IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

Kudrenko Valentina
March 2022

Outline

Executive Summary

Summary of methodologies

1. Data Collection through API
2. Data Collection with Web Scraping
3. Data Wrangling
4. Exploratory Data Analysis with Data Visualization
5. Exploratory Data Analysis with SQL
6. Data Collection with Web Scraping
7. Interactive Visual Analytics with Folium
8. Predictive Analysis (Classification)

Executive Summary

Summary of all results

1. Exploratory Data Analysis results
2. Interactive analytics in screenshots
3. Predictive Analytics results

Introduction

Summary of methodologies

In this work, we will predict whether the first stage of the Falcon 9 will land successfully. SpaceX advertises $62 million cost of Falcon 9 rocket launches on its website; other providers cost more than $165 million each, most of the savings is because Space X being able to reuse the first stage

Therefore, if we can determine whether the first stage successfully lands, this information could be used if an alternative company wants to bid against SpaceX for a rocket launch

The goal of this project is to create a machine learning pipeline to predict whether the landing of the first stage will be successful and what factors influence it

Problems we want to find answers

Factors that determine successful rocket landing

The effect of each relationship of rocket variables on outcome

Conditions which will aid SpaceX have to achieve the best results

Section 1

Methodology

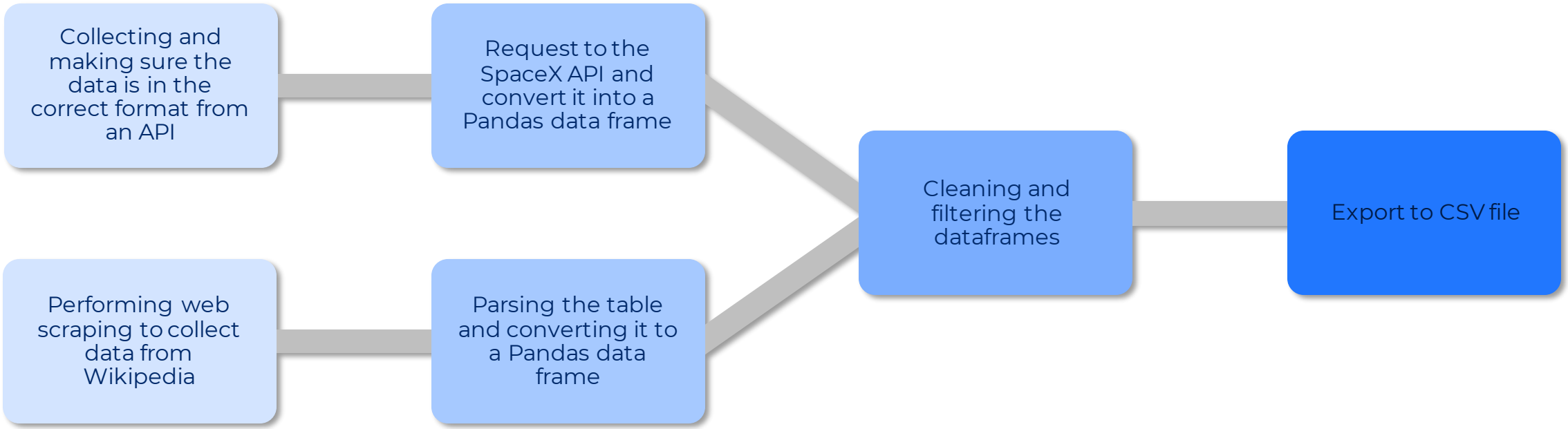
Methodology

Executive Summary

Data collection	• Data Collection through API • Data Collection with Web Scraping
Perform data wrangling	• Exploratory Data Analysis • Determine Training Labels
Perform exploratory data analysis (EDA) using visualization and SQL	• Exploratory Data Analysis • Preparing Data Feature Engineering
Perform interactive visual analytics using Folium and Plotly Dash	• Finding some geographical patterns about launch sites • Interactive visual analysis of SpaceX launch data in real time with Dashboard
Perform predictive analysis using classification models	• Performing exploratory Data Analysis and determine Training Labels • Building and evaluating classification models

Methodology

Data Collection



Data Collection – SpaceX API

```
data=pd.json_normalize(response.json())
```

```
launch_dict = {'FlightNumber': list(data['flight_number']),
'Date': list(data['date']),
'BoosterVersion':BoosterVersion,
...,
'Latitude': Latitude}
data_falcon=pd.DataFrame(launch_dict)
```

```
data_falcon9['PayloadMass'] = data_falcon9\
['PayloadMass'].replace(np.nan, Mean_PayloadMass)
```

