IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

IMADUDDIN
12 January 2023

Outline

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

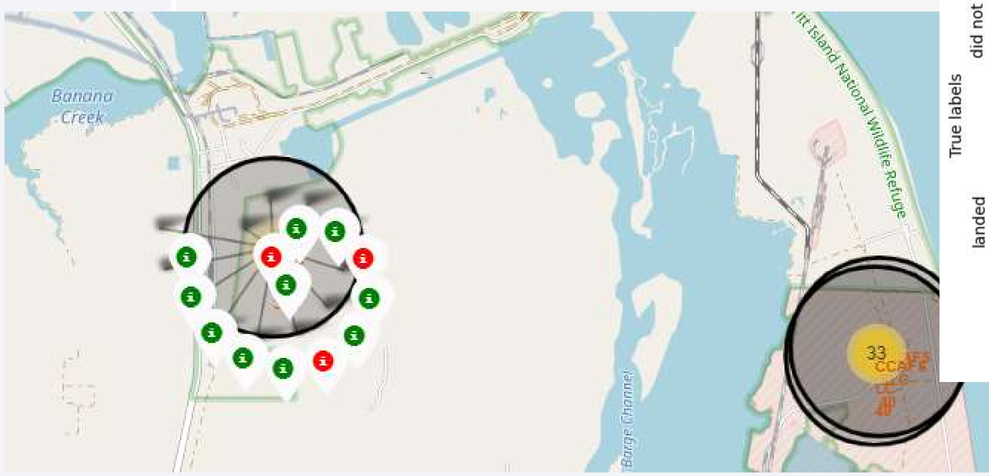
Executive Summary

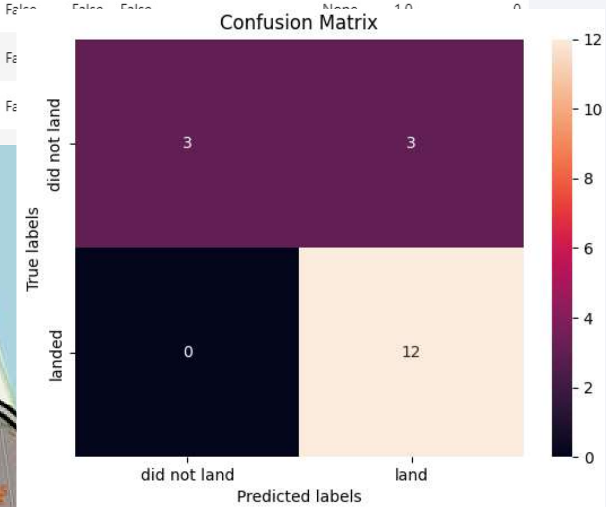
- **Methods :**
  - Data Collection using SpaceX API, Web Scrapping & Wrangling
  - Exploratory Data Analysis (EDA) : SQL, Matplotlib, Pandas
  - Visual & Dashboard (Folium, Analytics, Plotly Dash)
  - Machine Learning Prediction (Logistic, SVM, KNN, Tree)
- **Results :**
  - Data Extract & Analysis with Interactive Visualization
  - Best Model for Predictive Analysis

```
%sql select * from spacexdataset where launch_site like 'CCA%' limit 5;
```

```
* ibm_db_sa://pfw84467:***@815fa4db-dc03-4c70-869a-a9cc13f33084.bs2io90l08kqb1od8lcg.databases.appdomain.cloud:30367/BLUDB
  sqlite:///my_data1.db
```

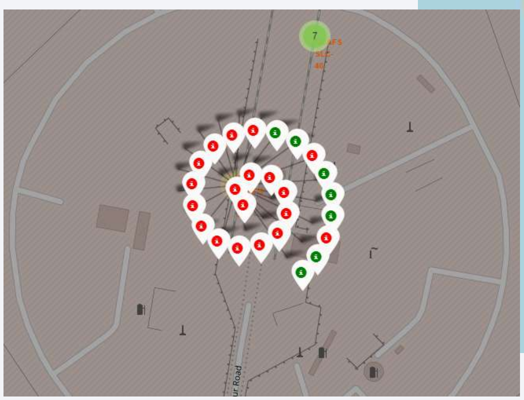
	FlightNumber	Date	BoosterVersion	PayloadMass	Orbit	LaunchSite	Outcome	Flights	GridFins	Reused	Legs	LandingPad	Block	ReusedCount
4	1	2010-06-04	Falcon 9	NaN	LEO	CCSFS SLC 40	None None	1	False	False	False	None	1.0	0
5	2	2012-05-22	Falcon 9	525.0	LEO	CCSFS SLC 40	None None	1	False	False	False	None	1.0	0
6	3	2013-03-01	Falcon 9	677.0	ISS	CCSFS SLC 40	None None	1	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
7	4	2013-09-29	Falcon 9	500.0	PO	VAFB SLC 4E	False Ocean	1	[illegible]					
8	5	2013-12-03	Falcon 9	3170.0	GTO	CCSFS SLC 40	None None	1	[illegible]					





Introduction

- Project background and context :
  - Objective : “evaluate the eligibility & viability of the new company ‘Space Y’ to compete ‘Space X’ by optimizing total cost for launches”
- Problems you want to find answers :
  - Best model to predict successful landing of 1st stage rocket with all determining factors (Launch Site, Orbit, etc..)



Section 1

Methodology

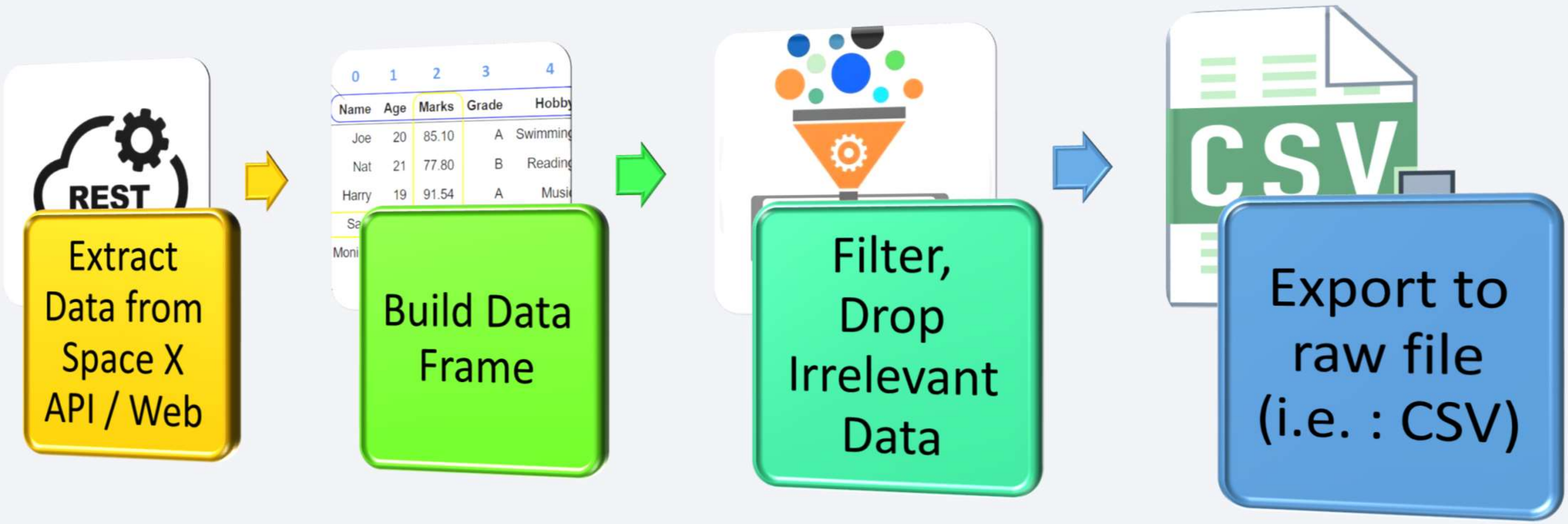
Methodology

Executive Summary

- Data collection methodology:
  - Source : Space X Rest API (https://api.spacexdata.com/v4/rockets/)
  - Web Scraping : https://en.wikipedia.org/wiki/List_of_Falcon/_9/_and_Falcon_Heavy_launches
- Perform data wrangling
  - Dropping irrelevant data
  - By evaluating feature data, create a landing outcome label based on the outcome data
- Perform exploratory data analysis (EDA) using visualization and SQL
- Perform interactive visual analytics using Folium and Plotly Dash
- Perform predictive analysis using classification models
  - Normalize data, divide train-test data set, evaluated by 4 models and judge by its best accuracy score

Data Collection

- Gather data from established system (API or Web), to extract relevant data to be processed further



Data Collection – SpaceX API

SpaceX provide public API, data can be extracted and processed further

1.a

```
spacex_url="https://api.spacexdata.com/v4/launches/past"
response = requests.get(spacex_url)
```

```
json_list = requests.get(static_json_url).json()
df_json_norm = pd.json_normalize(json_list)
```

1.b

2.a

getBoosterVersion(data)
getLaunchSite(data)
getPayloadData(data)
getCoreData(data)

2.c

```
falcon9 = pd.DataFrame(launch_dict)
```

2.b

```
launch_dict = {'FlightNumber': list(data['flight_number'])
'Date': list(data['date']),
'BoosterVersion':BoosterVersion,
'PayloadMass':PayloadMass,
'Orbit':Orbit,
'LaunchSite':LaunchSite,
'Outcome':Outcome,
'Flights':Flights,
'GridFins':GridFins,
'Reused':Reused,
'Legs':Legs,
'LandingPad':LandingPad,
'Block':Block,
'ReusedCount':ReusedCount,
'Serial':Serial,
'Longitude': Longitude,
'Latitude': Latitude}
```

3.a

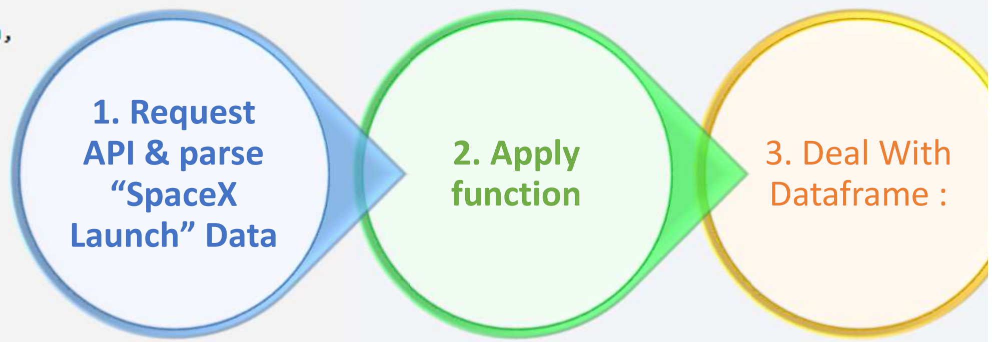
```
falcon9.drop(falcon9[falcon9['BoosterVersion']!='Falcon 9'].index, inplace = True)
falcon9.loc[:,'FlightNumber'] = list(range(1, falcon9.shape[0]+1))
```

3.b

```
# Calculate the mean value of PayloadMass column
AVG_PayloadMass = falcon9["PayloadMass"].astype("float").mean(axis=0)
# Replace the np.nan values with its mean value
falcon9["PayloadMass"].replace(np.nan, AVG_PayloadMass, inplace=True)
falcon9.isnull().sum()
```

3.c

```
falcon9.to_csv('dataset_part_1.csv', index=False)
```



a.Request, Get Response
b.Convert Response to .json

a.Get core data
b.Assign List to Dictionary
c.Create Dataframe

a.Filter only Falcon 9
b.Deal Missing Value
c.Export to File (ex: CSV,

	FlightNumber	Date	BoosterVersion	PayloadMass	Orbit	LaunchSite
4	1	2010-06-04	Falcon 9	NaN	LEO	CCSFS SLC 40
5	2	2012-05-22	Falcon 9	525.0	LEO	CCSFS SLC 40
6	3	2013-03-01	Falcon 9	677.0	ISS	CCSFS SLC 40
7	4	2013-09-29	Falcon 9	500.0	PO	VAFB SLC 4E
8	5	2013-12-03	Falcon 9	3170.0	GTO	CCSFS SLC 40
...	...	...	...	...	...	...
89	86	2020-09-03	Falcon 9	15600.0	VLEO	KSC LC 39A
90	87	2020-10-06	Falcon 9	15600.0	VLEO	KSC LC 39A
91	88	2020-10-18	Falcon 9	15600.0	VLEO	KSC LC 39A
[illegible]	[illegible]	2020-	[illegible]	[illegible]	[illegible]	CCSFS SLC

Source : https://github.com/iim05te/SpaceX_Imaduddin/blob/main/IM_W1_01_Collect_Data.ipynb

