Space X Landing Analysis

Ismael Coulibaly

2022 February 17

OUTLINE

- Executive Summary
- Introduction
- Methodology
- Results
  - Visualization – Charts
  - Dashboard
- Discussion
  - Findings & Implications
- Conclusion
- Appendix

EXECUTIVE SUMMARY

Methodologies

- Data Collection
- Data Wrangling and formatting
- EDA and visualisation
  - EDA with Data Visualisation
  - EDA with SQL
- Visual analytics with Folium
- Dasboard Presentation with Plotly dash
- Machine Learning Predictive Analysis

Results

- EDA results
- Interactive review of the dashboard
- Predictions results insights

INTRODUCTION

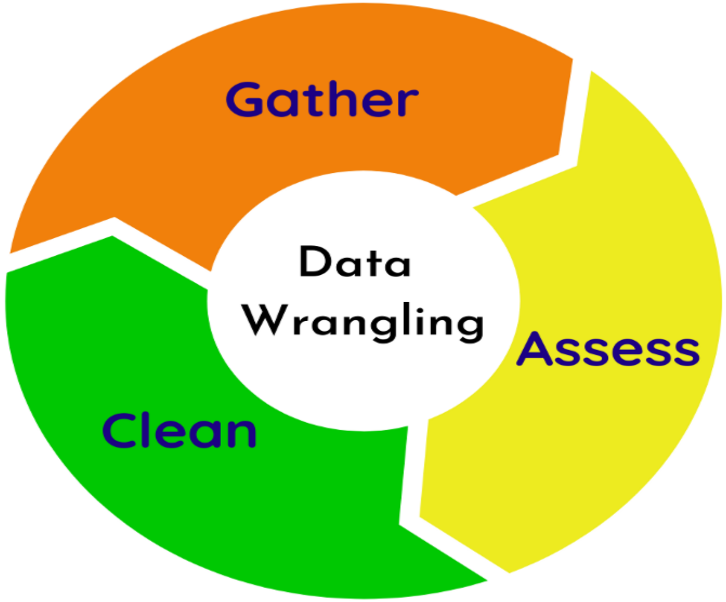
Context

- SpaceX company stated on their websites that the Falcon 9 rocket launches Costis around 62 M dollars while other providers cost upward of 165 million dollars each, much of the savings is because SpaceX can reuse the first stage. Being able to determine if the first stage will land will help determine the cost of a launch. Thus helping our company (SPACE Y) to compete with SpaceX

Problems to solve through this analysis

- What parameters determines the most a successful land?
- In Which conditions or company should experiment in order to get the best results

DATA WRANGLING METHODOLOGY



- The data Collection was done by making a request to the SpaceX API Ensuring a better accuracy about veracity of entries

```
In [6]: spacex_url="https://api.spacexdata.com/v4/launches/past"

In [7]: response = requests.get(spacex_url)
```

- Reformatting
  - Data was given in a json format so basics reformatttng was needed

```
In [11]: # Use json_normalize meethod to convert the json result into a dataframe
         data=response.json()
         data=pd.json_normalize(data)
```

  - As only data of Falcon 9 is revelant for us we make a filter

```
In [27]: # Hint data['BoosterVersion']!='Falcon 1'
         mask=data['BoosterVersion']!='Falcon 1'
         data_falcon9=data[mask]
```

- Missing Values
  - First Enquiry about the numbers of Missing Values
    - As a result landing pad has the most entries with nan = 26
  - Dealing with missing values
    - For doing so we fill any missing value with the mean value of the columns getting our final data clean

```
In [29]: data_falcon9.isnull().sum()
```

```
Out[29]: FlightNumber      0
         Date              0
         BoosterVersion    0
         PayloadMass       5
         Orbit             0
         LaunchSite        0
         Outcome           0
         Flights           0
         GridFins          0
         Reused            0
         Legs              0
         LandingPad       26
         Block             0
         ReusedCount       0
         Serial            0
         Longitude         0
         Latitude          0
         dtype: int64
```

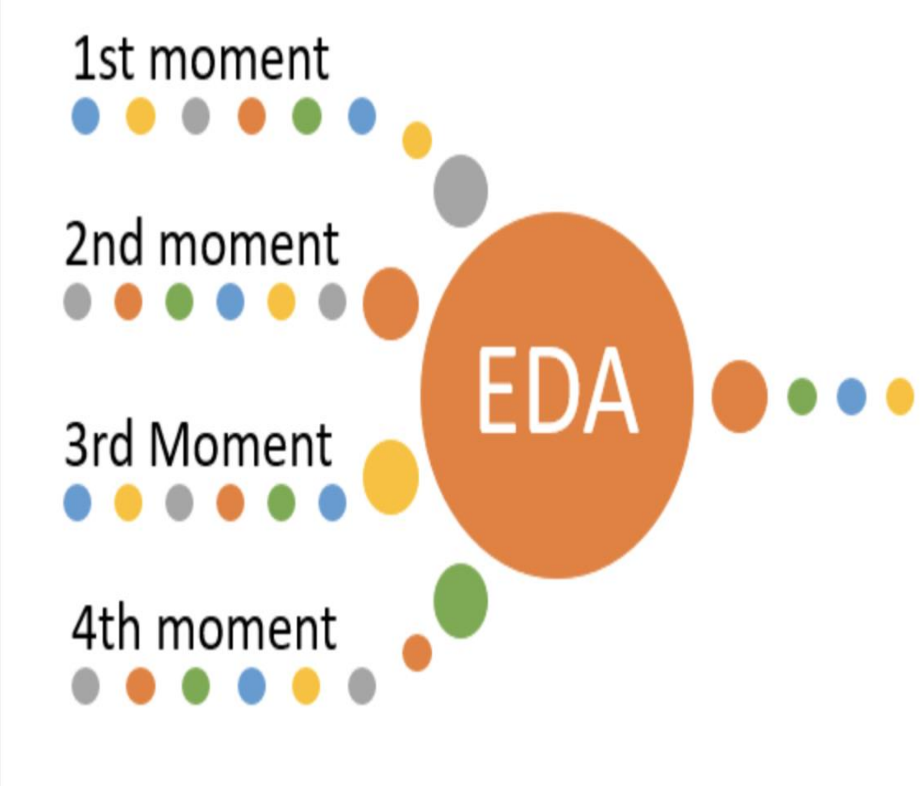
```
In [30]: # Calculate the mean value of PayloadMass column
         mean= data['PayloadMass'].mean()
         data.replace({np.nan:mean})
         # Replace the np.nan values with its mean value
```

Out[30]:

	FlightNumber	Date	BoosterVersion	PayloadMass	Orbit	LaunchSite	Outcome	Flights	GridFins	Reused	Legs	LandingPa
0	1	2006-03-24	Falcon 1	20.000000	LEO	Kwajalein Atoll	None None	1	False	False	False	5919.165341
1	2	2007-03-21	Falcon 1	5919.165341	LEO	Kwajalein Atoll	None None	1	False	False	False	5919.165341
2	4	2008-09-28	Falcon 1	165.000000	LEO	Kwajalein Atoll	None None	1	False	False	False	5919.165341
3	5	2009-07-13	Falcon 1	200.000000	LEO	Kwajalein Atoll	None None	1	False	False	False	5919.165341
4	6	2010-06-04	Falcon 9	5919.165341	LEO	CCSFS SLC 40	None None	1	False	False	False	5919.165341
...	...	...	...	...	...	...	...	...	...	...	...	...
89	102	2020-09-03	Falcon 9	15600.000000	VLEO	KSC LC 39A	True ASDS	2	True	True	True	5e9e3032383ecb6bb234e7c
90	103	2020-10-06	Falcon 9	15600.000000	VLEO	KSC LC 39A	True ASDS	3	True	True	True	5e9e3032383ecb6bb234e7c
91	104	2020-10-18	Falcon 9	15600.000000	VLEO	KSC LC 39A	True ASDS	6	True	True	True	5e9e3032383ecb6bb234e7c
92	105	2020-10-24	Falcon 9	15600.000000	VLEO	CCSFS SLC 40	True ASDS	3	True	True	True	5e9e3033383ecbb9e534e7c
93	106	2020-11-05	Falcon 9	3681.000000	MEO	CCSFS SLC 40	True ASDS	1	True	False	True	5e9e3032383ecb6bb234e7c

