IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

Marta Badi

March ,6/2024

Data Science Capstone Final Project Presentation

Outline

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

Executive Summary

- Summary of methodologies
  - Data collection with API, Web scraping and SQL
  - Data wrangling and analysis
  - Interactive with folium
  - Predictive analysis for every classification
- Summary of all results:
  - Data analysis with interactive visualization
  - Predictive model for analysis

Introduction

- This capstone project course will give you a taste of what data scientists go through in real life when working with real datasets.
- You will assume the role of a Data Scientist working for a startup intending to compete with SpaceX, and in the process follow the Data Science methodology involving data collection, data wrangling, exploratory data analysis, data visualization, model development, model evaluation, and reporting your results to stakeholders.
- Here we will predict if Falcon 9 first stage land successfully.
- Falcon 9 advertises if rockets launches on its websites, with the cost of 62 million dollars.
- This information will be used if an alternate company wants to bid against SpaceX for a rocket launch

❖**Problem we want to answer**

- With what factor the rocket will land?
- The effect of each relationship of rocket variable on outcome

Section 1

Methodology

Methodology

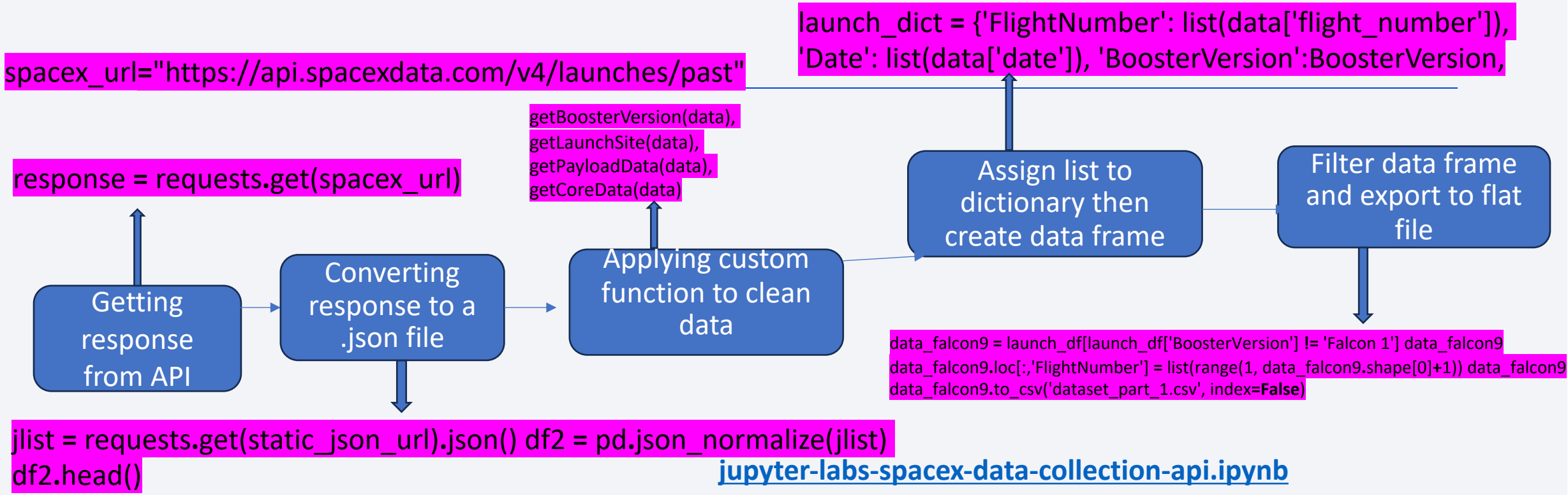
Executive Summary

- Data collection methodology:
  - Using SspaceX rest API
  - Web scraping Wikipedia
- 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: Meaning and Steps

- Data collection is the process of gathering and measuring information on targeted variables in an established system, which enables one to answer relevant questions and evaluate outcomes.
  1. Getting data from API or Webpage
  2. Make data frame from it.
  3. Filter data frame as per requirement
  4. Export to flat file

Data Collection – SpaceX API



[jupyter-labs-spacex-data-collection-api.ipynb]

FlightNumber	Date	BoosterVersion	PayloadMass	Orbit	LaunchSite	Outcome	Flights	GridFins	Reused	Legs	LandingPad	Block	ReusedCount	Serial	Longitude	Latitude	Class	
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	None None	
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	None None	
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	None None	
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	False Ocean	
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	None None	

Data Collection - Scraping

Response from HTML

Creating Beautiful object

Finding Tables

Getting Column names

Creation of Dictionary and appending data keys

Converting dictionary to data frame

Data frame to .CSV

static_url=https://en.wikipedia.org/w/index.php?title=List_of_Falcon_9_and_Falcon_Heavy_launches&oldid=1027686922, html_data = requests.get(static_url)

soup = BeautifulSoup(html_data.text, 'html.parser')

html_tables = soup.find_all('table'), first_launch_table = html_tables[2]

```
name = extract_column_from_header(element[row])
    if (name is not None and len(name) > 0):
        column_names.append(name)
```

launch_dict= dict.fromkeys(column_names) jupyter-labs-webscraping (1).ipynb

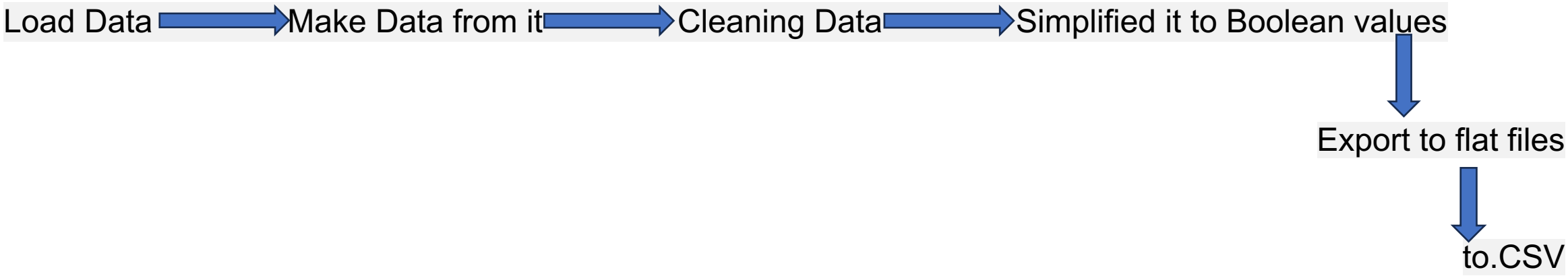
Flight No.	Launch site	Payload	Payload mass	Orbit	Customer	Launch outcome	Version Booster	Booster landing	Date	Time
0	1	CCAFS	Dragon Spacecraft Qualification Unit	0	LEO	SpaceX	Success\n	F9 v1.0B0003.1	Failure	4 June 2010	18:45
1	2	CCAFS	Dragon	0	LEO	NASA	Success	F9 v1.0B0004.1	Failure	8 December 2010	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 Meaning and the steps

Data wrangling is the process of cleaning and unifying messy and complex data sets for access and analysis
In this lab, we performed some Exploratory Data Analysis (EDA) to find some patterns in the data and determine what would be the label for training labels, ‘’1’’ means booster successfully landed else ‘’0’’.

```
df['Class'] = df['Outcome'].apply(lambda landing_class: 0 if landing_class in bad_outcomes else 1)
df[['Class']].head(8)
```

The Steps:



Data Wrangling

Calculate number of launches at site

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

```
CCAFS SLC 40     55
KSC LC 39A       22
VAFB SLC 4E      13
```

Calculate number and occurrence of each orbits

```
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
```

Calculate number and occurrence of each outcome per orbit type

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

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

```
df['Class'] = df['Outcome'].apply(lambda landing_class: 0 if landing_class in bad_outcomes else 1)
df[['Class']].head(8)
```

Create landing outcome label from outcome column

Export data as.CSV

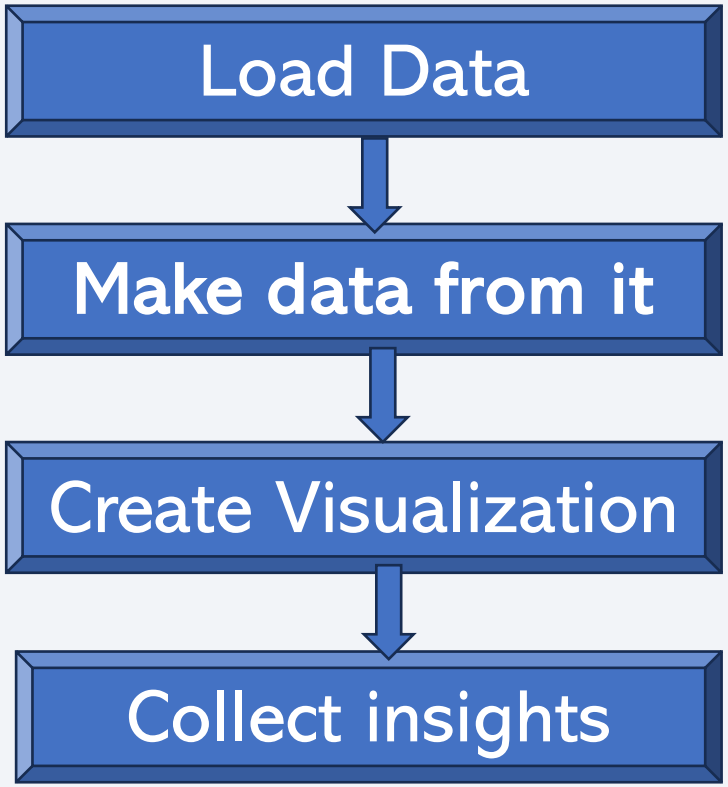
```
df.to_csv("dataset_part_2.csv", index=False)
```

[labs-jupyter-spacex-Data wrangling (1).ipynb]

	FlightNumber	Date	BoosterVersion	PayloadMass	Orbit	LaunchSite	Outcome	Flights	GridFins	Reused	Legs	LandingPad	Block	ReusedCount	Serial	Longitude	Latitude	Class
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	0
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	0
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	0
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	0
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	0