Data Collection - Scraping

Data obtained by scraping from [Wikipedia](#)

1.a

```
spacex_url="https://api.spacexdata.com/v4/launches/past"
response = requests.get(static_url)
```

```
soup = BeautifulSoup(response.text, 'html')
```

1.b

2.a

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

2.b

```
column_names = []
tc = first_launch_table.find_all('th')
for th in tc:
    name = extract_column_from_header(th)
    if name is not None and len(name) > 0:
        column_names.append(name)
```

```
print(column_names)
```

2.c

```
launch_dict= dict.fromkeys(column_names)

# Remove an irrelvant column
del launch_dict['Date and time ( )']

# Let's initial the launch_dict with each value to be an empty list
launch_dict['Flight No.'] = []
launch_dict['Launch site'] = []
launch_dict['Payload'] = []
launch_dict['Payload mass'] = []
launch_dict['Orbit'] = []
launch_dict['Customer'] = []
launch_dict['Launch outcome'] = []
# Added some new columns
launch_dict['Version Booster']=[]
launch_dict['Booster landing']=[]
launch_dict['Date']=[]
launch_dict['Time']=[]
```

3.a

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

	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
...	...	...	...	...	...	...	...	...	...	...	...
116	117	CCSFS	Starlink	15,600 kg	LEO	SpaceX	Success\n	F9 B5B1051.10	Success	9 May 2021	06:42
117	118	KSC	Starlink	~14,000 kg	LEO	SpaceX	Success\n	F9 B5B1058.8	Success	15 May 2021	22:56
118	119	CCSFS	Starlink	15,600 kg	LEO	SpaceX	Success\n	F9 B5B1063.2	Success	26 May 2021	18:59
			SpaceX CRS-22	3,328 kg	LEO	NASA	Success\n	F9 B5B1067.1	Success	3 June 2021	17:29
			SXM-8	7,000 kg	GTO	Sirius XM	Success\n	F9 B5	Success	6 June 2021	04:26

3.b

```
df.to_csv('spacex_web_scraped_tpf.csv', index=False)
```

1. Request Wiki & parse "SpaceX Launch" Data

a. Request, Get Response HTML
b. Create BeautifulSoup Object

2. Extract column/variable from HTML Table Header

a. Find table
b. Get Column name
c. Create Dictionary, append data to keys

3. Deal Dataframe :

a. Convert dictionary to data frame
b. Convert to CSV

Source : https://github.com/iim05te/SpaceX_Imaduddin/blob/main/IM_W1_01_Collect_WebScrap.ipynb

Data Wrangling

Data cleaned, unify messy and complex data set to ease access and analyze. Convert "Outcome column" into Training Labels for booster landing (1 : success, 0 : fail)

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

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

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

```
GTO       27
ISS       21
VLEO      14
PO         9
LEO        7
SSO        5
MEO        3
ES-L1      1
```

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

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

A. Calculate Number of Launces each Site

B. Calculate number and occurrence on each orbit

C. Calculate number and occurrence of mission outcome per orbit

D. Create Landing Outcome Label & Export to raw file (i.e. : CSV)

Outcome	Flights	GridFins	Reused	Legs	LandingPad	Block	ReusedCount	Serial	Longitude	Latitude	Class
None None	1	False	False	False	NaN	1.0	0	B0003	-80.577366	28.561857	0
None None	1	False	False	False	NaN	1.0	0	B0005	-80.577366	28.561857	0
None None	1	False	False	False	NaN	1.0	0	B0007	-80.577366	28.561857	0
False Ocean	1	False	False	False	NaN	1.0	0	B1003	-120.610829	34.632093	0
None None	1	False	False	False	NaN	1.0	0	B1004	-80.577366	28.561857	0

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

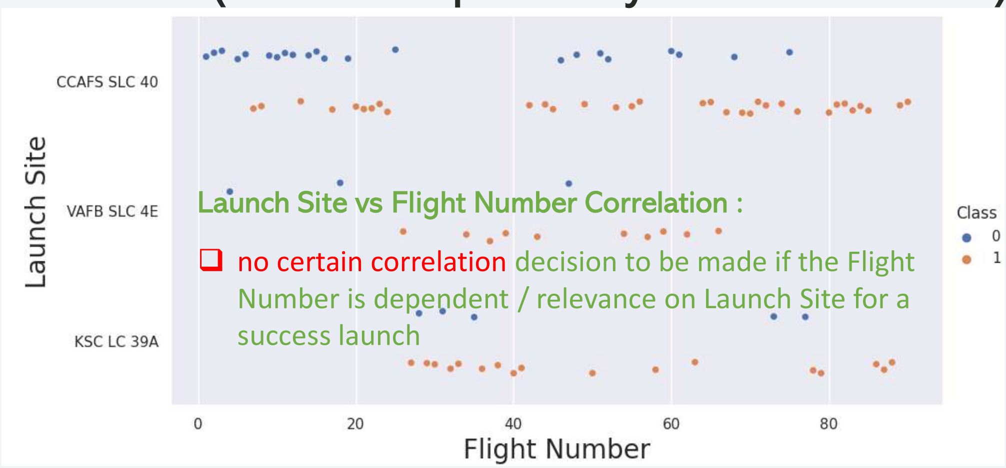
Source : https://github.com/iim05te/SpaceX_Imaduddin/blob/main/IM_W1_02_Wrangling.ipynb

EDA with Data Visualization (1/2)

Using Scatter, Bar Chart, Line Chart for EDA

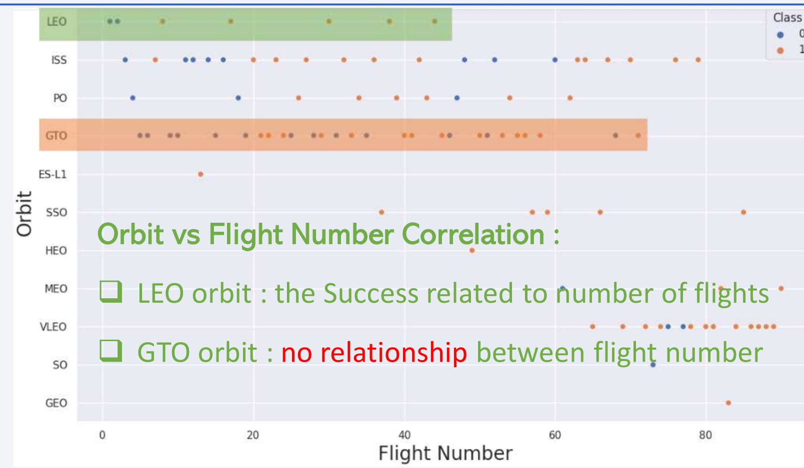
Source : https://github.com/iim05te/SpaceX_Imaduddin/blob/main/IM_W2_01_Visual.ipynb

I. Scatter (to show dependency of each attribute)



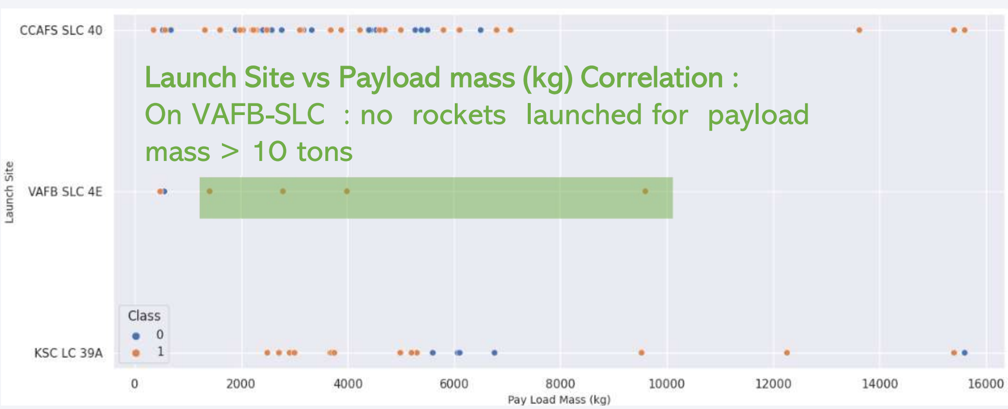
Launch Site vs Flight Number Correlation :

- ❑ no certain correlation decision to be made if the Flight Number is dependent / relevance on Launch Site for a success launch

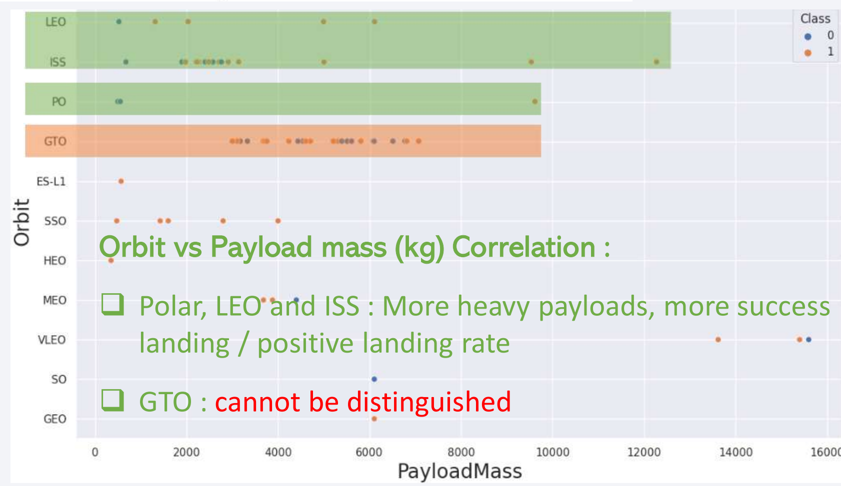


Orbit vs Flight Number Correlation :

- ❑ LEO orbit : the Success related to number of flights
- ❑ GTO orbit : no relationship between flight number



Launch Site vs Payload mass (kg) Correlation :
On VAFB-SLC : no rockets launched for payload mass > 10 tons



Orbit vs Payload mass (kg) Correlation :

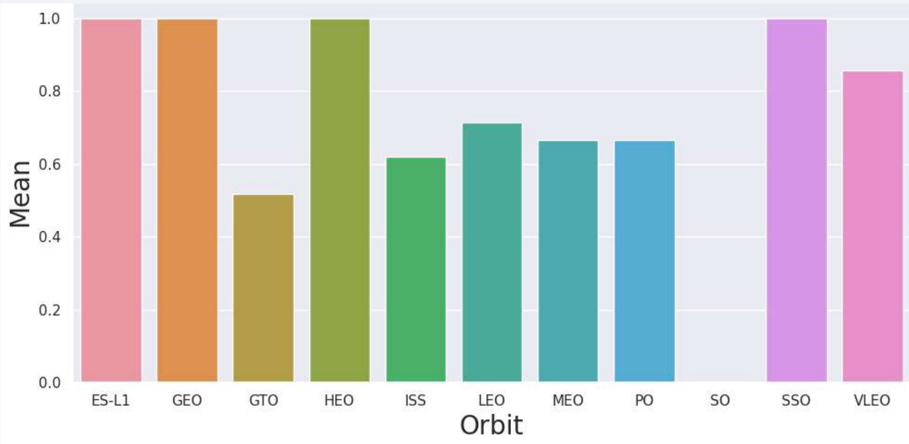
- ❑ Polar, LEO and ISS : More heavy payloads, more success landing / positive landing rate
- ❑ GTO : cannot be distinguished

EDA with Data Visualization (2/2)

II. Bar Chart (to show which Orbit with most successful landing)

Source : https://github.com/iim05te/SpaceX_Imaduddin/blob/main/IM_W2_01_Visual.ipynb

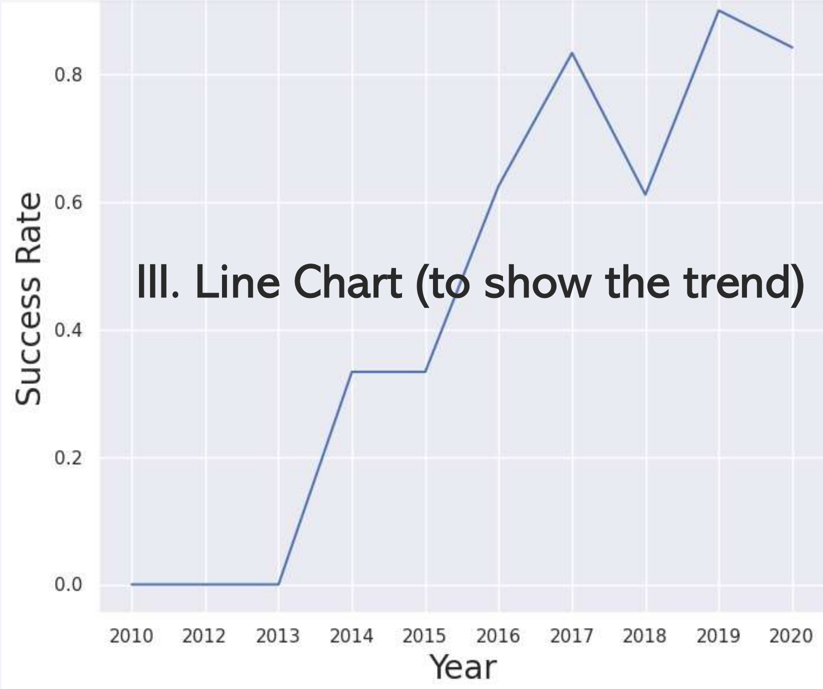
```
Orbit_Mean=df.groupby("Orbit",as_index=False)["Class"].mean()
```