94 rows × 17 columns

EDA AND VISUALISATION METHODOLOGY



- EDA and visualization methodology using libraries
- EDA with SQL

EDA

- Determining the number of each launch Sites

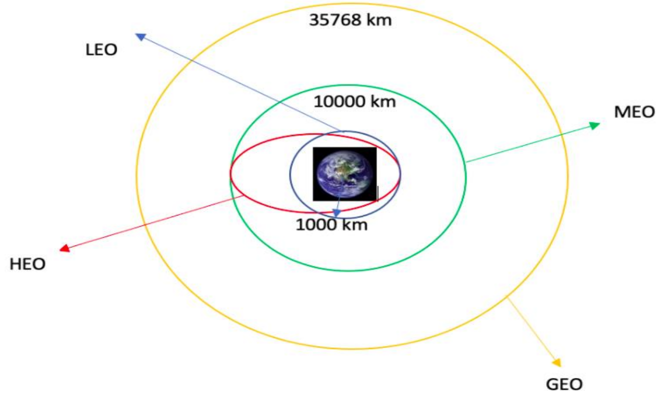
```
In [12]: # Apply value_counts() on column LaunchSite
         df['LaunchSite'].value_counts()

Out[12]: CCAFS SLC 40    55
         KSC LC 39A      22
         VAFB SLC 4E     13
         Name: LaunchSite, dtype: int64
```

```
In [13]: # Apply value_counts on Orbit column
         df['Orbit'].value_counts()

Out[13]: GTO      27
         ISS      21
         VLEO     14
         PO        9
         LEO       7
         SSO       5
         MEO       3
         HEO       1
         GEO       1
         ES-L1     1
         SO        1
         Name: Orbit, dtype: int64
```

- Determining numbers of orbits
    - Each launch site aims to an orbit



EDA

- **Create a Landing Label**
  - Label Creation is helpful for letting know to the predictions algorithm 'what defines success ' in our case 1 for success and 0 for failure

In [19]:
```
df['Class']=landing_class
df[['Class']].head(8)
```

Out[19]:

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

In [21]:
```
df.head(5)
## edit
df['Class'].value_counts()
## end edit
```

Out[21]:
```
1    60
0    30
Name: Class, dtype: int64
```

We can use the following line of code to determine the success rate:

In [22]:
```
df["Class"].mean()
```

Out[22]:
```
0.6666666666666666
```

We can now export it to a CSV for the next section,but to make the answers consistent, in the next lab we will provide data in a pre-selected date range.

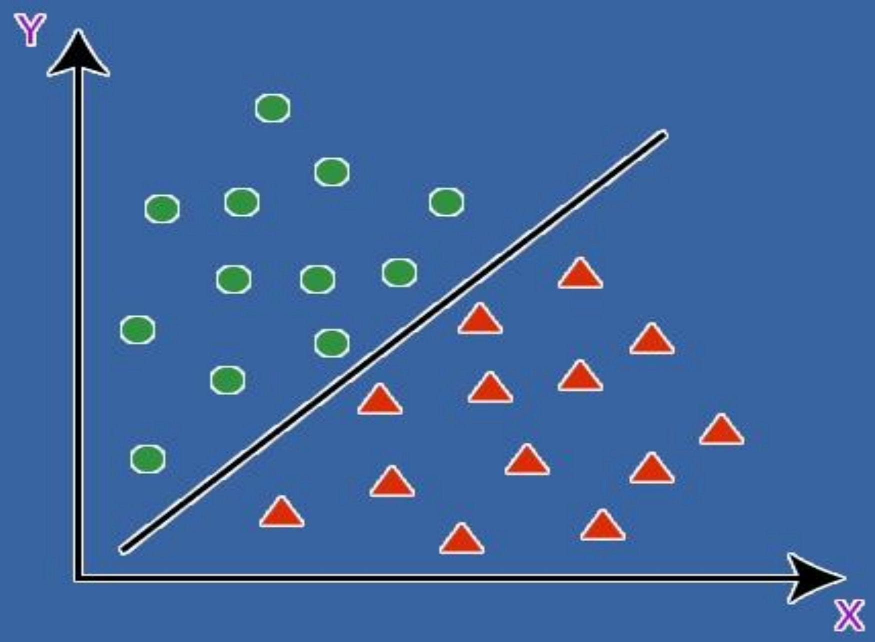
```
df.to_csv("dataset_part\_2.csv", index=False)
```

✓ As insight we notice that we have around 66% of Success Rate

EDA WITH SQL

- **Using SQL to enquiry the database stored on, IBM Db2 cloud**
    - Listing unique Launching sites
    - *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 booster versions which have carried the maximum payload mass*

Predicttive ANalysis methodology



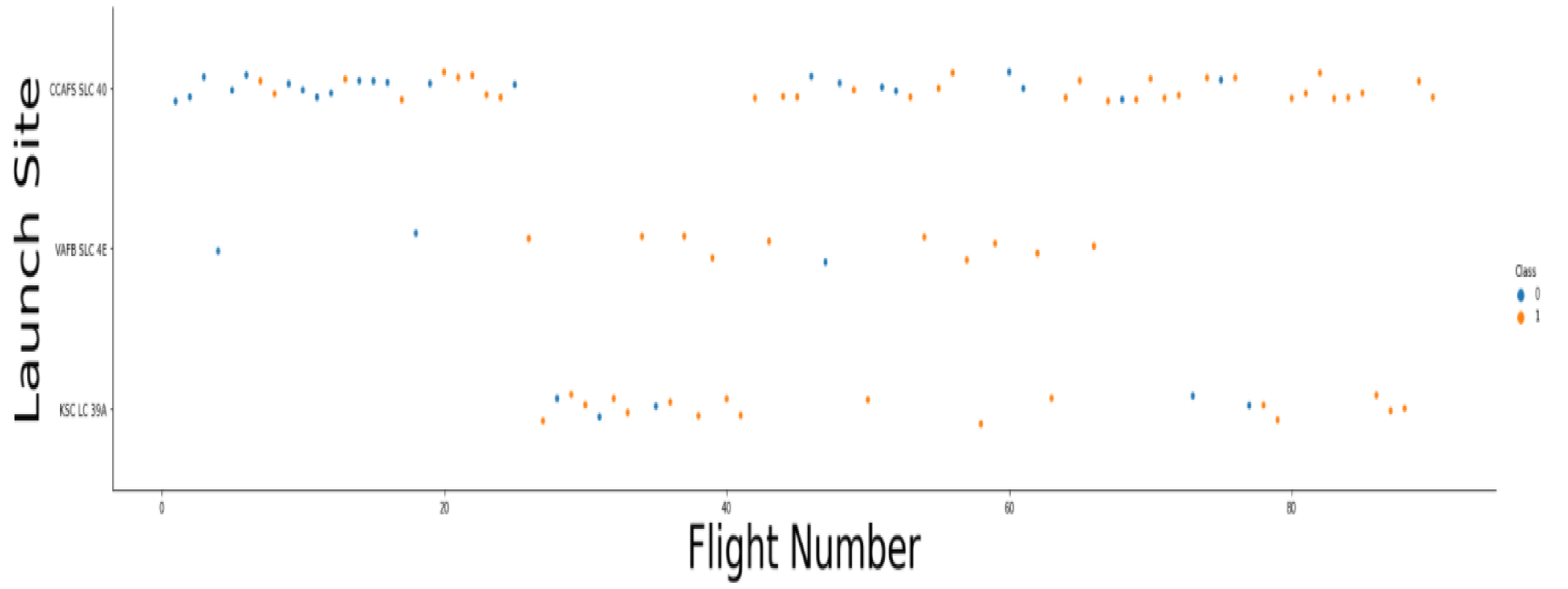
Building	Evaluating	Improving	Finding
Building model • Loading and Standardize data • Split in train and test sdataset • Applying SVM, decision tree ,logistic regression and KNN algorithm • Fit the model using GridSearchCV	Evaluating model • Get accuracy • Plot Confusion Matrix	Improving the model • Udsing hyperparameter tuning to try to get the best **performing model**	Finding the best model **by ranking them based on accuracy**

RESULTS

- EDA with visualization
- EDA with SQL results
- interactive map with Folium
- Plotly Dash dashboard
- Predictive analysis results

