DETERMINE THE SPACEX’S FALCON 9 FIRST STAGE LANDING SUCCESSFULLY

AKASH NAVET

OUTLINE

- Executive Summary
- Introduction
- Methodology
- Results
- Conclusion

EXECUTIVE SUMMARY

- WE WANT TO DETERMINE THE SPACEX'S FALCON 9 FIRST STAGE SUCCESSFUL LANDING AND THUS WE WILL BUILD MACHINE LEARNING MODEL FOR CORRECT PREDICTION. THE CORRECT PREDICTION CAN LEAD US TO FINANCIAL ADVANTAGE.
- THE DATA WAS COLLECTED THROUGH SPACEX API AND WIKIPEDIA WEB SOURCE. EDA WAS PERFORMED WITH SQL AND VISUALIZATION. INTERACTIVE MAPS WITH FOLIUM AND PLOTLY DASH WAS USED TO UNDERSTAND DATA IN MORE DEPTH.
- DIFFERENT MACHINE MODELS WERE USED TO BEST PREDICTION. DECISION TREE CLASSIFIER WITH AN ACCURACY OF 0.889 WAS THE BEST MACHINE LEARNING MODEL BUILD TO PREDICT SUCCESSFUL LANDING.
- THERE IS A HUGE SCOPE TO IMPROVE THE MODEL.

INTRODUCTION

THE COMMERCIAL SPACE AGE IS HERE, COMPANIES ARE MAKING SPACE TRAVEL AFFORDABLE.

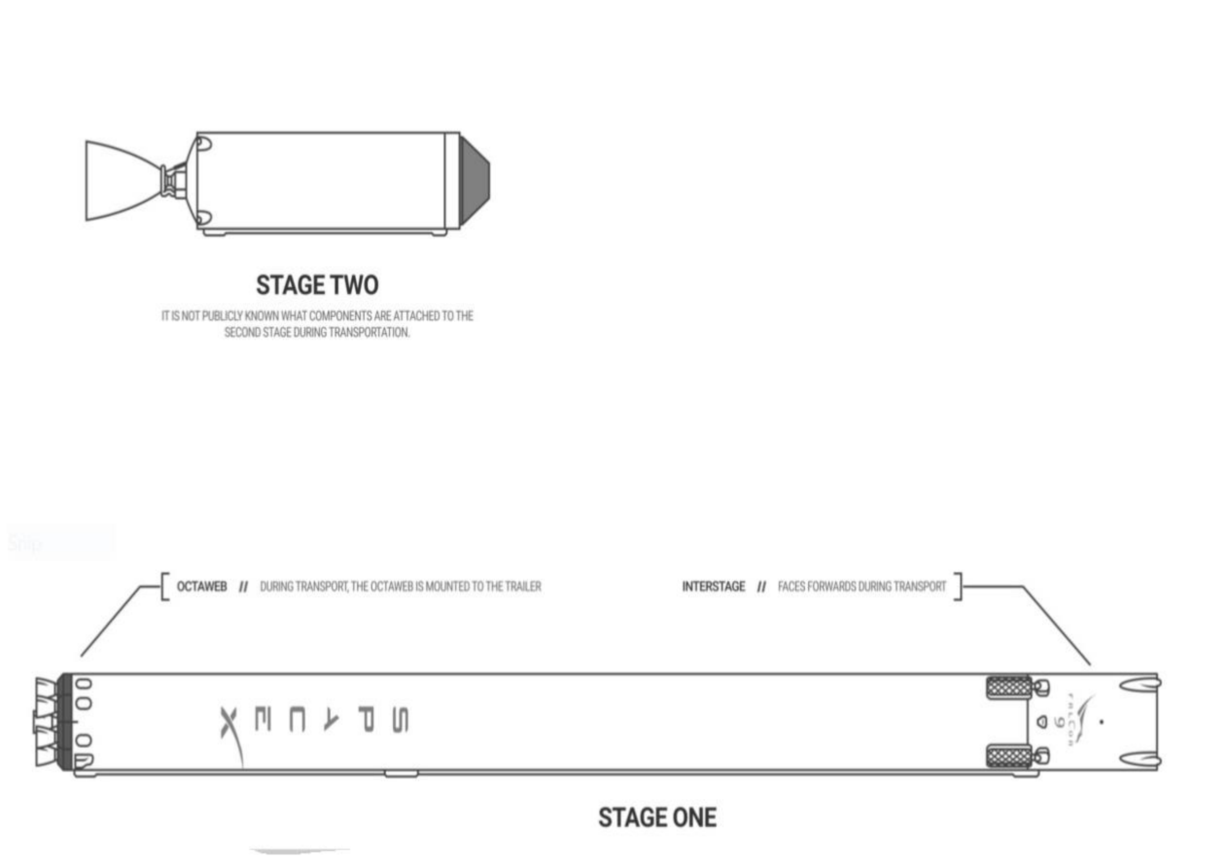
SPACEX ADVERTISES ITSELF AS MOST COST EFFECTIVE.

INTRODUCTION

- SPACEX'S CLAIMS FALCON 9 LAUNCHES WITH A COST OF $62 MILLION AS COMPARE TO OTHER PROVIDERS WITH $165 MILLION LAUNCH.

- SPACEX'S FALCON 9 ROCKETS HAS TWO STAGES FOR LAUNCH

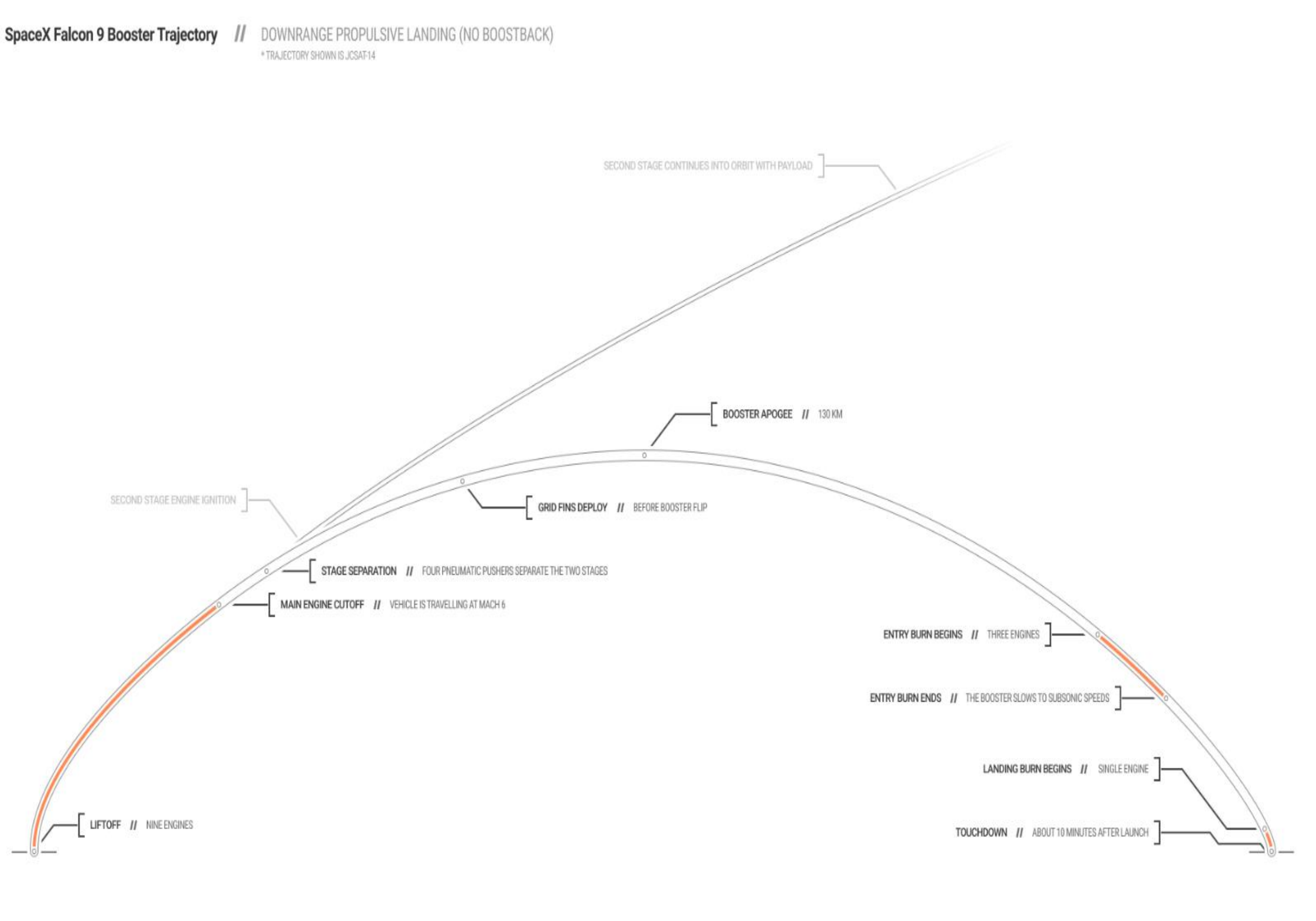
1. FIRST STAGE
2. SECOND STAGE



INTRODUCTION

- THE MAIN REASON THAT SPACEX IS MORE COST EFFECTIVE THAN OTHER PROVIDERS IS BECAUSE IT REUSES THE FIRST STAGE.