Getting Response from API using the GET request

Converting Response into a Pandas dataframe using .json_normalize()

Apply custom functions to clean data

Assign list to dictionary then create dataframe

Filter the dataframe

Data Wrangling (dealing with Missing Values)

Export to CSV file

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

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

```
data_falcon9 = data_falcon\
[data_falcon.BoosterVersion == 'Falcon 9']
data_falcon9.loc[:,'FlightNumber'] = list\
(range(1, data_falcon9.shape[0]+1))
```

```
data_falcon9.to_csv('dataset_part_1.csv', index=False)
```

Received Pandas dataframe

	FlightNumber	Date	BoosterVersion	PayloadMass	Orbit	LaunchSite	Outcome	Flights	GridFins	Reused	Legs	LandingPad	Block	ReusedCount	Serial	Longitude	Latitude
4	1	2010-06-04	Falcon 9	6123.547647	LEO	CCSFS SLC 40	None None	1	False	False	False	None	1.0	0	B0003	-80.577366	28.561857
5	2	2012-05-22	Falcon 9	525.000000	LEO	CCSFS SLC 40	None None	1	False	False	False	None	1.0	0	B0005	-80.577366	28.561857
6	3	2013-03-01	Falcon 9	677.000000	ISS	CCSFS SLC 40	None None	1	False	False	False	None	1.0	0	B0007	-80.577366	28.561857
7	4	2013-09-29	Falcon 9	500.000000	PO	VAFB SLC 4E	False Ocean	1	False	False	False	None	1.0	0	B1003	-120.610829	34.632093
8	5	2013-12-03	Falcon 9	3170.000000	GTO	CCSFS SLC 40	None None	1	False	False	False	None	1.0	0	B1004	-80.577366	28.561857

More details you can see in the Jupyter Notebook "Data Collection API"

Data Collection - Scraping

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

```
column_names = []
temp = soup.find_all('th')
for x in range(len(temp)):
    try:
     name = extract_column_from_header(temp[x])
     if (name is not None and len(name) > 0):
        column_names.append(name)
    except:
     pass
```

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

Request the Falcon9 Launch Wiki page from its URL

Creating BeautifulSoup object from the HTML response

Finding tables

Extract all column/variable names from the HTML table header

Create a data frame by parsing the launch HTML tables

Export to CSV file

```
static_url = "https://en.wikipedia.org/w/index.\
php?title=List_of_Falcon_9_and_Falcon_Heavy_\
launches&oldid=1027686922"
response = requests.get(static_url)
```

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

```
launch_dict= dict.fromkeys(column_names)
df = pd.DataFrame.from_dict(launch_dict)
```

Received Pandas dataframe

	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

More details you can see in the Jupyter Notebook "Complete the Data Collection with Web Scraping lab"

Data Wrangling

```
df.Orbit.value_counts()
```

```
landing_class = []
for outcome in df['Outcome']:
    if outcome in bad_outcomes:
        landing_class.append(0)
    else:
        landing_class.append(1)
```

Calculate the number of launches on each site

Calculate the number and occurrence of each orbit

Calculate the number and occurence of mission outcome per orbit type

Create a landing outcome label from Outcome column

Export to CSV file

```
df.LaunchSite.value_counts()
```

```
landing_outcomes = df.Outcome.value_counts()
```

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

Received Pandas dataframe

	FlightNumber	Date	BoosterVersion	PayloadMass	Orbit	LaunchSite	Outcome	Flights	GridFins	Reused	Legs	LandingPad	Block	ReusedCount	Serial
0	1	2010-06-04	Falcon 9	6104.959412	LEO	CCAFS SLC 40	None None	1	False	False	False	NaN	1.0	0	B0003
1	2	2012-05-22	Falcon 9	525.000000	LEO	CCAFS SLC 40	None None	1	False	False	False	NaN	1.0	0	B0005
2	3	2013-03-01	Falcon 9	677.000000	ISS	CCAFS SLC 40	None None	1	False	False	False	NaN	1.0	0	B0007
3	4	2013-09-29	Falcon 9	500.000000	PO	VAFB SLC 4E	False Ocean	1	False	False	False	NaN	1.0	0	B1003
4	5	2013-12-03	Falcon 9	3170.000000	GTO	CCAFS SLC 40	None None	1	False	False	False	NaN	1.0	0	B1004

More details you can see in the Jupyter Notebook "Complete the EDA lab"

EDA with Data Visualization

Visualizing the data makes it very easy to predict which factors will lead to the highest probability of success both at launch and at landing