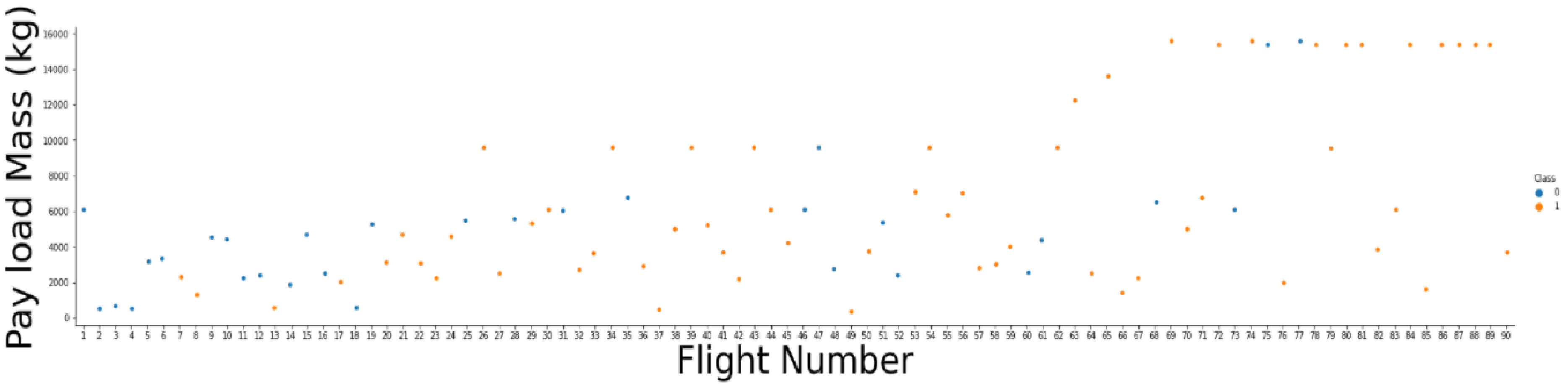
EDA with visualization

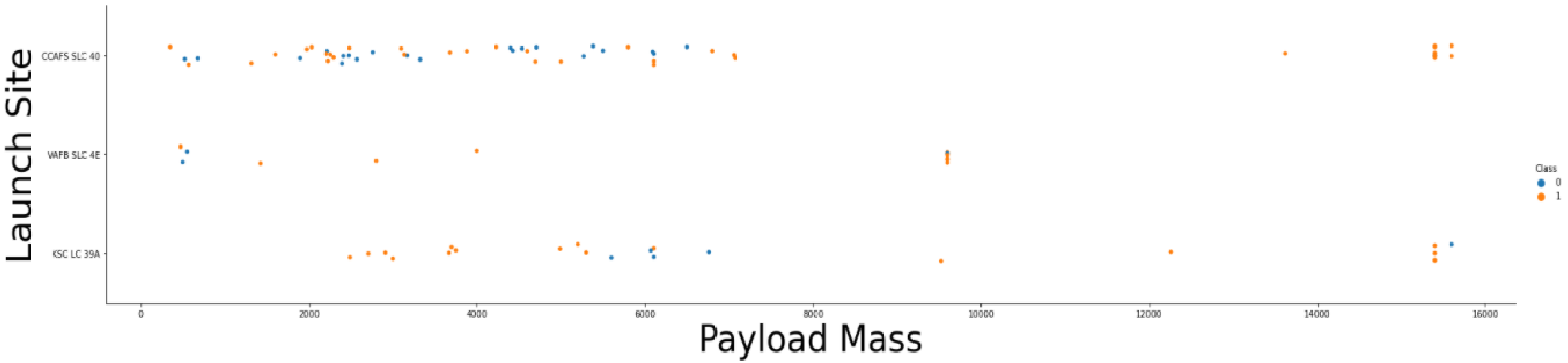
Flight number vs launch Sites



- ✓ This indicates The more a site have launches the more they will be successful