- HENCE, IF WE CAN DETERMINE THE FIRST STAGE SUCCESSFUL LANDING OF SPACEX'S FALCON 9, WE CAN DETERMINE THE COST OF THE LAUNCH.



INTRODUCTION

PROBLEM STATEMENT

TO DETERMINE SPACEX'S FALCON 9 FIRST STAGE SUCCESSFUL LANDING.

ANSWER WOULD BE ADDRESSED

DATA SCIENCE MACHINE LEARNING MODEL AND PUBLIC INFORMATION WILL BE USED TO PREDICT SPACEX'S FALCON 9 FIRST STAGE SUCCESSFUL LANDING.

METHODOLOGY

DATA SOURCES AND RESEARCH METHODS USED FOR THE ANALYSIS.

METHODOLOGY

- DATA COLLECTION METHODOLOGY

1. SPACEX LAUNCH DATA COLLECTED THROUGH SPACEX REST API.
2. DATA COLLECTED THROUGH WEB SCRAPPING FALCON 9 AND FALCON HEAVY LAUNCHES RECORDS FROM WIKIPEDIA.

HTTPS://EN.WIKIPEDIA.ORG/WIKI/LIST_OF_FALCON_9_AND_FALCON_HEAVY_LAUNCHES

- PERFORM DATA WRANGLING

1. REVIEW SOME ATTRIBUTES LIKES PAY LOAD MASS, LAUNCH SITE, OUTCOME ETC.
2. LANDING OUTCOME WAS CONVERTED INTO CLASSES (0 OR 1). 0 FOR BAD OUTCOME AND 1 FOR GOOD OUTCOME.

METHODOLOGY

1. PERFORM EXPLORATORY DATA ANALYSIS (EDA) USING SQL AND VISUALIZATION

2. PERFORM INTERACTIVE VISUAL ANALYTICS USING FOLIUM AND PLOTLY DASH

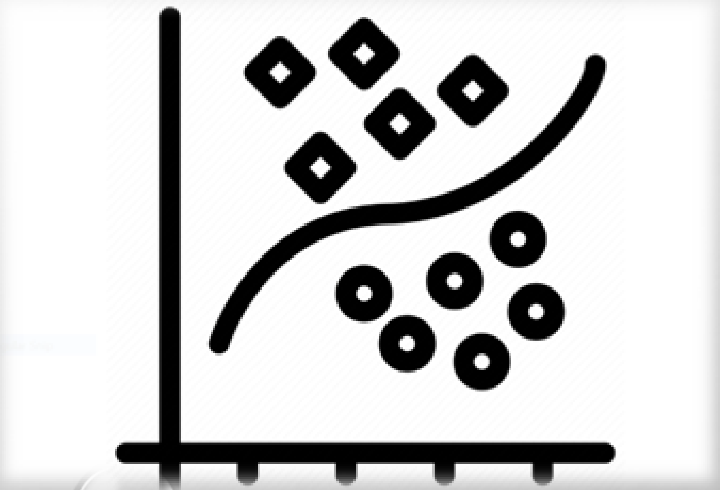
3. PERFORM PREDICTIVE ANALYSIS USING CLASSIFICATION MODELS

a) LOGISTIC REGRESSION

b) SUPPORT VECTOR MACHINE

c) DECISION TREE CLASSIFIER

d) K NEAREST NEIGHBORS



DATA COLLECTION WITH SPACEX REST API

Getting response from API and converting response to json file

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

Applying custom functions to clean data

```python
# Call getBoosterVersion
getBoosterVersion(data)
```

Assign list to dictionary and then create dataframe for falcon 9 rockets only

```python
# Hint data['BoosterVersion']!='Falcon 1'
data_falcon9 = df.loc[df['BoosterVersion'] != 'Falcon 1']
```

	FlightNumber	Date	BoosterVersion	PayloadMass	Orbit	LaunchSite	Outcome	Flights	GridFins	Reused	Legs	LandingPad	Block	ReusedCount	Serial
4	1	2010-06-04	Falcon 9	NaN	LEO	CCSFS SLC 40	None None	1	False	False	False	None	1.0	0	B0003
5	2	2012-05-22	Falcon 9	525.0	LEO	CCSFS SLC 40	None None	1	False	False	False	None	1.0	0	B0005
6	3	2013-03-01	Falcon 9	677.0	ISS	CCSFS SLC 40	None None	1	False	False	False	None	1.0	0	B0007
7	4	2013-09-29	Falcon 9	500.0	PO	VAFB SLC 4E	False Ocean	1	False	False	False	None	1.0	0	B1003
8	5	2013-12-03	Falcon 9	3170.0	GTO	CCSFS SLC 40	None None	1	False	False	False	None	1.0	0	B1004
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
89	86	2020-09-03	Falcon 9	15600.0	VLEO	KSC LC 39A	True ASDS	2	True	True	True	5e9e3032383ecb6bb234e7ca	5.0	11	B1060
90	87	2020-10-06	Falcon 9	15600.0	VLEO	KSC LC 39A	True ASDS	3	True	True	True	5e9e3032383ecb6bb234e7ca	5.0	11	B1058
91	88	2020-10-18	Falcon 9	15600.0	VLEO	KSC LC 39A	True ASDS	6	True	True	True	5e9e3032383ecb6bb234e7ca	5.0	11	B1051
92	89	2020-10-24	Falcon 9	15600.0	VLEO	CCSFS SLC 40	True ASDS	3	True	True	True	5e9e3033383ecbb9e534e7cc	5.0	11	B1060
93	90	2020-11-05	Falcon 9	3681.0	MEO	CCSFS SLC 40	True ASDS	1	True	False	True	5e9e3032383ecb6bb234e7ca	5.0	6	B1062

[Github link-1]

DATA COLLECTION WITH WEB SCRAPPING

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

```
beautiful_soup = BeautifulSoup(html_data, 'html5lib')
```

Getting HTML response → Creating Beautiful soup

Finding table and getting columns → Creating dictionary and appending data

Converting dictionary to data frame

[Github Link- 2]

```
first_launch_table = html_tables[2]
print(first_launch_table)
```

```
headings = []
for key,values in dict(launch_dict).items():
    if key not in headings:
        headings.append(key)
    if values is None:
        del launch_dict[key]

def pad_dict_list(dict_list, padel):
    lmax = 0
    for lname in dict_list.keys():
        lmax = max(lmax, len(dict_list[lname]))
    for lname in dict_list.keys():
        ll = len(dict_list[lname])
        if  ll < lmax:
            dict_list[lname] += [padel] * (lmax - ll)
    return dict_list

pad_dict_list(launch_dict,0)

df = pd.DataFrame.from_dict(launch_dict)
df.head()
```

	Flight No.	Launch site	Payload	Payload mass	Orbit	Customer	Launch outcome	Version Booster	Booster landing	Date	Time
0	1	CCAFS	Dragon Spacecraft Qualification Unit	0	LEO	SpaceX	Success\n	F9 v1.0B0003.1	Failure	4 June 2010	18:45
1	2	CCAFS	Dragon	0	LEO	NASA	Success	F9 v1.0B0004.1	Failure	8 December 2010	15:43
2	3	CCAFS	Dragon	525 kg	LEO	NASA	Success	F9 v1.0B0005.1	No attempt\n	22 May 2012	07:44
3	4	CCAFS	SpaceX CRS-1	4,700 kg	LEO	NASA	Success\n	F9 v1.0B0006.1	No attempt	8 October 2012	00:35
4	5	CCAFS	SpaceX CRS-2	4,877 kg	LEO	NASA	Success\n	F9 v1.0B0007.1	No attempt\n	1 March 2013	15:10

DATA WRANGLING

Calculate the percentage of missing value from each attribute.

Calculate the number and occurrences of each orbit.

Calculate the number of launches from each site.

```
df.isnull().sum()/df.count()*100
```

```
FlightNumber       0.000
Date               0.000
BoosterVersion     0.000
PayloadMass        0.000
Orbit              0.000
LaunchSite         0.000
Outcome            0.000
Flights            0.000
GridFins           0.000
Reused             0.000
Legs               0.000
LandingPad        40.625
Block              0.000
ReusedCount        0.000
Serial             0.000
Longitude          0.000
Latitude           0.000
dtype: float64
```

```
# Apply value_counts on Orbit column
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
Name: Orbit, dtype: int64
```

```
# 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
```