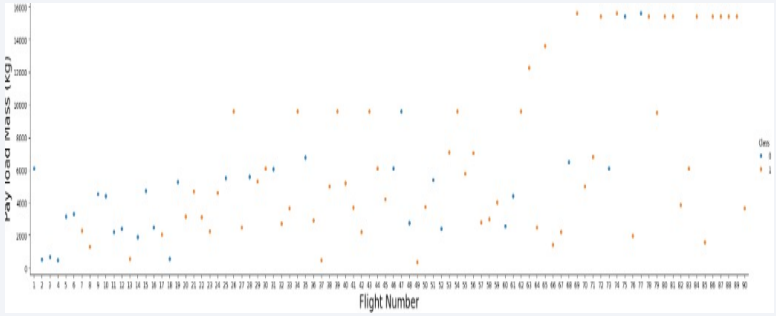
EDA with Data Visualization

- **EDA meaning and Steps:**
- Exploratory data analysis is an approach to analyze data and summarize their main characteristics using statistical graphing

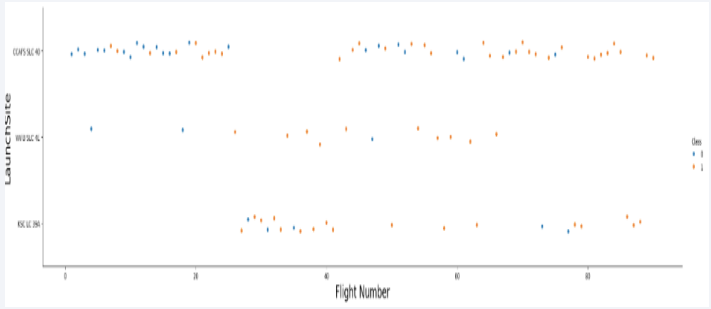


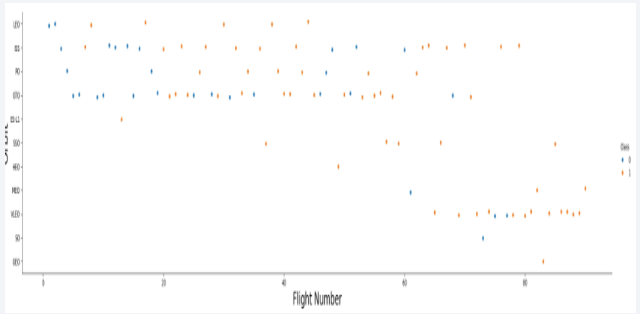
EDA with Data Visualization

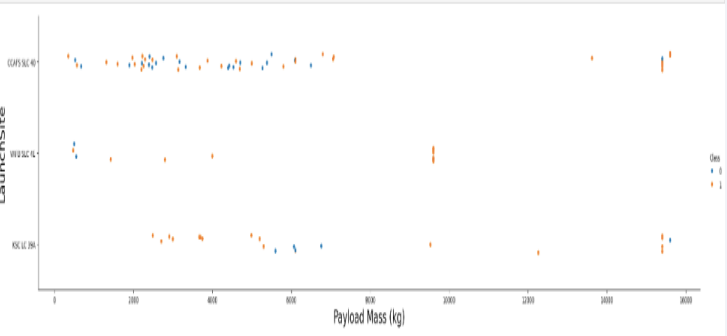
- **Scatter graphs were Drawn:**
  - **Payload and flight number**
  - **Flight number and launch site**
  - **Pay load and Launch site**
  - **Flight number and orbit type**
  - **Payload and orbit type**

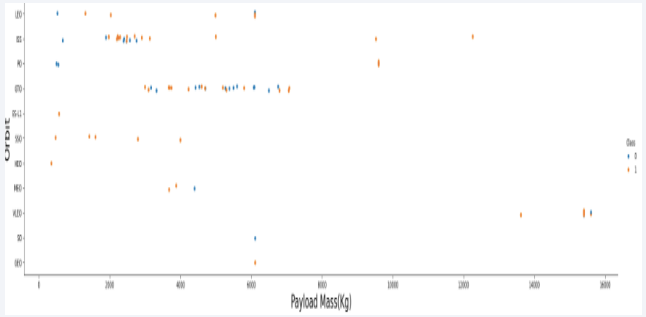


- **Scatter plots show dependency of attributes on each other**

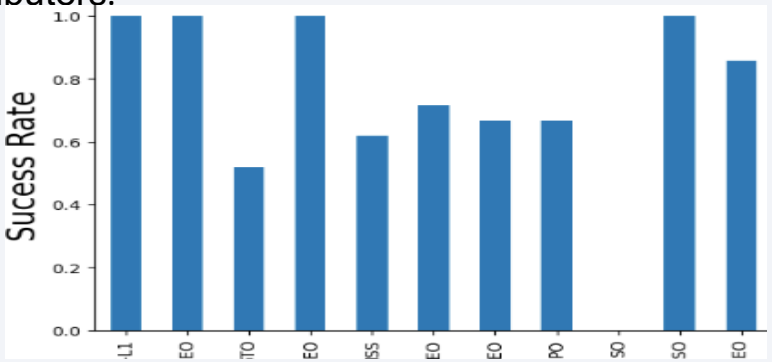






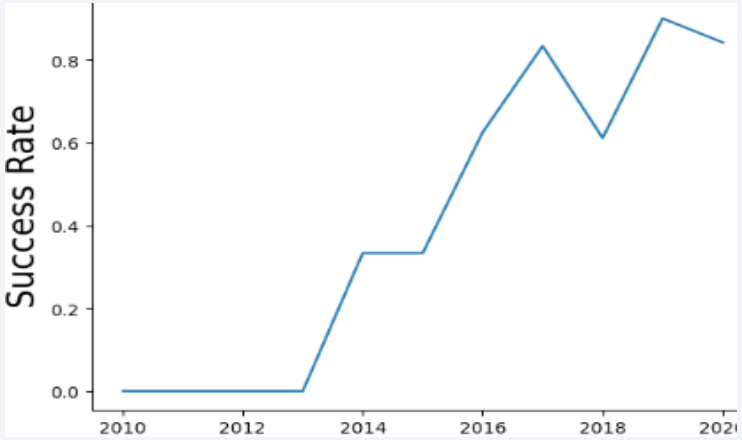


- Bar graph Drawn
  - Success rate vs orbit type
  - Bar graph is easiest to interpret a relationship between the attributers.



- **Line graph drawn**
  - **Launch success vs yearly trend**

[jupyter-labs-eda-dataviz.ipynb.jupyterlite.ipyn]



EDA with SQL

- SQL is an indispensable tool for Data Scientist and Analysists as most of the real-world data is stored in databases. It is not only the standard language for Relational Database operations, but also an incredibly powerful tool. For analyzing data and drawing useful insights from it.
- !pip install sqlalchemy==1.3.9
- !pip install ibm_db_sa
- !pip install ipython-sql
- %load_ext sql
- We performed SQL queries to gather information from given dataset.
    - Displayinf the names of unique launch sites in the space mission
    - Display. Records where launch sites begin with the string ‘CCA
    - Displaying the total payload mass carried by boosters launched by NASA
    - Listing the date where the successful landing outcomes in drone was achieved
    - Listing the names of the booster which have which have succussing ground pad and have payload mass greater than 4000 but less than 6000
    - Listing the total number of successful and failure mission outcomes
    - Listing names of the booster versions which have carried the maximum payload mass
    - Listing the failed landing outcomes in drone ship the booster versions and launch sites names for the year 2015
    - Ranking the count of outcomes(such as failure drone ship) or success (ground pad) between the date 2010-06-04 and 2017-03-20 in descending order.

jupyter-labs-eda-sql-coursera_sqllite (1).ipynb

Build an Interactive Map with Folium

- Folium makes it easy to visualize data that has been manipulated in python on an interactive leaflet map. We use the latitude and longitude coordinates for each launch site and added a circle marker around each site with a label of the name of the launch site. It is also easy to visualize the number of success and failure for each launch site with Green and red master on the map.

Map objects	Code	Result
Map marker	Folium.Marker(	Map object to make a map
Icon marker	FoliumIcon(	Create an icon on map
Circle markers	FoliumCircle(	create a circle where marker is being placed
Polyline	FoliumPolyline(	Create a line between points
Marker cluster	MarkerCluster()	This is a good way to simplify a map containing many markers having the same coordinate.
Antpath	Foliumpluginf Antpath(	Create an animated line between points

lab_jupyter_launch_site_location.ipynb

Build a Dashboard with Plotly Dash

- Summarize what plots/graphs and interactions you have added to a dashboard
- Explain why you added those plots and interactions
- Add the GitHub URL of your completed Plotly Dash lab, as an external reference and peer-review purpose

Predictive Analysis (Classification)

❖ **Building model:**

- Load out feature engineered data into data frame
- Transform it in to Numpy arrays
- Standards and transform data
- Split data in to training and test data sets
- Check how many test samples has been created
- List down machine learning algorisms we want to use
- Set our parameters algorism to GridSearchCV
- Fit our datasets into the GridSearchCV objects and

train our model

```
Y = data['Class'].to_numpy()
X= preprocessing.StandardScaler().fit(X).transform(X)
X_train, X_test, Y_train, Y_test = train_test_split( X, Y, test_size=0.2, random_state=2) print ('Train set:', X_train.shape, Y_train.shape) print ('Test set:', X_test.shape, Y_test.shape)
```

❖ **Finding the best performing classification model**

- The model with best accuracy score wins the best performing
- model

```
lr=LogisticRegression() grid_search = GridSearchCV(lr, parameters, cv=10) logreg_cv = grid_search.fit(X_train, Y_train)
parameters = {'kernel':('linear', 'rbf','poly','rbf', 'sigmoid'), 'C': np.logspace(-3, 3, 5), 'gamma':np.logspace(-3, 3, 5)}
svm = SVC()
parameters = {'n_neighbors': [1, 2, 3, 4, 5, 6, 7, 8, 9, 10], 'algorithm': ['auto', 'ball_tree', 'kd_tree', 'brute'], 'p': [1,2]}
KNN = KNeighborsClassifier()
```

❖ **Evaluating model**

- Check accuracy for each model
- Get best hyperparameters for each type of algorism
- Plot confusion matrix.