To see these relationships, were plotted:

Scatter point charts

showing patterns between:
- Flight Number vs. Payload
- Flight Number vs. Launch Site
- Launch Site vs. Payload
- Flight Number vs. Orbit
- Payload vs. Orbit

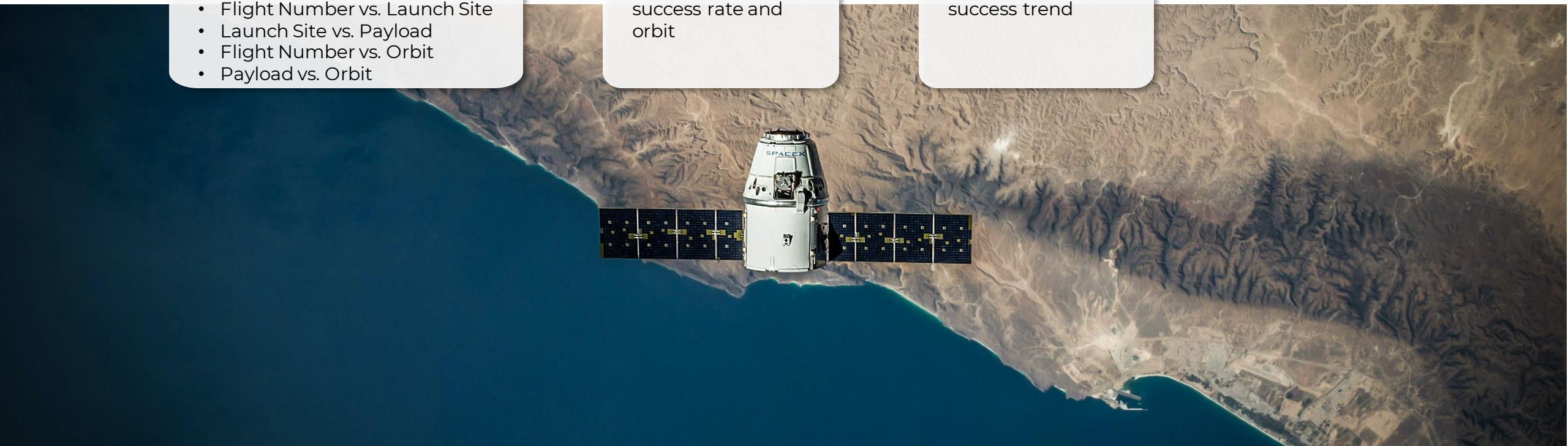
Bar chart

showing patterns between success rate and orbit

Line chart

showing the average launch success trend

More details you can see in the Jupyter Notebook "EDA with Data Visualization"



EDA with SQL

After connecting to the database, we wrote and executed the following SQL queries to get insight from the data

1. Display the names of the unique launch sites in the space mission
2. 5 records where launch sites begin with the string 'CCA'
3. The total payload mass carried by boosters launched by NASA (CRS)
4. Average payload mass carried by booster version F9 v1.1
5. The date when the first successful landing outcome in ground pad was achieved
6. The names of the boosters which have success in drone ship and have payload mass greater than 4000 but less than 6000
7. The total number of successful and failure mission outcomes
8. The names of the booster versions which have carried the maximum payload mass
9. The failed landing outcomes in drone ship, their booster versions, and launch site names for in year 2015
10. The count of landing outcomes (such as Failure (drone ship) or Success (ground pad)) between the date 2010-06-04 and 2017-03-20, in descending order (first 5 results)

More details you can see in the Jupyter Notebook "EDA with SQL"

Build an Interactive Map with Folium

Folium makes it easy to visualize data processed in Python on an interactive map

I used latitude and longitude coordinates for each launch site and added a circle marker around each launch site labeled with the name of the launch site. It's also easy to visualize the number of successes and failures for each launch pad using the green and red markers on the map

Map Objects	Code	Result
Map Marker	folium.Marker()	Created marks on the map
Icon Marker	folium.Icon()	Created icons on the map
Circle Marker	folium.Circle()	Created circles where the markers are
PolyLine	folium.PolyLine()	Created lines between the points
Marker Cluster Object	MarkerCluster()	Simplified maps containing many markers with the same coordinates
AntPath	folium.Plugins.AntPath()	Created animated lines between the points