DATA WRANGLING

Landing Outcomes and counts

- Different types of landing outcomes
- The counts of different types of 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
Name: Outcome, dtype: int64
```

[Github Link 3]

Changing all different landing outcomes into binary class

- 0 landing outcomes means bad landing outcome
- 1 landing outcomes means good landing outcome

	Class
0	0
1	0
2	0
3	0
4	0
5	0
6	1
7	1

EDA WITH SQL

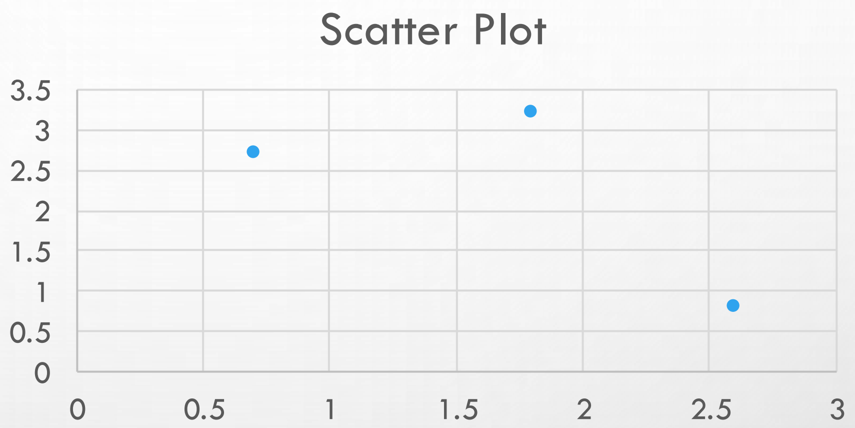
- Displaying the names of the unique launch sites in the space mission.
- Displaying the total payload mass carried by boosters launched by NASA (CRS).
- Displaying average payload mass carried by booster version F9 v1.1.
- Listing the date when the first success full landing outcome in ground pad was achieved.
- Listing the total number of successful and failure mission outcomes.
- Listing the records which will display the month names, failure landing outcomes in drone ship ,booster versions, launch site for the months in year 2015.
- Ranking the count of successful landing outcomes between the date 04-06-2010 and 20-03-2017 in descending order.

Github Link-4

EDA WITH VISUALIZATION

- Scatter Plots are drawn to show the dependency of attributes on each other

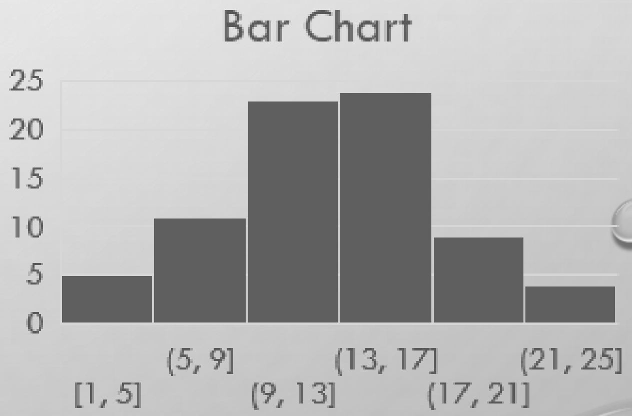
a) Payload Mass VS Flight number
b) Flight Number VS Launch Site
c) Payload Mass VS Launch Site
d) Flight Number VS Orbit Type
e) Payload Mass Vs Orbit Type



[Github Link-5]

- Launch Success Yearly Line Graph Drawn to show clear trend and aid future prediction

- Bar Graph drawn for success rate and Orbit Type to interpret the different success rate among different orbit types



BUILD INTERACTIVE MAP WITH FOLIUM

We use longitude and latitude coordinates to build interactive map with folium. Each Launch site is added in the map with circular marker and labeled with name of launch site. The success and failure of the launch is marked with green (success) and red (failure) is also added on map.

Map Objects	Code	Results
Map Marker	folium.Marker(	Map Object to make marker on map
Icon Marker	folium.Icon(	Create an icon on map
Circle Marker	foilum.Circle(	Create circle where marker is placed
Polyline	folium.Polylline(	Create line between points
Marker Cluster	MarkerCluster(	Create cluster on map where many markers around them

Github link -6

BUILD A DASHBOARD WITH PLOTLY DASH

- Pie chart showing percentage of total Success for all sites or by certain launch site.
- Sactter Graph showing correlation between payload and success for all sites or by certain launch site.

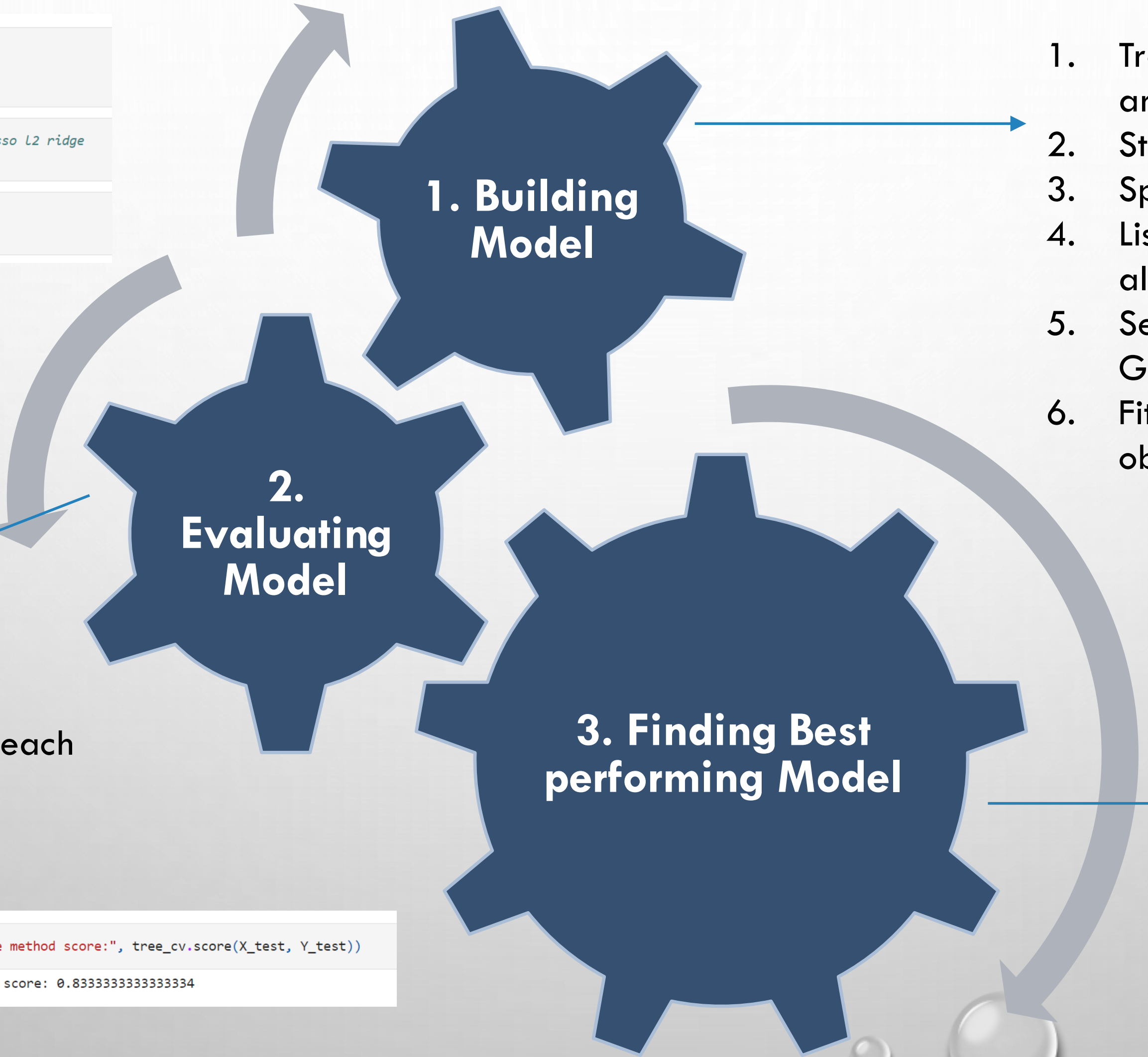
Map Objects	Code	Results
PLotly	Import plotly.express as px	Plot the graph with interactive plotly library
Dropdown	dcc.Dropdown(	Create a dropdown for launch sites
Rangeslider	dcc.RangeSlider(	Create a rangeslider for Payload mass range
Pie Chart	px.pie(	Create pie chart for success percentage
Scatter Chart	Pc.scatter(	Create scatter chart for correlation display

PREDICTIVE ANALYSIS (CLASSIFICATION)

```
parameters ={'C':[0.01,0.1,1],
             'penalty':['l2'],
             'solver':['lbfgs']}
```

```
parameters ={"C":[0.01,0.1,1],'penalty':['l2'], 'solver':['lbfgs']}# l1 lasso l2 ridge
lr=LogisticRegression()
```

```
gs_cv = GridSearchCV(lr, parameters, scoring='accuracy', cv=10)
logreg_cv = gs_cv.fit(X_train, Y_train)
```



1. Transform Dataframe into Numpy array.
2. Standardize and transform data.
3. Spilt data into training and test set.
4. List down Machine learning algorithms we want to use.
5. Set our parameters and algorithms to GridSearchCV
6. Fit our dataset into the GridsearchCV objects and train our model

[Github Link-7]

1. Check accuracy of each model
2. Find best hyperparameters for each type of algorithms.
3. Plot confusion matrix.