EDA with visualization

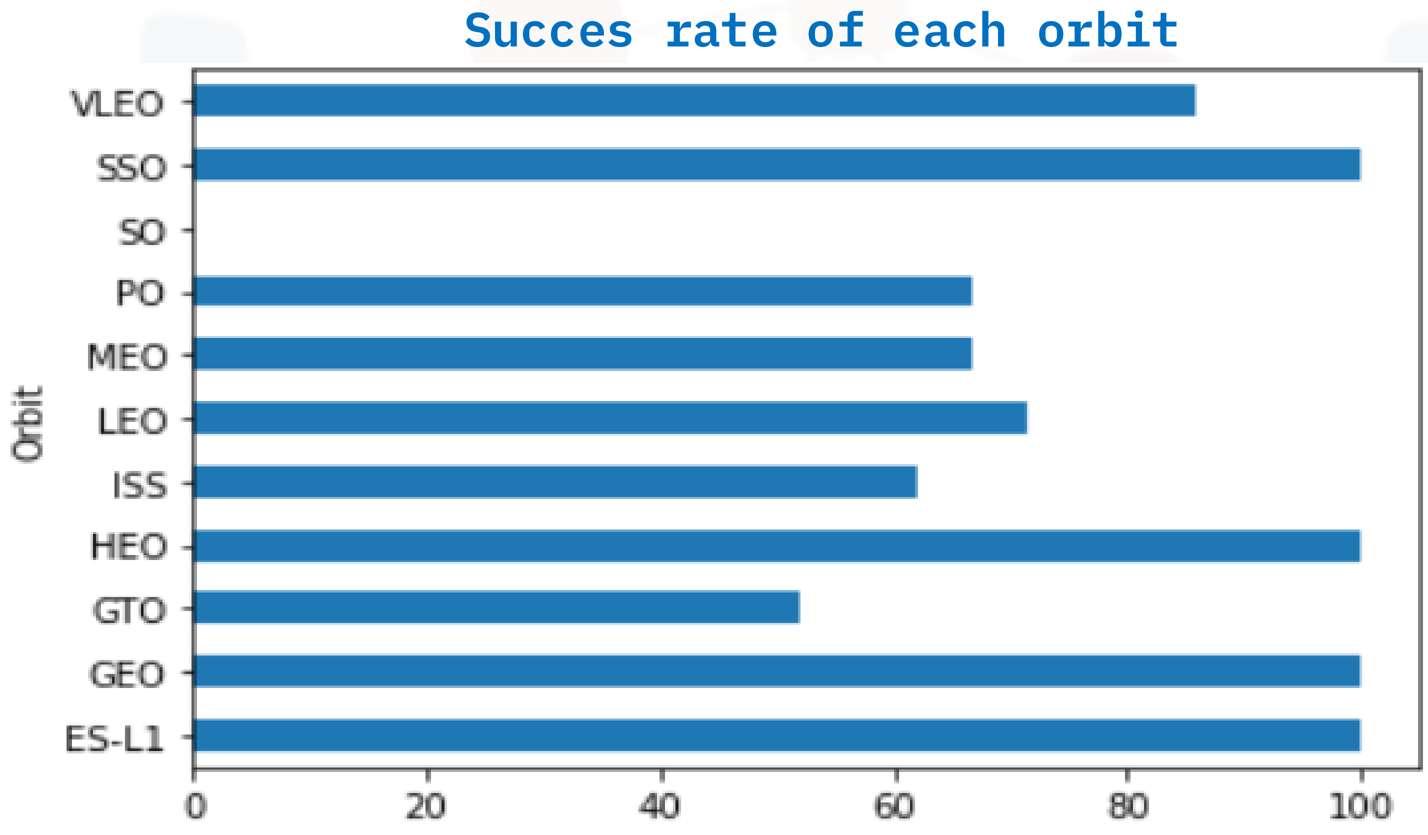




- Different launch sites have different success rates

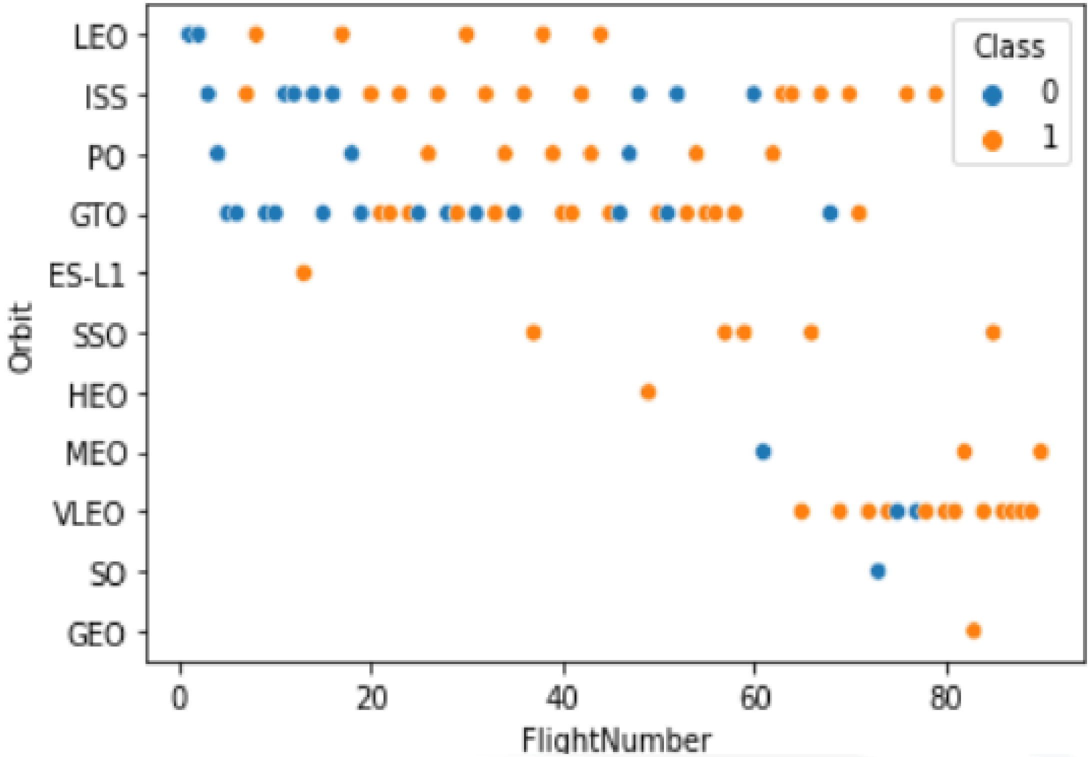
- Some Launch Sites does not welcome heavy payload

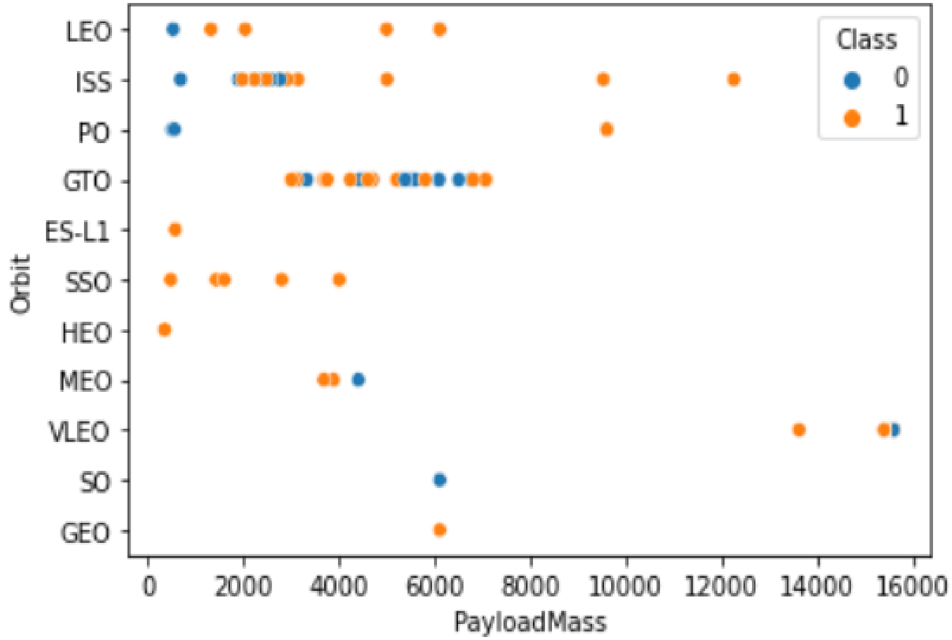
EDA with visualization



- ✓ SSO , GEO and ES-L1 have 100% Success Rate

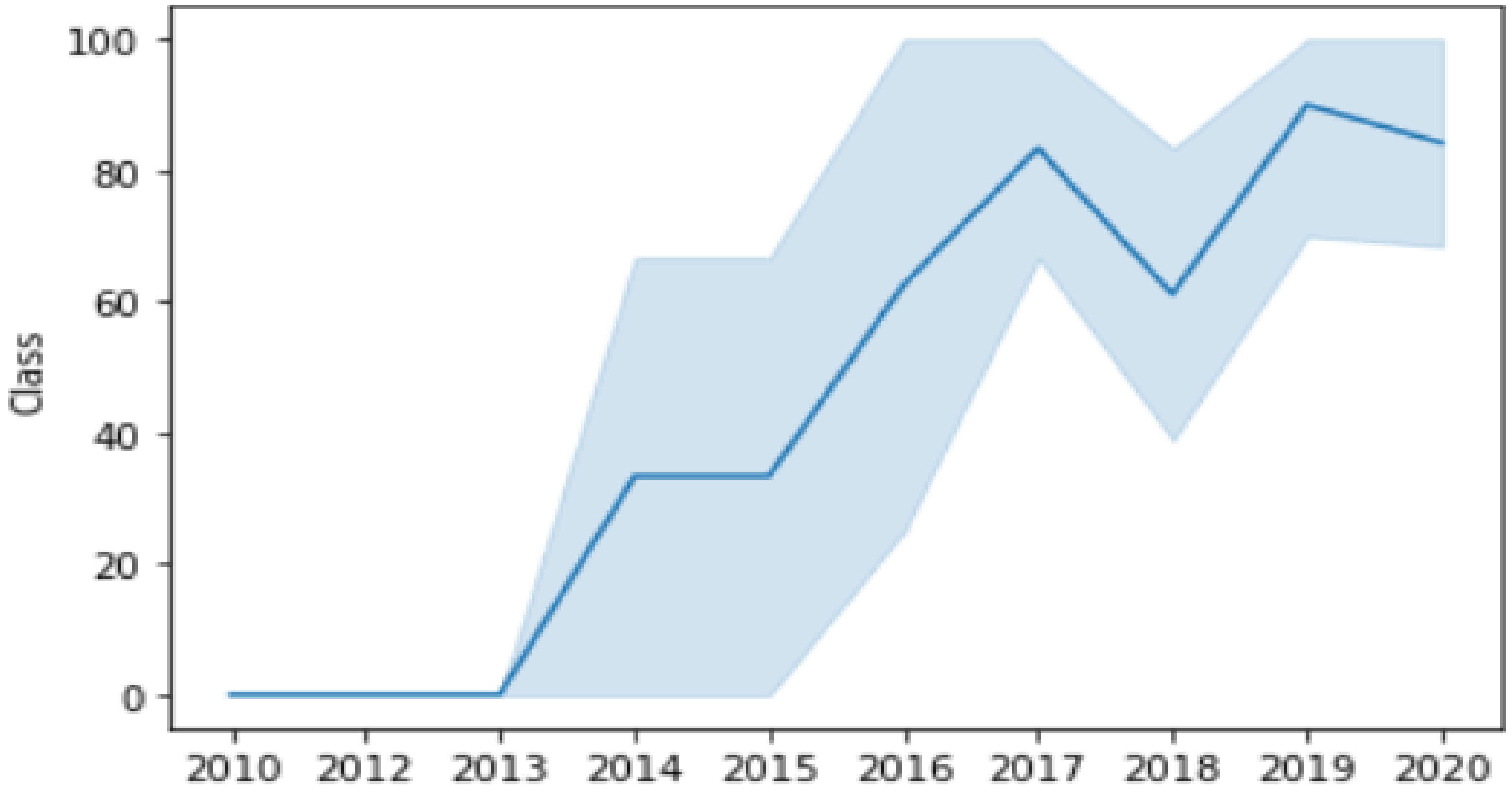
EDA with visualization





EDA with visualization

Yearly success rate



- ✓ Success Rate was increasing from 2013 till 2021

EDA with SQL

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

```
]: %sql SELECT AVG(payload_mass__kg_ ) FROM SPACEXTBL WHERE booster_version LIKE 'F9 v1.1%'
```

```
 * ibm_db_sa://bnx43768:***@2d46b6b4-cbf6-40eb-bbce-6251e6ba0300.bs2io90l08kqb1od8lcg.databases.appdomain.cloud:32328/bludb
Done.
```

]:

1
2534

✓ Average payload mass is about 2534 Kg

EDA with SQL

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

Hint:Use min function

```
%sql SELECT MIN (DATE)\
FROM SPACEXTBL \
WHERE landing__outcome='Success (ground pad)'
```

```
 * ibm_db_sa://bnx43768:***@2d46b6b4-cbf6-40eb-bbce-6251e6ba0300.bs2io90l08kqb1od8lcg.databases.appdomain.cloud:32328/bludb
Done.
```

