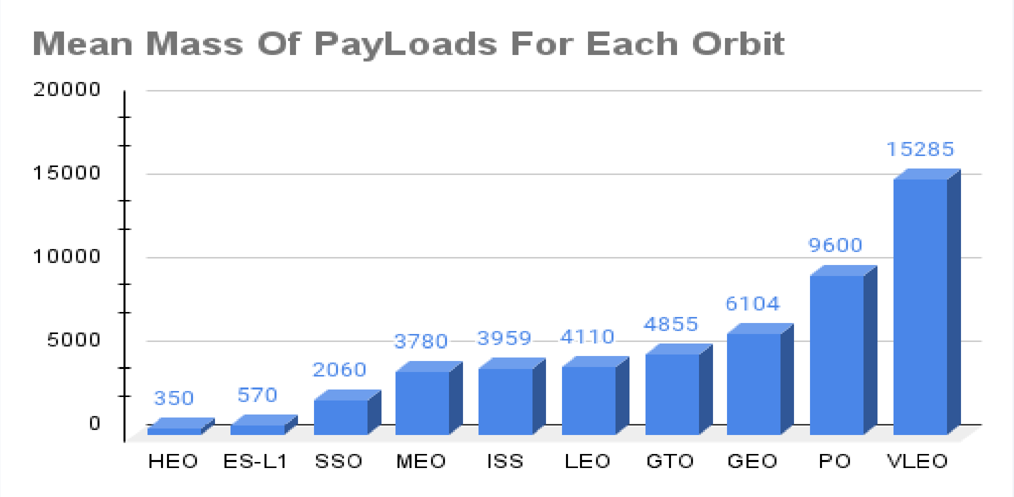
LAUNCH SITES

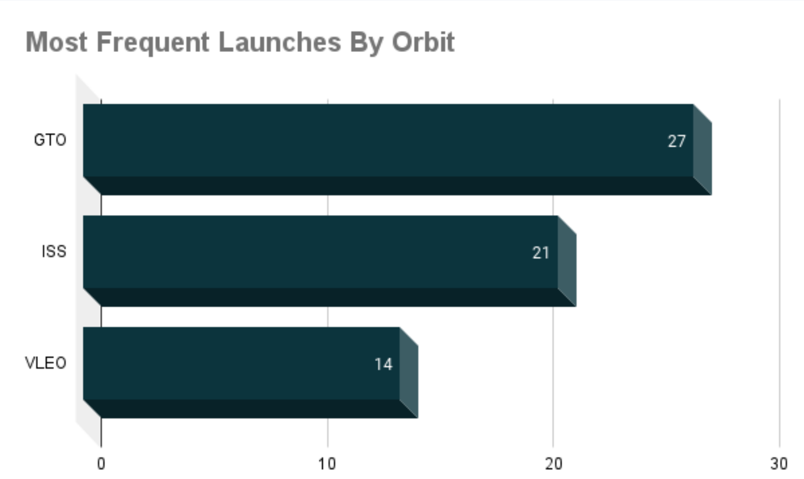


Most number of launches performed at:
CCAFS SLC 40
With Success Rate **28%**
While **KSC LC 39A & VAFB SLC 4E**
With Success Rate **36%**

MOST FOCUSED LAUNCHES

Successful Orbits Launch With Their Average Payload Mass





We are given not that much scientific data but breaking down payload masses and orbit types data we find out *Decision Tree Classifier* as best ml model.
Low payload masses for are ***more favorable ES-L1, HEO, SSO*** for upcoming bidding while *GTO ISS VLEO* requires more time to make things work ***smoothly.***

Introduction

- We use python request module to scrap data from public API
- By performing data wrangling and data preprocessing we obtain analyzable and visualizable form.
- We plotted various matplotlib charts to prepare content for executive summary meanwhile used pandas group-by function on Orbits type , frequency of successful launches against each year by using
- Used visual geographic charts also plotted some charts on google data studio to show parameters against each other.

Section 1

Methodology

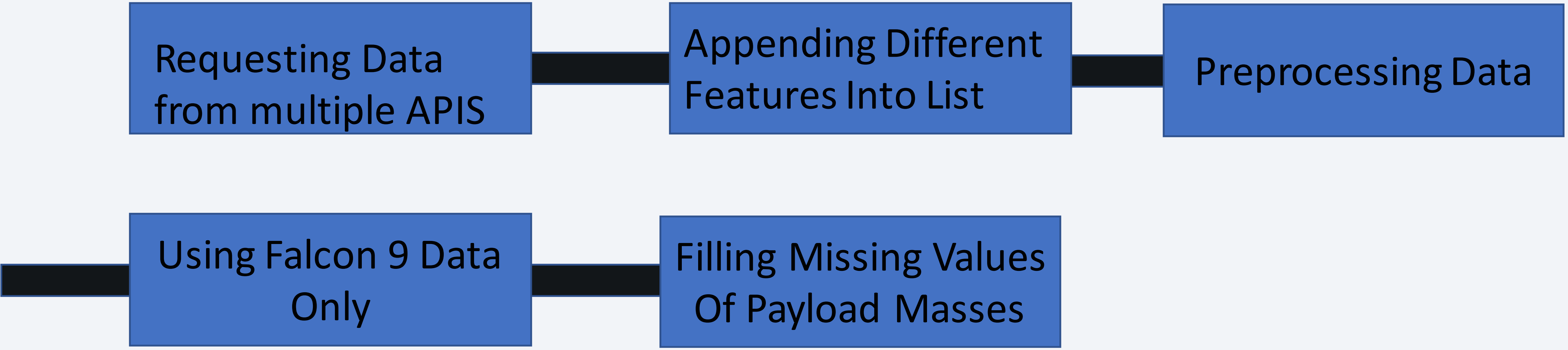
Methodology

Executive Summary

- Data collection methodology:
    - Describe how data was collected
- 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

- Describe how data sets were collected.
- You need to present your data collection process use key phrases and flowcharts

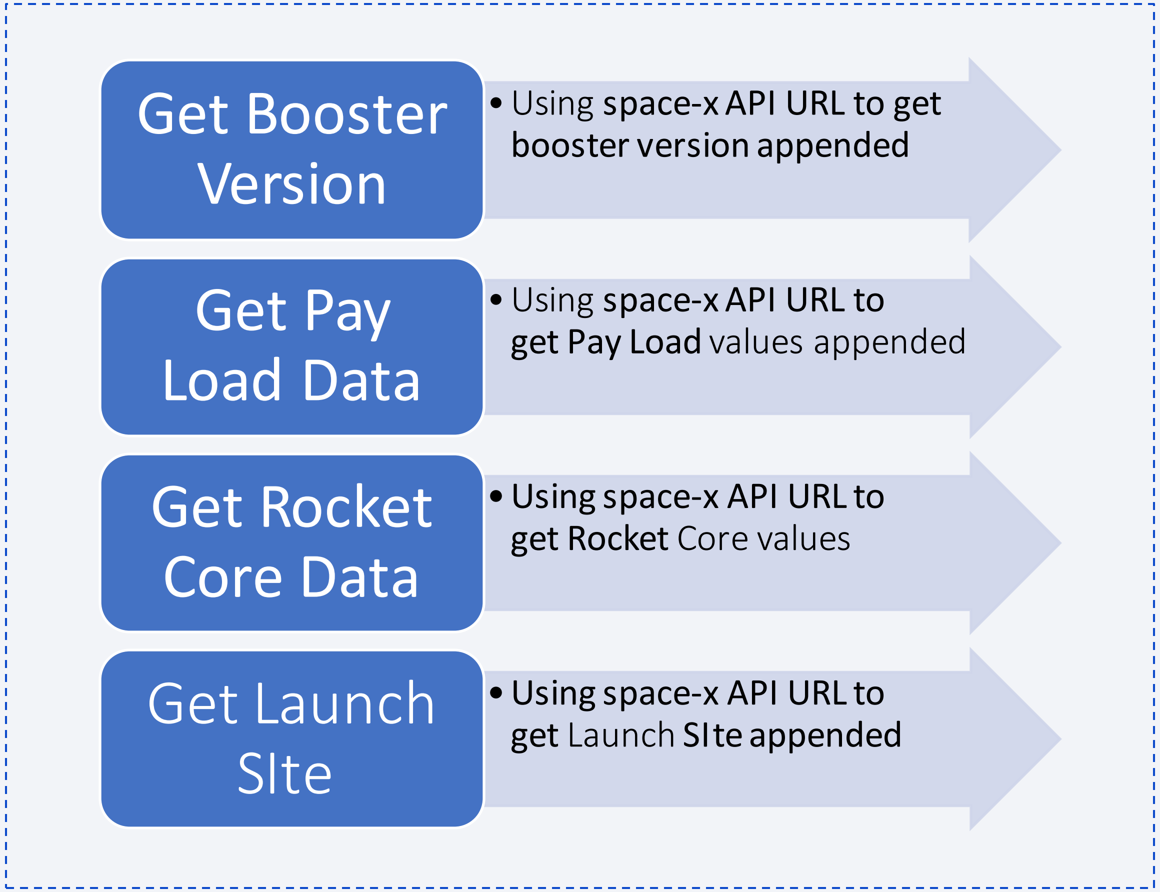


https://1drv.ms/b/s!AoRSVscc8IEsgzzdv1KgAHdyjRUR?e=93sbnB

Data Collection – SpaceX API

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

- Add the GitHub URL of the completed SpaceX API calls notebook (must include completed code cell and outcome cell), as an external reference and peer-review purpose