```
print("Accuracy for decision tree classifier on the test data using the method score:", tree_cv.score(X_test, Y_test))
```

```
Accuracy for decision tree classifier on the test data using the method score: 0.8333333333333334
```

The model with the best accuracy score is the best performing model

```
algorithms = {'KNN':knn_cv.best_score_,'Decision Tree':tree_cv.best_score_,'Logistic Regression':logreg_cv.best_score_,'SVM':svm_cv.best_score_}
best_algorithm = max(algorithms, key= lambda x: algorithms[x])

print('The method which performs best is \"',best_algorithm,'\" with a score of',algorithms[best_algorithm])
```

```
The method which performs best is " Decision Tree " with a score of 0.8888888888888888
```

RESULTS

RESULTS FROM THE DIFFERENT ANALYSIS PERFORMED

EDA WITH SQL

Sql QUERIES WERE RUN TO UNDERATND THE DATA MORE IN DEPTH

Display the names of the unique launch sites in the space mission

```
%sql SELECT DISTINCT Launch_Site FROM SPACEX_2;
```

```
 * ibm_db_sa://pmw92122:***@ea286ace-86c7-4d5b-8580-3fbfa46b1c66.bs2io90l08kqb1od8lcg.databases.appdomain.cloud:31505/bludb
   sqlite:///my_data1.db
Done.
```

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

Display the total payload mass carried by boosters launched by NASA (CRS)

```
%%sql SELECT SUM(PAYLOAD_MASS__KG_) AS TOTAL_PAYLOAD
FROM SPACEX_2
WHERE Customer LIKE 'NASA (CRS)';
```

```
 * ibm_db_sa://pmw92122:***@ea286ace-86c7-4d5b-8580-3fbfa46b1c66.bs2io90l08kqb1od8lcg.databases.appdomain.cloud:31505/bludb
   sqlite:///my_data1.db
Done.
```

total_payload
45596

Display 5 records where launch sites begin with the string 'CCA'

```
%%sql SELECT * FROM SPACEX_2
WHERE Launch_Site LIKE 'CCA%' LIMIT 5;
```

```
 * ibm_db_sa://pmw92122:***@ea286ace-86c7-4d5b-8580-3fbfa46b1c66.bs2io90l08kqb1od8lcg.databases.appdomain.cloud:31505/bludb
   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	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

Display average payload mass carried by booster version F9 v1.1

```
%%sql SELECT AVG(PAYLOAD_MASS__KG_) AS AVERAGE_PAYLOAD_MASS
FROM SPACEX_2
WHERE Booster_Version LIKE 'F9 v1.1';
```

```
 * ibm_db_sa://pmw92122:***@ea286ace-86c7-4d5b-8580-3fbfa46b1c66.bs2io90l08kqb1od8lcg.databases.appdomain.cloud:31505/bludb
   sqlite:///my_data1.db
Done.
```

average_payload_mass
2928

EDA WITH SQL

Sql QUERIES WERE RUN TO UNDERATND THE DATA MORE IN DEPTH

List the date when the first succesful landing outcome in ground pad was acheived.

Hint:Use min function

```
%%sql SELECT MIN(Date) AS First_Successfull_landing_date
FROM SPACEX_2
WHERE landing__outcome = 'Success (ground pad)';
```

```
 * ibm_db_sa://pmw92122:***@ea286ace-86c7-4d5b-8580-3fbfa46b1c66.bs2io90l08kqb1od8lcg.databases.appdomain.cloud:31505/bludb
   sqlite:///my_data1.db
Done.
```

first_successfull_landing_date
2015-12-22

List the names of the boosters which have success in drone ship and have payload mass greater than 4000 but less than 6000

```
%%sql SELECT BOOSTER_VERSION FROM SPACEX_2
WHERE LANDING__OUTCOME = 'Success (drone ship)'
AND payload_mass__kg_ BETWEEN 4000 AND 6000;
```

```
 * ibm_db_sa://pmw92122:***@ea286ace-86c7-4d5b-8580-3fbfa46b1c66.bs2io90l08kqb1od8lcg.databases.appdomain.cloud:31505/bludb
   sqlite:///my_data1.db
Done.
```

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

List the total number of successful and failure mission outcomes

```
%%sql SELECT COUNT(MISSION_OUTCOME) AS "Successful Mission"
FROM SPACEX_2 WHERE MISSION_OUTCOME LIKE 'Success%';
```

```
 * ibm_db_sa://pmw92122:***@ea286ace-86c7-4d5b-8580-3fbfa46b1c66.bs2io90l08kqb1od8lcg.databases.appdomain.cloud:31505/bludb
   sqlite:///my_data1.db
Done.
```

Successful Mission
100

```
%%sql SELECT COUNT(MISSION_OUTCOME) AS "Failure Mission"
FROM SPACEX_2 WHERE MISSION_OUTCOME LIKE 'Failure%';
```

```
 * ibm_db_sa://pmw92122:***@ea286ace-86c7-4d5b-8580-3fbfa46b1c66.bs2io90l08kqb1od8lcg.databases.appdomain.cloud:31505/bludb
   sqlite:///my_data1.db
Done.
```

Failure Mission
1

EDA WITH SQL

Sql QUERIES WERE RUN TO UNDERATND THE DATA MORE IN DEPTH

List the names of the booster_versions which have carried the maximum payload mass. Use a subquery

```
%%sql SELECT DISTINCT BOOSTER_VERSION AS "Booster Versions which carried the Maximum Payload Mass",
PAYLOAD_MASS__KG_ as "Maximum Payload Mass" FROM SPACEX_2
WHERE PAYLOAD_MASS__KG_ =(SELECT MAX(PAYLOAD_MASS__KG_) FROM SPACEX_2);
```

```
 * ibm_db_sa://pmw92122:***@ea286ace-86c7-4d5b-8580-3fbfa46b1c66.bs2io90l08kqb1od8lcg.databases.appdomain.cloud:31505/bludb
   sqlite:///my_data1.db
Done.
```

Booster Versions which carried the Maximum Payload Mass	Maximum Payload Mass
F9 B5 B1048.4	15600
F9 B5 B1048.5	15600
F9 B5 B1049.4	15600
F9 B5 B1049.5	15600
F9 B5 B1049.7	15600
F9 B5 B1051.3	15600

List the records which will display the month names, failure landing_outcomes in drone ship ,booster versions, launch_site for the months in year 2015.

Note: SQLLite does not support monthnames. So you need to use substr(Date, 4, 2) as month to get the months and substr(Date,7,4)='2015' for year.

```
%%sql SELECT MONTHNAME(DATE) as "Month", BOOSTER_VERSION, LAUNCH_SITE FROM SPACEX_2
WHERE DATE LIKE '2015-%' AND
LANDING__OUTCOME = 'Failure (drone ship)';
```

```
 * ibm_db_sa://pmw92122:***@ea286ace-86c7-4d5b-8580-3fbfa46b1c66.bs2io90l08kqb1od8lcg.databases.appdomain.cloud:31505/bludb
   sqlite:///my_data1.db
Done.
```

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

EDA WITH SQL

Sql QUERIES WERE RUN TO UNDERATND THE DATA MORE IN DEPTH

Rank the count of successful landing_outcomes between the date 04-06-2010 and 20-03-2017 in descending order.

```
%%sql SELECT LANDING__OUTCOME as "Landing Outcome", COUNT(LANDING__OUTCOME) AS "Total Count"
FROM SPACEX_2
WHERE DATE BETWEEN '2010-06-04' AND '2017-03-20'
GROUP BY  LANDING__OUTCOME
ORDER BY COUNT(LANDING__OUTCOME) DESC ;
```

```
 * ibm_db_sa://pmw92122:***@ea286ace-86c7-4d5b-8580-3fbfa46b1c66.bs2io90l08kqb1od8lcg.databases.appdomain.cloud:31505/bludb
   sqlite:///my_data1.db
Done.
```

Landing Outcome	Total 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

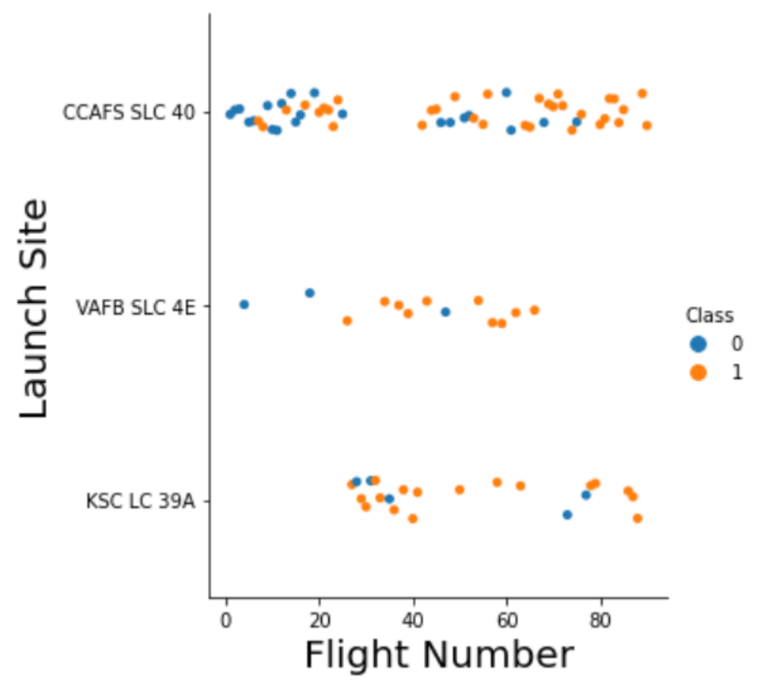
```
%%sql SELECT LANDING__OUTCOME as "Success Landing Outcome", COUNT(LANDING__OUTCOME) AS "Total Count"
FROM SPACEX_2
WHERE DATE BETWEEN '2010-06-04' AND '2017-03-20'
AND LANDING__OUTCOME LIKE '%Success%'
GROUP BY  LANDING__OUTCOME
ORDER BY COUNT(LANDING__OUTCOME) DESC ;
```

```
 * ibm_db_sa://pmw92122:***@ea286ace-86c7-4d5b-8580-3fbfa46b1c66.bs2io90l08kqb1od8lcg.databases.appdomain.cloud:31505/bludb
   sqlite:///my_data1.db
Done.
```