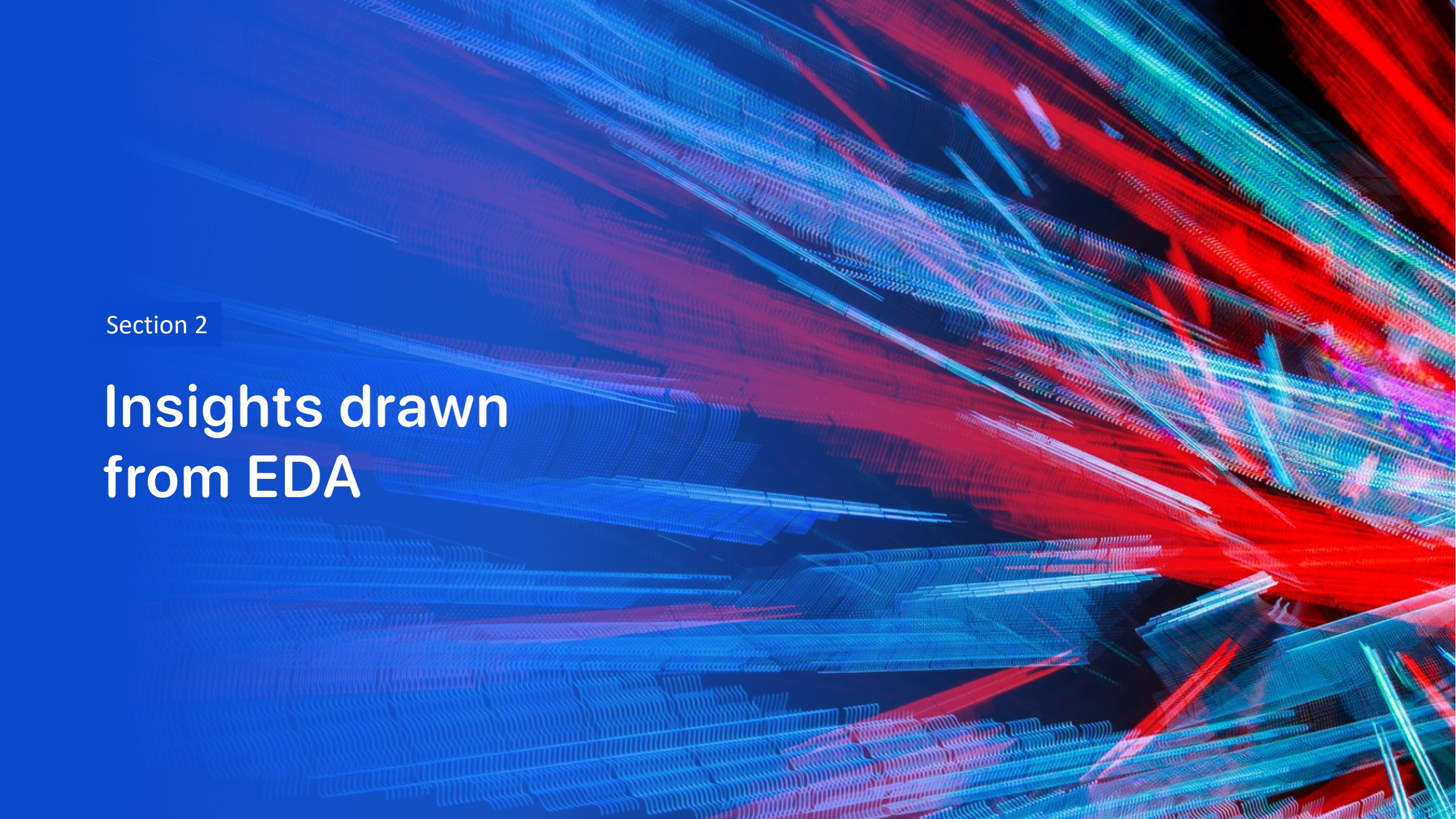
```
yhat = knn_cv.predict(X_test) , plot_confusion_matrix(Y_test,yhat)
```

❖ **Best model**

SpaceX_Machine_Learning_Prediction_Part_5.jupyterlite (3).ipynb

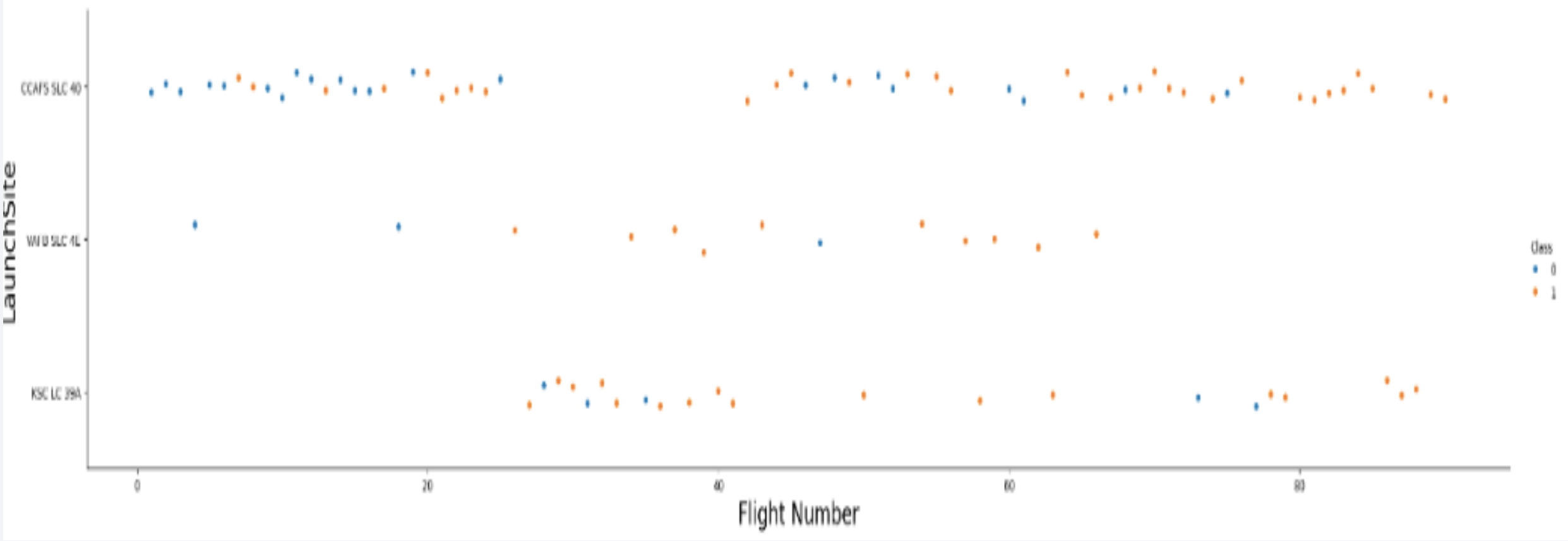
Results

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


Section 2
Insights drawn from EDA

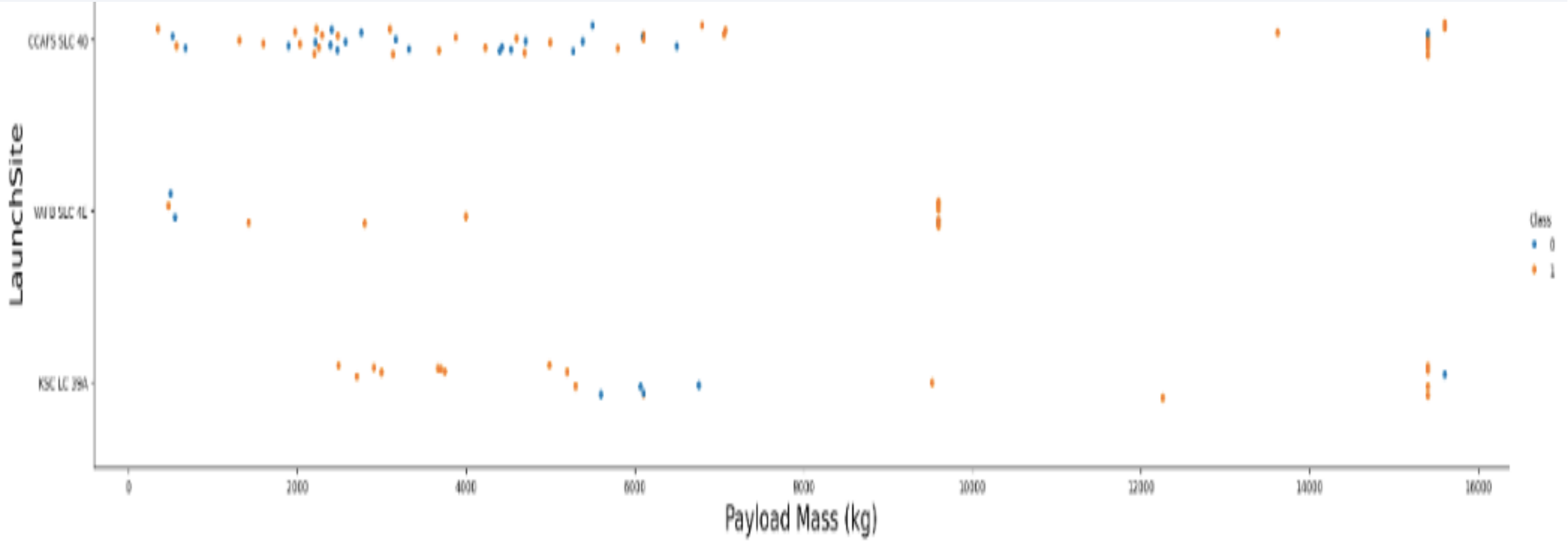
Flight Number vs. Launch Site

- With higher flight numbers (greater than 30) the success rate for the Rochet is increasing