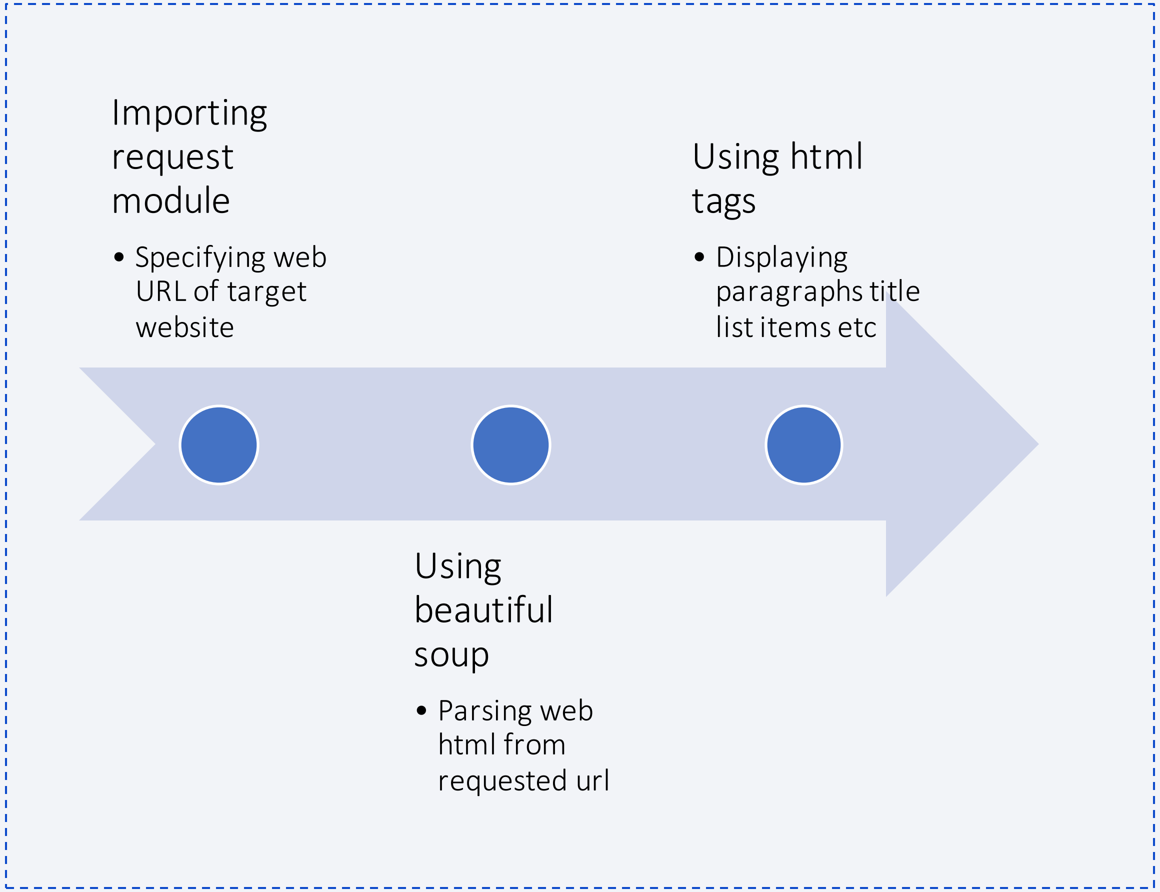


https://1drv.ms/b/s!AoRSVscc8IEsgzzdv1KgAHdyjRUR?e=93sbnB

Data Collection - Scraping

- Present your web scraping process using key phrases and flowcharts

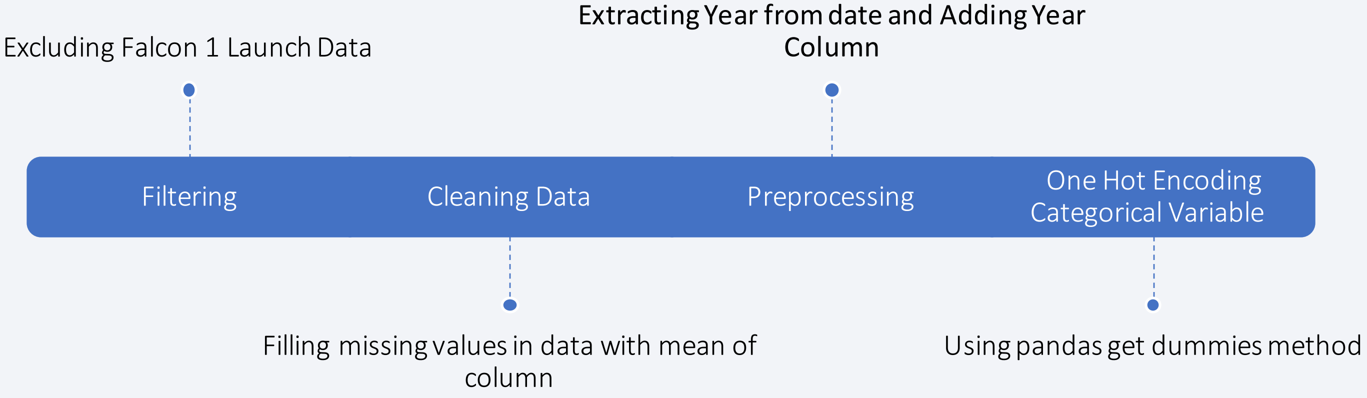
- Add the GitHub URL of the completed web scraping notebook, as an external reference and peer-review purpose



https://1drv.ms/b/s!AoRSVscc8IEsgzzdv1KgAHdyjRUR?e=93sbnB

Data Wrangling

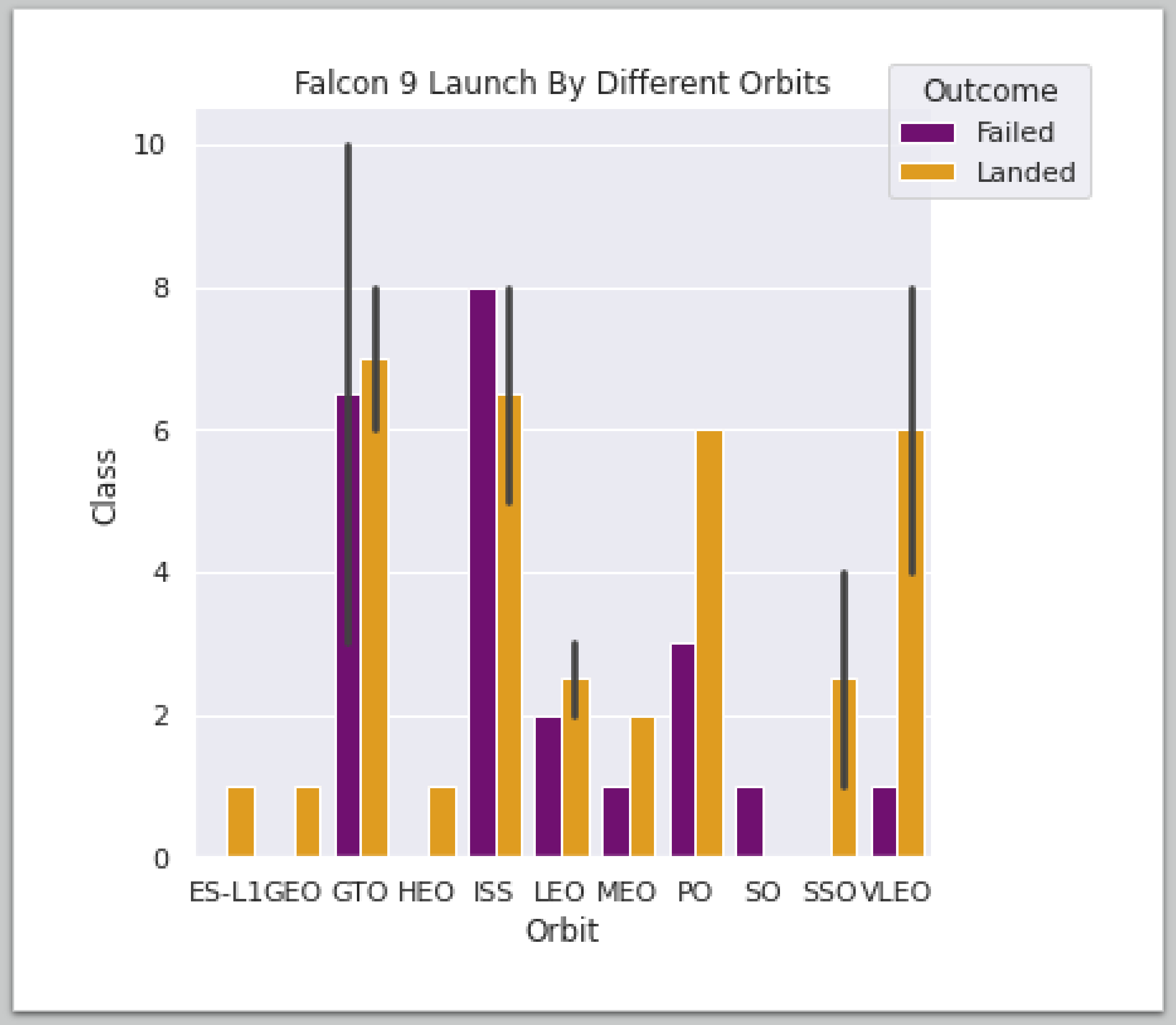
- Describe how data were processed
- You need to present your data wrangling process using key phrases and flowcharts
- Add the GitHub URL of your completed data wrangling related notebooks, as an external reference and peer-review purpose



https://1drv.ms/b/s!AoRSVscc8IEsgzzdv1KgAHdyjRUR?e=93sbnB

EDA with Data Visualization

- We plotted several charts to understand continuous values relationship with categorical data especially in case of orbits type with payload masses.
- Charts are also created using grouping some columns to give them exclusive focus and displaying them to show a particular insight of a chosen scenario.



https://1drv.ms/b/s!AoRSVscc8IEsgz1bx-BTIbfh9UxT?e=bwXgAI

EDA with SQL

- select min(payload_mass__kg_) from SPACEXTBL
- Give us minimum continuous value of falcon 9 payload.

- SELECT avg(PAYLOAD_MASS__KG_) as Average_Payload_Mass from SPACEXTBL
- Gives us average masses of all launches since 2013

- SELECT * FROM SPACEXTBL where ('Orbit')='LEO' LIMIT 5
- Gives us most recent records of falcon 9 launch into Low Earth Orbit

Build an Interactive Map with Folium

- Used Folium Map method specified location in latitude and longitudes and set zoom
- We want to add circle to make our given location coordinate prominent also added marker:
- to help in giving identity elements like launch site success rate total launches and pop ups
- Got an exposure of formula for longitude and latitude calculation.
- lat1 = radians(lat1) ; lon1 = radians(lon1); lat2 = radians(lat2); lon2 = radians(lon2)
- dlon = lon2 – lon1 ; dlat = lat2 – lat1
- a = sin(dlat / 2)**2 + cos(lat1) * cos(lat2) * sin(dlon / 2)**2
- c = 2 * atan2(sqrt(a), sqrt(1 - a))
- distance = R * c

https://1drv.ms/b/s!AoRSVscc8IEsgz4as2jXxPCpfcbT?e=0hwbKc

Build a Dashboard with Plotly Dash

- We used plotly express method to plot our sub data frame
- Used dash components like doc dropdown method and doc slider specifying start and ending range with increment size too
- Also use doc graph method

- Setting up app layout and using Html Div method for background color H1 heading or title of visualization with sub heading inside children list having H1 and Div method associated with html

Predictive Analysis (Classification)

- We first scaled data using sci-kit learn standard scaler providing values using fit method and then transform method
- Train test data set using sci-kit learn to distribute data for training and some data for validation using sci-kit-learn model selection method
- To get best possible accuracy we try fitting our models Grid Search Cross Validation (from sklearn.model_selection import GridSeatchCV Method.