Success Landing Outcome	Total Count
Success (drone ship)	5
Success (ground pad)	3

EDA WITH VISUALIZATION

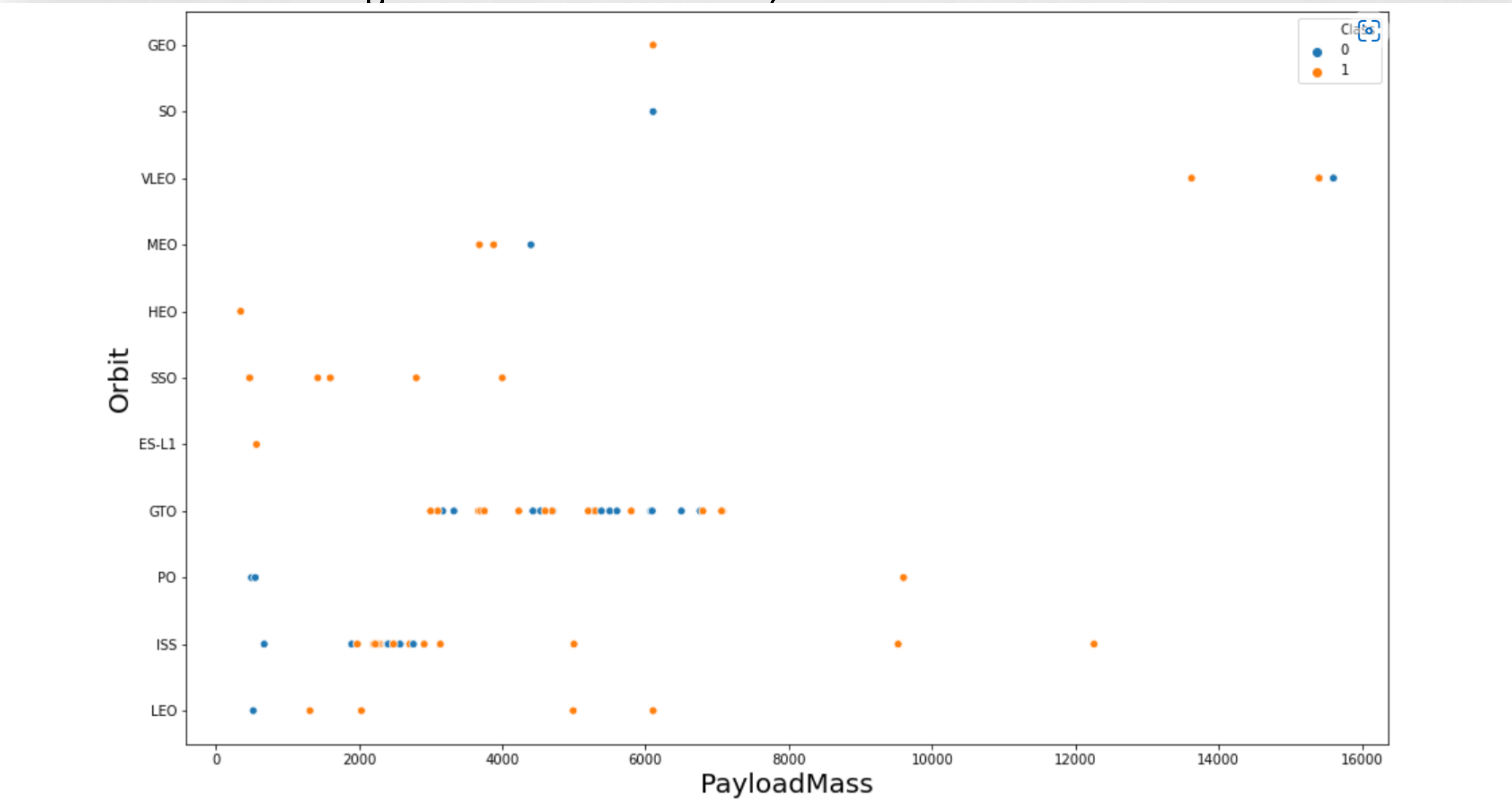
Now if we observe Launch Site vs Flight number scatter point chart we will that CCAFS SLC 40 is most prefer launch site.



Now if we observe Payload Vs. Launch Site scatter point chart we will find for the VAFB-SLC launch site there are no rockets launched for heavy payload mass(greater than 10000)

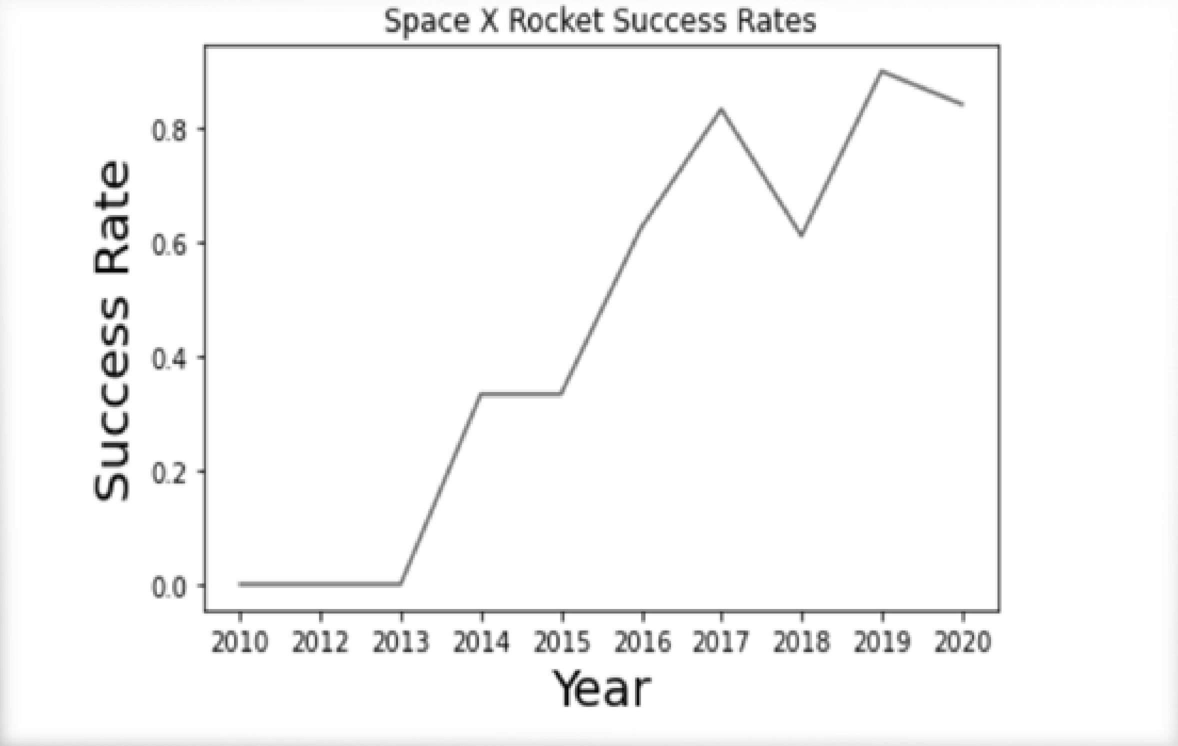
EDA WITH VISUALIZATION

Now if we observe Orbit Vs Payload Mass scatter point chart we will find the heavy payloads the successful landing or positive landing rate are more for Polar,LEO and ISS..

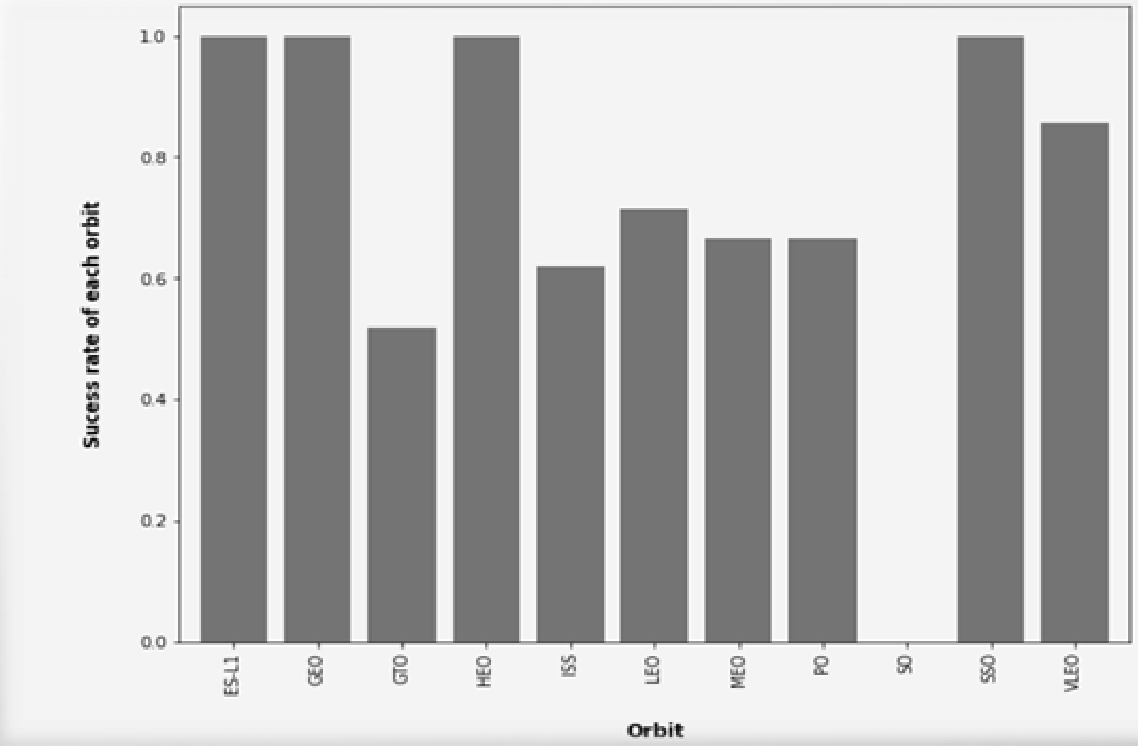


EDA WITH VISUALIZATION

Now if we observe success rate vs Year bar chart, we will find a positive trend in the success rate as year increase.