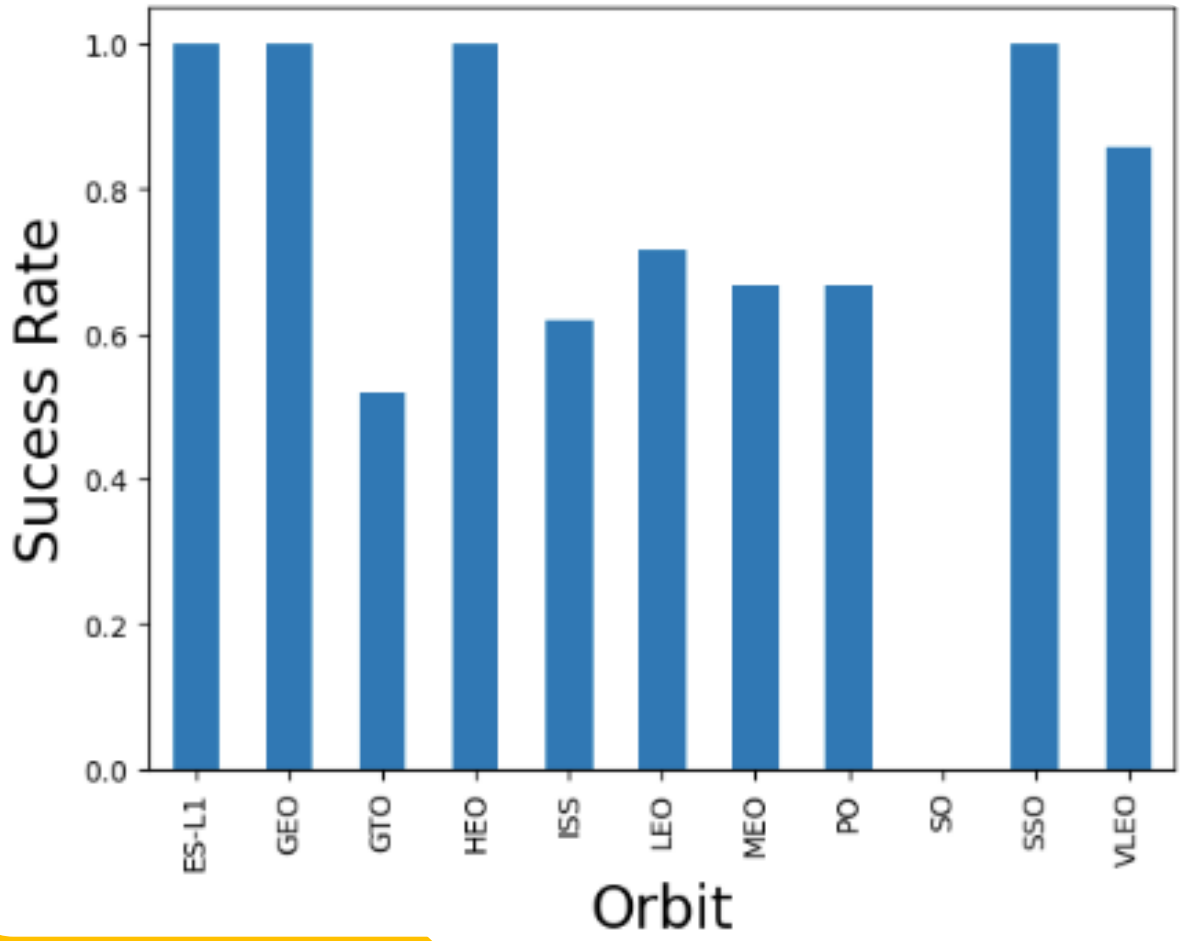


Payload vs. Launch Site

The greater the payload mass (700kg) higher the success rate for the Rochet. But, there is no clear pattern to take decision, if the launch site is dependent on the payload mass for a success launch.



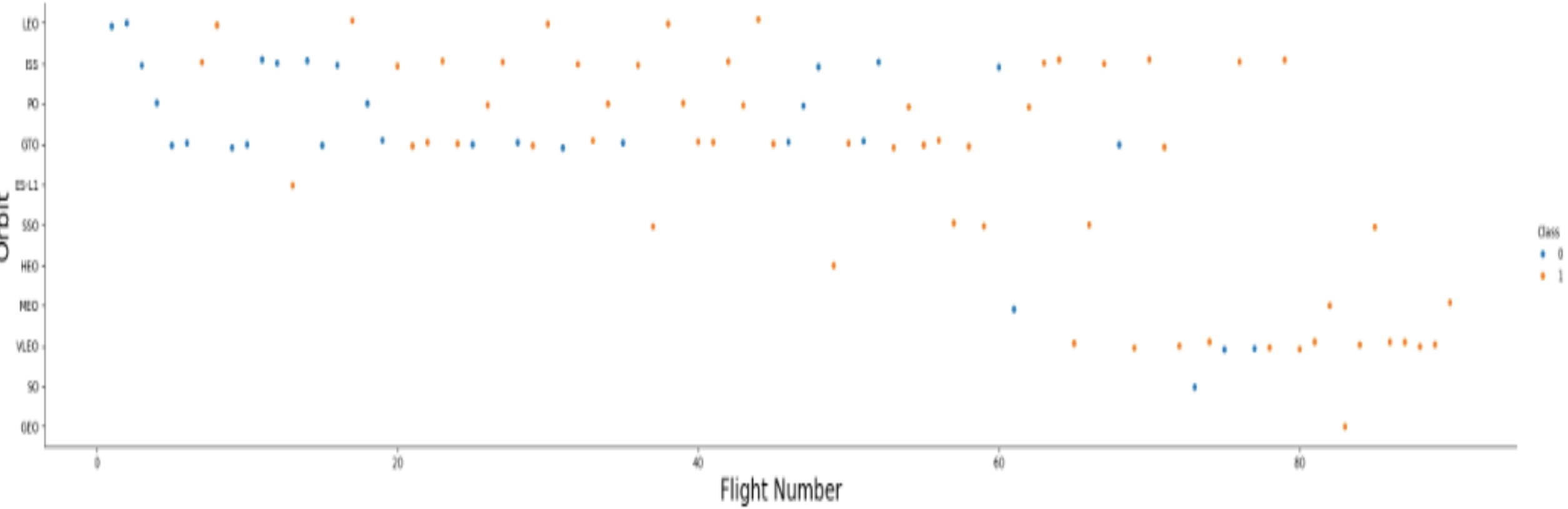
Success Rate vs. Orbit Type



- ES.L1, GEO, HEO, SSO has higher success rate

Flight Number vs. Orbit Type

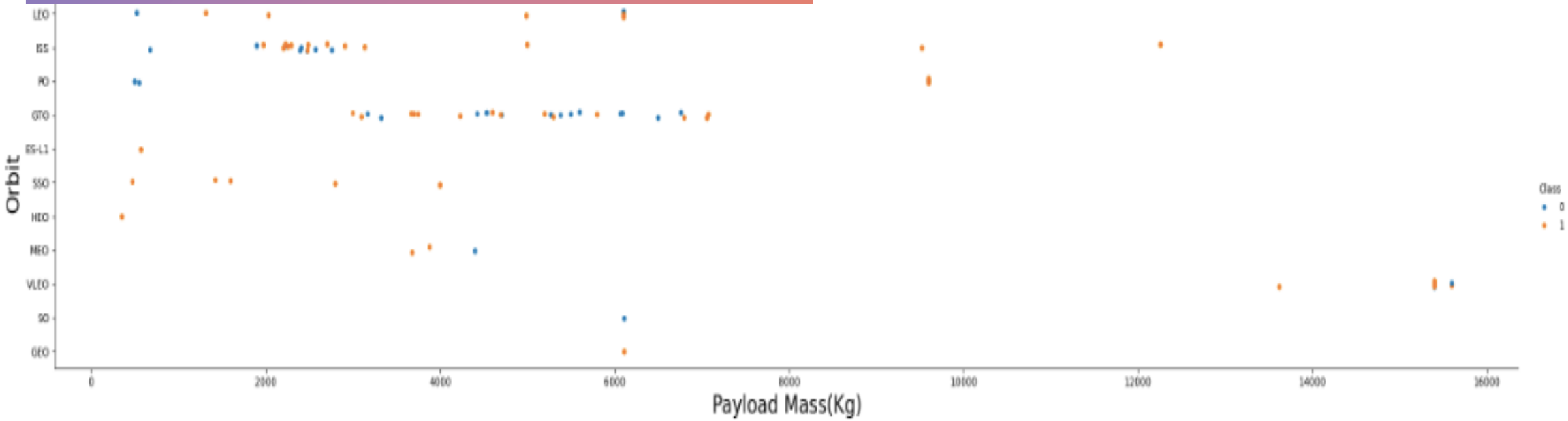
- We see that for LEO orbit the success increases with the number of flights.
- On the other hand, there seems to be no relationship between flight number and GTO orbit.