Logistic Regression

- From sklearn.linear_model import LogisticRegression

Support Vector Machines

- From sklearn.svm import SVC

Decision Tree Classifier

- From sklearn.tree import DecisionTreeClassifier

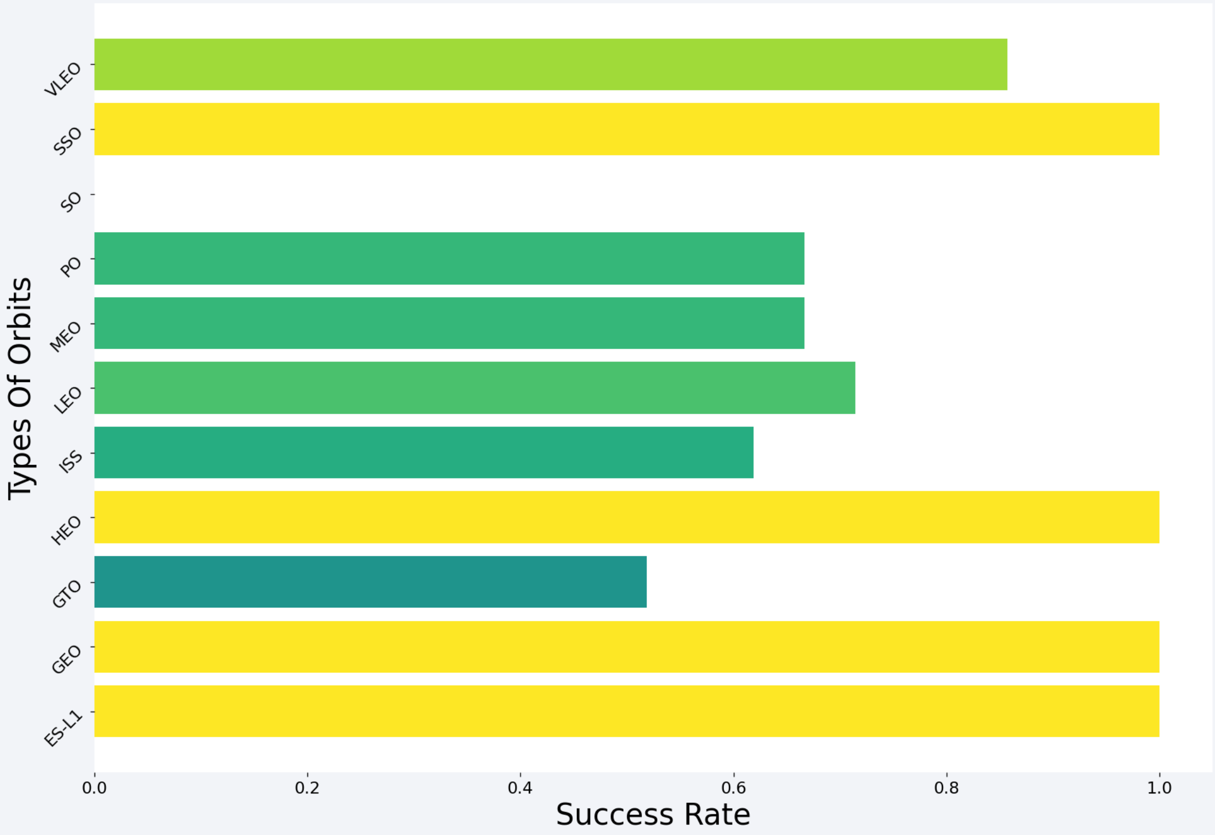
K Nearest Neighbors classification

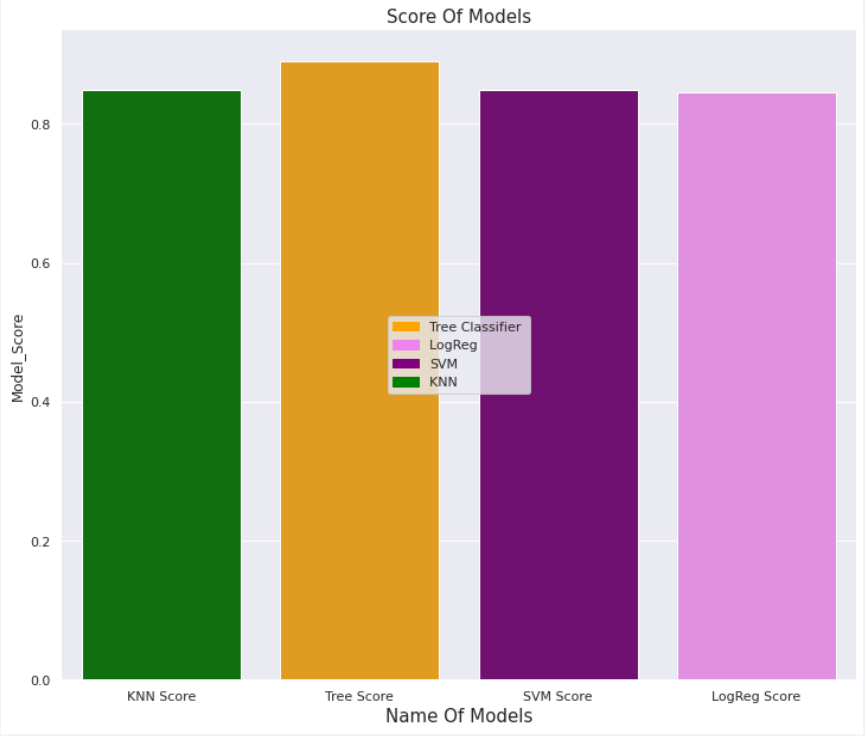
- From sklearn.neighbors import KNeighborsClassifier

https://1drv.ms/b/s!AoRSVscc8IEsgz-j2MZVft0YW7Ii?e=DYPkZQ

Results

- Success Rate By Orbits





- Decision Tree Classifier Dominance

Section 2

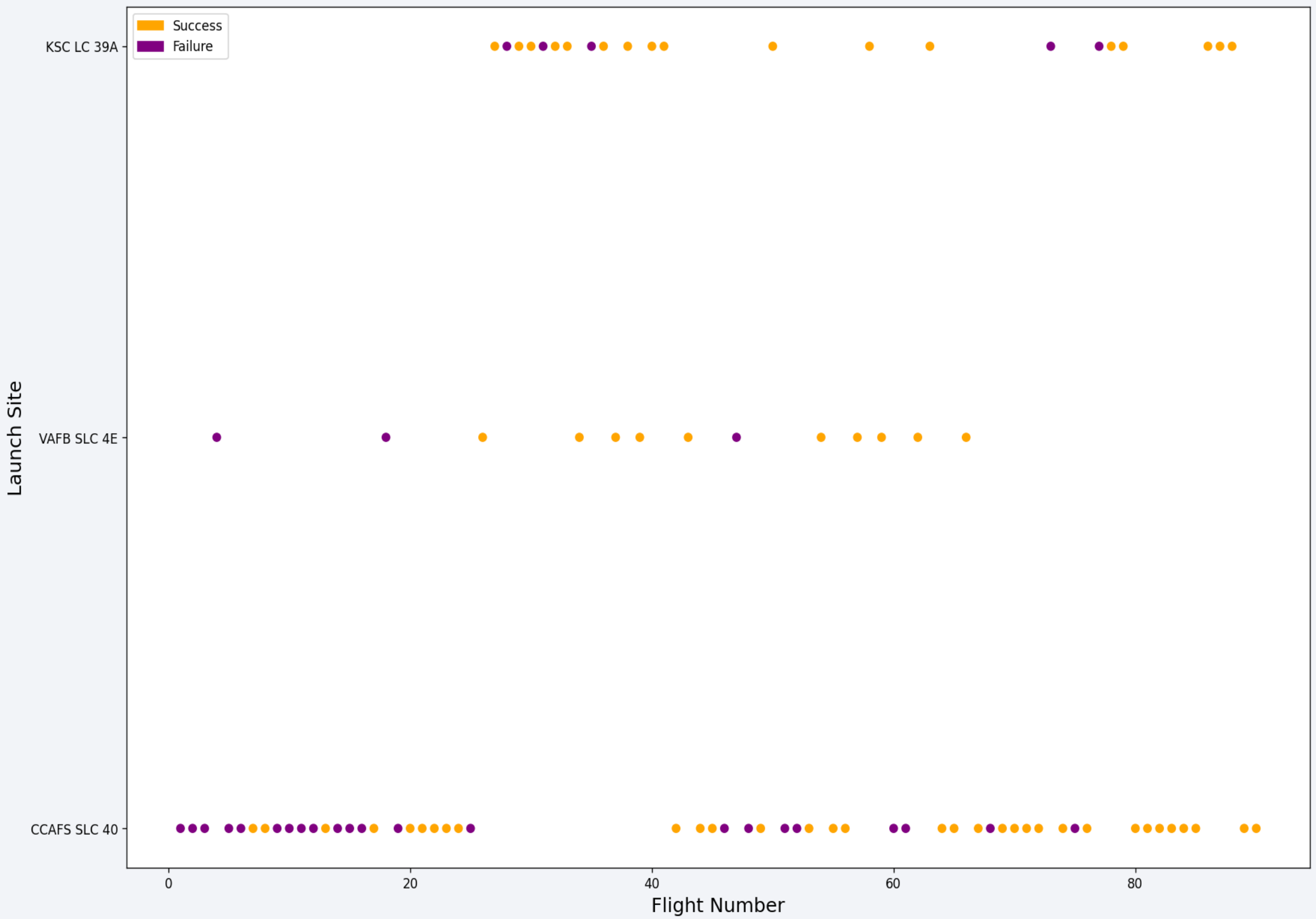
Insights drawn from EDA

Flight Number vs. Launch Site

- You can see consecutive pattern for each launch site in Flight Number vs. Launch Site scatter plot

- Most consecutive flight launch pattern is shown by CCAFS LC 40

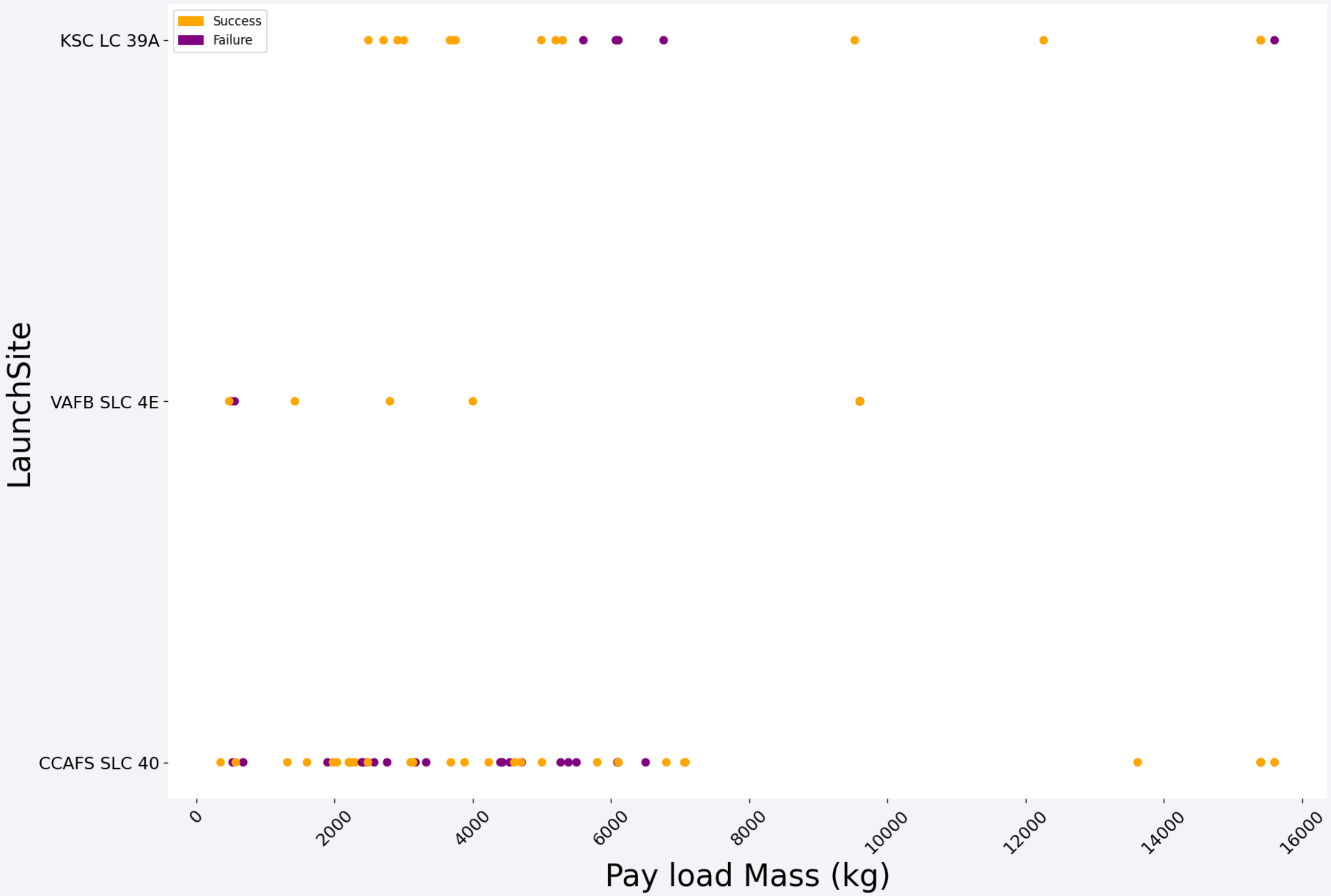
- Least consecutive pattern is shown by VAFB SLC 4E