Conclusions

1. Launch sites are near the equator line to minimize fuel consumption by taking advantage of the Earth's eastward rotation to help spacecraft into orbit
2. Launch sites are near the coastline so they can fly over the ocean during launch for two safety reasons
   - the crew can abort the launch and try to land on the water;
   - minimize the risk of debris falling on people and buildings
3. The launch sites are near highways, which makes it easy to transport the necessary people and equipment
4. Launch sites are not near cities, but not in the cities themselves, minimizing the danger to densely populated areas

More details you can see in the Jupyter Notebook "Interactive Visual Analytics with Folium lab"

Build a Dashboard with Plotly Dash

Plotly Dash app

We created the Plotly Dash app to enable users to perform real-time interactive visual analysis of SpaceX launch data

Input components

This dashboard app contains input components such as dropdown and range slider to interact with pie chart and scatter chart

Solved questions

After visual analysis using the dashboard, we will be able to get some information to answer the following five questions

- Which site has the most successful launches?
- Which site has the highest launch success rate?
- Which payload range(s) has the highest percentage of successful launches?
- Which payload range(s) has the lowest success rate?
- Which version of F9 Booster (v1.0, v1.1, FT, B4, B5, etc.) has the highest launch success rate?

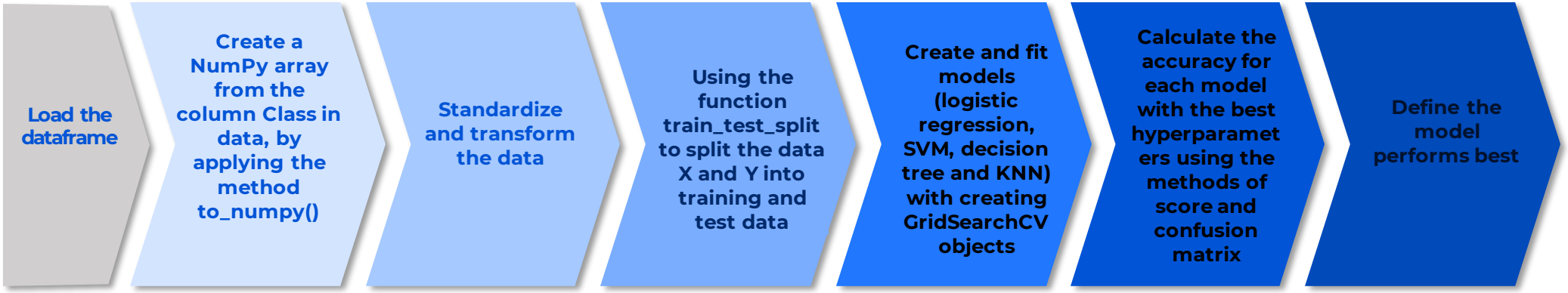
More details you can see in the Jupyter Notebook "SpaceX_dash_app"

Predictive Analysis (Classification)

```
Y = data["Class"].to_numpy()
```

```
X_train, X_test, Y_train, \
Y_test = train_test_split\
(X, Y, test_size=0.2, random_state=2)
```

```
logreg_cv.score(X_test, Y_test)
yhat=logreg_cv.predict(X_test)
plot_confusion_matrix(Y_test,yhat)
svm_cv.score(X_test, Y_test)
yhat=svm_cv.predict(X_test)
plot_confusion_matrix(Y_test,yhat)
tree_cv.score(X_test, Y_test)
yhat = tree_cv.predict(X_test)
plot_confusion_matrix(Y_test,yhat)
knn_cv.score(X_test, Y_test)
yhat = knn_cv.predict(X_test)
plot_confusion_matrix(Y_test,yhat)
```