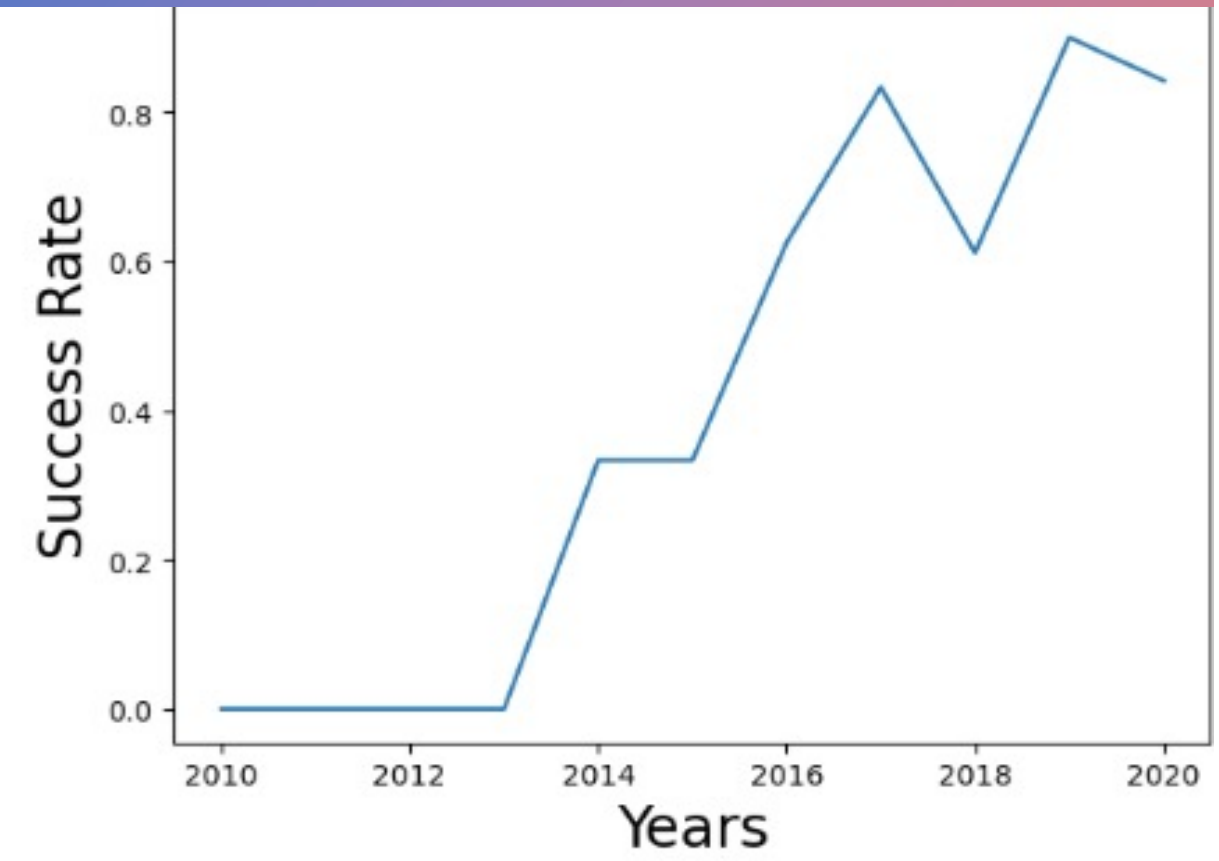
Payload vs. Orbit Type

We observe that heavy payload have a negative influence on MEO, GTO, VLEO orbits.

Positive on LEO, ISS orbits



Launch Success Yearly Trend



- We observe that success rate since 2013 kept increasing relatively though there is slight dip after 2019.

All Launch Site Names

- SQL Query

```
%sql select Unique(LAUNCH_SITE) from SPACEXTBL;
```

- Description
- Using the word UNIQUE in the query we pull unique values for the launch site column from table SPAXE.

Launch Site
CCAFS LC-40
CCAFS SLC-40
KSC LC-39A
VAFB SLC-4E

Launch Site Names Begin with 'CCA'

Launch_Site
CCAFS LC-40
CCAFS LC-40
CCAFS LC-40
CCAFS LC-40
CCAFS LC-40

- **SQL Query**

```
%sql SELECT LAUNCH_SITE from SPACEXTBL where (LAUNCH_SITE) LIKE 'CCA%' LIMIT 5;
```

- **Description**
- Using the word ‘LIMIT 5’ in the query we fetch 5 records SpaceX with condition LIKE key word with wild card: ‘CCA%’.
- The percentage in the end suggests that the LAUNCH_SITES name must stars with CCA.

Total Payload Mass

- **SQL Query**

```
%sql select sum(PAYLOAD_MASS__KG_) as payloadmass from SPACEXTBL;
```

- **Description:**

Using the function ‘SUM’ calculates the total in the column PAYLOAD_MASS_KG_ and WHERE clause filters the data to fetch Customer’s by name “NASA(CRS)”

Pay load mass
619967

Average Payload Mass by F9 v1.1

SQL query

```
%sql sCelect avg(PAYLOAD_MASS__KG_) as payloadmass from SPACEXTBL;
```

Description

Using the function AVG works out the average in the column PAYLOAD_MASS_KG

The WHERE clauses the dataset to only perform calculations on BOOSTER_version “F9v1.1”.

payloadmass
6138.287128712871

First Successful Ground Landing Date

SQL Query

```
%sql select min(DATE) from SPACEXTBL;
```

Description

Using the function MIN works out the minimum date in the column Date and

WHERE clause filters the data to only perform calculations on landing.

Outcomes with values “Success(ground pad)”

min(DATE)
2010-06-04

Successful Drone Ship Landing with Payload between 4000 and 6000

SQL Query

```
%sql SELECT BOOSTER_VERSION FROM SPACEX WHERE LANDING__OUTCOME = 'Success (drone ship)' \
AND PAYLOAD_MASS__KG_ > 4000 AND PAYLOAD_MASS__KG_ < 6000;
```

Selecting only Booster_Version, Where clause filters the datasets to Landing_Outcome = Success(drone ship)

And clauses specifies additional filters conditions

Payload_MASS_KG>4000 AND Payload_MASS_KG_<6000

boosterversion
F9 B5 B1048.4
F9 B5 B1049.4
F9 B5 B1051.3
F9 B5 B1056.4

Total Number of Successful and Failure Mission Outcomes

SQL Query

```
%sql SELECT BOOSTER_VERSION FROM SPACEX WHERE LANDING__OUTCOME = 'Success (drone ship)' \
AND PAYLOAD_MASS__KG_ > 4000 AND PAYLOAD_MASS__KG_ < 6000;
```

Description

Selecting multiple count is a complex query. I have used case clause within sub query for getting both success and failure counts in same query.

Case when MISSION_OUTCOME LIKE '%Success% then 1 else 0 end' returns a Boolean value which we sum to get the result needed.

Successful Mission	Failure Mission
100	1

Boosters Carried Maximum Payload

- SQL Query

```
%sql select BOOSTER_VERSION as boosterversion from SPACEXTBL where PAYLOAD_MASS__KG_=(select max(PAYLOAD_MASS__KG
```

➢Description

- Using the function MAX works the maximum payload in
- the column PAYLOAD_MASS_KG_ in sub query.
- WHERE clauses filters Booster Version which had
- maximum payload.

boosterversion
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

2015 Launch Records

SQL Query

```
%sql SELECT MONTH(DATE),MISSION_OUTCOME,BOOSTER_VERSION,LAUNCH_SITE FROM SPACEXTBL where EXTRACT(YEAR FROM DATE)='2015';
```

- Description:
- We need to 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.
- Via year function we extract the year and future where clause “Failure(drone ship)” fetches our required values.

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

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

SQL Query

```
%sql SELECT LANDING__OUTCOME FROM SPACEXTBL WHERE DATE BETWEEN '2010-06-04' AND '2017-03-20' ORDER BY DATE
```

COUNT count records in column LANDIG_OUTCOME

WHERE filters data with '%Success%'

AND DATE>'2010-06-04'

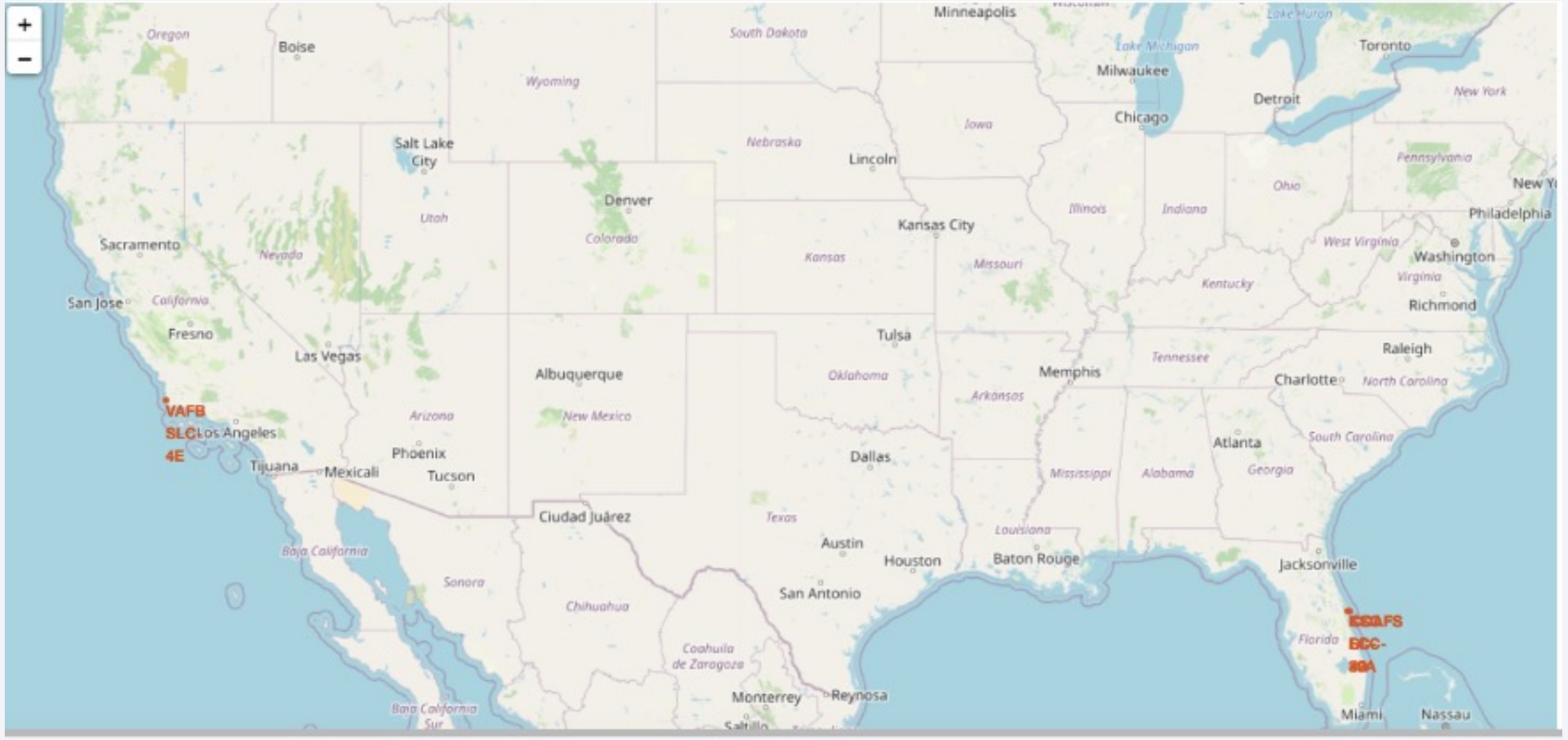
AND DATE<'2017-03-20'