Orbit with Mean of Success Rate :

ES-L1,GE,HEO, SSO have highest mean success rate

III. Line Chart (to show the trend)




```
year=[]
def Extract_year(date):
    for i in df["Date"]:
        year.append(i.split("-")[0])
    return year
df['year']=Extract_year(df["Date"])
df_groupby_year=df.groupby("year",as_index=False)["Class"].mean()
```

Launch Success Yearly Trend :

Success rate keep increasing from 2013 to 2020 (generally)

EDA with SQL

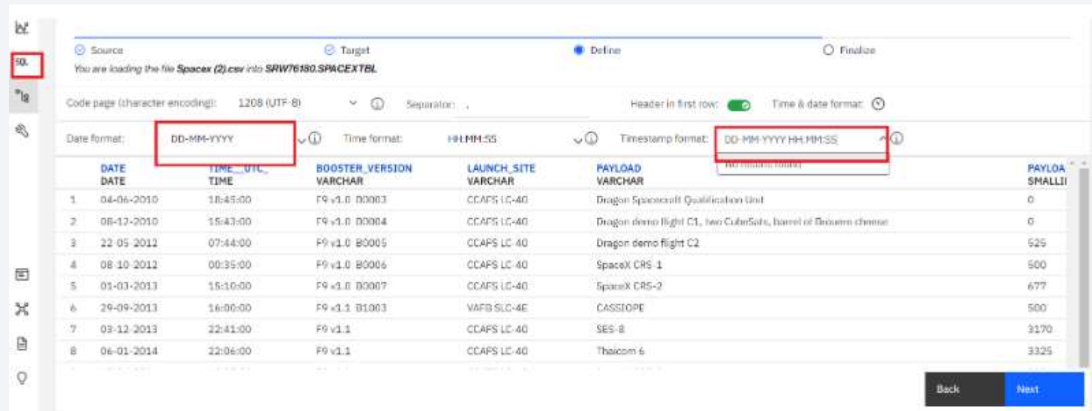
SQL queries using Db2 Cloud

Source : https://github.com/iim05te/SpaceX_Imaduddin/blob/main/IM_W2_01_SQL.ipynb

IBM Db2 Cloud

1. Names of launch sites;
2. Top 5 launch sites whose name begin with the string 'CCA';
3. Total payload mass carried by boosters launched by NASA (CRS);
4. Average payload mass carried by booster version F9 v1.1;
5. Date when the first successful landing outcome in ground pad was achieved;
6. Names of the boosters which have success in drone ship and have payload mass between 4000 and 6000 kg;
7. Total number of successful and failure mission outcomes;
8. Names of the booster versions which have carried the maximum payload mass;
9. Failed landing outcomes in drone ship, their booster versions, and launch site names for in year 2015;
10. Rank of the count of landing outcomes (such as Failure (drone ship) or Success (ground pad)) between the date 2010-06-04 and 2017-03-20

```
* ibm_db_sa:/            ]815fa4db-dc03-4c70-869a-a9cc13f33084.bs2io90l08kqb1od8lcg.
  sqlite:///my_data1.db
```




```
%sql select * from spacexdataset where launch_site like 'CCA%' limit 5;
```

Python

```
* ibm_db_sa:/            :03-4c70-869a-a9cc13f33084.bs2io90l08kqb1od8lcg.databases.appdomain.cloud:30367/BLUDB
  sqlite:///my_data1.db
Done.
```

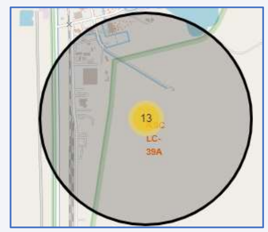
DATE	time_utc_	booster_version	launch_site	payload	payload_mass_kg_	orbit	customer	mission_outcome	landing_outcome
2010-06-04	18:45:00	F9 v1.0 B0003	CCAFS LC-40	Dragon Spacecraft Qualification Unit	0	LEO	SpaceX	Success	Failure (parachute)
2010-12-08	15:43:00	F9 v1.0 B0004	CCAFS LC-40	Dragon demo flight 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

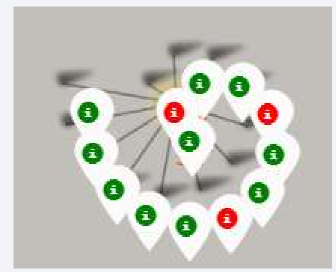
Build an Interactive Map with Folium

1. Markers indicate points like launch sites;
2. Circles indicate highlighted areas around specific coordinates, (i.e. : NASA)
3. Marker clusters object to simplify a map containing multiple markers in the same coordinates; indicates groups of events in each coordinate (i.e. : success / failure launches in a launch site)
4. Lines are used to indicate distances between two coordinates (to anticipate railway, highway, city and coastline risk mitigation impact on failure landing)

1
```
site_map.add_child(marker)
```

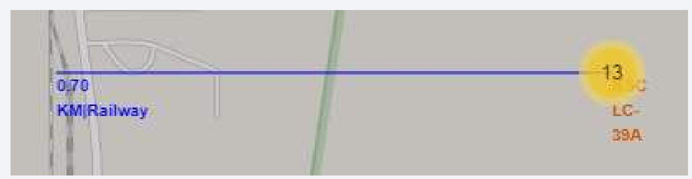
2
```
site_map.add_child(circle)
```





3
```
marker_cluster = MarkerCluster()
```

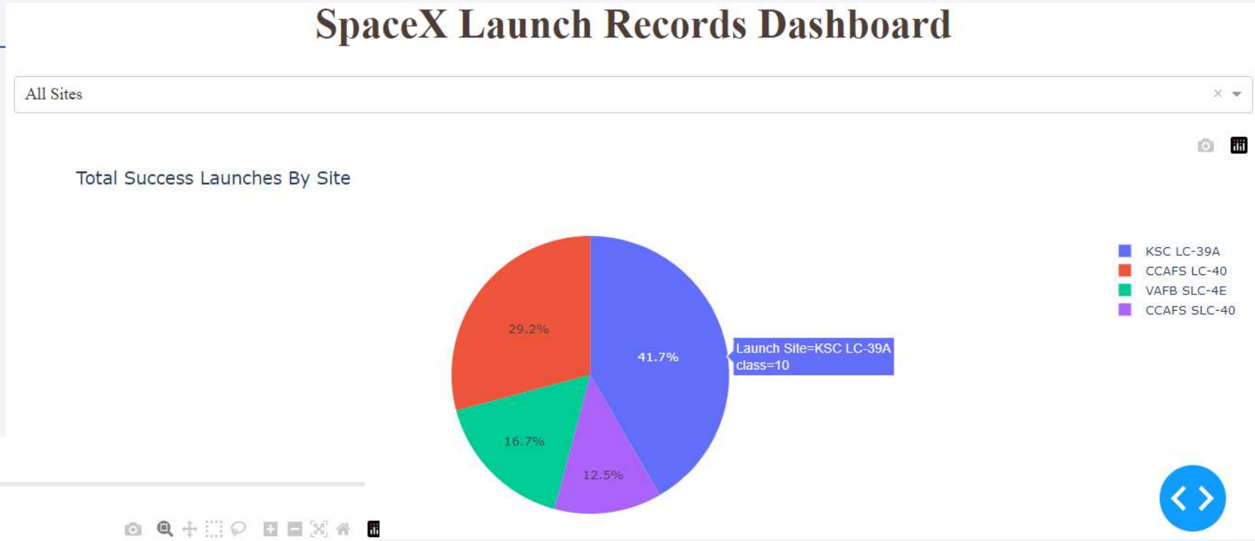
4
```
site_map.add_child(lineKSC_Rail)
```

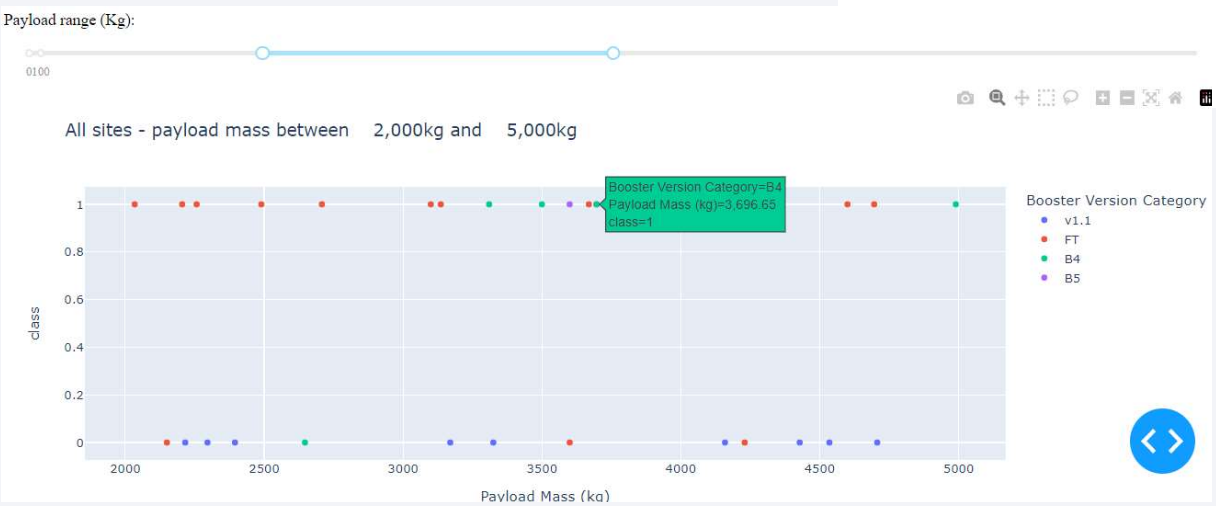


Source : https://github.com/iim05te/SpaceX_Imaduddin/blob/main/IM_W3_01_Folium.ipynb

Build a Dashboard with Plotly Dash

- **Pie Chart** to summarize total success launch for all launch site & dropdown on certain site in percentage





- **Scatter Plot** to show correlation between success rate vs booster version category on sites by sliding Payload mass (kg) on all launch sites to help decide best launch site according to payload mass

Source : https://github.com/iim05te/SpaceX_Imaduddin/blob/main/IM_W3_plotly/IM_W3_plotly.py

Predictive Analysis (Classification)

4 Method used (LR / Logistic Regression, SVM / Support Vector Machine, KNN / K - Nearest Neighbors, DT / Decision Tree), compare its **Accuracy Score**, then choose the best score.