Payload vs. Launch Site

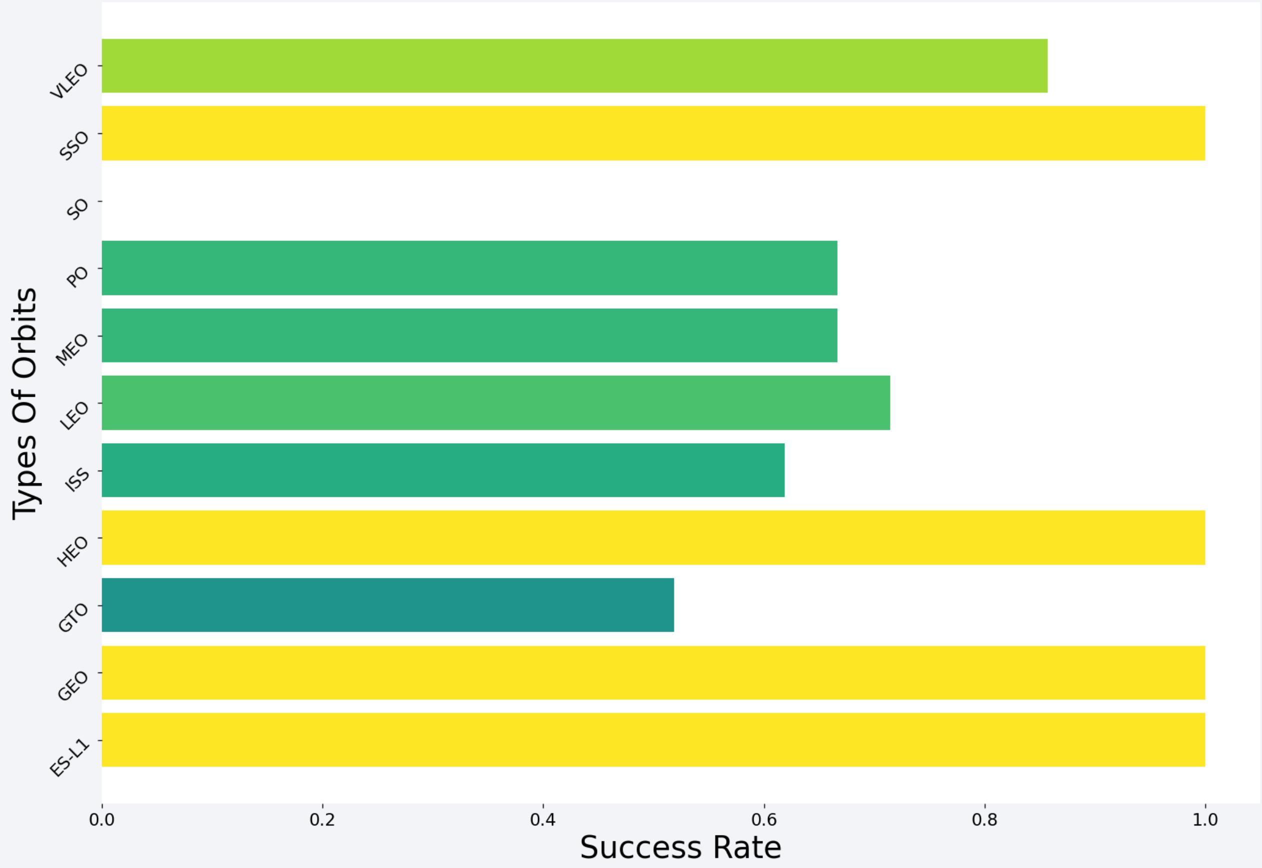
- Here we shown a scatter plot of Payload vs. Launch Site

- We can clearly see purple spots are more common in low to medium range of payload masses especially in CCAFS SLC 40 launch site but after repetitive launches we measured a big change.



Success Rate vs. Orbit Type

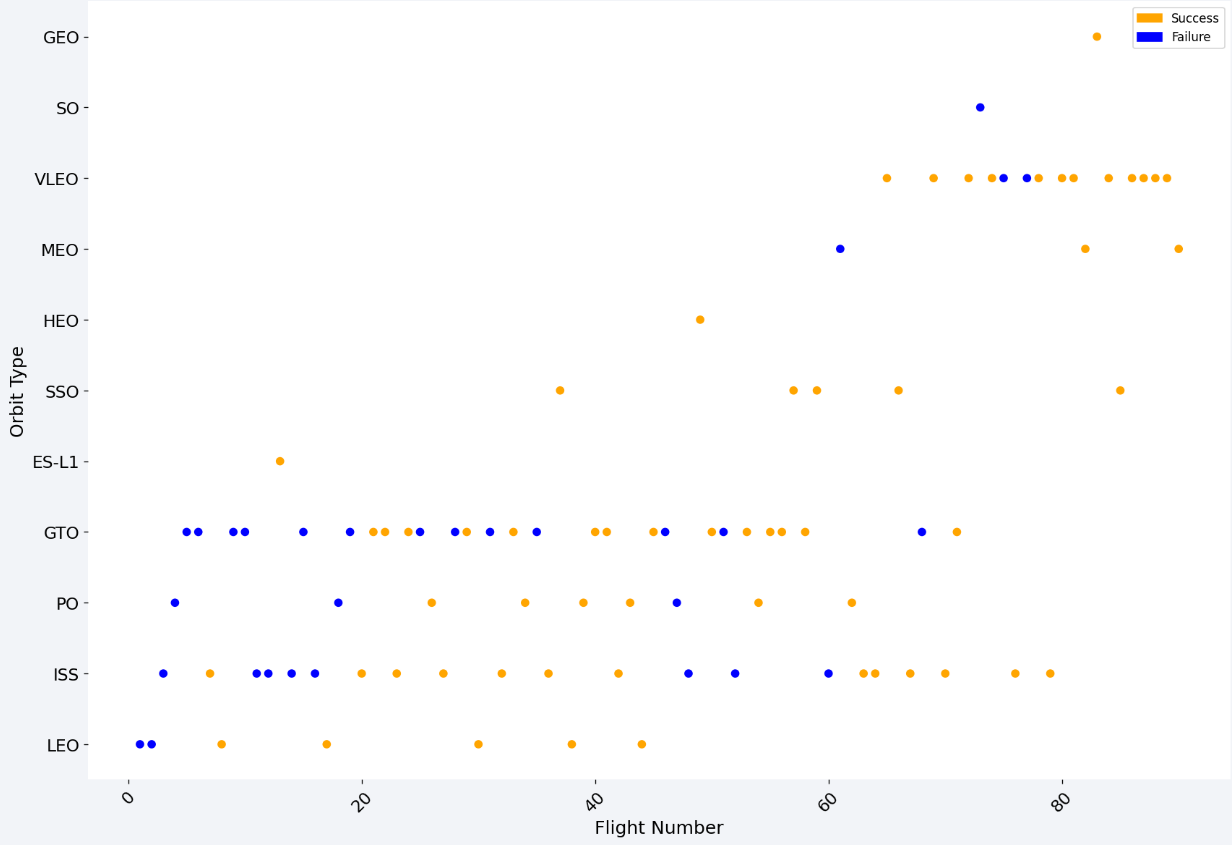
- You can take a look at the success rate of each orbit type
- Some orbits never went to failed outcome and green bars showed improved efficiency with passage of time



Flight Number vs. Orbit Type

- Here we can see Flight number vs. Orbit type of all launches

- It is clear that successful outcomes are increasing and become dominating over fail outcomes with successive flight numbers