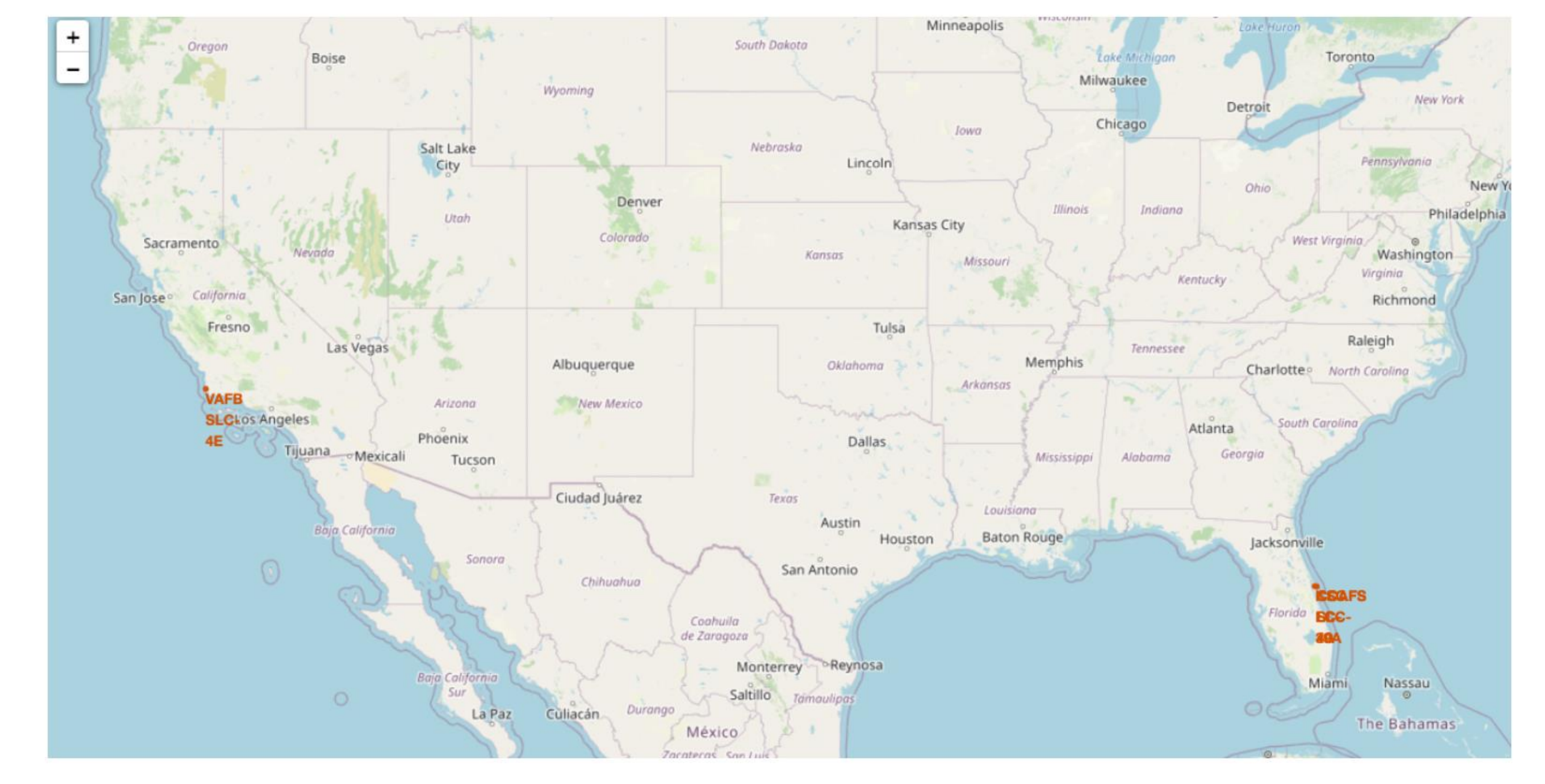


Now if we observe Orbit Vs success rate of each orbit bar chart, we will find the ESL-1, GEO, HEO, SSO has the highest success rate and SO has no success rate.

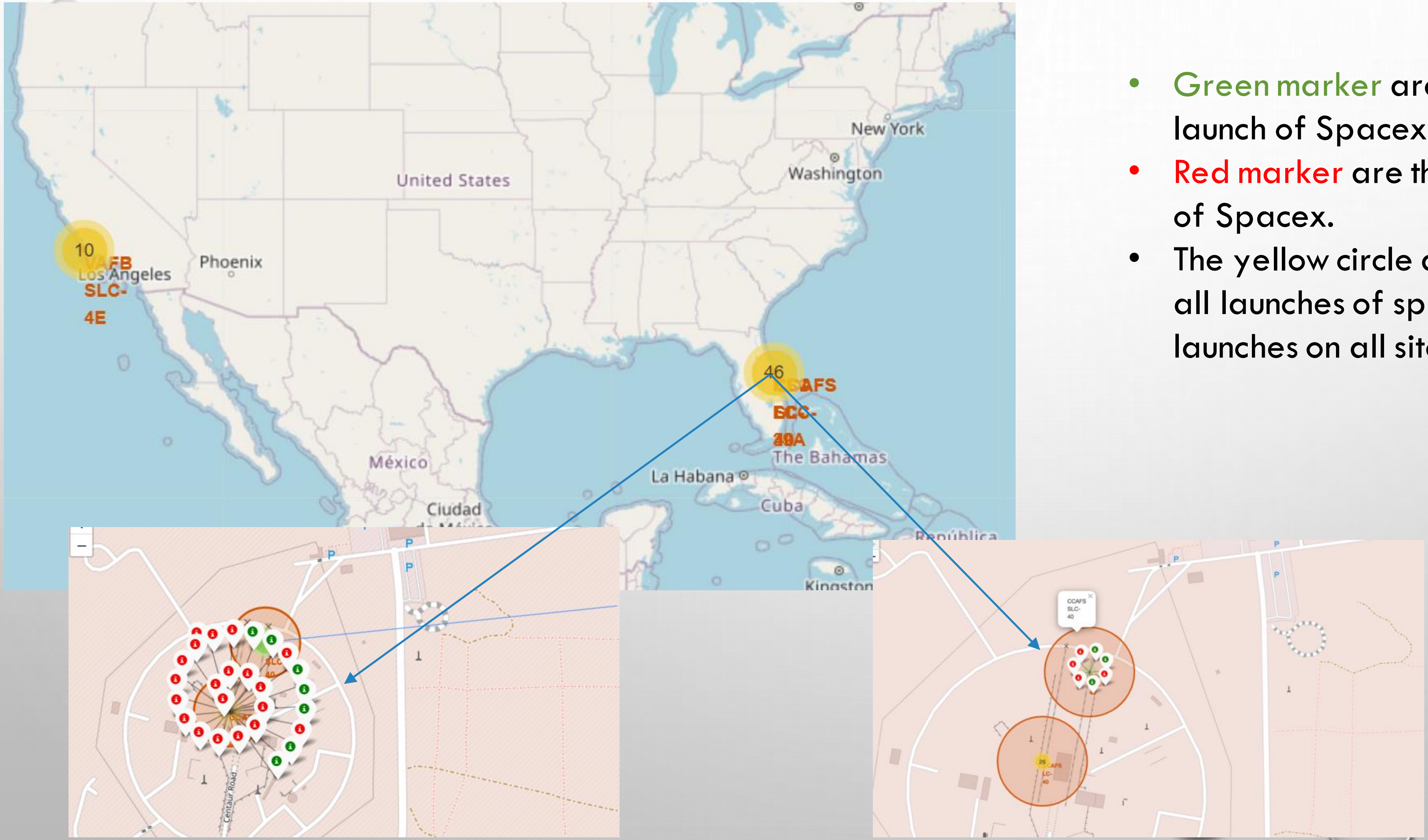


ANALYSIS WITH FOLIUM

We can see spacex launch sites on the USA interactive map with folium.