1. Modelling

- Load to Dataframe
- Transform into NumPy Array
- Standardize & Transform
- Split (Train-Test)
- Pick a Method (1 by 1)
- Set Parameter & Algorithm on GridSearchCV
- Fit data into GridSearchCV object & train model

2. Evaluate on each Model

- Check accuracy Score
- Get best hyperparameter
- Plot Confusion Matrix

3. Choose Best Score

- Compare each Accuracy Score
- Choose the best Accuracy Score

1

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

2

```
print("tuned hpyerparameters :(best parameters) ",logreg_cv.best_params_)
print("accuracy :",logreg_cv.best_score_)
print('Accuracy Score on Log.Reg:', logreg_cv.score(X_test, Y_test))
yhat=logreg_cv.predict(X_test)
plot_confusion_matrix(Y_test,yhat)
```

3

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

	Accuracy Score
KNN	0.848214
Decision Tree	0.889286
Logistic Regression	0.846429
SVM	0.848214

Source : https://github.com/iim05te/SpaceX_Imaduddin/blob/main/IM_W4_01_ML.ipynb

Results (Recap)

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

Results (1/3 : EDA)

- Visualization :
  - Scatter Plot Dependency Attribute :
    a. Payload vs Launch Site : no rockets launched from VAFB-SLC for payload mass > 10 tons
    b. Orbit vs Number of Flight : the Success on LEO orbit related to number of flights
    c. Orbit vs Payload : in PO, LEO and ISS : more heavy payloads, more success landing / positive landing rate
  - Bar Chart Categorical Orbit : ES-L1,GE,HEO, SSO have highest mean success rate
  - Plot Line Trend : The number of landing outcomes became as better as years passed since 2013

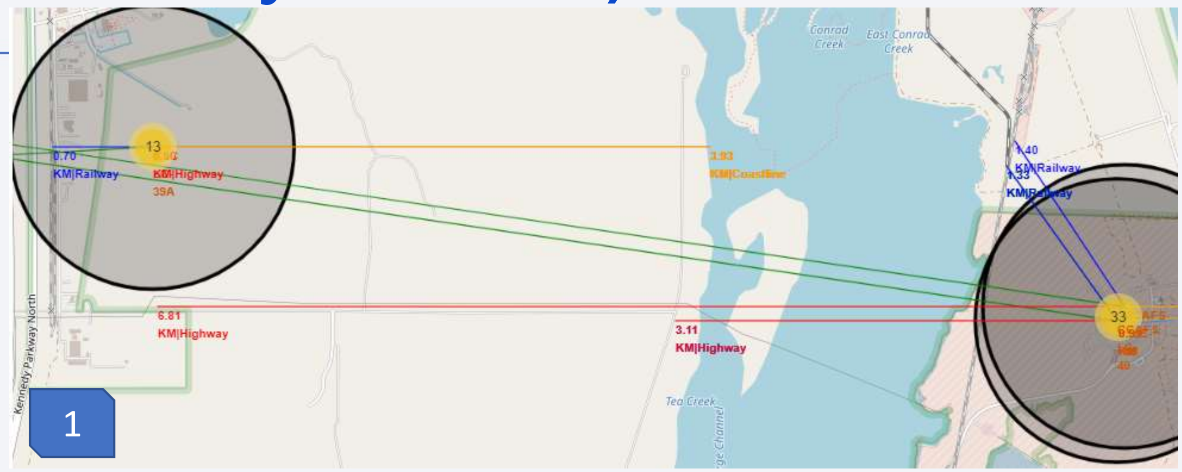
- SQL :

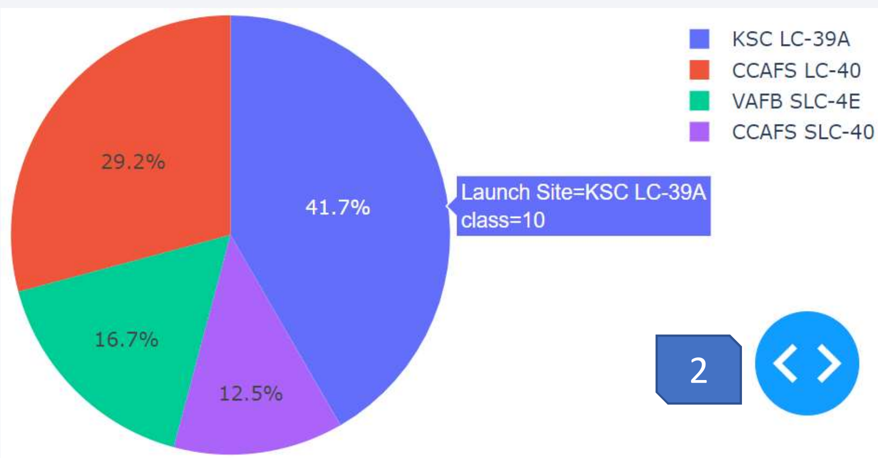
1. Space X uses 4 different launch sites
2. The first launches were done to Space X itself and NASA
3. Average payload mass of F9 v1.1 booster is 2,928 kg
4. 1st success landing in 2015 five year after the first launch
5. Many Falcon 9 booster versions were successful at landing in drone ships having payload above the average
6. Almost 100% of mission outcomes were successful
7. Two booster versions failed at landing in drone ships in 2015: F9 v1.1 B1012 and F9 v1.1 B1015

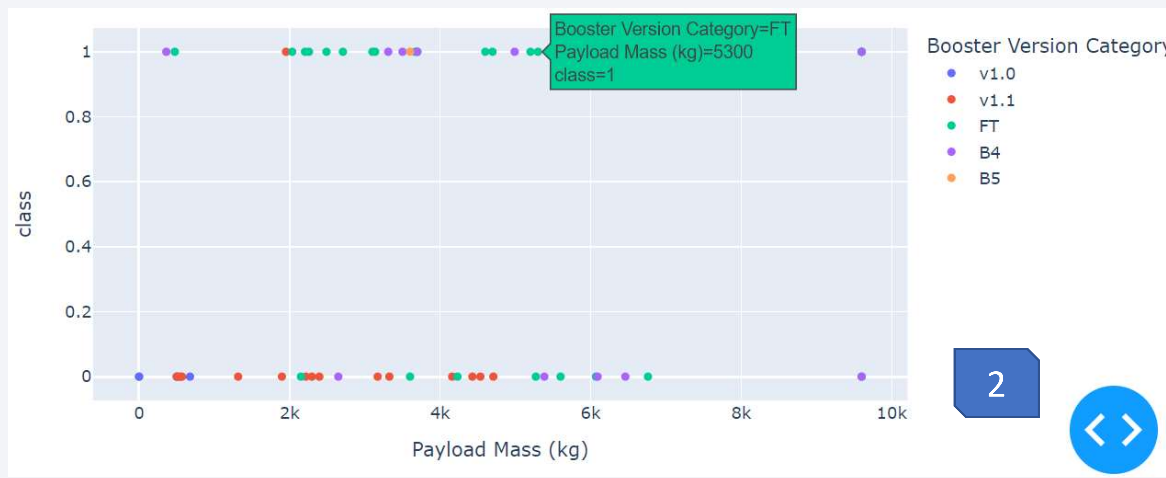
Results (2/3 : Interactive Analytic Demo)

1. Folium : showing Best & Safety Place to launch (in distance)
- near coastline and far from city to minimize the risk
- near logistic infrastructure around

2. Dash :
- 10 success launch in KSC LC-39A (41.7 %)
- Booster Version Category FT have more success launch







Results (3/3 : Predictive analysis results)

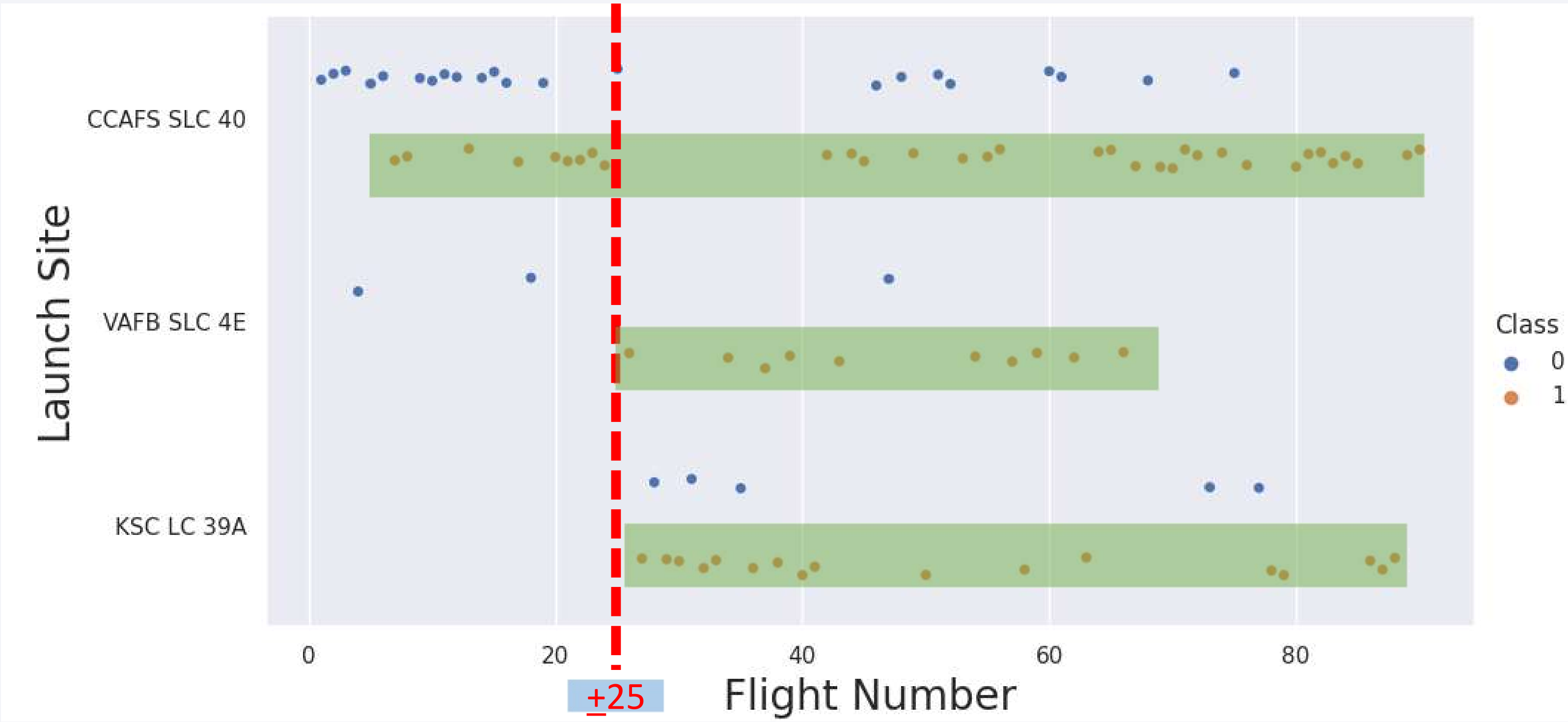
Predictive analysis showed that Decision Tree has the best Accuracy Score for a model to predict successful landings, Accuracy Score 88.93 %

	Accuracy Score
KNN	0.848214
Decision Tree	0.889286
Logistic Regression	0.846429
SVM	0.848214