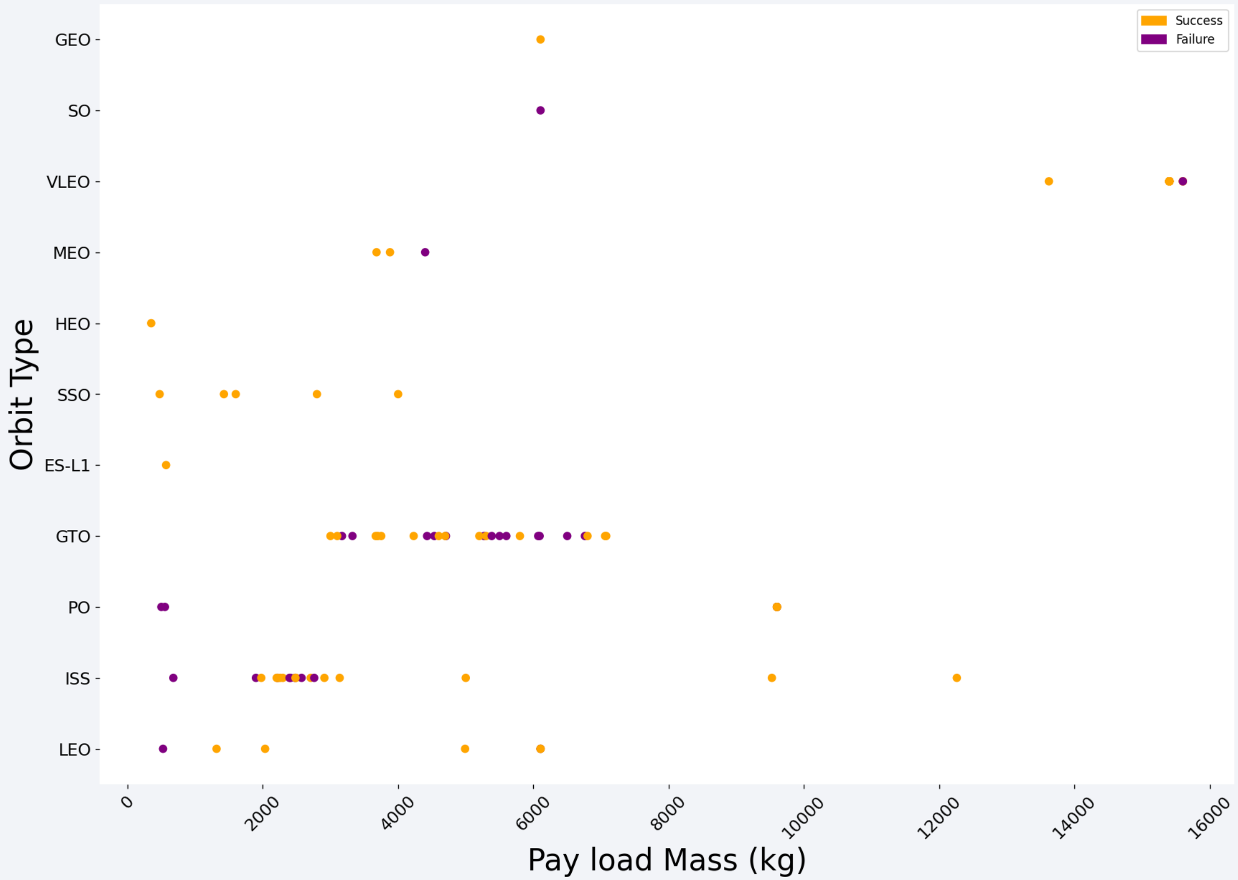


Payload vs. Orbit Type

- Take a sneak peek of a scatter plot of payload vs. orbit type

- Low Height Orbits are working even with very large payload masses

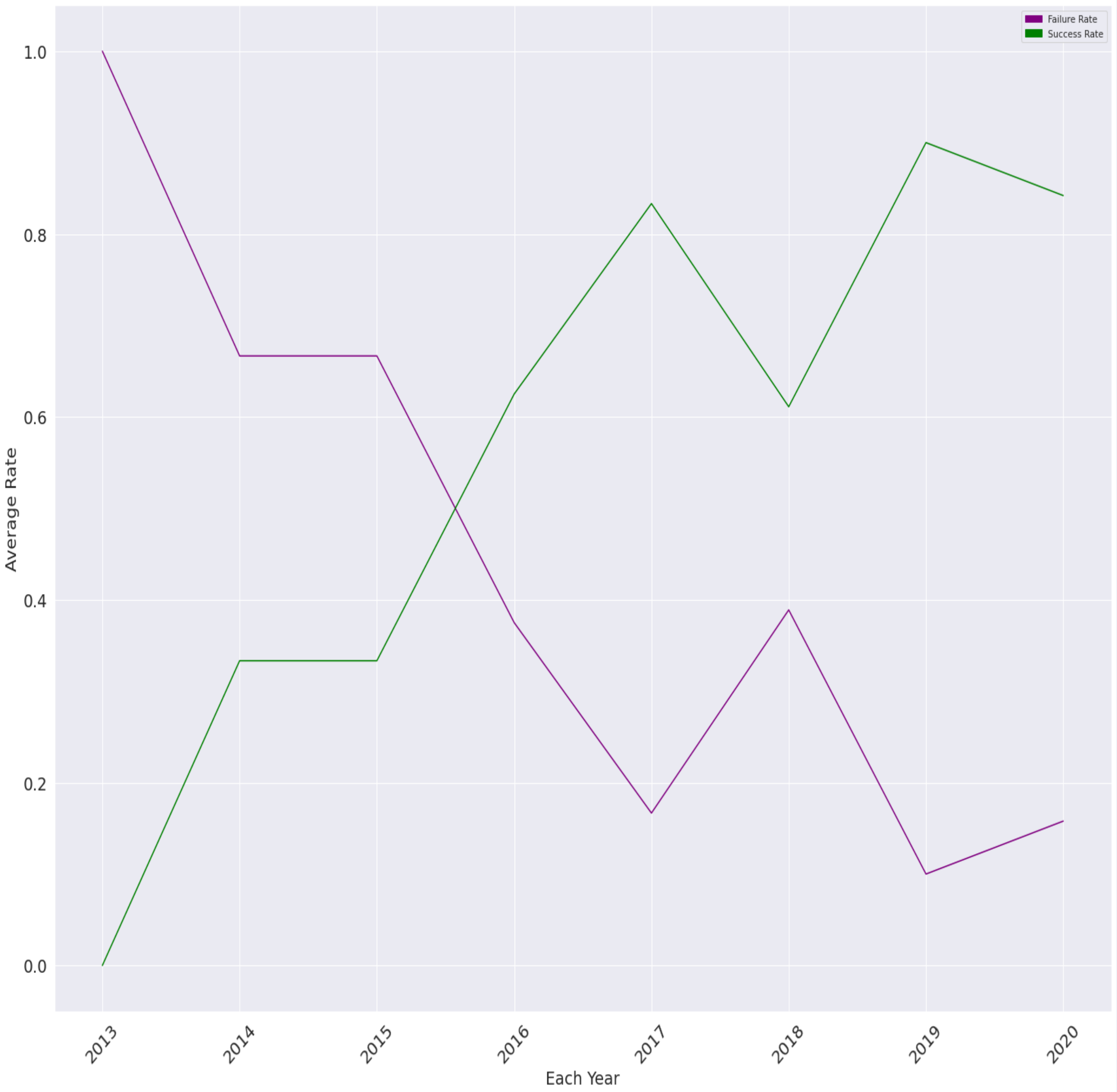
- Most number of launches of orbits between 2,000 and 10,000



Launch Success Yearly Trend

- *You can see falcon 9 reusability success rate is improving every year.*

- **We are ignoring scientific data which also holds importance which is kept confidential by SpaceX here we focus on mass of payload and launch orbit more closely of Falcon 9 data since 2013**



All Launch Site Names

- Used group by pandas data frame method to collect all launch sites with their success rate
  - VAFB SLC 4E
  - KSC LC 39A
  - CCAFS SLC 40

```
site_success = df.groupby('LaunchSite' , as_index= False )['Class'].mean()
# = ['CCAFS SLC 40', 'KSC LC 39A','VAFB SLC 4E  ']
total =  sum(site_success['Class'])
site_success['Percentage'] =  site_success['Class'].map(lambda x : (x/total)*100 )
site_success
```

	LaunchSite	Class	Percentage
0	CCAFS SLC 40	0.400000	46.617767
1	KSC LC 39A	0.227273	26.487368
2	VAFB SLC 4E	0.230769	26.894866

Launch Site Names Begin with 'KSC'

- To make sure our columns type we used data frame d-types if string operation is to be performed then type of columns must be changed to string first then do any string operation.
- We use substring method str. Contains to find all column entries which include 'KSC'
- In order to present only 5 of them we stored results in variable and sliced them [:5]

```
df['LaunchSite'] = df['LaunchSite'].astype('string')

z = df[ df['LaunchSite'].str.contains('KSC')  ]
z[:5]
```

	FlightNumber	Date	BoosterVersion	PayloadMass	Orbit	LaunchSite	Outcome	Flights	GridFins	Reused	Legs	
26	27	2017-02-19	Falcon 9	2490.000000	ISS	KSC LC 39A	1	1	True	False	True	5e9(
27	28	2017-03-16	Falcon 9	5600.000000	GTO	KSC LC 39A	0	1	False	False	False	
28	29	2017-03-30	Falcon 9	5300.000000	GTO	KSC LC 39A	1	2	True	True	True	5e9(
29	30	2017-05-01	Falcon 9	6104.959412	LEO	KSC LC 39A	1	1	True	False	True	5e9(
30	31	2017-05-15	Falcon 9	6070.000000	GTO	KSC LC 39A	0	1	False	False	False	

Total Payload Mass

- We simply used sum function over falcon 9 to obtain one single value

```
sum(df['PayloadMass'])
```

```
549446.3470588236
```

- Almost 550 hundred thousand mass kg was carried by boosters when summed up all launches of falcon 9

Average Payload Mass by F9

- Using falcon 9 data frame we calculated average with as usual method.

```
total =  sum(df['PayloadMass'])
num = len((df['PayloadMass']))
avg = total/num
avg
```

```
6104.959411764707
```

- We came to know average mass of our payload close to 6 thousand kg

First Successful Ground Landing Date

- We set outcome equal to 1 means landed and to make sure it has landing pad data available we used drop NA pandas method to not show null values
- We decided to include some important information only to ease readability

```
# df[df['LandingPad'] != NaN ]
df_drop_na = df[df['Class'] == 1]

df_drop_na[['Date' , 'Class', 'LandingPad']].dropna()
```

	Date	Class	LandingPad
16	2015-12-22	1	5e9e3032383ecb267a34e7c7
19	2016-04-08	1	5e9e3032383ecb6bb234e7ca
20	2016-05-06	1	5e9e3032383ecb6bb234e7ca
21	2016-05-27	1	5e9e3032383ecb6bb234e7ca
22	2016-07-18	1	5e9e3032383ecb267a34e7c7
23	2016-08-14	1	5e9e3032383ecb6bb234e7ca
25	2017-01-14	1	5e9e3033383ecbb9e534e7cc
26	2017-02-19	1	5e9e3032383ecb267a34e7c7
28	2017-03-30	1	5e9e3032383ecb6bb234e7ca
29	2017-05-01	1	5e9e3032383ecb267a34e7c7
31	2017-06-03	1	5e9e3032383ecb267a34e7c7

Successful Drone Ship Landing with Payload between 4000 and 6000

- The names of boosters which have successfully landed on drone ship and had payload mass greater than 4000 but less than 6000 all are ***Falcon 9***

```
df_drop_na['PayloadMass'].astype('int')
```

```
df_payloadrange =  df_drop_na[df_drop_na['PayloadMass'] >= 4000 ]
df_payloadrange = df_payloadrange[df_payloadrange['PayloadMass'] <= 6000 ]
df_payloadrange[['FlightNumber','Date','BoosterVersion','PayloadMass','Orbit','LaunchSite' ,'Outcome']].dropna()
```

	FlightNumber	Date	BoosterVersion	PayloadMass	Orbit	LaunchSite	Outcome
20	21	2016-05-06	Falcon 9	4696.0	GTO	CCAFS SLC 40	True ASDS
23	24	2016-08-14	Falcon 9	4600.0	GTO	CCAFS SLC 40	True ASDS
28	29	2017-03-30	Falcon 9	5300.0	GTO	KSC LC 39A	True ASDS
37	38	2017-09-07	Falcon 9	4990.0	LEO	KSC LC 39A	True RTLS
39	40	2017-10-11	Falcon 9	5200.0	GTO	KSC LC 39A	True ASDS
44	45	2018-01-31	Falcon 9	4230.0	GTO	CCAFS SLC 40	True Ocean
54	55	2018-08-07	Falcon 9	5800.0	GTO	CCAFS SLC 40	True ASDS
58	59	2018-12-03	Falcon 9	4000.0	SSO	VAFB SLC 4E	True ASDS
69	70	2019-12-05	Falcon 9	5000.0	ISS	CCAFS SLC 40	True ASDS

Total Number of Successful and Failure Mission Outcomes

- Calculate the total number of successful and failure mission outcomes
- Present your query result with a short explanation here

```
orbit_df = stage_2_df.groupby('Orbit' , as_index= False , sort=True )['Class'].count()
print(sum(orbit_df['Class']))
orbit_df.sort_values('Class' , ascending=False)
```