1
2015-12-22

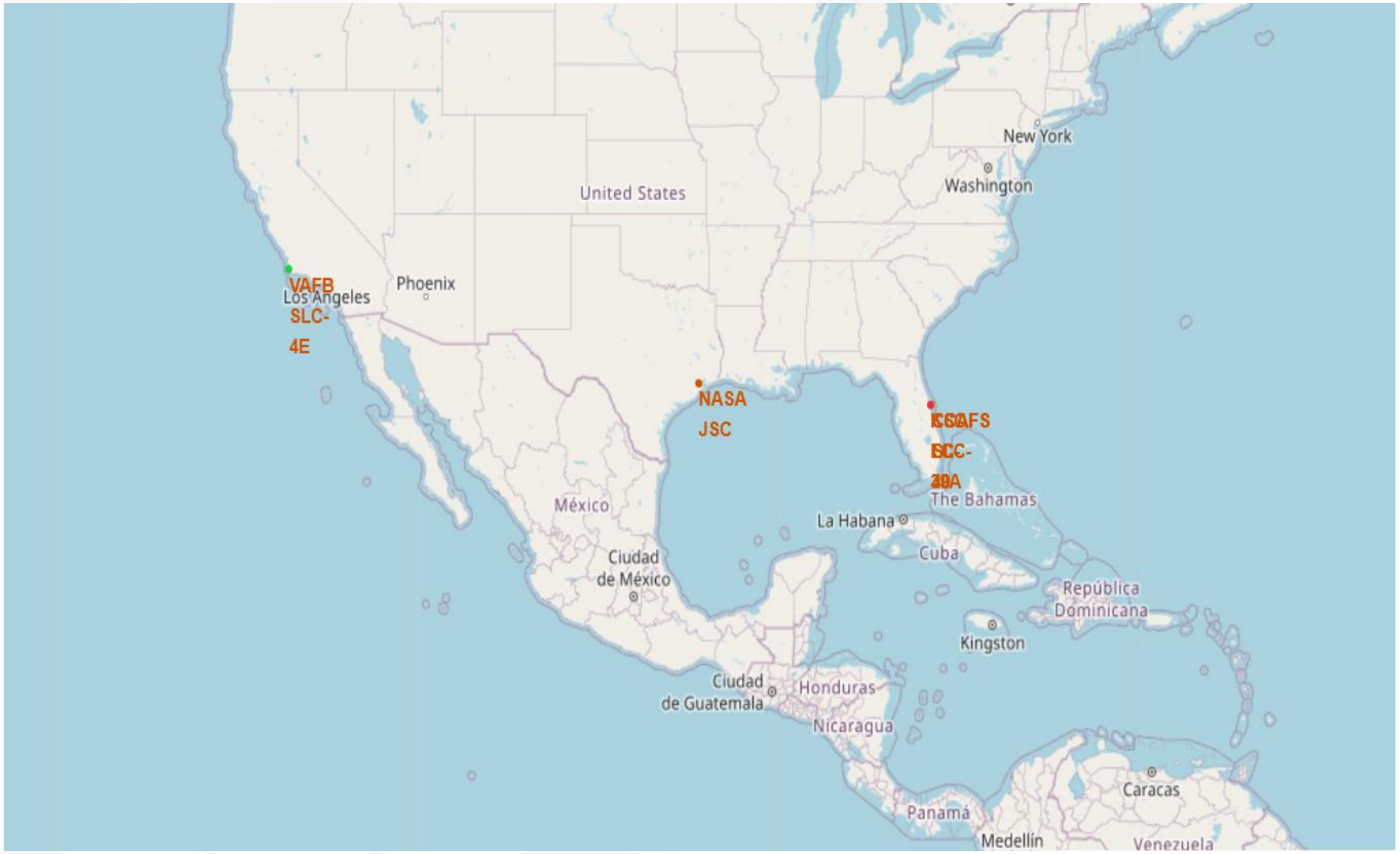
Visual Analysis with Folium

IBM Developer

SKILLS NETWORK

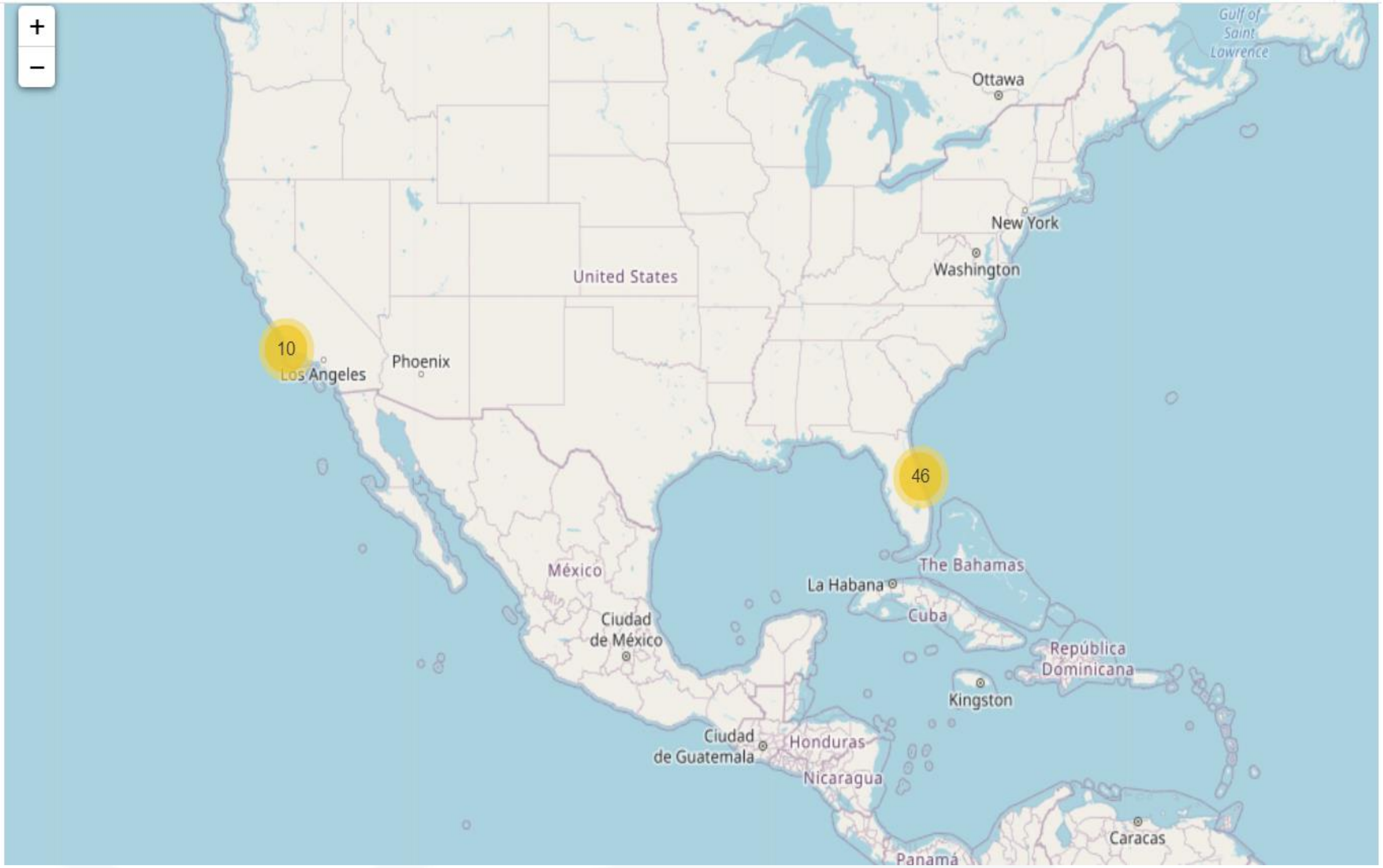
Visual Analysis with Folium

- Representing each Launch Sites an NASA center on the Map



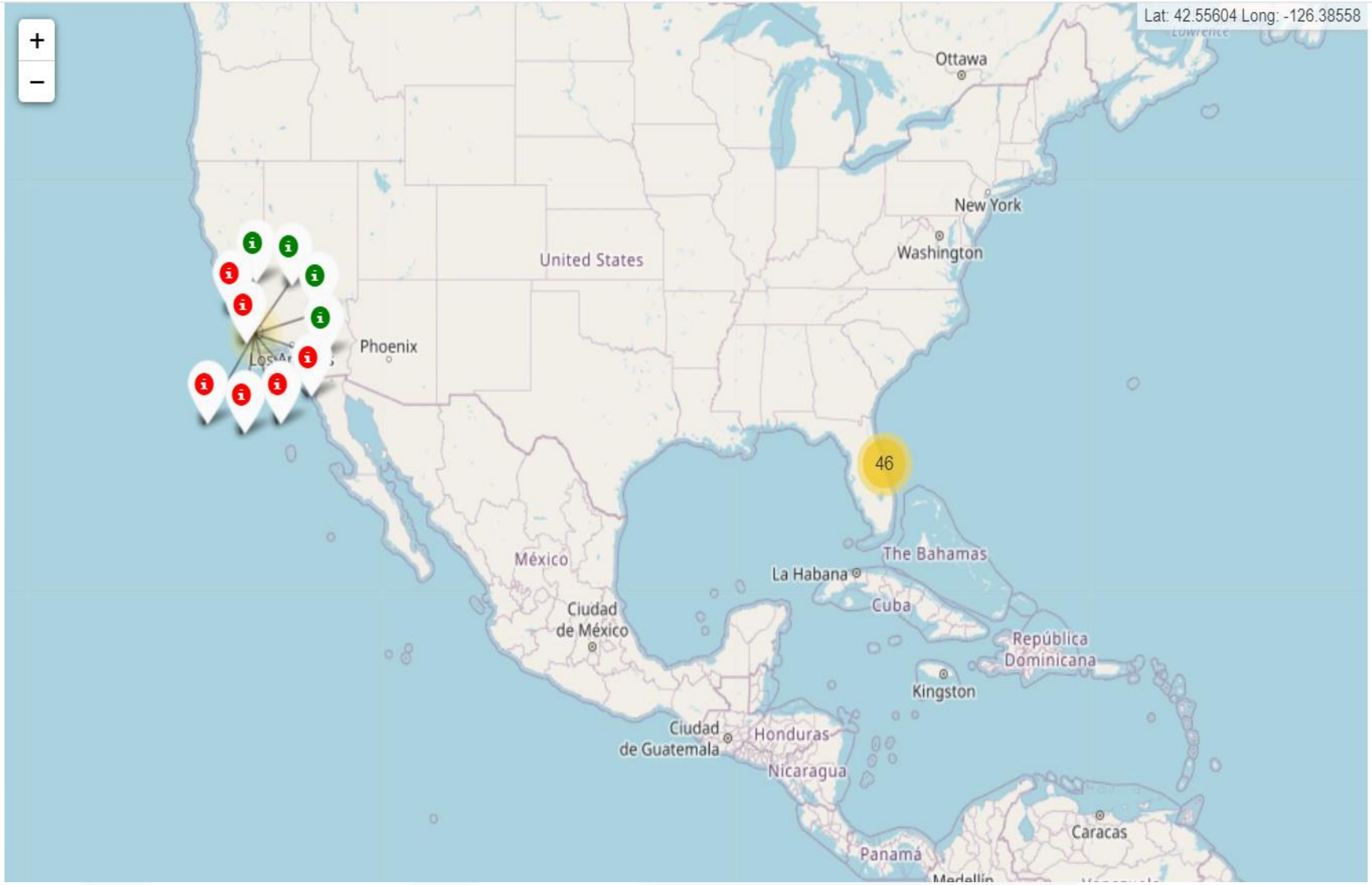
Visual Analysis with Folium

❖ Number of launch on each Sites



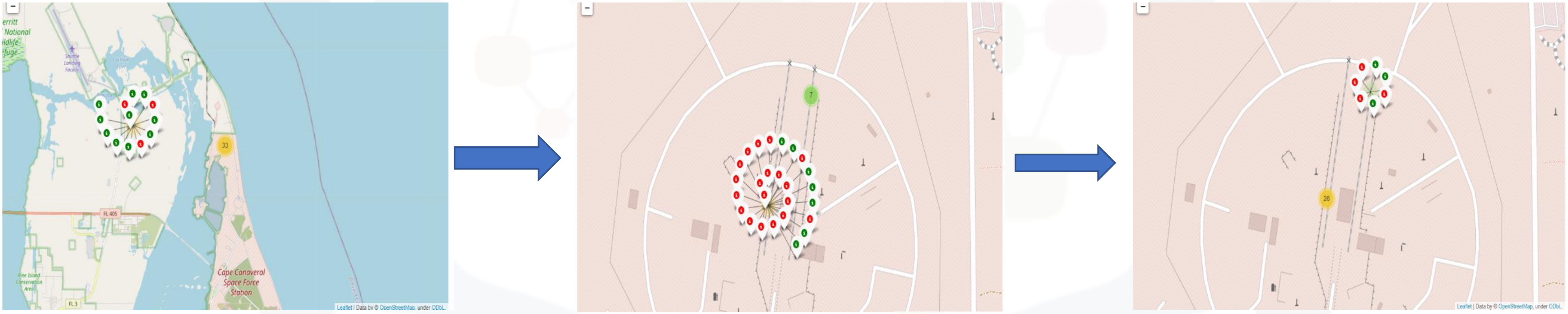
Visual Analysis with Folium

- Visualisation of Successful and uncessfull lanch on Western sites

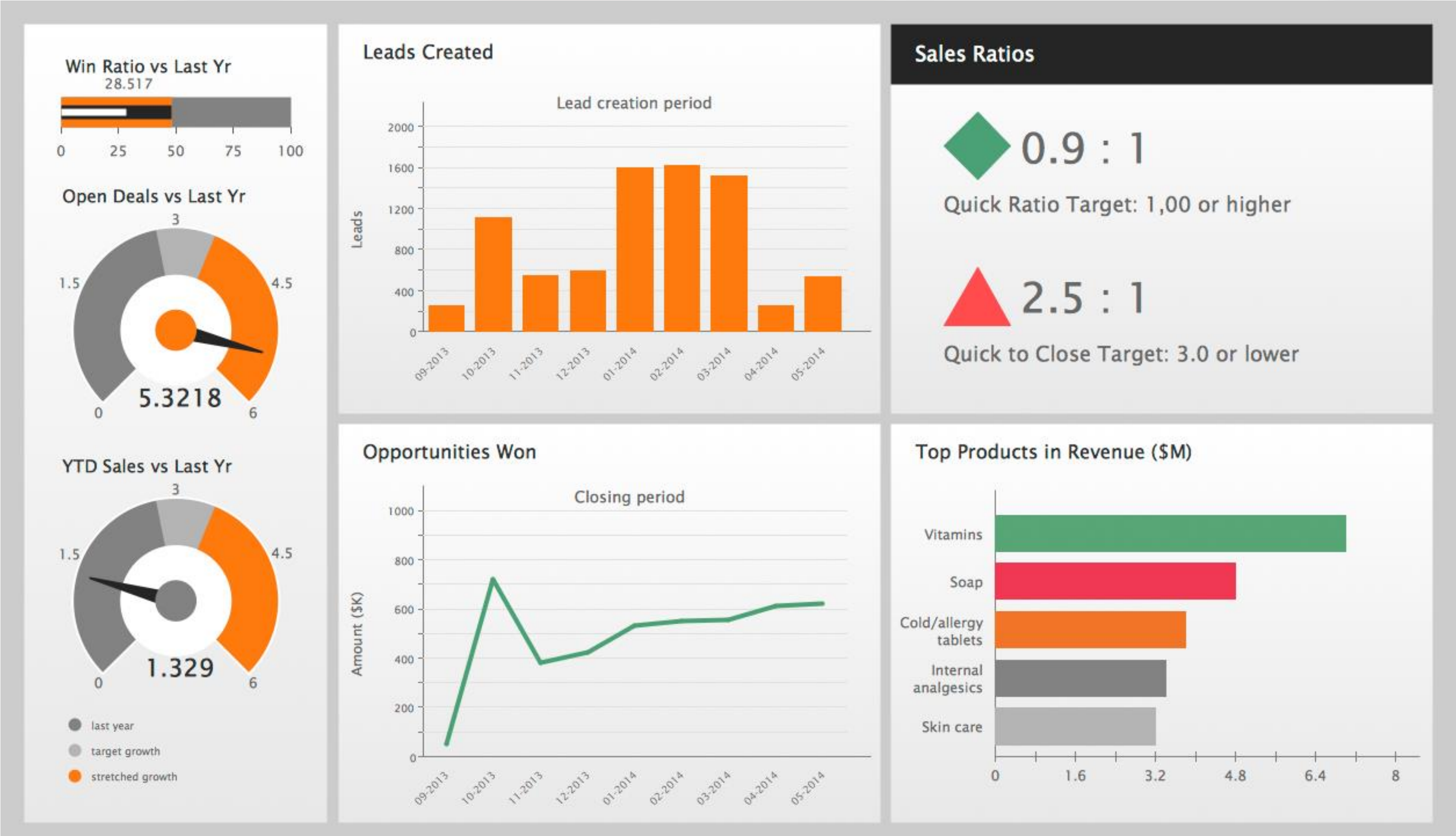


Visual Analysis with Folium

❖ Visualization of Successful and unsuccessful landing on Western sites

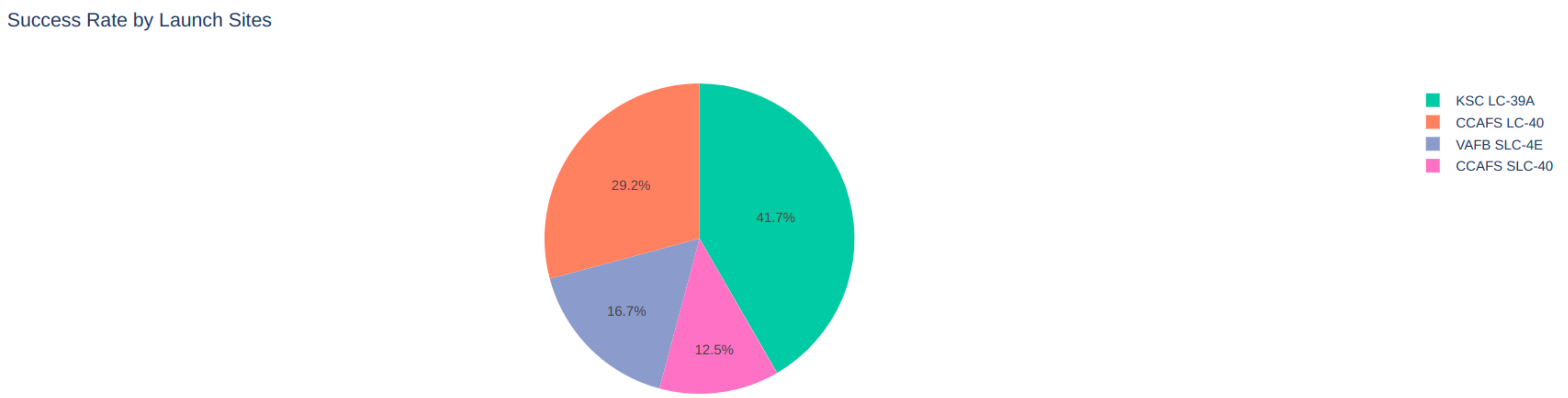


Plotly DASHBOARD



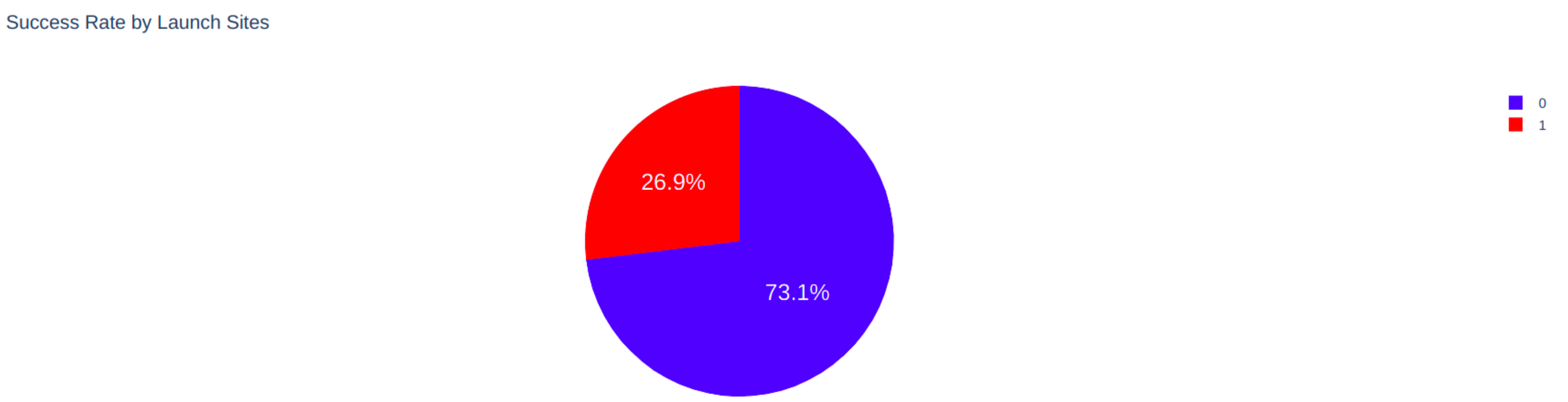