Section 2

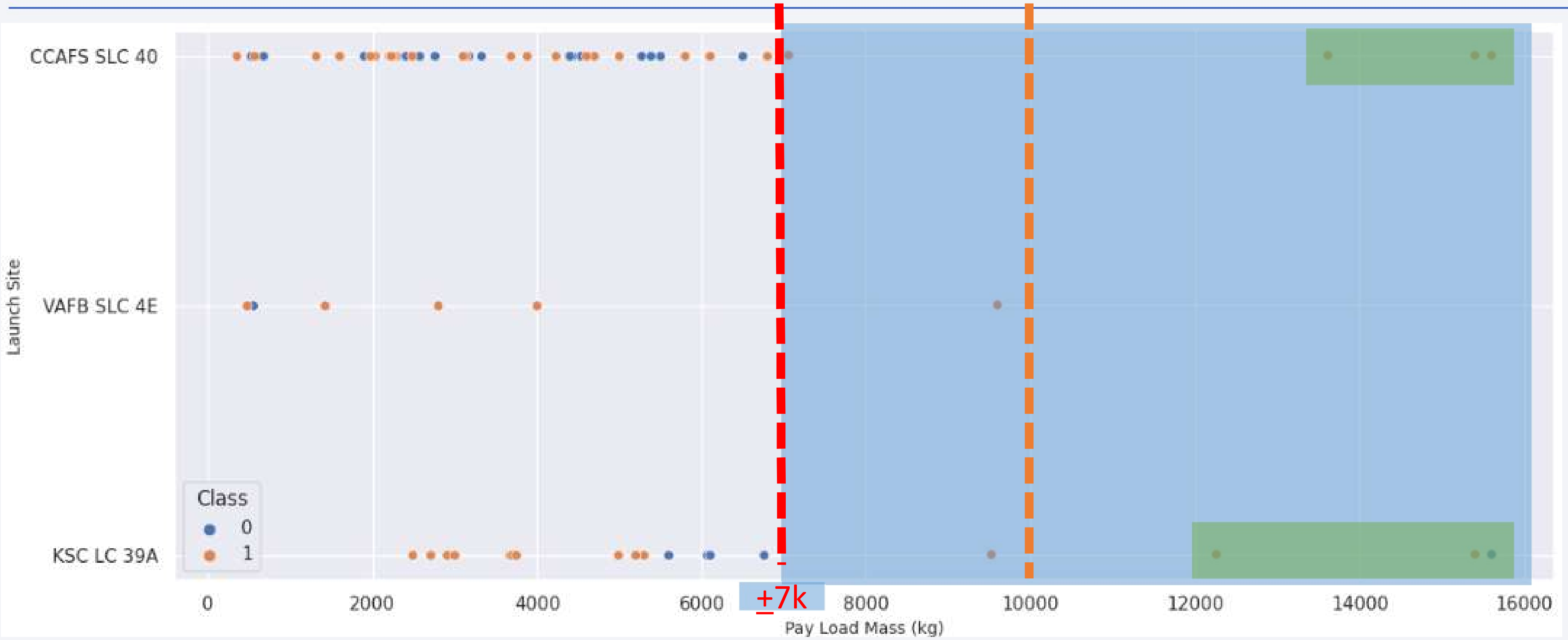
Insights drawn from EDA

Flight Number vs. Launch Site



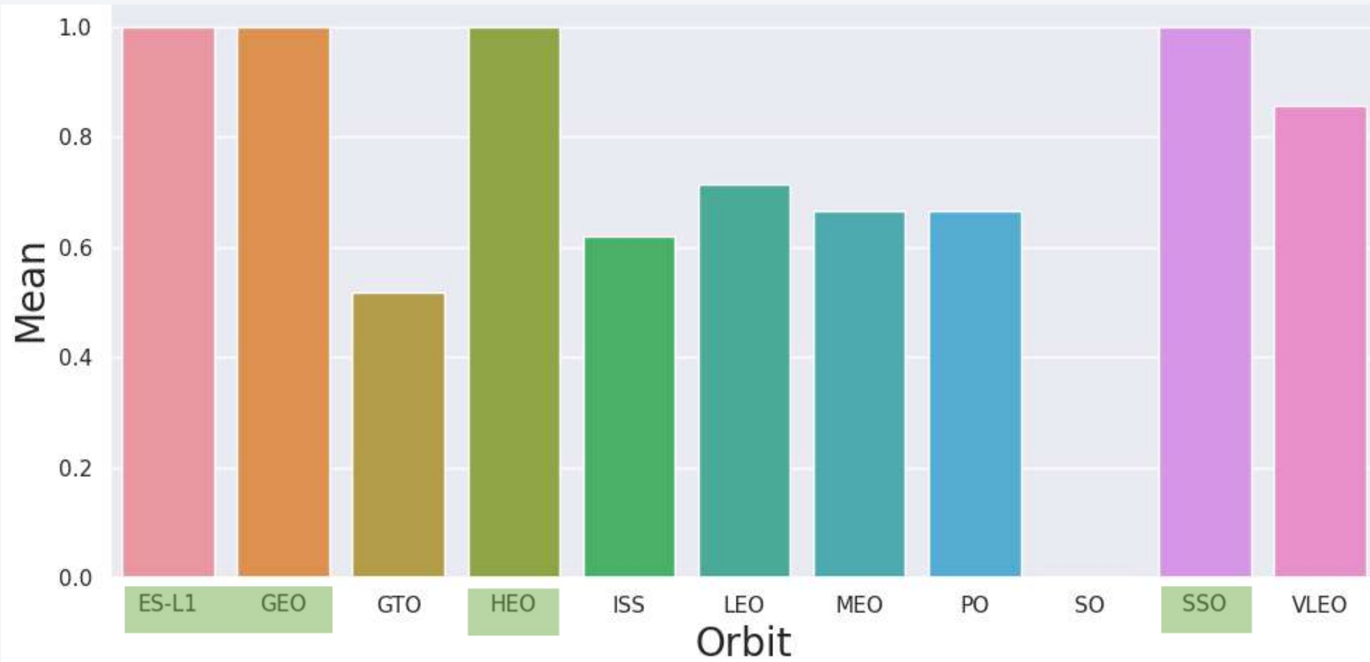
- ❑ most of launch on CCAFS SLC-40, but only 60% successful, KSC LC 39A & VAFB SLC 4E has 77 % success rate, need to be verified further
- ❑ After flight number +25, KSC LC 39A & VAFB SLC 4E have higher success rate
- ❑ But no certain correlation decision to be made if the Flight Number is dependent / relevance on Launch Site for a success launch

Payload vs. Launch Site



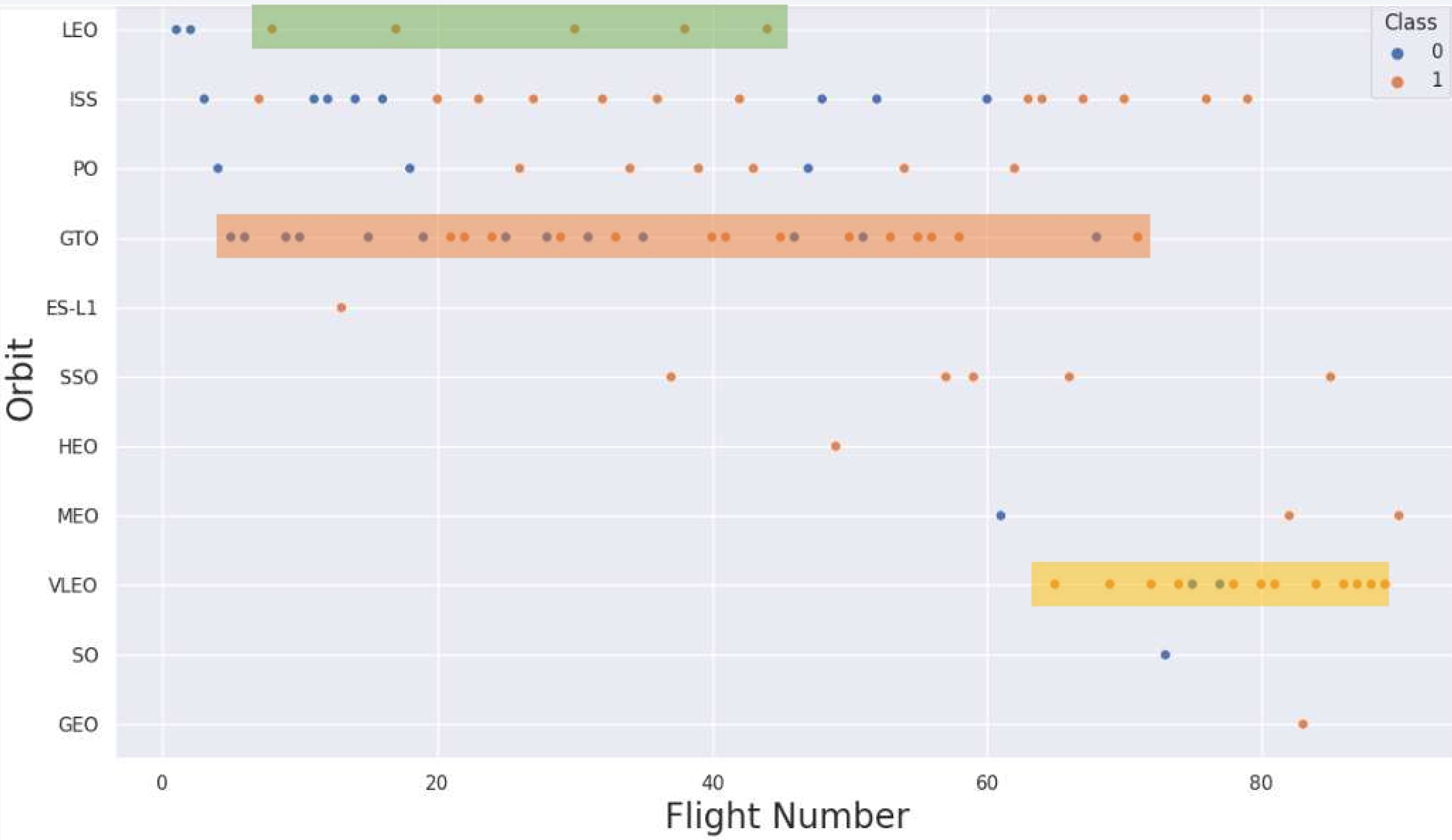
- ❑ no rockets launched from VAFB-SLC for payload mass > 10 tons
- ❑ Payloads mass > ± 7 tons have most success rate;
- ❑ Payloads mass > ± 10 tons seems only possible launched from CCAFS SLC 40 and KSC LC 39A
- ❑ no certain pattern if the launch site dependent on Pay Load Mass for a success launch

Success Rate vs. Orbit Type



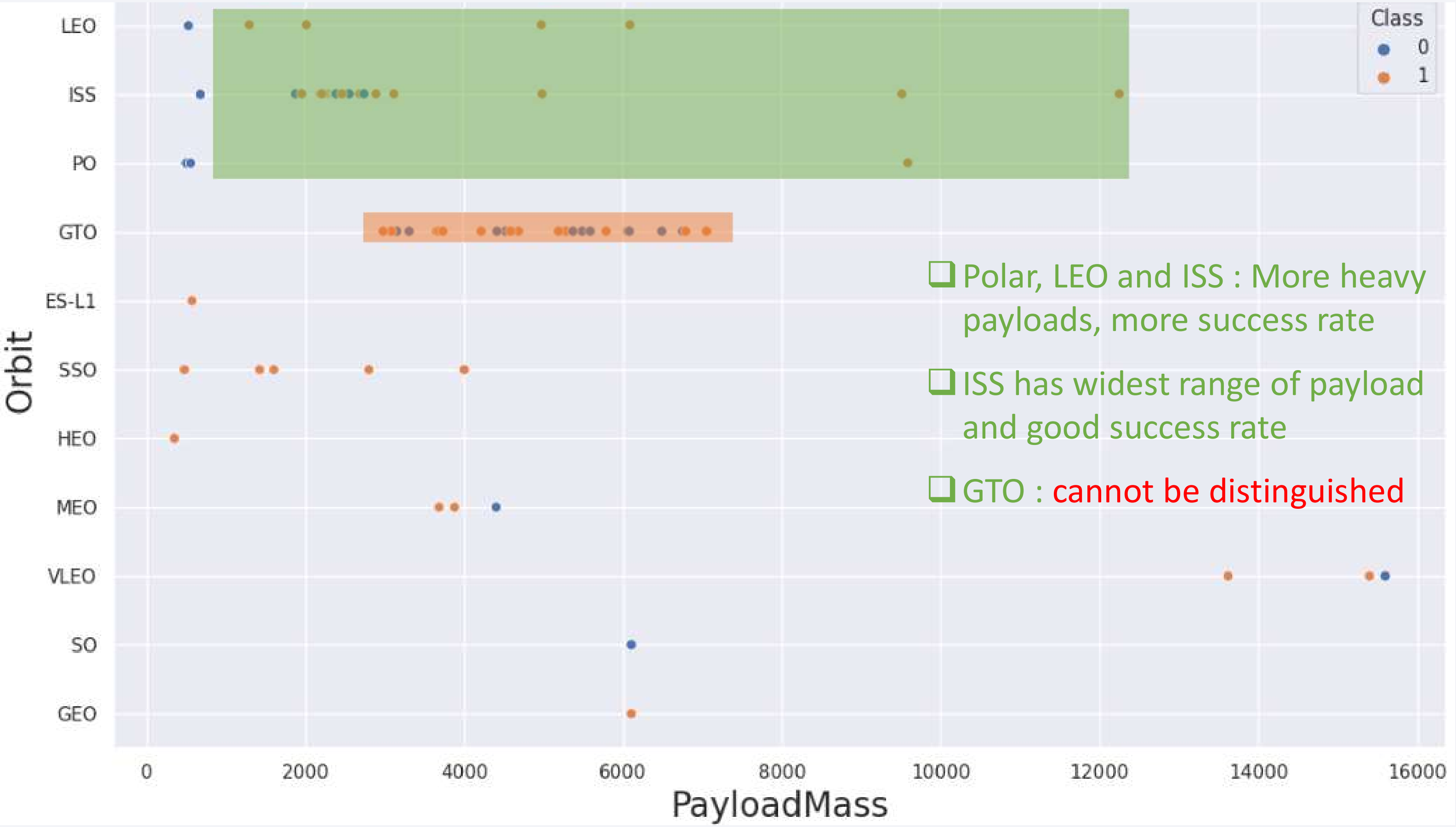
- ES-L1,GE,HEO, SSO have highest mean success rate