60

	Orbit	Class
2	GTO	14
4	ISS	13
9	VLEO	12
7	PO	6
5	LEO	5
8	SSO	5
6	MEO	2
0	ES-L1	1
1	GEO	1
3	HEO	1

```
failed_orbitdf = failed_df.groupby('Orbit' , as_index= False , sort=True )['Class'].count()
print(sum(failed_orbitdf['Class']))
failed_orbitdf.sort_values('Class' , ascending=False)
```

30

	Orbit	Class
0	GTO	13
1	ISS	8
4	PO	3
2	LEO	2
6	VLEO	2
3	MEO	1
5	SO	1

Boosters Carried Maximum Payload

- The names of the booster which have carried the maximum payload mass is **Falcon 9**
- We applied sort values on data frame column payload mass and set in descending order

```
[65] # Booster Carried Max Payload
     df.sort_values('PayloadMass' , ascending=False )
```

	FlightNumber	Date	BoosterVersion	PayloadMass	Orbit	LaunchSite	Outcome	Flights	GridFins	Reused	Legs	
68	69	2019-11-11	Falcon 9	15600.0	VLEO	CCAFS SLC 40	1	4	True	True	True	5e9e
73	74	2020-01-29	Falcon 9	15600.0	VLEO	CCAFS SLC 40	1	3	True	True	True	5e9e
76	77	2020-03-18	Falcon 9	15600.0	VLEO	KSC LC 39A	0	5	True	True	True	5e9e
71	72	2020-01-07	Falcon 9	15400.0	VLEO	CCAFS SLC 40	1	4	True	True	True	5e9e

2017 Launch Records

- Following are the records which will displaying the month names, Launch Sites and Booster Version for the months in year 2017
- 16 March , 2017
- 15 May , 2017
- 5 July , 2017

```
two17 = df[df['Year'] == 2017 ]
two17 = two17[two17['Class'] == 1  ]
two17[['Date','LaunchSite', 'BoosterVersion','Longitude','Latitude']]
```

	Date	LaunchSite	BoosterVersion	Longitude	Latitude
27	2017-03-16	KSC LC 39A	Falcon 9	-80.603956	28.608058
30	2017-05-15	KSC LC 39A	Falcon 9	-80.603956	28.608058
34	2017-07-05	KSC LC 39A	Falcon 9	-80.603956	28.608058

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

- Following image show us the ranked count of successful landing outcomes between the date 2010-06-04 and 2017-03-20 in descending order

- We find out pandas date time type and group by with year method as important ingredient.

- Then sort values using descending order.

```python
# Changing date type to Pandas Date time
df['Date'] = pd.to_datetime(df['Date'])
df.dtypes

# Start Date And End Date
start_date = '2010-06-04'
end_date = '2017-03-20'
mask = (df['Date'] >= start_date ) & (df['Date'] <= end_date  ) & (df['Class'] == 1 )
outcomes_10_17 = df.loc[mask]
# Counting By Year To Show Them In Descending Order
outcomes_10_17_rank = outcomes_10_17.groupby('Year')['Outcome'].count()
# it returned us a series converting back into Data Frame
outcomes_10_17_rank =  pd.DataFrame(outcomes_10_17_rank)
outcomes_10_17_rank.sort_values('Outcome' , ascending=False)
```

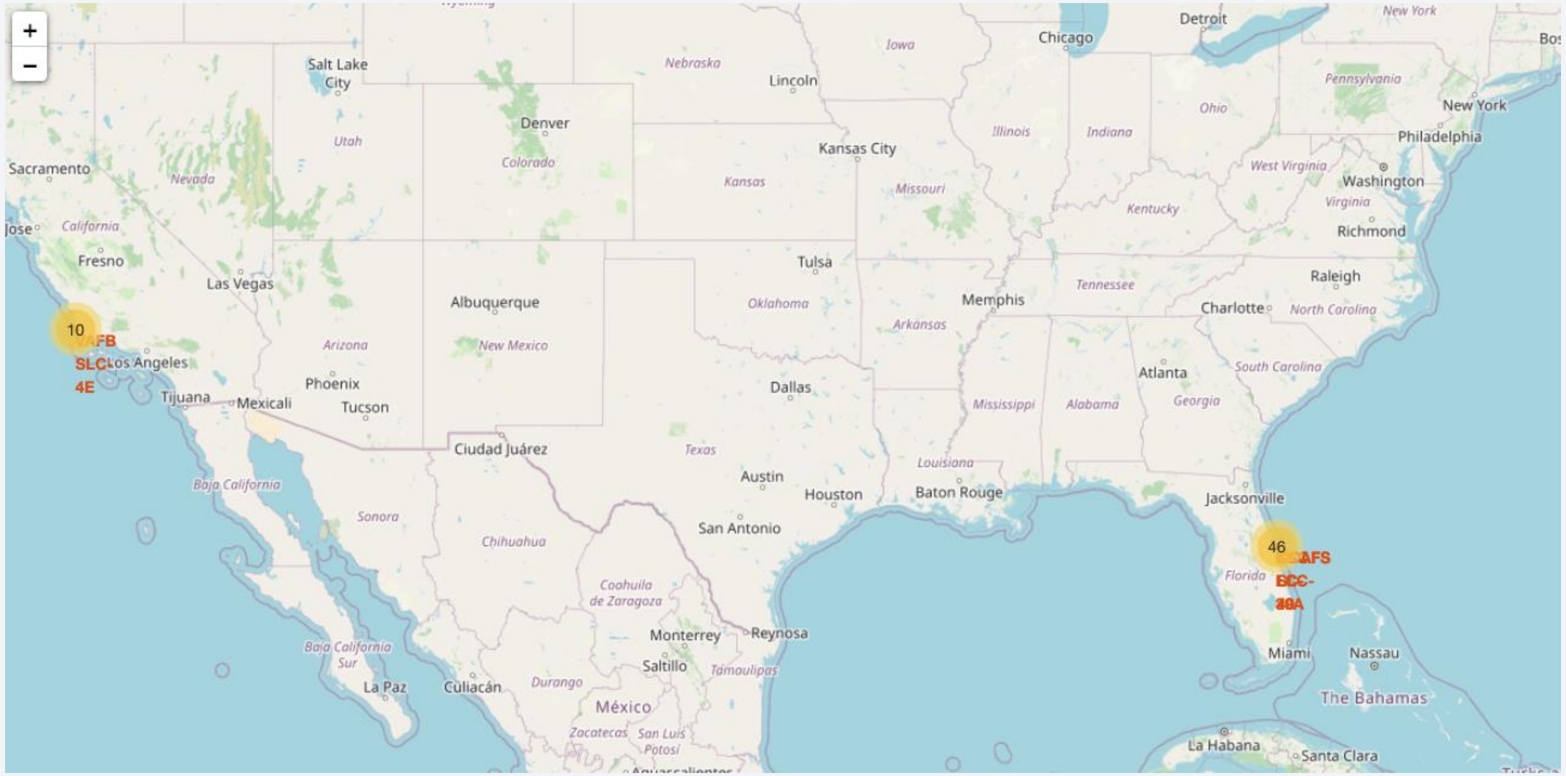
	Outcome
Year	
2016	5
2014	2
2015	2
2017	2

Section 3

Launch Sites Proximities Analysis

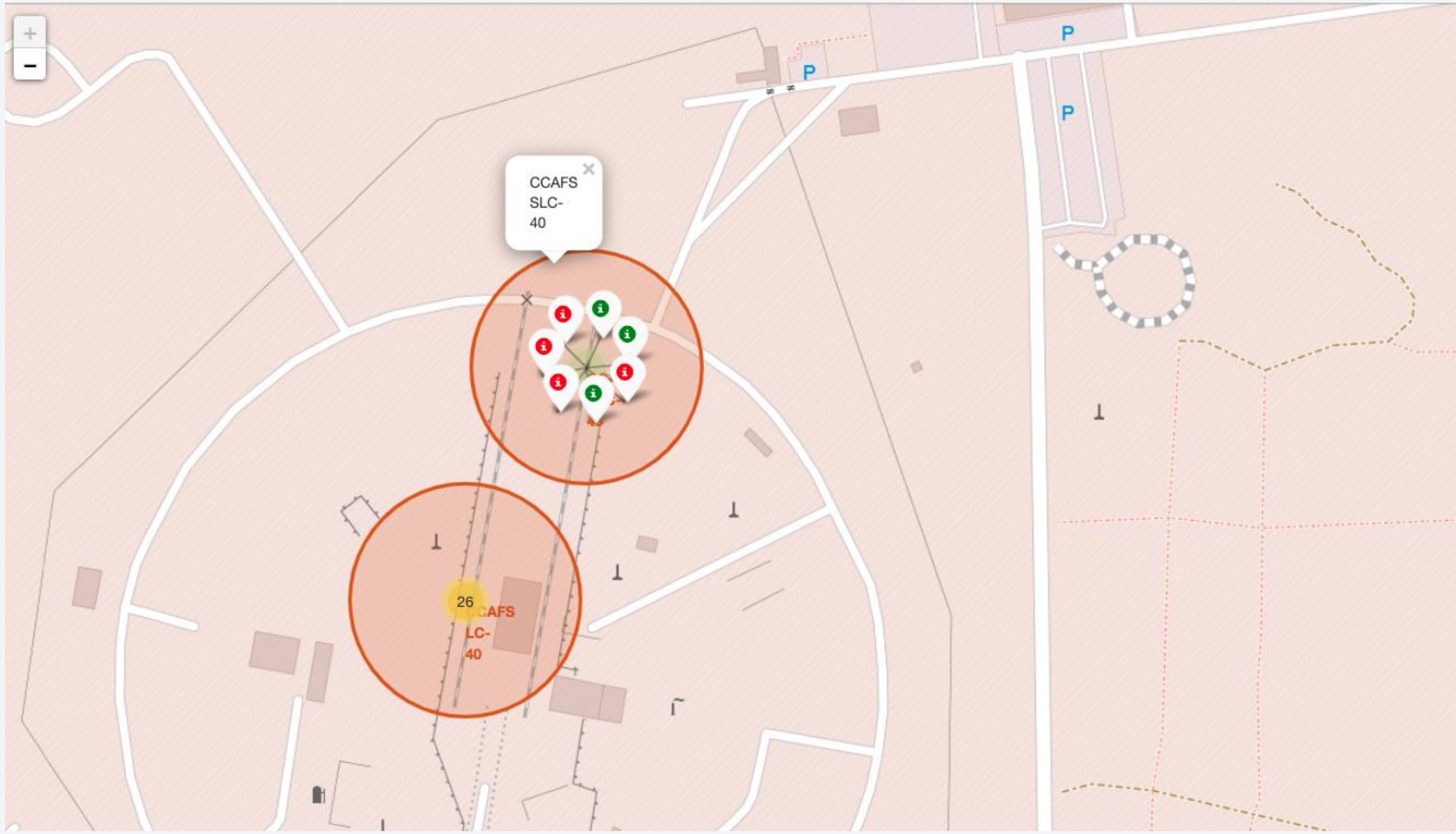
Map Of U.S East Coast And West Coast Launch Sites

Circles are used to display launch sites as we can see they are located very close to ocean bay



Diameter Of Circle Covering Launches Coordinates

- Markers are indicating success with green color and failed ones are red.