Rank success count between 2010-06-04 and 2017-03-20
8


Section 3
Launch Sites
Proximities Analysis

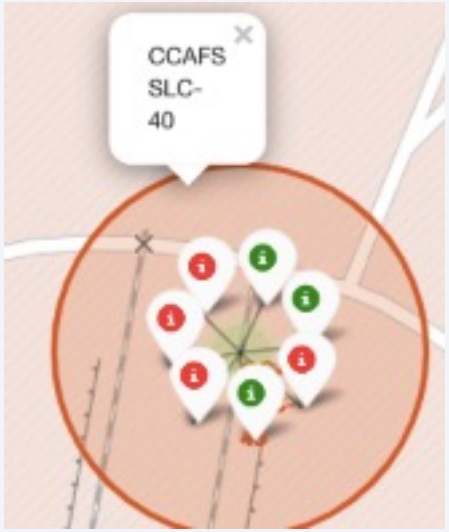
All Launch sites on Folium Map

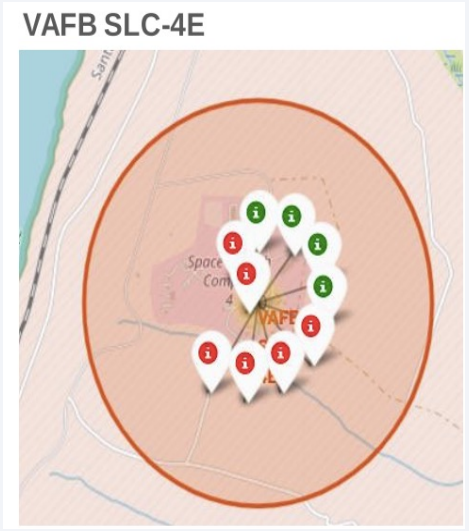
- We can see the spacex launch sites are near to the United States of America coasts i.e Florida and Canada regions.

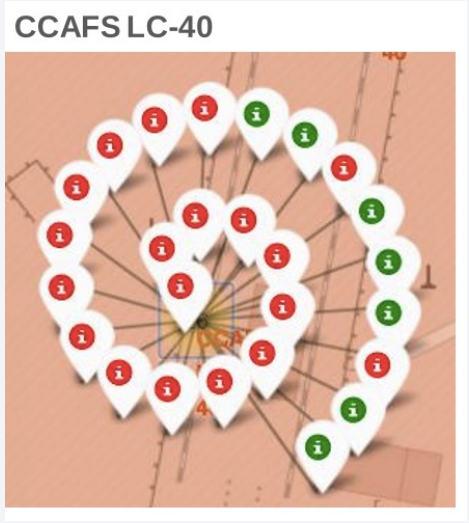


Color labeled Launch Records

- Green Marker i Shows successful launchs
- Red Marker i shows failures.
- From these screenshots its easily to understand successful and failure launchs. KSC LC-39A has has high probability of success.

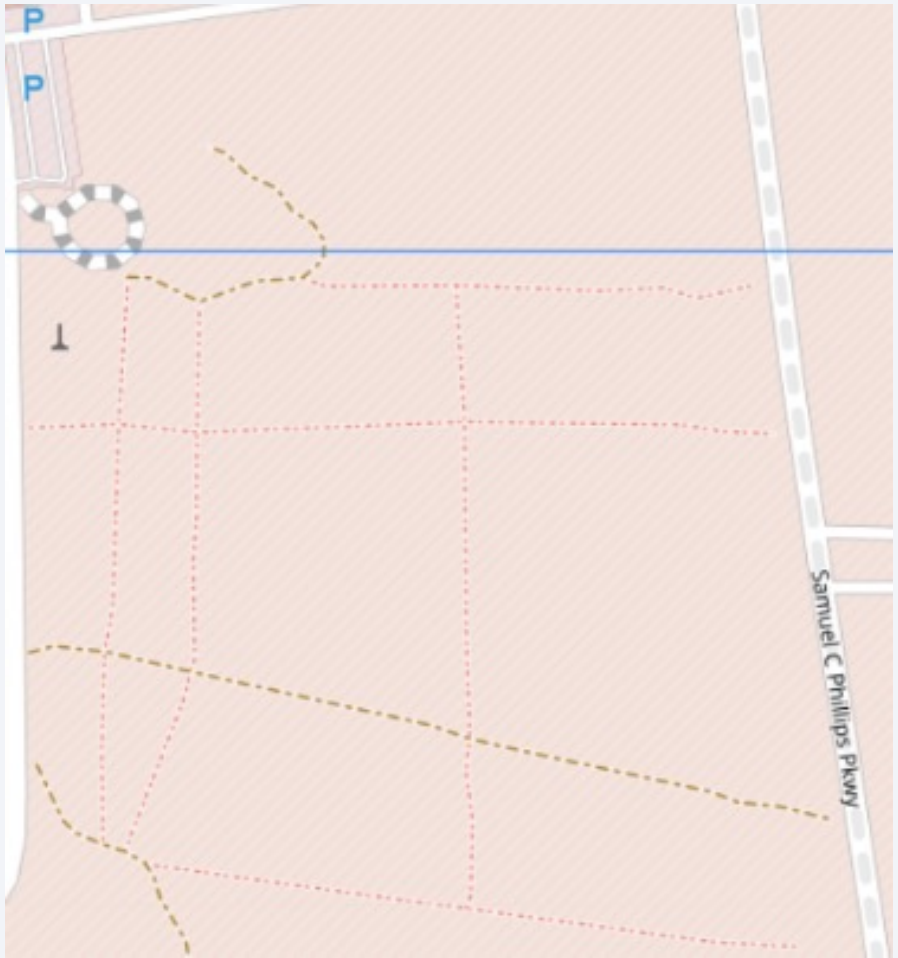


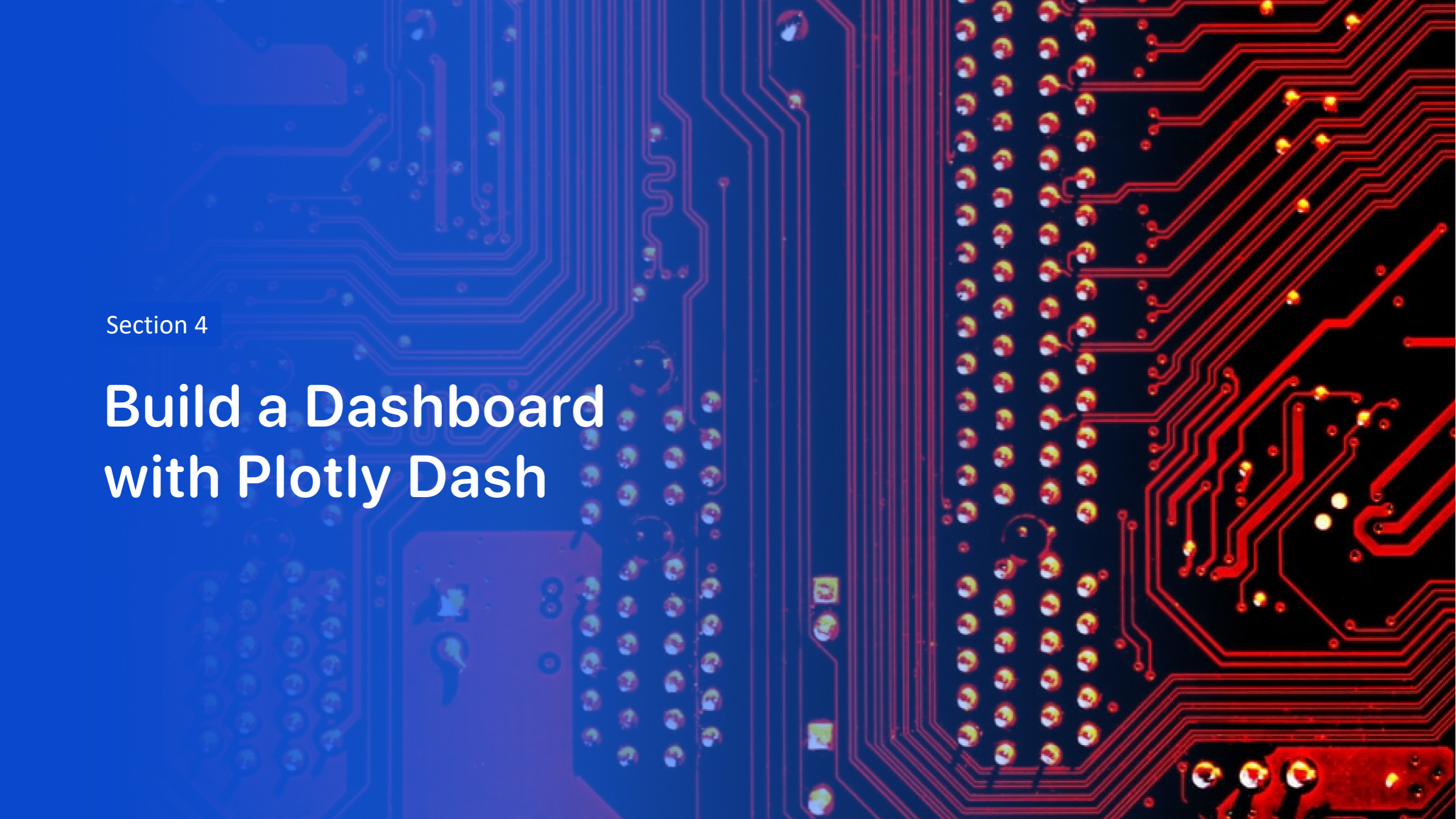




Launch site distance from equator to railway

- Distance from equator is greater than 3000km for all sites.
- Distance for all launch sites from railway tracks are greater than .7km for all sites. So, launch sites are not so far away from railway tracks.