Flight Number vs. Orbit Type



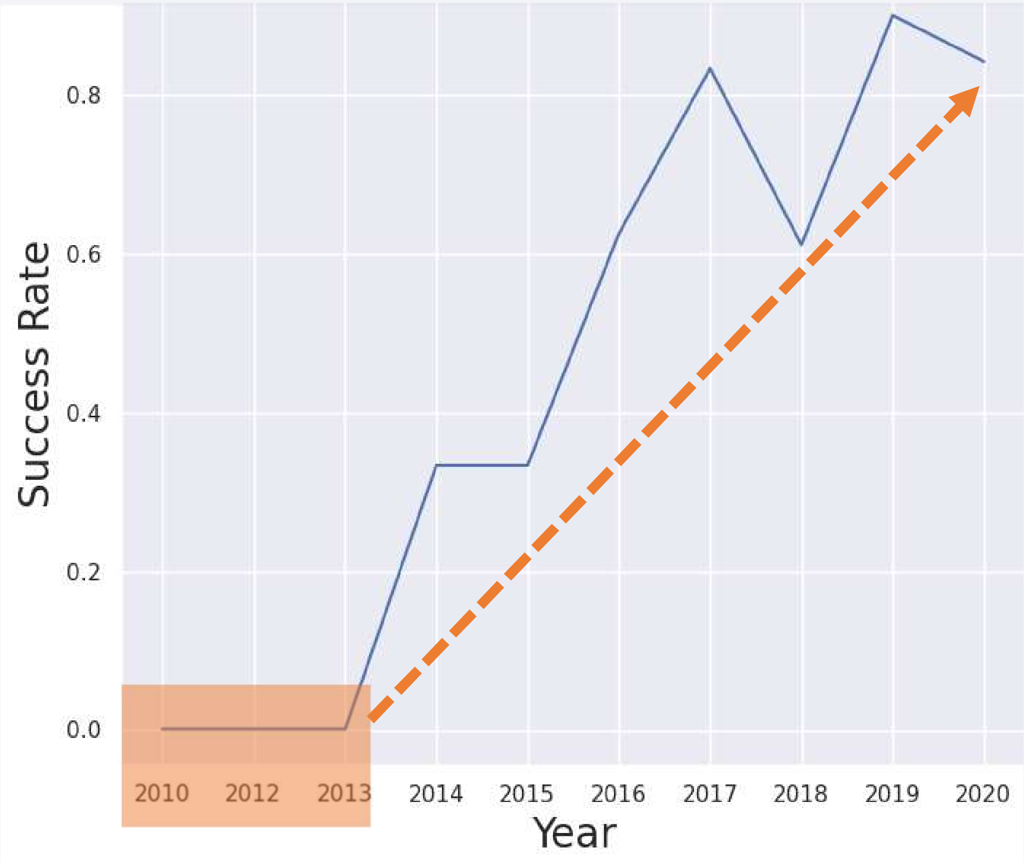
- Success rate improved over time on all orbit (flight number increases on each orbit)
- the Success on LEO orbit related to number of flights
- VLEO could be opportunity, since recent frequent flight
- no real pattern on GTO

Payload vs. Orbit Type



- Polar, LEO and ISS : More heavy payloads, more success rate
- ISS has widest range of payload and good success rate
- GTO : cannot be distinguished

Launch Success Yearly Trend



- ❑ Success rate keep increasing from 2013 to 2020 (generally)
- ❑ Seems 2010-2013 were a period time for adjustment & improvement analysis (budget and technology

All Launch Site Names

```
%sql select distinct launch_site from spacexdataset;
```

- %sql to use SQL extension on notebook
- select distinct : to pull unique values of launch_site column from table spacexdataset

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

Launch Site Names Begin with 'CCA'

```
%sql select * from spacexdataset where launch_site like 'CCA%' limit 5;
```

- %sql to use SQL extension on notebook
- select * : to pull all values of column from table spacexdataset
- where launch_site like 'CCA%' : keyword wild card only launch_site begin with CCA, % in the end means can be any word after
- limit 5 : only fetch 5 records

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

Total Payload Mass

```
%sql select sum(payload_mass__kg_) as "Total Payload Mass (kg) by NASA (CRS)" from spacexdataset where customer='NASA (CRS)';
```

- sum : to pull sum in kg of payload mass column from table spacexdataset, summing all values
- as “Total …” : alias to rename the new column as “Total …”
- where customer= : keyword wild card customer exactly same as ‘NASA (CRS)’

Total Payload Mass (kg) by NASA (CRS)
45596

Average Payload Mass by F9 v1.1

```
%sql select avg(payload_mass__kg_) as "Average Payload Mass (kg) by booster version F9 v1.1" from spacexdataset where booster_version like 'F9 v1.1%';
```

- avg : to pull averagein kg of payload mass column from table spacexdataset, calculating all values in col.
- as “Average…” : alias to rename the new column as “Average …”
- select * : to pull all values of column from table spacexdataset
- where booster_version like ‘F9 v1.1%’ : keyword wild card only booster_version begin with F9 v1.1, % in the end means can be any word after

Average Payload Mass (kg) by booster version F9 v1.1
2534

First Successful Ground Landing Date

```
%sql select min(DATE) as "1st Succesful Landing Outcome in Ground Pad" from spacexdataset where  landing__outcome='Success (ground pad)';
```

- min(DATE) : to pull minimum value on DATE column (MIN = the oldest date)
- as “1st …” : alias to rename the new column as “1st …”
- where landing_outcome= : keyword wild card customer exactly same as ‘Success (ground pad)’

1st Succesful Landing Outcome in Ground Pad
2015-12-22

Successful Drone Ship Landing with Payload between 4000 and 6000

```
%sql select booster_version from spacexdataset where landing__outcome='Success (drone ship)' and payload_mass__kg_ between 4000 and 6000;
```

- select booster_version : to pull only booster version column from table spacexdataset
- where landing_outcome= : keyword wild card landing_outcome exactly same as 'Success (drone ship)'
- and … between 4000 and 6000 : additional filter between 2 values

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

Total Number of Successful and Failure Mission Outcomes

```
%sql select mission_outcome,count(mission_outcome) as "Total Number Mission Outcomes" from spacexdataset group by mission_outcome;
```

- ❑ select mission_outcome, count.. : to pull 1 column mission outcome and count all kind of it
- ❑ as “Total …” : alias to rename the new column as “Total …”
- ❑ group by ..= : to group by mission outcome and giving each counting records, summary

mission_outcome	Total Number Mission Outcomes
Failure (in flight)	1
Success	99
Success (payload status unclear)	1

Boosters Carried Maximum Payload

```
%sql select distinct booster_version as "Booster Versions carried Max Payload
Mass (kg)" from spacexdataset where payload_mass__kg_=(select
max(payload_mass__kg_) from spacexdataset);
```

- select distinct : to pull unique values of booster_version column from table spacexdataset
- as "Booster …" : alias to rename the new column as "Booster …"
- where payload .. = : to filter with subquery regarding value of payload
- ..=(select max ..) : subquery pull maximum value of payload column, then being filter in main query

Booster Versions carried Max Payload Mass (kg)
F9 B5 B1048.4
F9 B5 B1048.5
F9 B5 B1049.4
F9 B5 B1049.5
F9 B5 B1049.7
F9 B5 B1051.3
F9 B5 B1051.4
F9 B5 B1051.6
F9 B5 B1056.4
F9 B5 B1058.3
F9 B5 B1060.2
F9 B5 B1060.3

2015 Launch Records

```
%sql select monthname(date) as "Month in 2015 fail in Drone Ship" 
,landing__outcome,booster_version,launch_site from spacexdataset where 
landing__outcome='Failure (drone ship)' and year(date)=2015;
```

- monthname(date) : to pull month name value on DATE column
- as “Month …” : alias to rename the new column as “Month …”
- selected column : month name, landing outcome, booster version, launch site
- where landing_outcome= : keyword wild card customer exactly same as ‘Failure (drone ship)’
- and year(date)=2015 : additional filter by value of year on DATE column to be exactly 2015

Month in 2015 fail in Drone Ship	landing__outcome	booster_version	launch_site
January	Failure (drone ship)	F9 v1.1 B1012	CCAFS LC-40
April	Failure (drone ship)	F9 v1.1 B1015	CCAFS LC-40

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

```
%sql select landing__outcome,count(landing__outcome) as "Total Successful
Landing" from spacexdataset where landing__outcome like 'Success%' and date
between '2010-06-04' and '2017-03-20' group by landing__outcome order by
count(landing__outcome) Desc;
```

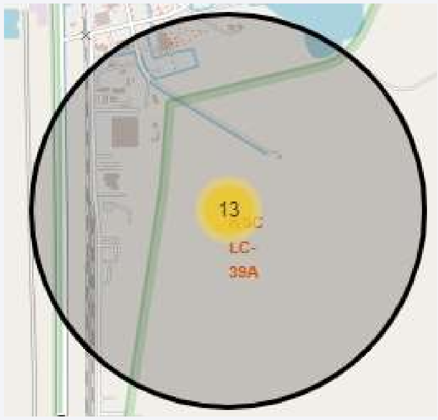
- select landing_outcome, count(landing_outcome) : to pull landing outcome column and its count
- as "Total …" : alias to rename the new column as "Total …"
- where landing_outcome like.. : keyword wild card customer start with 'Success.., weather on drone ship or ground pad
- between : additional filter by value of DATE column between 2 dates value
- group by : make a new tabel grouped by landing_outcome column
- order by count .. desc. : sort the new row descending based on count

landing_outcome	Total Successful Landing
Success (drone ship)	5
Success (ground pad)	3

Section 3

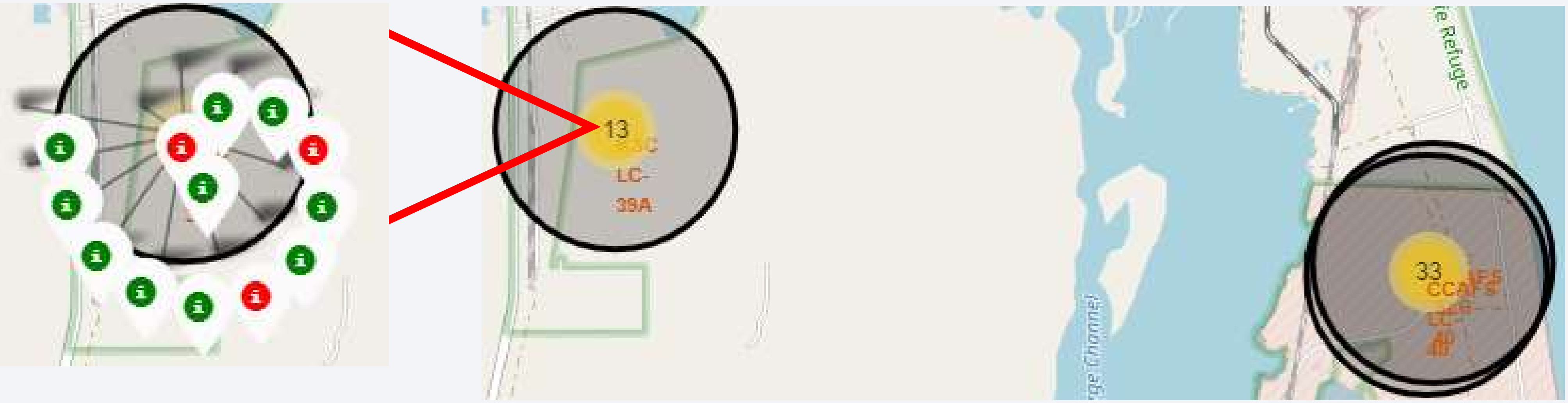
Launch Sites Proximities Analysis

All Launch Sites



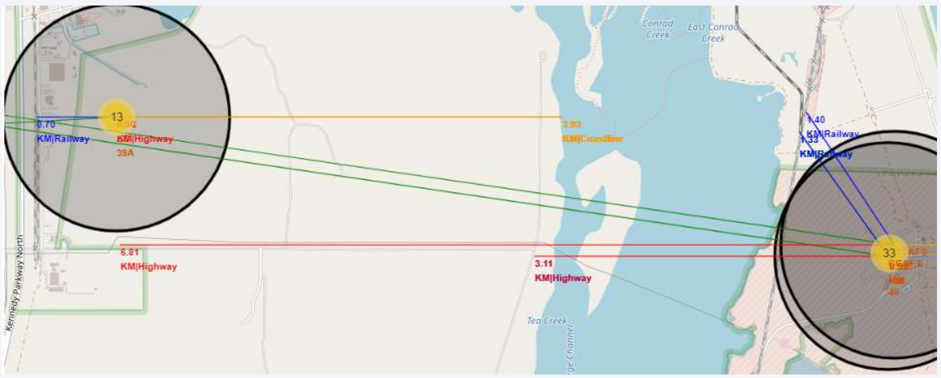
- Markers indicate launch sites
- Circles indicate highlighted areas around specific coordinates, (i.e. : NASA)
- Launch site near sea / coast line and far from big city, for safety analysis, but not too far from logistic needs such as highway / railroads

Launch Outcomes by Launch Sites



- Ex : KSC LC-39A launch site
- 13 launch in KSC, 10 success launch with green mark, 3 failure with red mark using marker cluster, the best probability of successful launch