Distance From Ocean

We marked two distinct locations that are associated with launches they both comes in CAFS SLC-40

We know gravity at sea level is bit more than

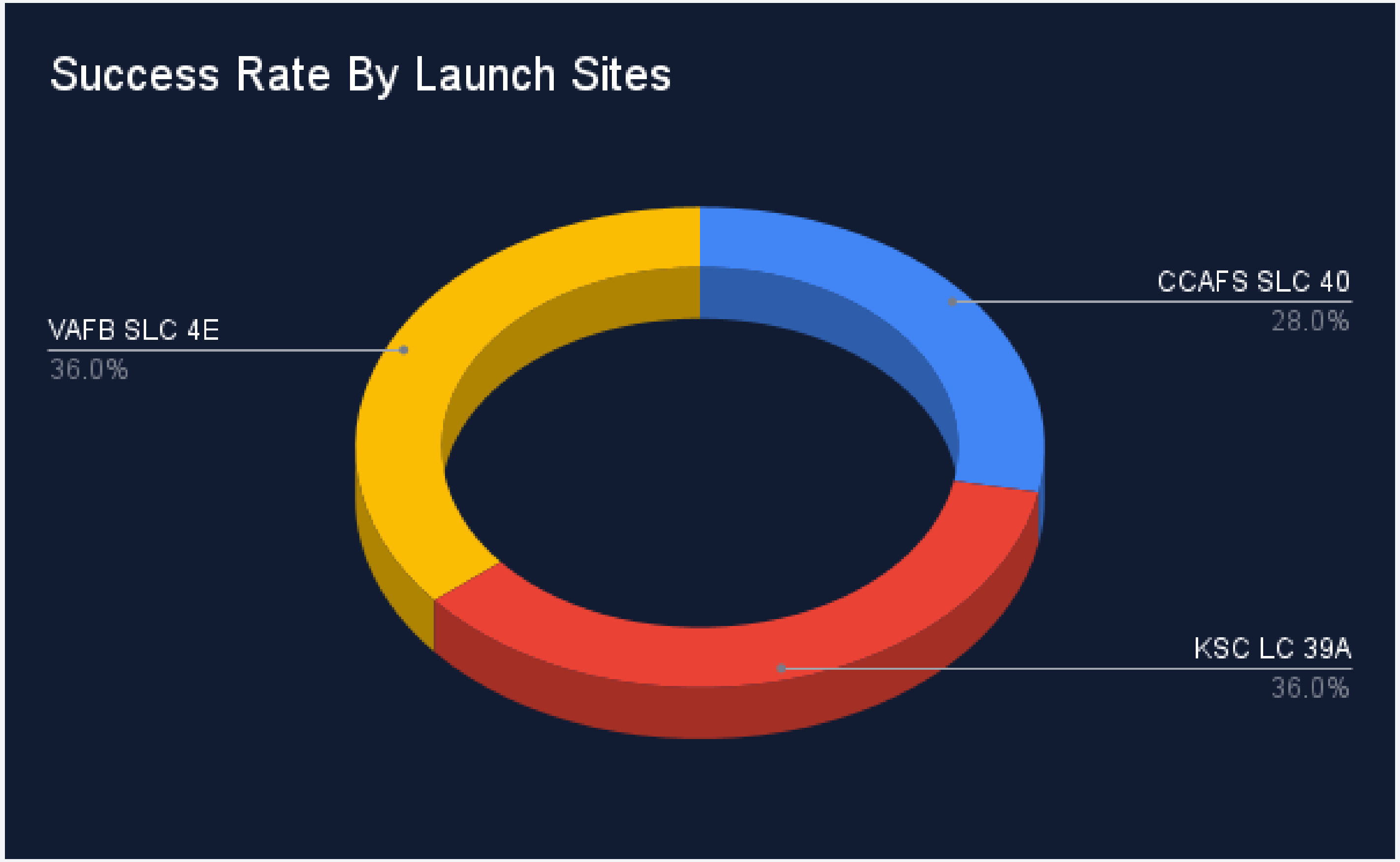
what it observed at mountainous regions

Section 4

Build a Dashboard with Plotly Dash

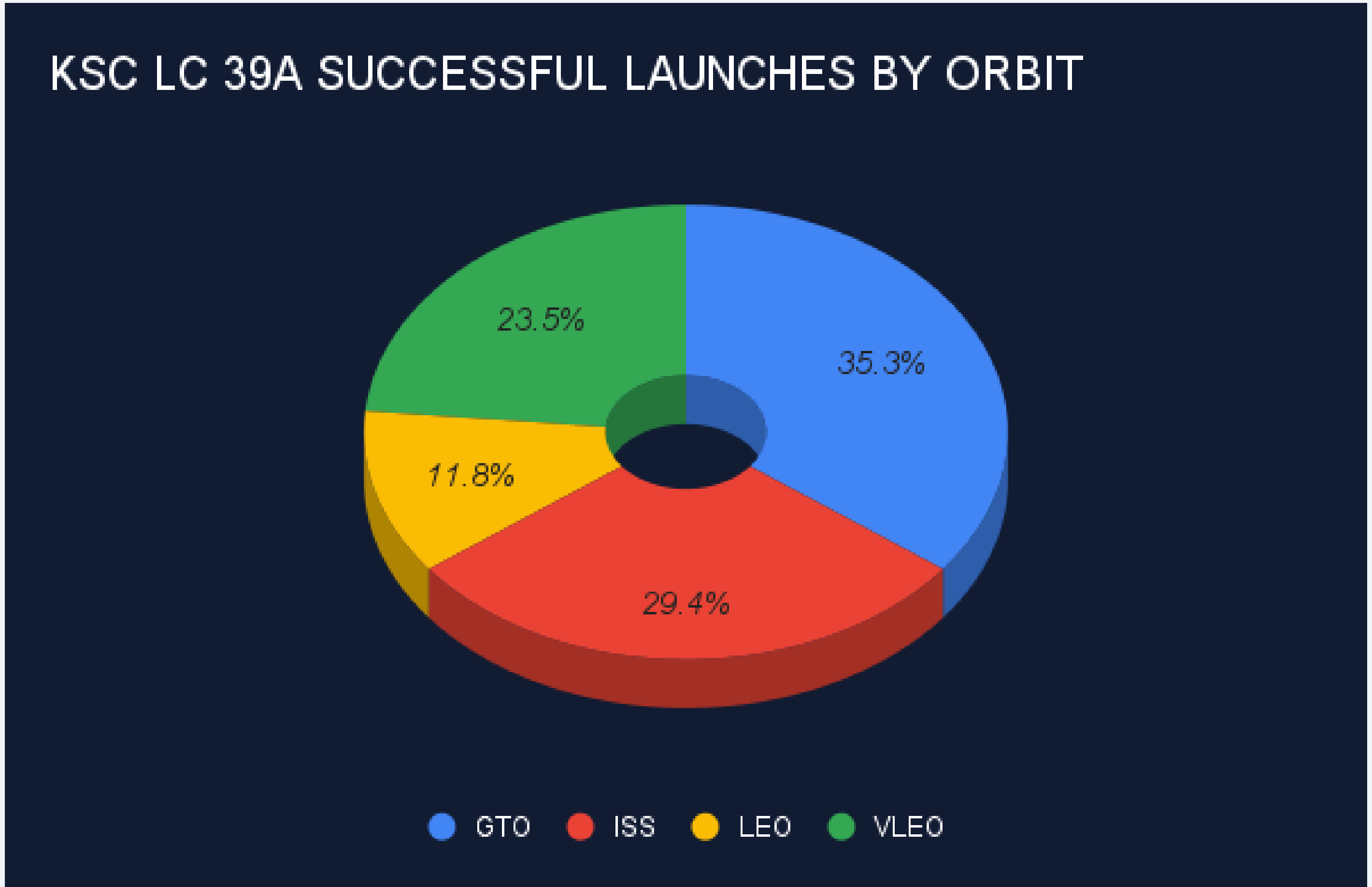
Launch Sites Rate Of Success

- VAFB SLC 4E and KSC LC 39A *topped chart* with 36% and luckily equal in success/launch ratio.
- CCAFS SLC 40 *bottoms chart* with 28% in success/launch ratio.



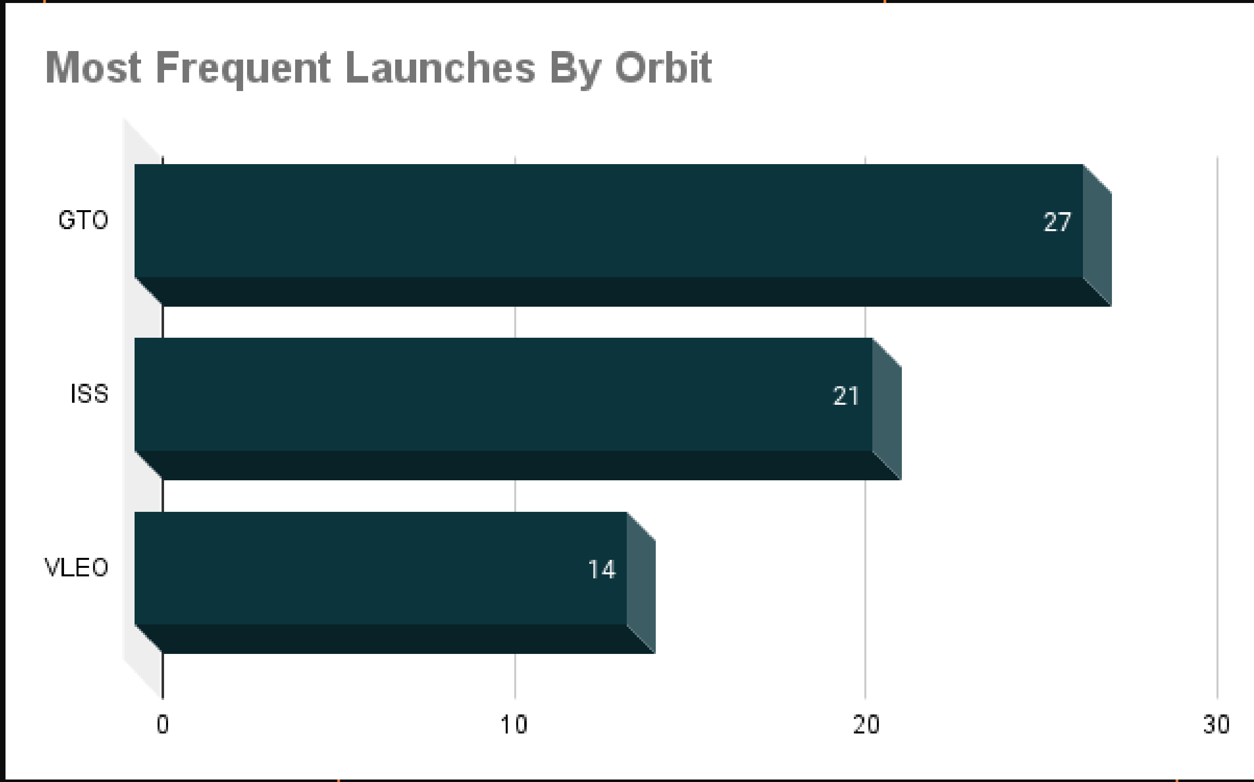
Pie Chart Of Most Successful Launch Site By Orbits