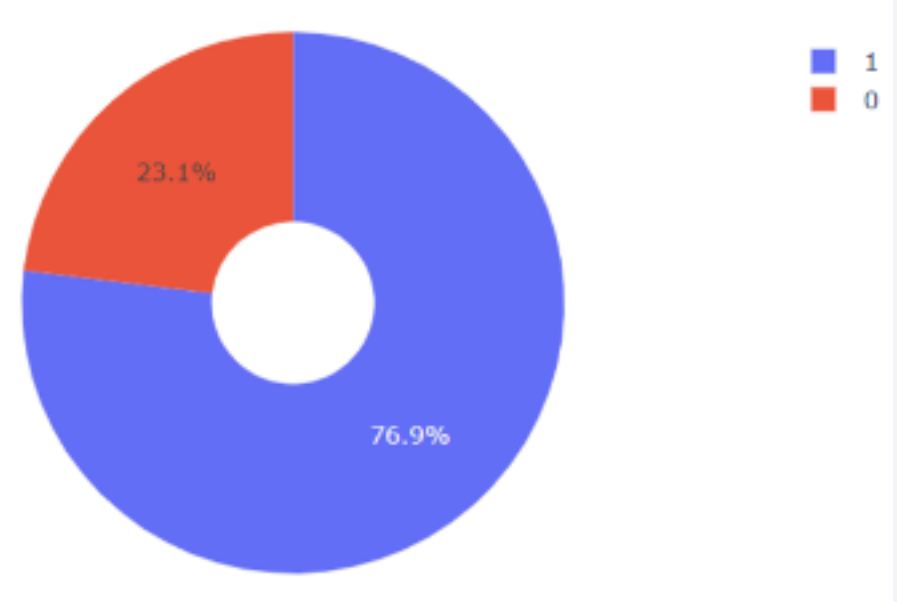

Section 4
Build a Dashboard with Plotly Dash

Launch success count for all

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

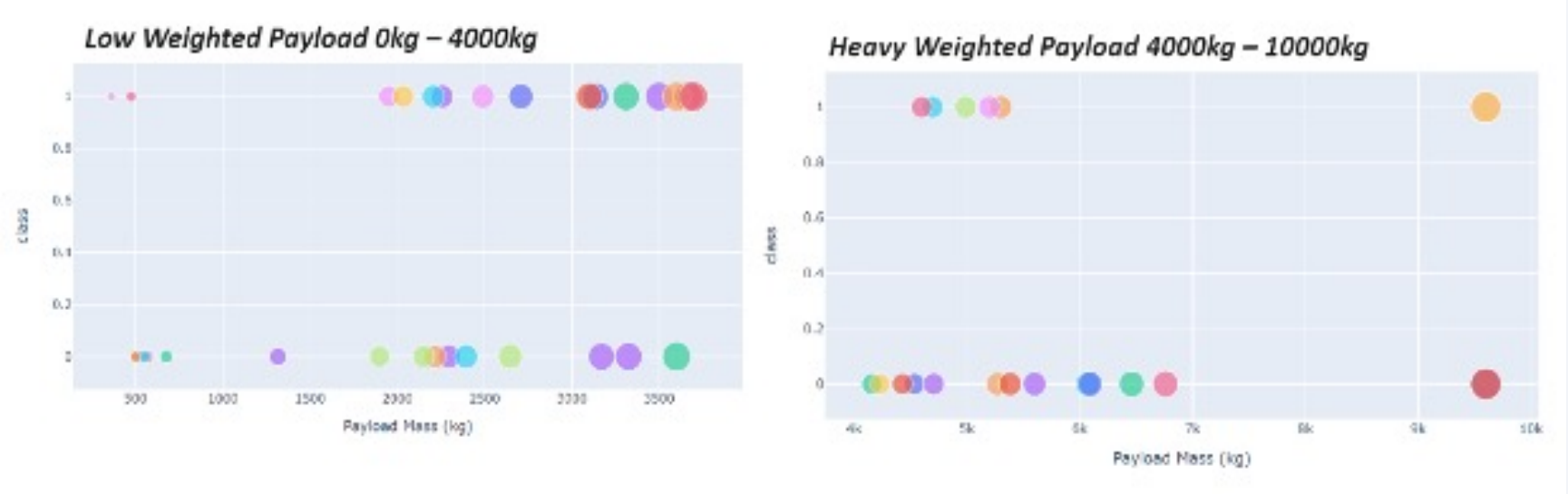
Launch Site with Highest Launch Success Ratio

- KSL LC-39A ACHIEVED A 76.9% Success rate while getting a 23.% failure rate
- After visual analysis using the dashboard, we are able to obtain some insights to answer these questions:
- White site has the highest launch success rate? KSC LC-39A.
- Which payload range has the highest launch success rate? 2000-10000kg
- Which payload range has lowest has the lowest launch success rate? 0-1000kg
- Which F9 Booster version (v1.0, v1.1, FT, B4, B5, etc) has the highest launch success rate? FT



Payloads vs. Launch outcomes Scatter Plot for all Sites

- We can see the success rates for low weighted payloads is higher than the heavy weighted payloads

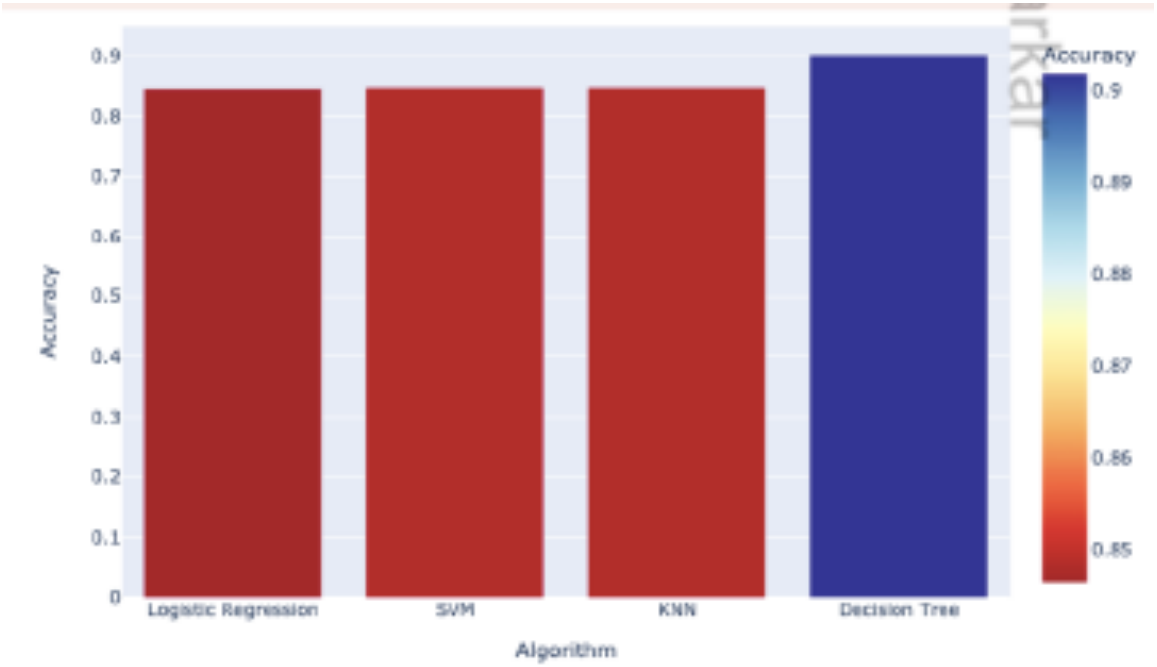


Section 5

Predictive Analysis (Classification)

Classification Accuracy

- As you can see our accuracy is extremely close, but we do have a clear winner which performs best – “Decision Tree” with a score of 0.90178.
- We trained four different models which each had an 83% accuracy rate.



Algorism	Accuracy	Accuracy on Test Data	Tuned Hyperparameter
Logistic regression	0.846429	0.833334	(‘C’:0.01,penality’: ‘12’ , ‘solver’, ‘lbgs’)
SVM	0.848214	0.833334	(‘C’: 1.0’gamma’: 0.031612277660168379, ‘kernel’: ‘sigmoid’))
KNN	0.848214	0.833334	[‘algorism’: ‘auto’: ‘n_neighbors’: 10,’p’: 1]
Decision Tree	0.901786	0.833334	[‘criterion’: ‘gin’,: max_depth’, ‘sgrt’, min_samples_leaf’,: 1, min_samples_split’: 2, “splitter’: ‘best’)

Confusion Matrix

- Out here for all models unfortunates, we have same confusion matrix.

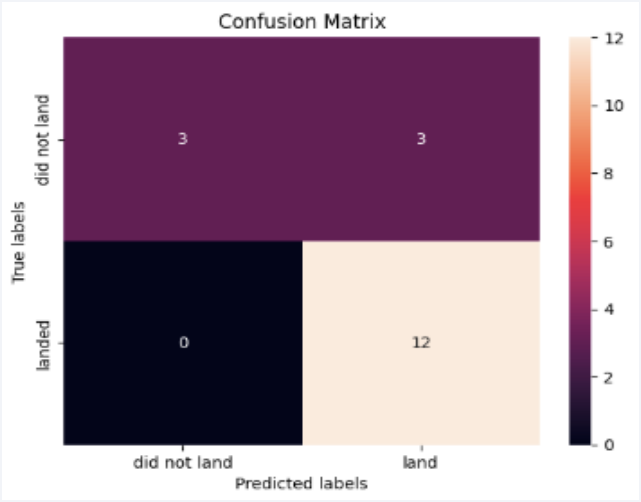
Predicted Values

Actual values

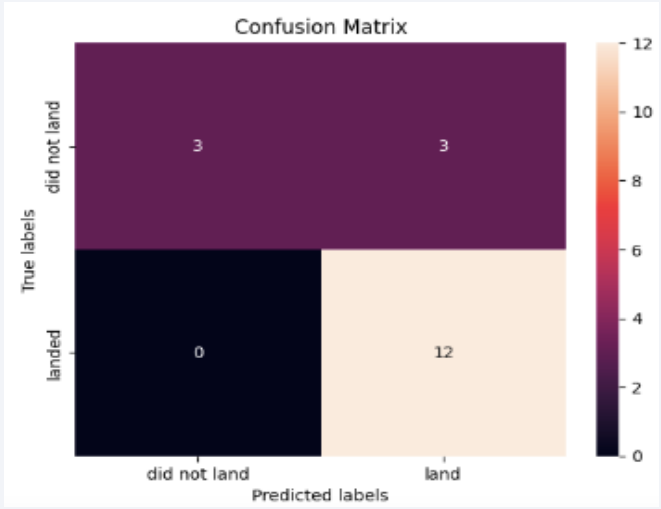
	Predicted No	Yes	
Actual: No	True Negative = 3	False Positive = 3	6
Yes	False Negative = 0	True Positive = 12	12
	3	15	Total cases 18