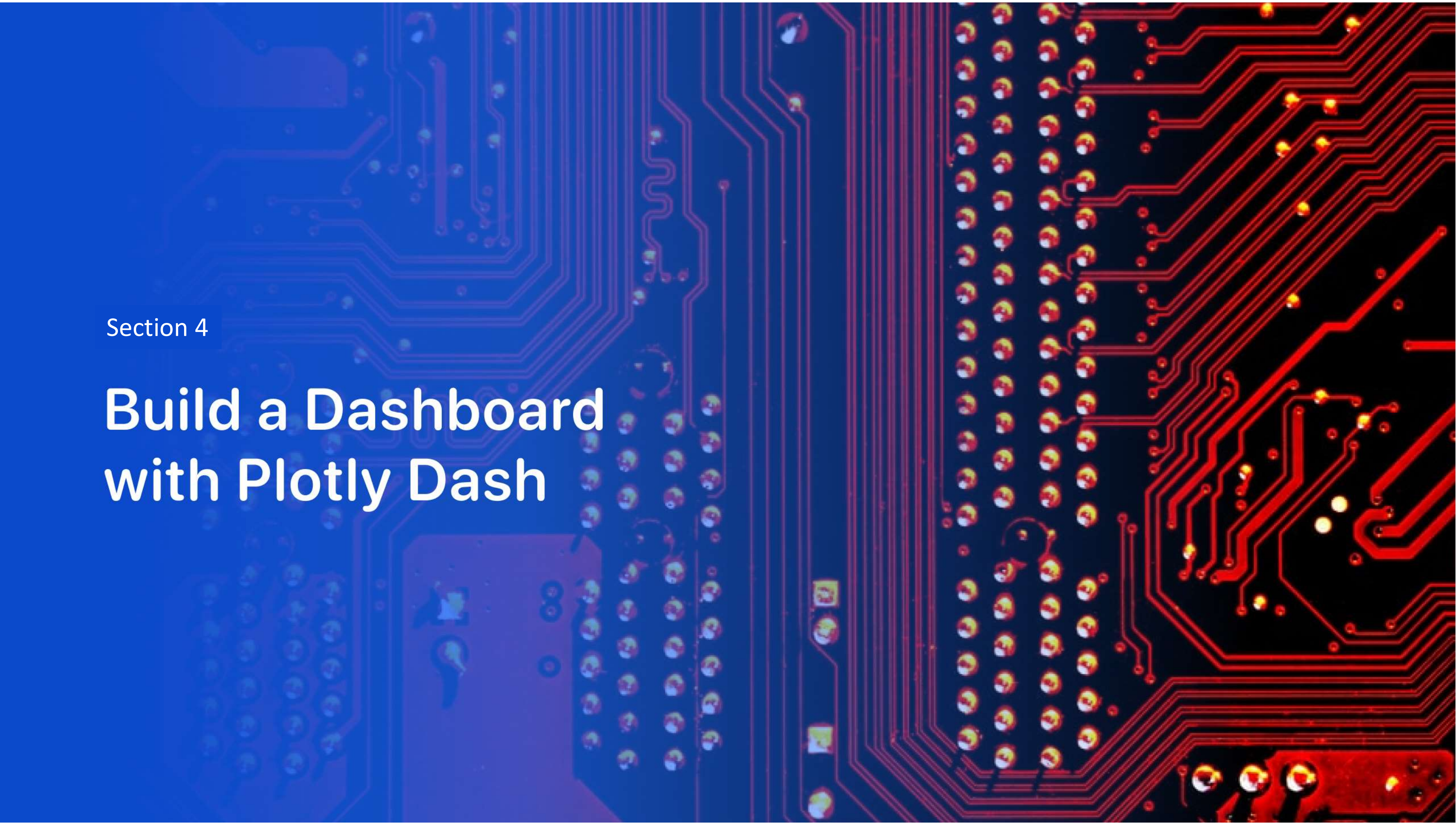
Proximity Map (Logistic & Safety)



	Launch Site	distance to Railway	distance to Highway	distance to Coastline	distance to City
0	CCAFS LC-40	1.334019	3.112840	0.948757	77.164049
1	CCAFS SLC-40	1.397330	6.808158	0.889001	77.201193
2	KSC LC-39A	0.704753	0.000056	3.927363	72.389724
3	VAFB SLC-4E	1.375983	14.613598	1.404180	14.332950

showing Best & Safety Place to launch (in distance)

- near coastline and far from city to minimize the risk of accident / inhabited area
- near logistic infrastructure around
- Distance from Cities : > 14 km
- Distance from Coastline : < 4 km
- Distance from Highway : VAFB most far 14.6 km, else < 7 km
- Distance from Railway : <1.4 km