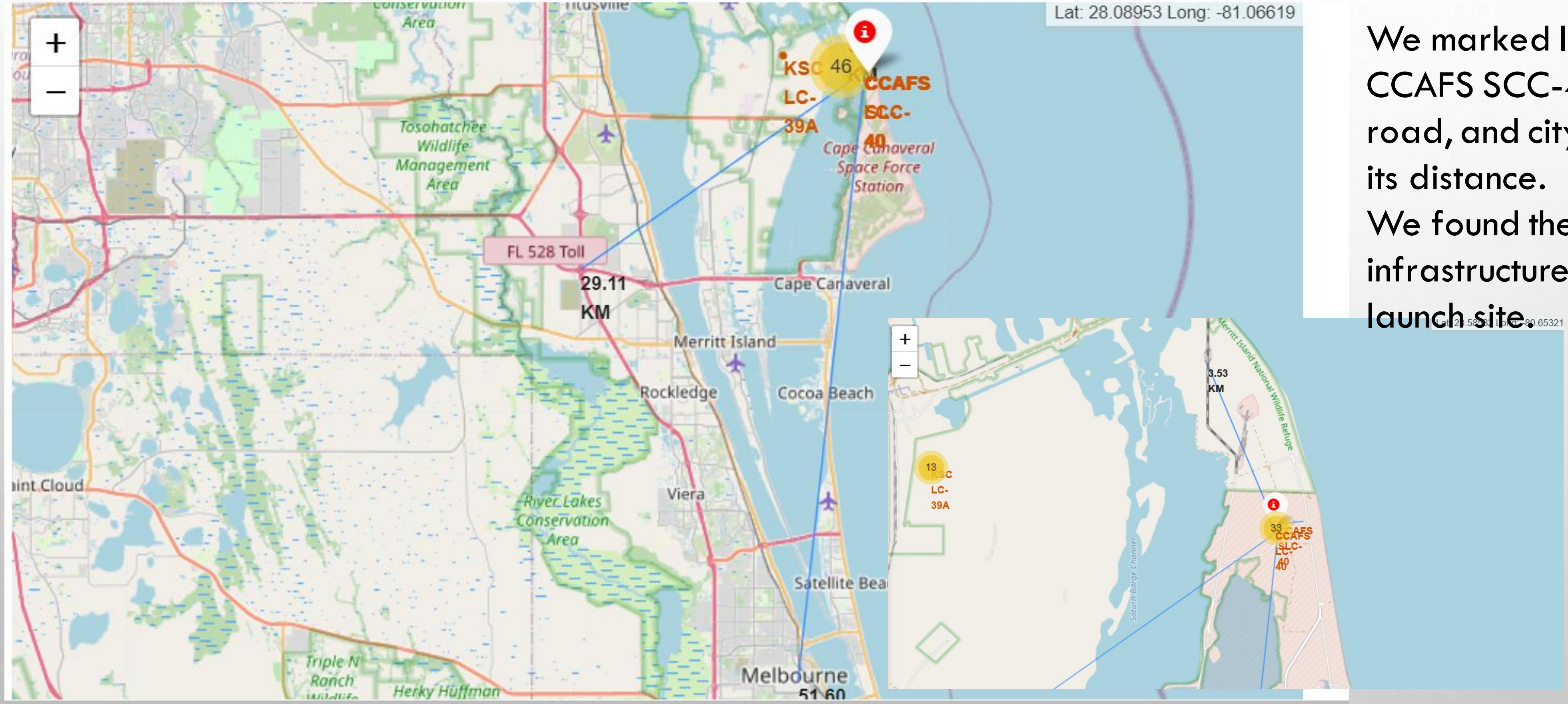


ANALYSIS WITH FOLIUM



- Green marker are the successful launch of Spacex.
- Red marker are the failed launch of Spacex.
- The yellow circle are the cluster of all launches of spacex falcon 9 launches on all sites.

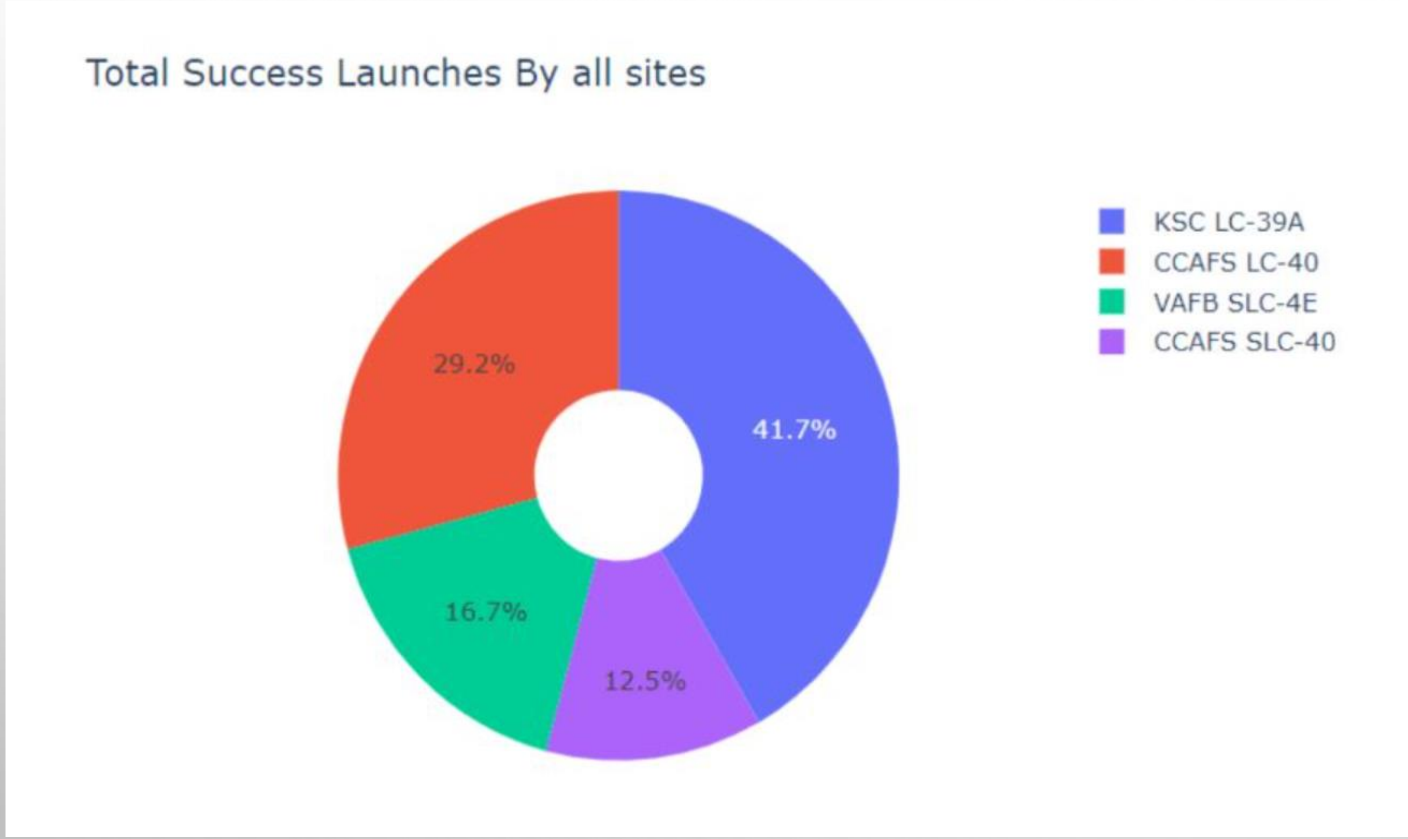
ANALYSIS WITH FOLIUM

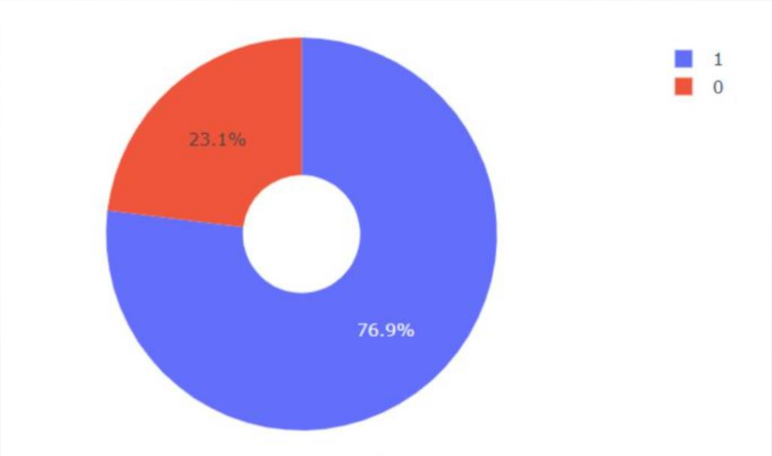


We marked line from launch site CCAFS SCC-40 to coast, railwayline, road, and city(Melbourne) and marked its distance.

We found the road, rail and coast infrastructure are not far from the launch site.