Succes Rates of all sites

All Sites

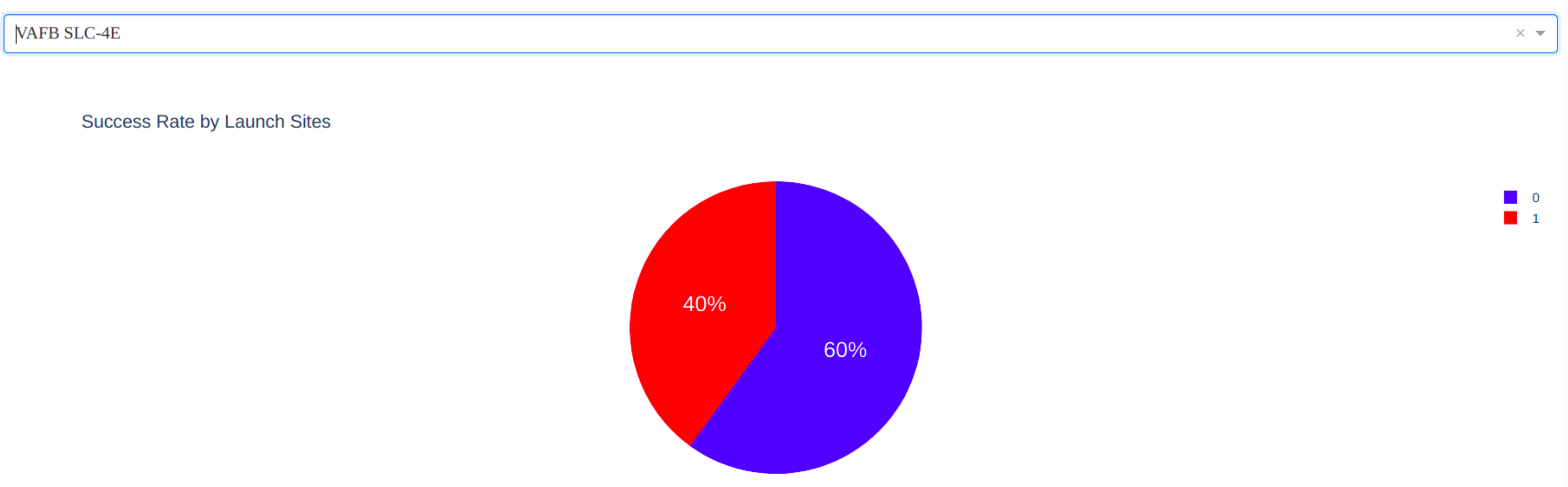


Success Rate of CCAFS LC-40

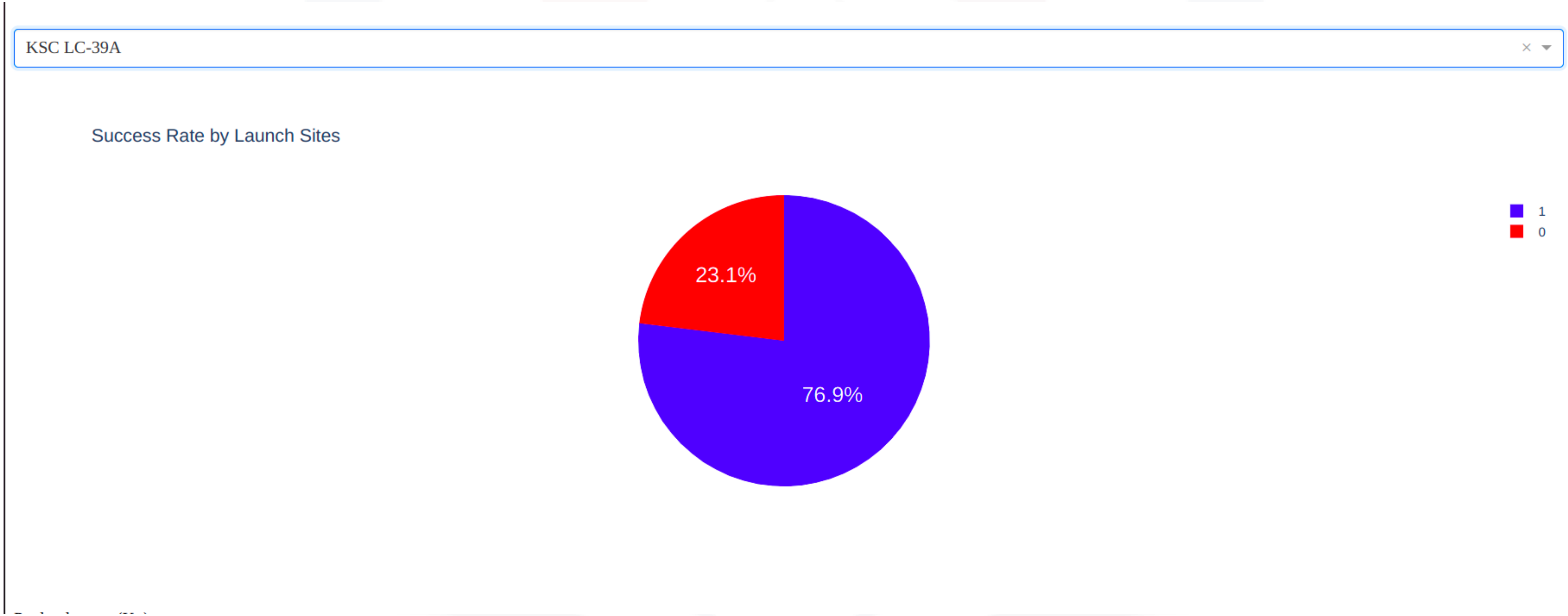
CCAFS LC-40



Succes Rate of VAFB SLC-4B

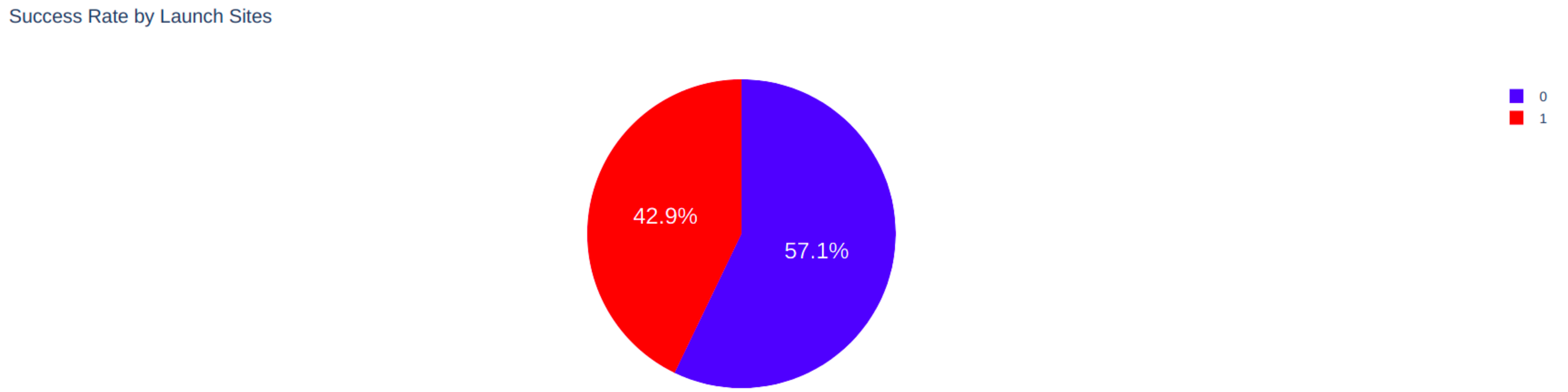


Succes Rate of KSC LC-39A



Success Rate of CCAFS SLC-40

CCAFS SLC-40



Success Corellation by Payload Mass & Bosster Version

Payload range (Kg):

Predictive Analysis Results

Predictive analysis results

- After choosing the best parameters for each algorithm

```
In [15]: parameters ={"C":[0.01,0.1,1],'penalty':['l2'], 'solver':['lbfgs']}# l1 lasso l2 ridge
         lr=LogisticRegression()
         logreg_cv = GridSearchCV(lr, parameters,cv=10)
         logreg_cv.fit(X_train, Y_train)