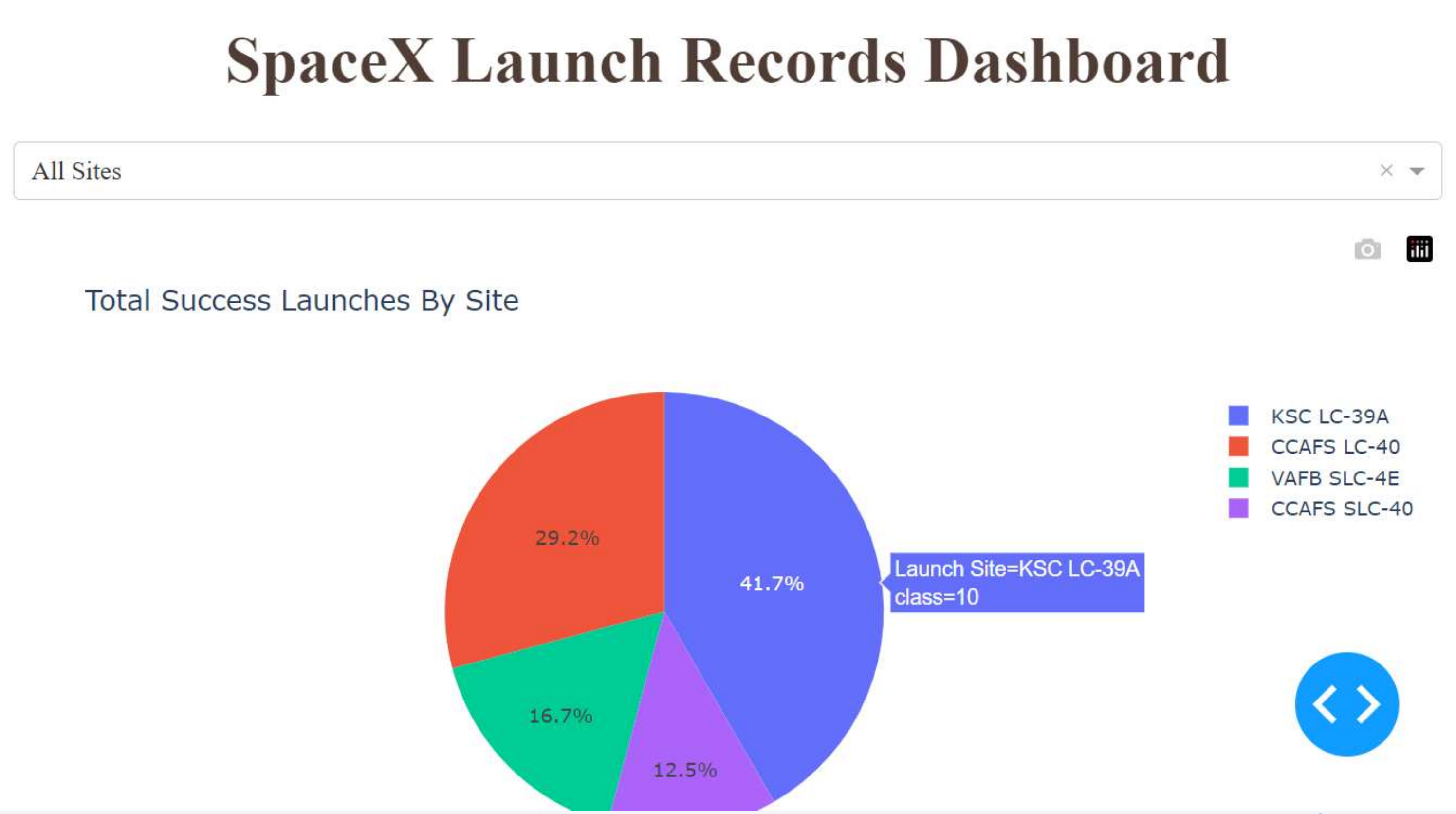


Section 4

Build a Dashboard with Plotly Dash

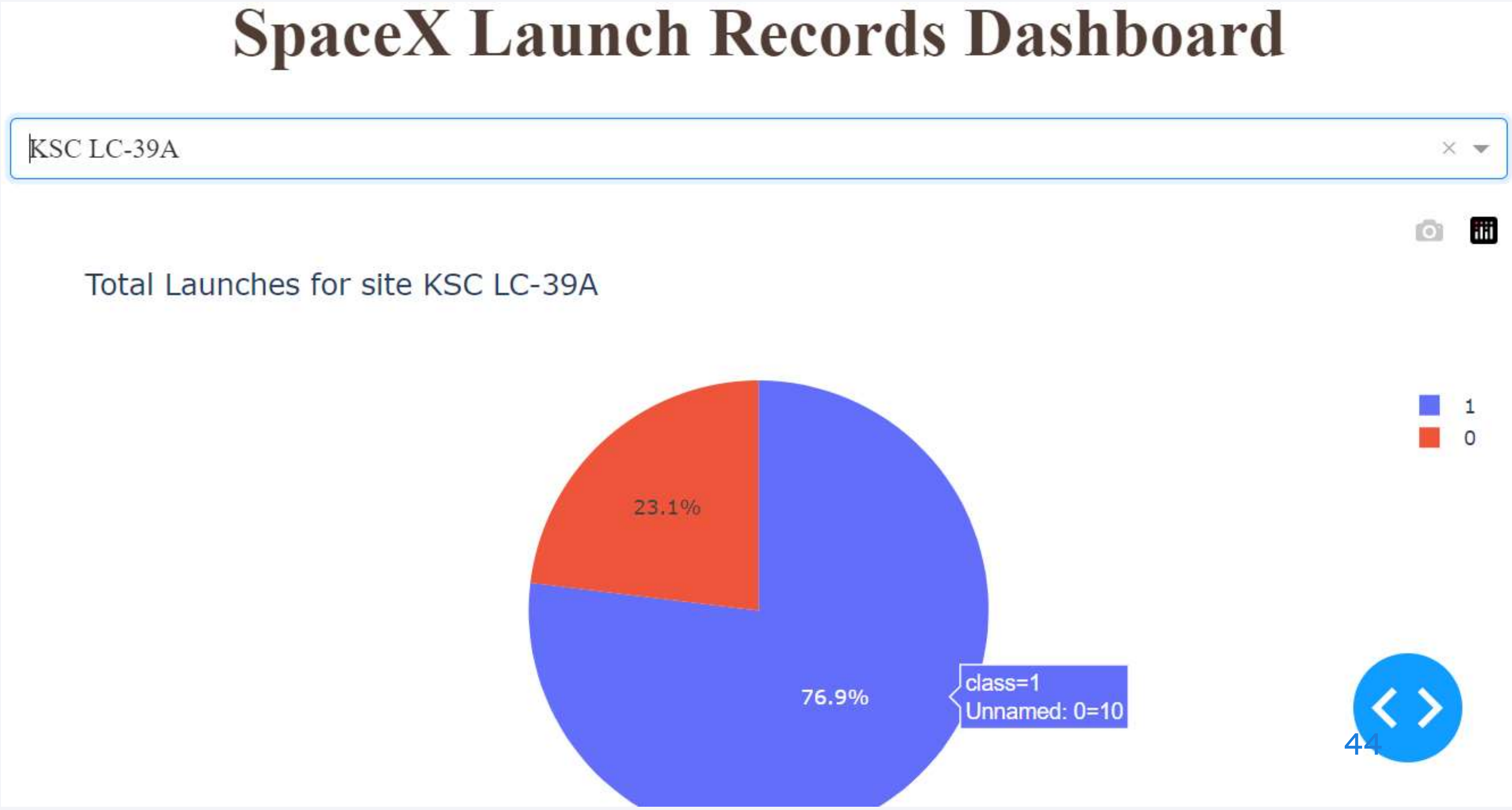
Successful Launches by Site Dashboard

- 10 of 24 success launch in KSC LC-39A (41.7 %) above the others



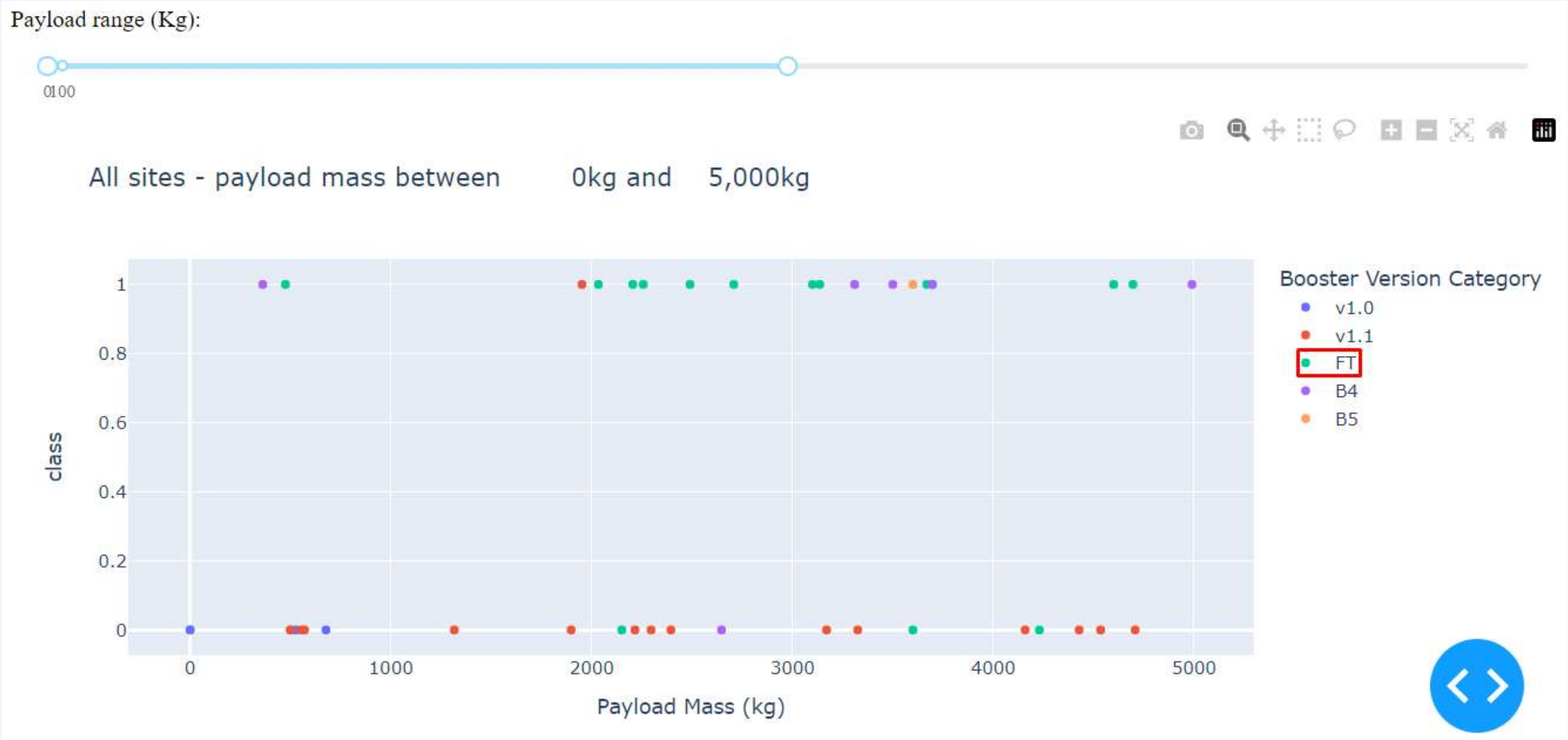
Launches Success Ratio on Most Success Rate Site

- 10 of 13 success launch in KSC LC-39A (76.9 %)



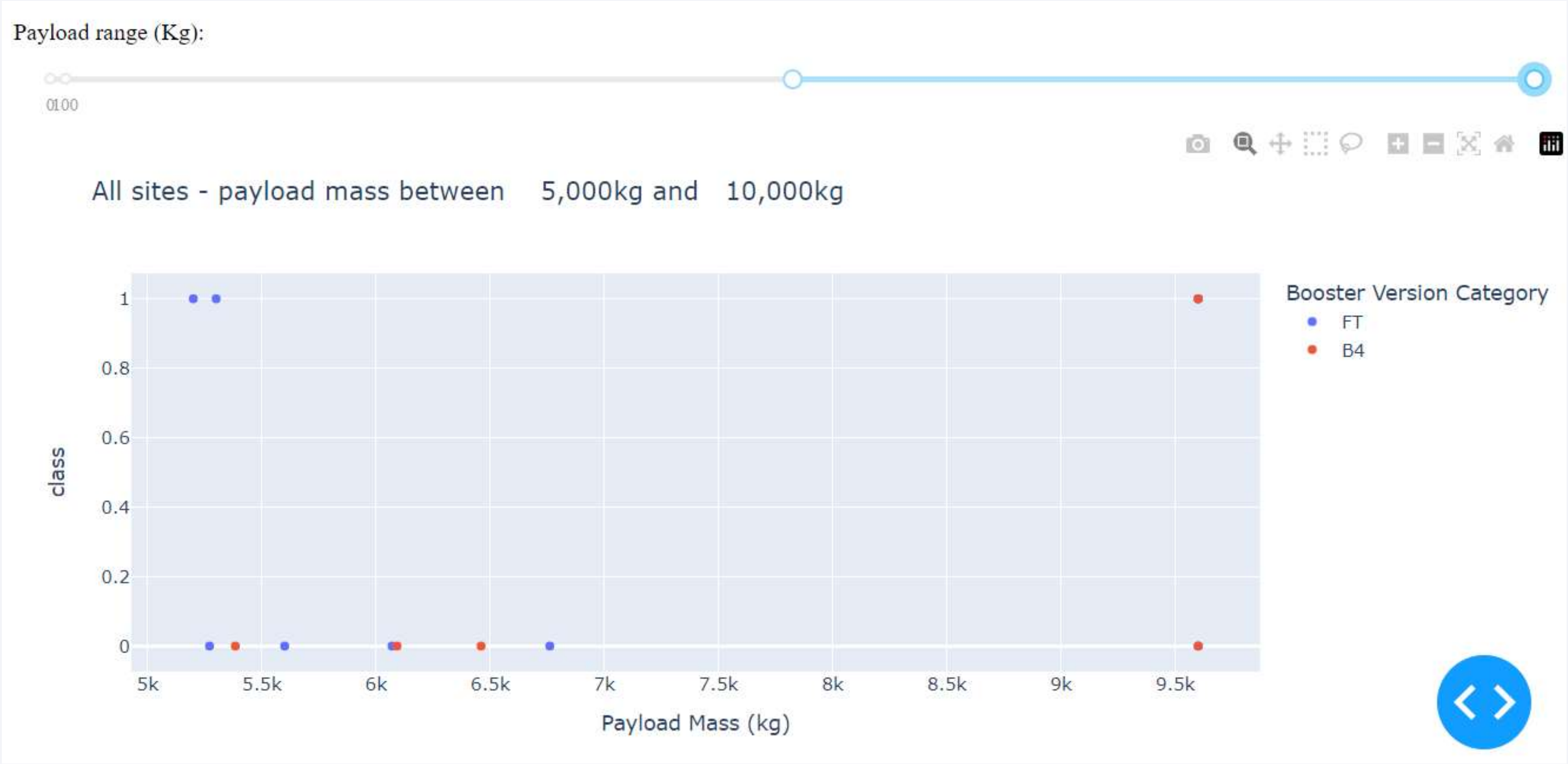
Payload vs Launch Outcome Dashboard (1/2)

- < 5 tons, FT booster prove most success launch



Payload vs Launch Outcome Dashboard (2/2)

> 5 tons, not enough data to estimate the best Booster option



Section 5

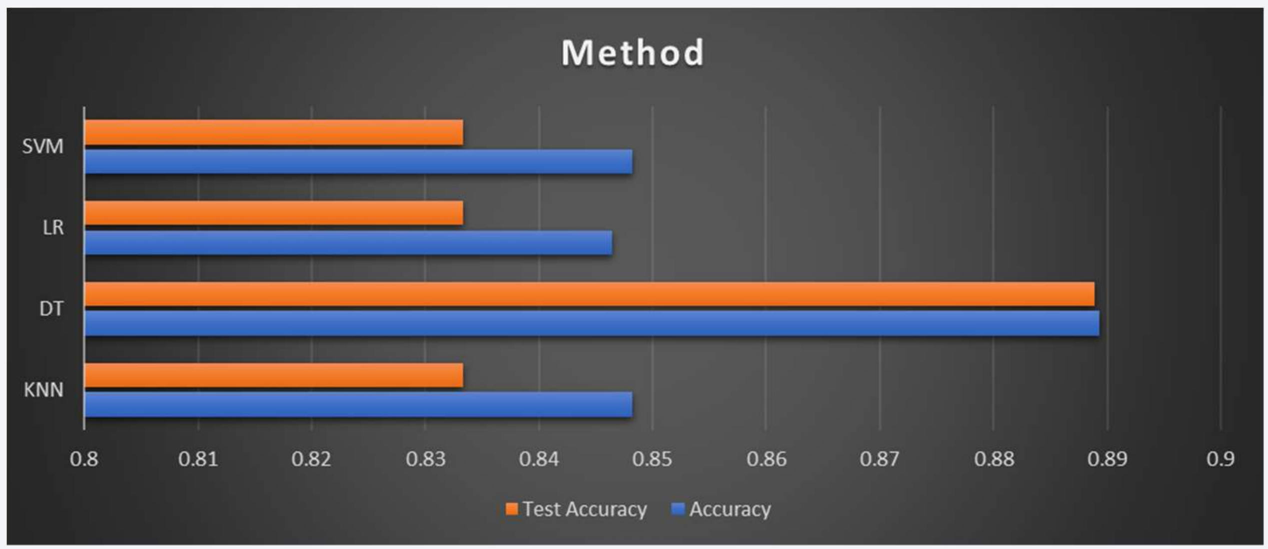
Predictive Analysis (Classification)

Classification Accuracy

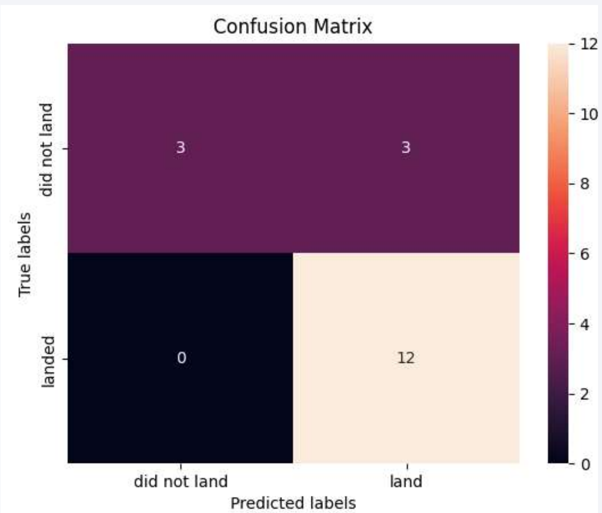
The model with the highest classification accuracy is Decision Tree (88.93%)

Method	Accuracy	Test Accuracy	Tuned Hyperparameter
KNN	0.8482142	0.8333333	{'algorithm': 'auto', 'n_neighbors': 10, 'p': 1}
DT	0.8892857	0.8888889	{'criterion': 'gini', 'max_depth': 6, 'max_features': 'sqrt', 'min_samples_leaf': 4, 'min_samples_split': 10, 'splitter': 'random'} accuracy : 0.8892857142857145
LR	0.8464285	0.8333333	{'C':0.01, 'penalty':'l2', 'solver':'lbfgs'}
SVM	0.8482142	0.8333333	{'C': 1.0, 'gamma': 0.03162277660168379, 'kernel': 'sigmoid'}

	Accuracy Score
KNN	0.848214
Decision Tree	0.889286
Logistic Regression	0.846429
SVM	0.848214



Confusion Matrix



		Predicted		
		didn't land (NO)	land (YES)	
Actual	didn't land (NO)	True Negative (TN) : 3	False Postive (FP) : 3	6
	land (YES)	False Negative (FN) : 0	True Postive (FP) : 12	12
		3	15	18

No	Parameter	Function		Calc	Result
1	**Accuracy**	(TP+TN)/Total	:	(12+3)/18	0.833333
2	**Misc Rate**	(FP+FN)/Total	:	(3+0)/18	0.166667
3	**TN Rate**	TN/Actual No	:	3/6	0.5
4	**TP Rate**	TP/Actual Yes	:	12/12	1
5	**FN Rate**	FN/Actual Yes	:	0/12	0
6	**FP Rate**	FP/Actual No	:	3/6	0.5
7	**Precision**	TP/Predicted Yes	:	12/15	0.8
8	**Prevalence**	Actual Yes/Total	:	12/18	0.666667

- In 4 models, all confusion matrix is the same. It is successfully 100 % predict true positive with 83 % Accuracy

Conclusions

1. The best launch site is KSC LC-39A;
2. Orbit ES-L1, GEO, HEO, SSO has highest mean success rate;
3. Most successful booster version : FT category
4. Payload mass > 5 tons are less risky and have most success rate;
5. Best & Safety Place to launch (in distance) is near coastline and far from city to minimize the risk of accident / inhabited area and also near logistic infrastructure around
6. Success rate keep increasing from 2013 to 2020 (generally, assuming a period time for adjustment & improvement analysis for budget and technology)
7. Decision Tree Classifier is the best model in this scenario with accuracy score 88.93 % and can be used to predict successful landings

Appendix

- Folium didn’t show maps on Github even it’s already mark Trusted, I use screen shoot
- Almost all of the figures and code are represented above with the Github source link

Thank you!