- Explain You can witness GEO orbit with 35% most contribution in success rate of KSC LC 39A then ISS with 29% contribution



Launches Most Frequently By Orbits

- Most number of successful outcomes belong to following orbits:
- GTO
  - *27 outcomes*
- ISS
  - *21 outcomes*
- VLEO
  - *14 outcomes*



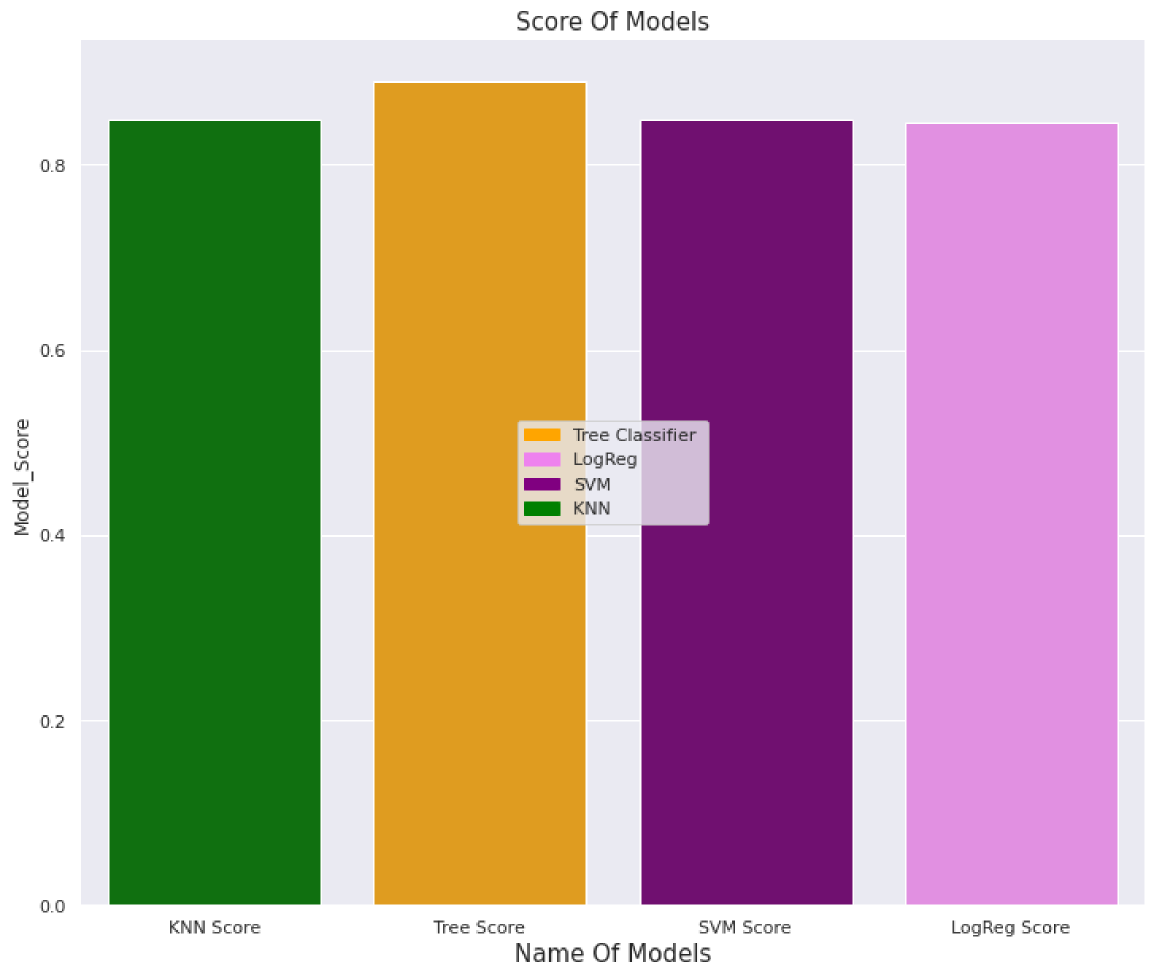
Section 5

Predictive Analysis (Classification)

Classification Accuracy

- Visualization of the built model accuracy for all classification models, in a bar chart

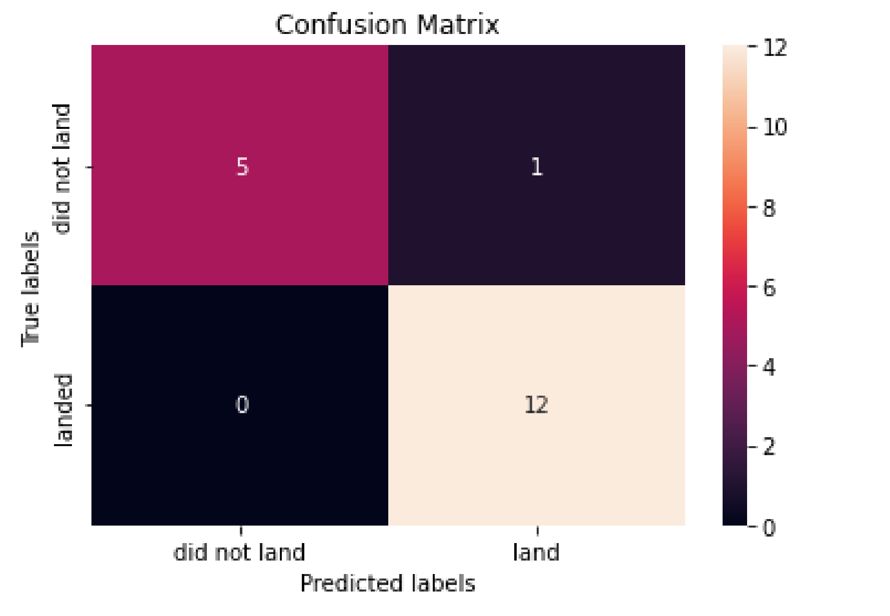
- You can see clear dominance of Decision Tree Classifier 0.87 over other models which showing almost same efficiency close to 0.84 0.85 .



Confusion Matrix

- Confusion matrix is showing very positive outcome for untrained data features which and for decision tree classifier we know it's efficiency is best one.
- This indicates with passage of time scientific hurdles are eliminated to a greater extent and outcome has become success dominant.
- For future results outcome results favor more for positivity

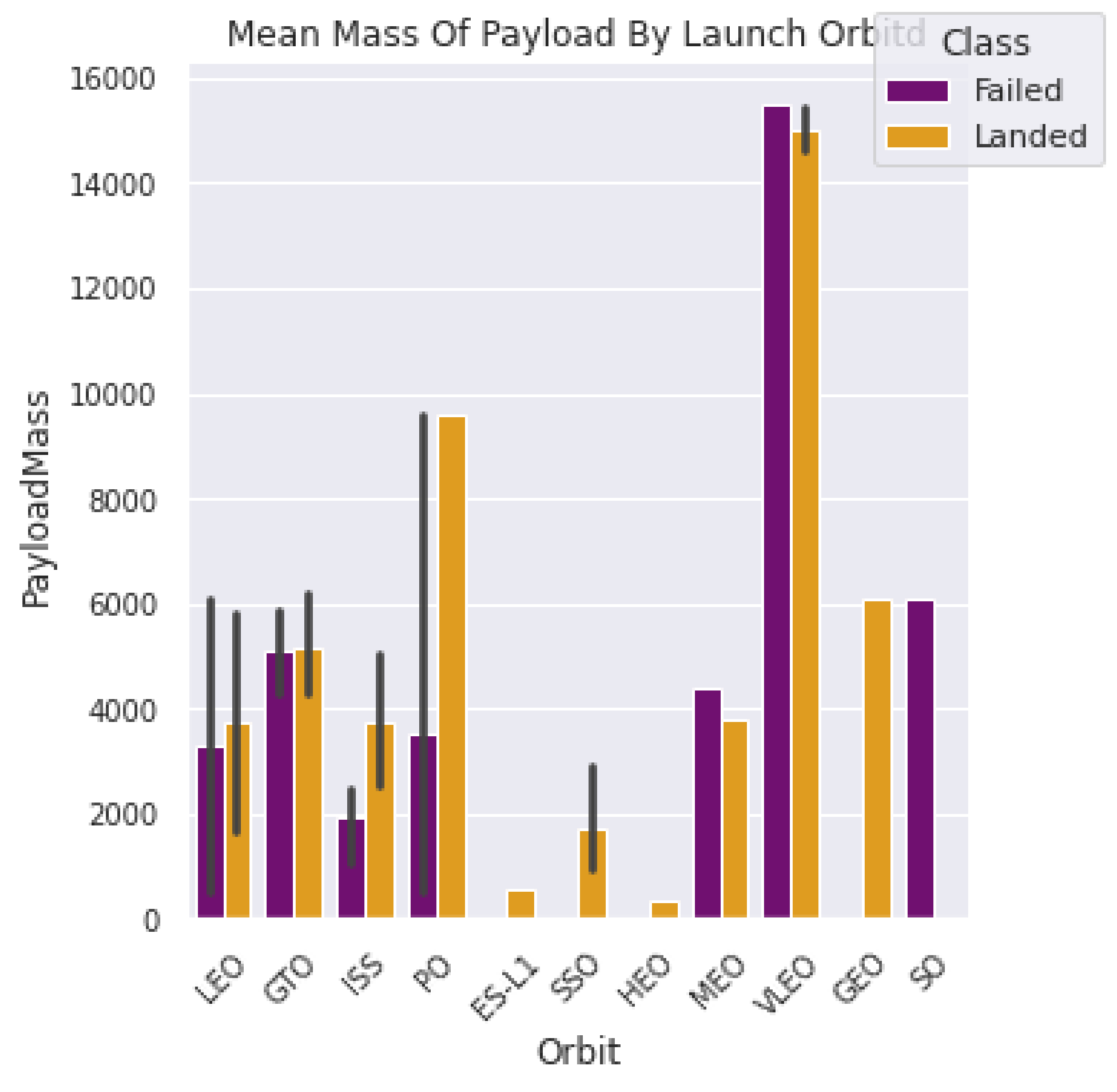
```
yhat=tree_cv.predict(X_test)
plot_confusion_matrix(Y_test,yhat)
```



Conclusions

- Poorly performing orbits launches of falcon 9 showed most improvements
- Scatter Chart of payload masses tells us variation in payload mass affects outcome of launch
- It should be noted that our model is based on statistics data of payload mass and type of orbit which are main factors responsible for stage 1 reusability so other **unexplored factors** should *not be ignored*
- Launch Sites : CCLAFS 40 showing most consistent positive outcome currently
- We should not bid for least successful orbits and avoid taking part in bidding when pay load mass is too high.
- Most favorable launch orbits looks ***PO* , *ES-L1* , *SSO* , *GEO* , *VLEO***

Appendix



Mean Mass Of Payload By Launch Orbitd
Class
Failed
Landed
PayloadMass
16000
14000
12000
10000
8000
6000
4000
2000
0
LEO
GTO
ISS
PO
ES-L1
SSO
HEO
MEO
VLEO
GEO
SO
Orbit

Thank you!