ANALYSIS WITH PLOTLY DASH



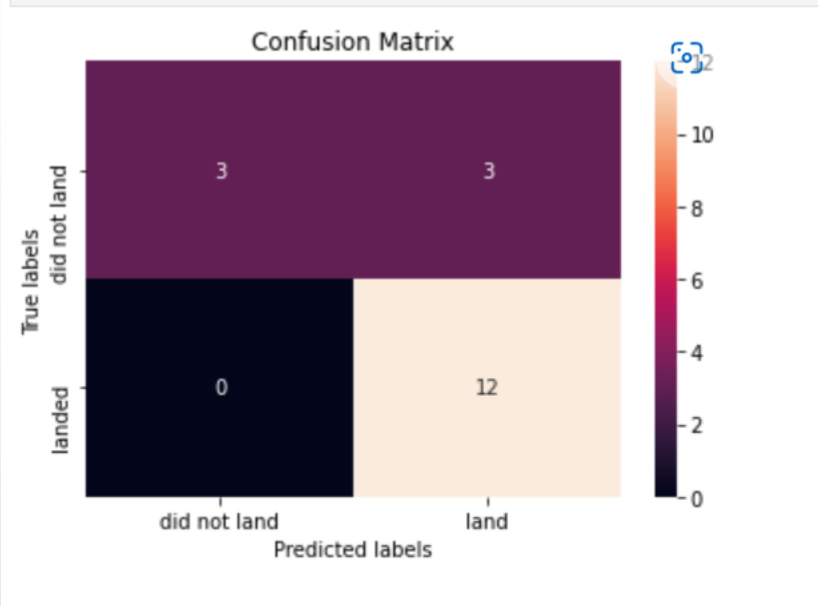


KSC LC -39A had achieved a 76.9% success rate while getting a 23.1% failure rate.

We can see that KSC LC -39A had most successful launches from all the sites.

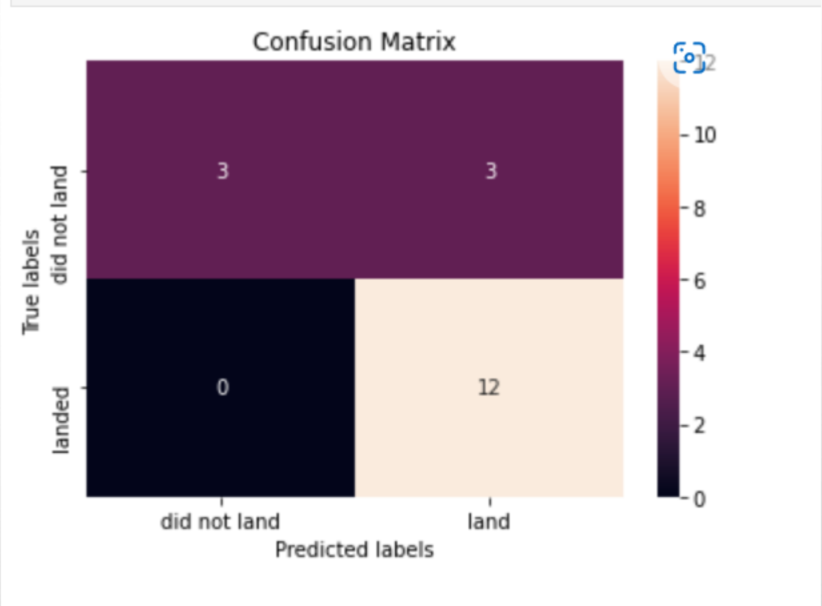
PREDICTIVE ANALYSIS (CLASSIFICATION)

Logistic Regression



Best Parameters	C-0.01, Penalty-12, solver- lbfgs
Train set accuracy	0.84722222
Test set accuracy	0.83333334

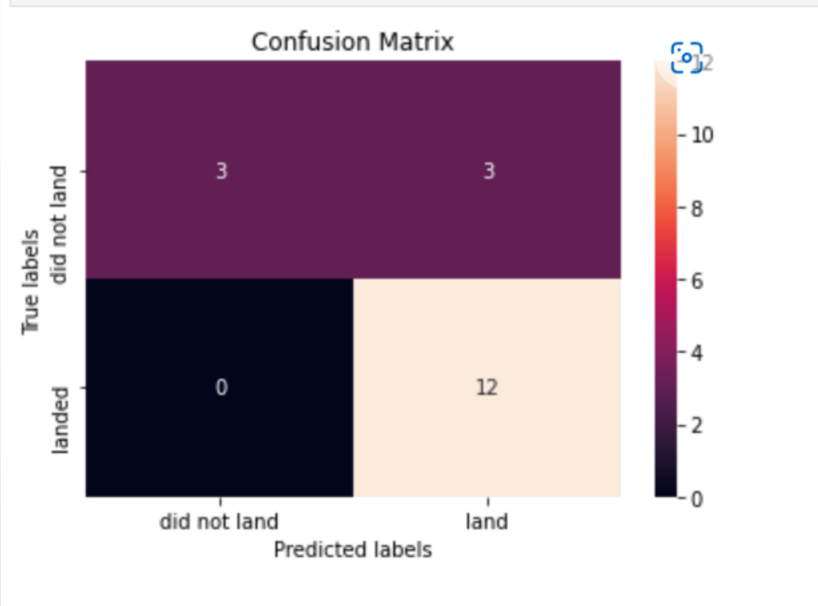
Support Vector Machine



Best Parameters	C-1.0, gamma- 0.031622, kernel- sigmoid
Train set accuracy	0.84722222
Test set accuracy	0.83333334

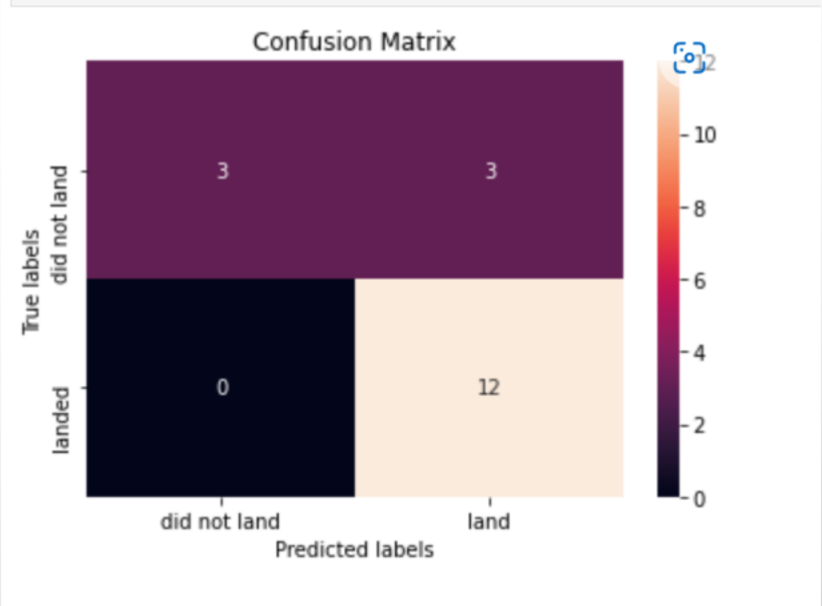
PREDICTIVE ANALYSIS (CLASSIFICATION)

Decision Tree



Best Parameters	Criterion- gini, max_depth-6, min sample leaf- 1, min sample split- 10, splitter- best
Train set accuracy	0.88888888
Test set accuracy	0.83333334

K-nearest Neighbour



Best Parameters	Alogrithum- auto, n neighbor- 9, P- 1
Train set accuracy	0.84722222
Test set accuracy	0.83333334

CONCLUSION

CONCLUSION FROM THE ABOVE ANALYSIS

CONCLUSION

- SUCCESS RATE OF SPACEX FALCON 9 FIRST STAGE LANDING HAS BEEN INCREASING EACH YEAR. IT IS A POSITIVE TREND
- KSC LC-39A HAD MOST SUCCESSFUL LAUNCHES.
- ORBITS ES-L1,GEO,HEO,SSO HAS THE HIGHEST SUCCESS RATES FOR THE LANDING FIRST STAGE.

- THE BEST MODEL TO PREDICT SUCCESSFUL LANDING OF FIRST STAGE OF FALCON 9 SPACEX IS DECISION TREE CLASSIFIER WITH ACCURACY OF 0.89

	index	Accuracy
0	KNN	0.847222
1	Decision Tree	0.888889
2	Logistic Regression	0.847222
3	SVM	0.847222

CONCLUSION

- ✓ THE TEST SET FOR ALL MACHINE LEARNING MODELS PRODUCED SAME ACCURACY OF 0.8334 AND HENCE WE HAVE SELECTED THE BEST MACHINE LEARNING MODEL TO PREDICT THE FIRST STAGE LANDING OF SPACEX FALCON 9 ON THE BASIS OF TRAIN SET MODEL.
- ✓ WE CAN ONLY PREDICT 88.89% TIMES THE CORRECT LANDING OF FIRST STAGE OF SPACEX FALCON 9 GIVEN THAT SUCCESS OR FAILURE OF LANDING IS COMPLETELY DEPENDED ON THE FEATURE SET SELECTED BY US IN BUILDING THE MODEL.
- ✓ *THERE IS HUGE SCOPE TO IMPROVE THE MODEL*. A COMPLETE STUDY IS NEEDED TO IDENTIFY MORE FEATURES ON WHICH FIRST STAGE LANDING IS DEPENDED. THE ACCURACY OF THE MODEL CAN BE IMPROVED ONCE WE WILL HAVE MORE CLASSIFIED DATA TO STUDY FROM SPACEX.

THANK YOU!