Out[15]: GridSearchCV(cv=10, estimator=LogisticRegression(),
                      param_grid={'C': [0.01, 0.1, 1], 'penalty': ['l2'],
                                  'solver': ['lbfgs']})
```

Predictive analysis results

- ✓ Logistic Regression Model perform better with an accuracy of around 83%

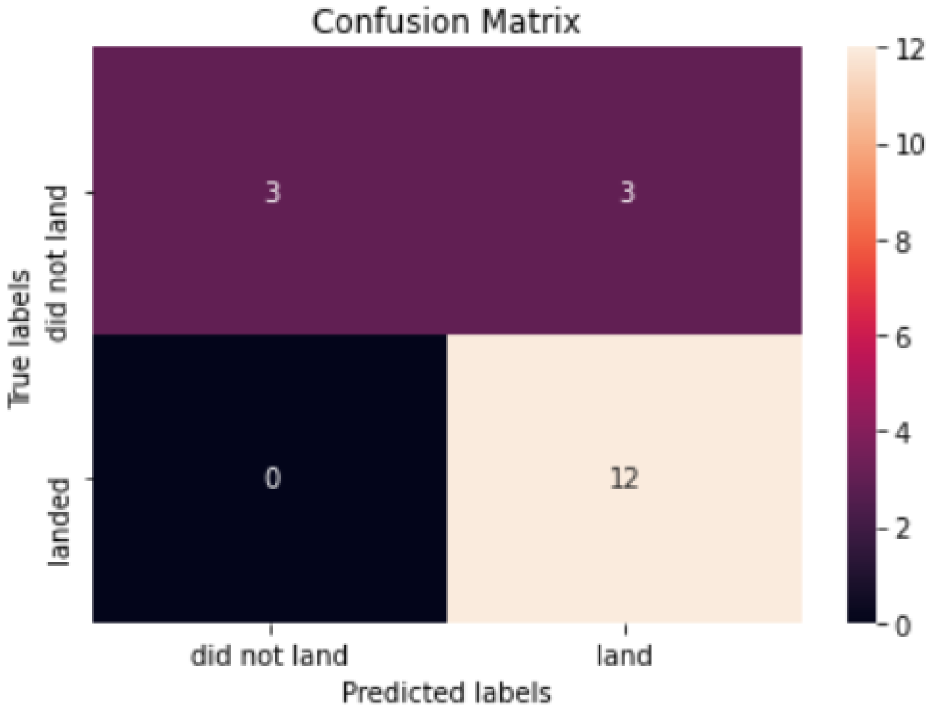
```
In [43]: models=['logreg','svm','tree','kNN']
         results = [logreg_cv.score(X_test,Y_test),svm_cv.score(X_test,Y_test),tree_cv.score(X_test,Y_test),knn_cv.score(X_test,Y_test)]
         results_df=pd.DataFrame(list(zip(models,results)),columns= ['Model name' , 'Results'])
         results_df.sort_values('Results',ascending=False)
```

Out[43]:

	Model name	Results
0	logreg	0.833333
1	svm	0.833333
3	kNN	0.833333
2	tree	0.777778

Predictive analysis results

Confusion Matrix Analysis



- ✓ We got 3 False positives values

OVERALL FINDINGS

SSO , **GEO** and **ES-L1** have **100%** Success Rate

The more a site have launches the more they will be successful

After **2020** success rates have been decreasing

Average payload mass is about **2534 Kg**

We achieved **83%** accuracy using Logistic Regression

3 False Positive

CONCLUSION

3 Classifiers performed well(logreg ,SVM and Knn)

Space X succes rates tend to increase over the year meaning the learning process from previous launches is effective

SSO , GEO and ES-L1 are the best orbits for test

KSC LC-39 A is the best launching site among all the availables