- Accuracy: (TP+TN/TOTAL=12+3/18 =0.83333)
- Misclassification rate: (FP+FN/TOTAL=(3+0)/18=0.1667
- True positive rate: TP/Actual Yes=12/12=1
- False positive rate: FP/Actual No=3/6= 0.5
- True Negative rate: TN/Actual No=3/6=0.5
- Precision: Actual/Total= 12/15=0.8
- Prevalence: Actual Yes/Total= 12/18=0.6667

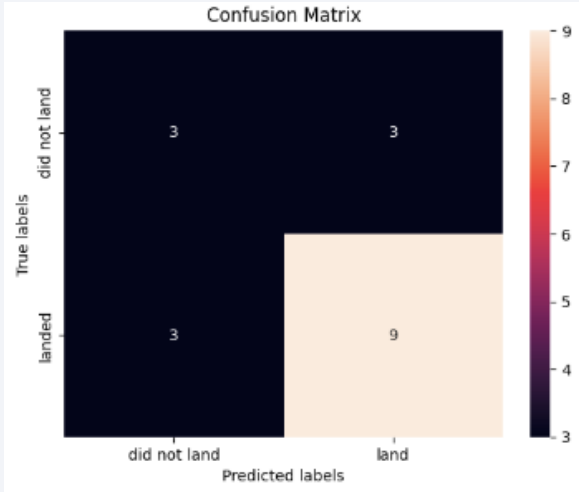
Logistic regression



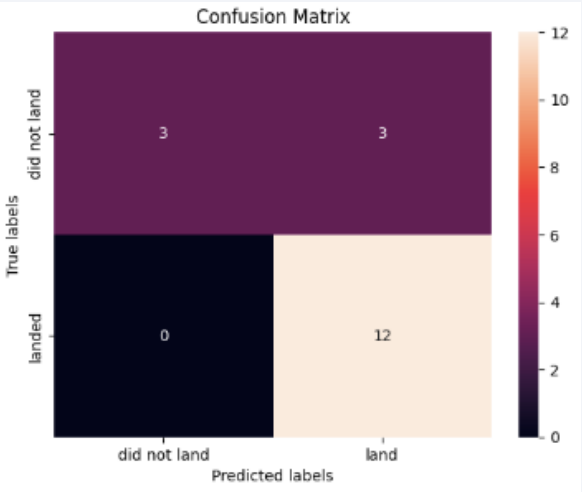
SVM



Decision tree

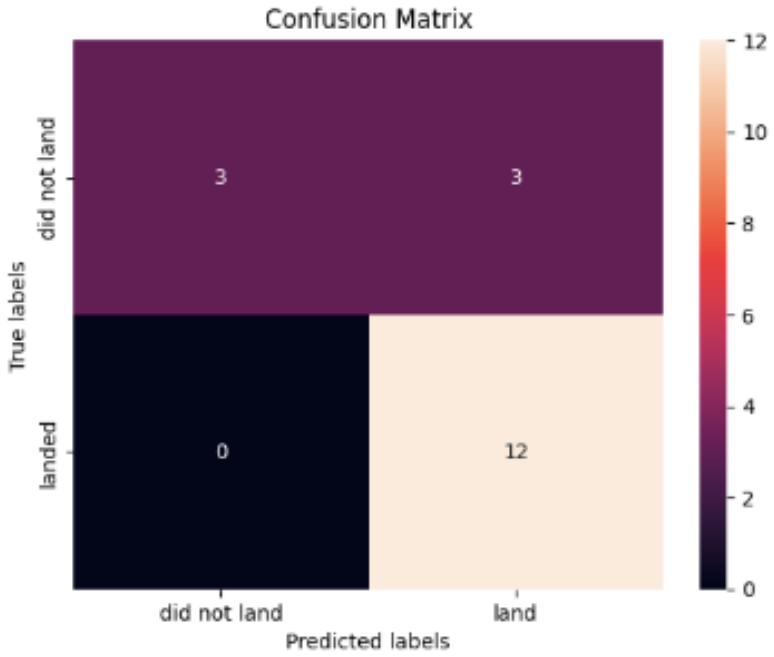


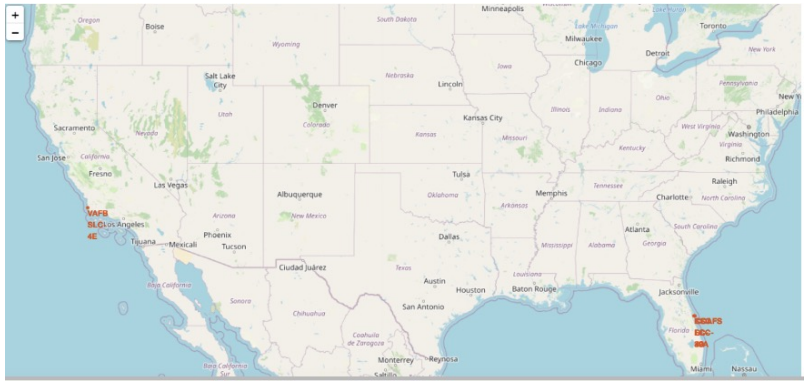
KNN

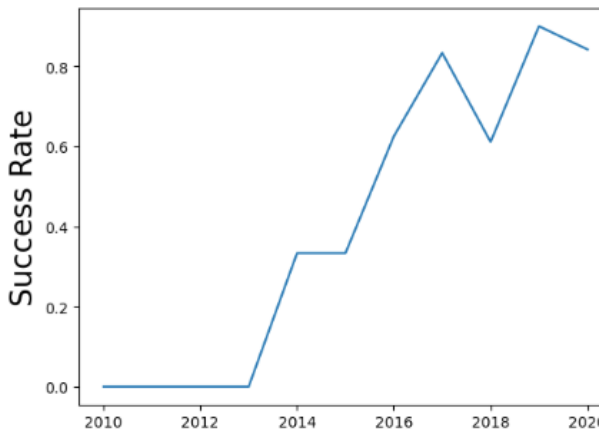


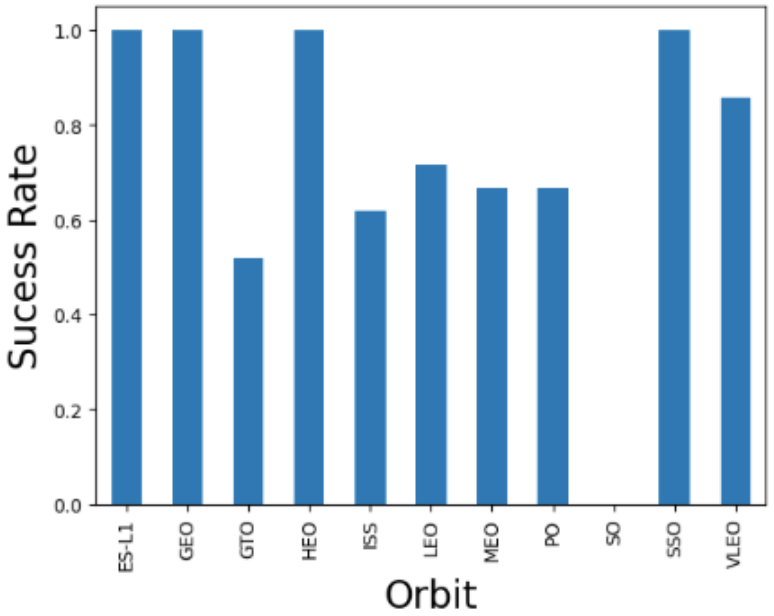
Conclusions

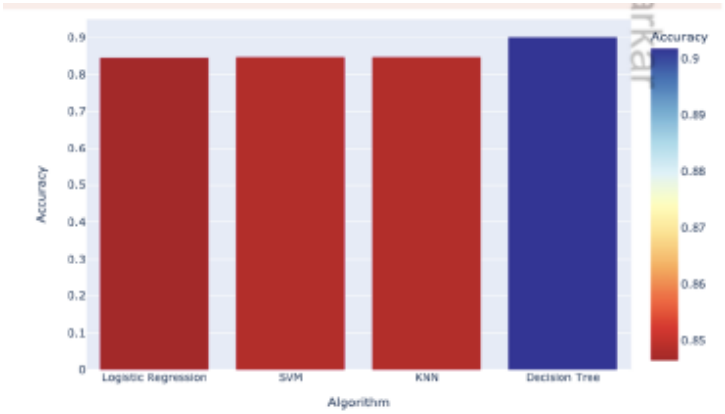
- Point 1: Orbits ES-L1, GEO, HEO, SSO has highest Success rates
- Point 2: Success rates for SpaceX launches has been increasing relatively with time and
- it looks like soon they will reach the required target
- Point 3: KSL LC-39A had the most successful launches but increasing payload mass
- seems to have negative impact on success
- Point 4: decision tree classifiers Algorism is the best for machine Learning model for
- provided data set.


Confusion Matrix
True labels
did not land
landed
3
3
0
12
did not land
land
Predicted labels
12
10
8
6
4
2
0




Success Rate
0.0
0.2
0.4
0.6
0.8
2010
2012
2014
2016
2018


Sucess Rate
0.0
0.2
0.4
0.6
0.8
1.0
ES-L1
GEO
GTO
HEO
ISS
LEO
MEO
PO
SO
SSO
VLEO
Orbit


Accuracy
Logistic Regression
SVM
KNN
Decision Tree
Algorithm

Appendix

Thank you!