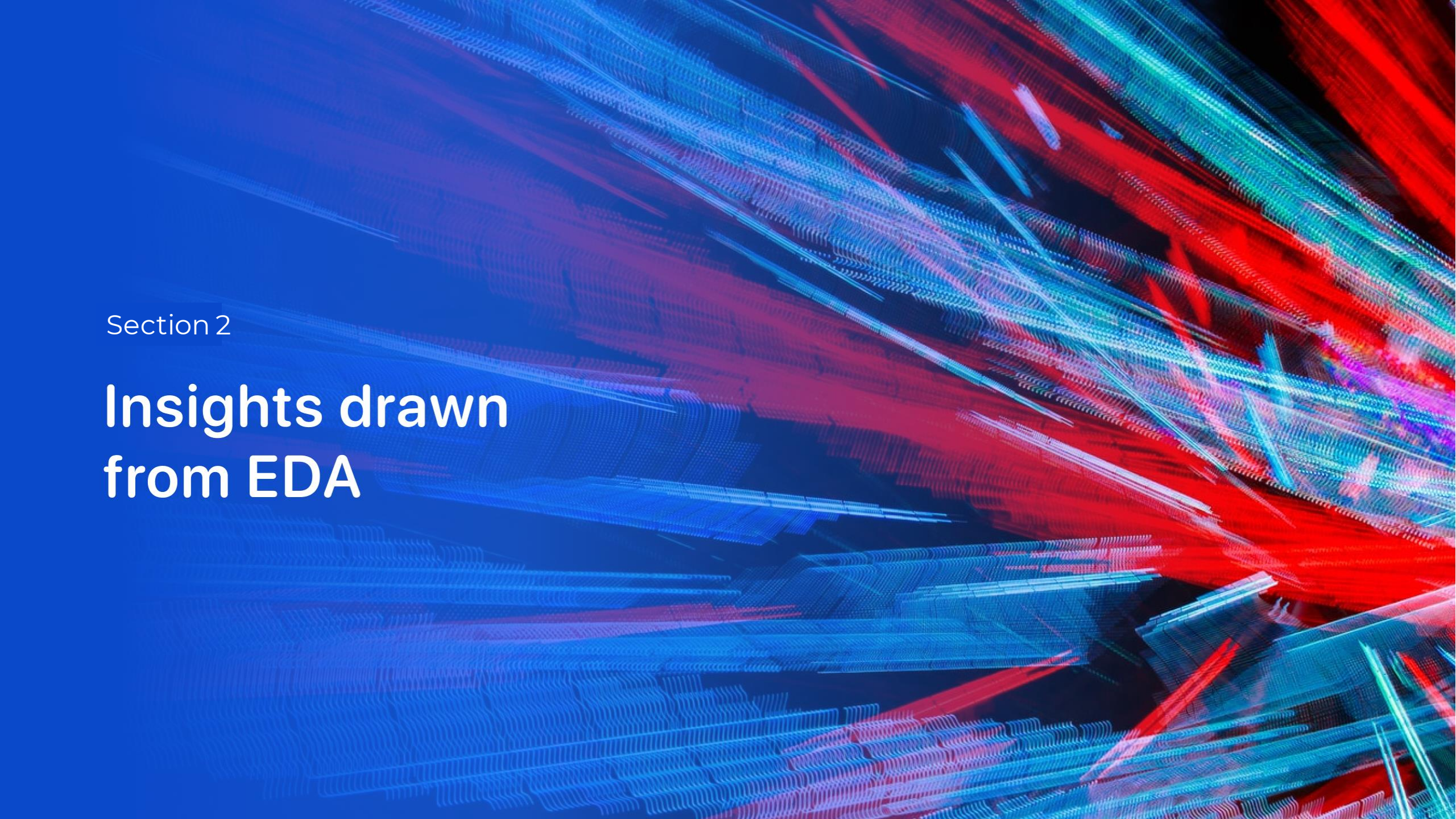

```
data = pd.read_csv("https://cf-courses-data.\
s3.us.cloud-object-storage.appdomain.cloud/\
IBM-DS0321EN-SkillsNetwork/datasets/\
dataset_part_2.csv")
X = pd.read_csv('https://cf-courses-data.s3.\
us.cloud-object-storage.appdomain.cloud/\
IBM-DS0321EN-SkillsNetwork/datasets/\
dataset_part_3.csv')
```

```
transform = preprocessing.StandardScaler()
X = transform.fit_transform(X)
```

```
lr=LogisticRegression()
logreg_cv = GridSearchCV(lr, parameters, cv=10)
logreg_cv.fit(X_train, Y_train)
svm = SVC()
svm_cv = GridSearchCV(svm, parameters, cv=10)
svm_cv.fit(X_train, Y_train)
tree = DecisionTreeClassifier()
tree_cv = GridSearchCV(tree, parameters, cv=10)
tree_cv.fit(X_train, Y_train)
KNN = KNeighborsClassifier()
knn_cv = GridSearchCV(KNN, parameters, cv=10)
knn_cv.fit(X_train, Y_train)
```

```
algorithms = {'KNN':knn_cv.best_score_,
              'DecisionTree':tree_cv.best_score_,
              'LogisticRegression':logreg_cv.best_score_,
              'SVM':svm_cv.best_score_}
bestalgorithm = max(algorithms, key=algorithms.get)
fig = px.bar(algo_df, x='Algorithm', y='Accuracy',
             hover_data=['Algorithm', 'Accuracy'],
             color='Accuracy',
             color_continuous_scale='rdylgn')
fig.update_layout(title='Algorithm vs. Accuracy',
                  xaxis_title='Algorithm',
                  yaxis_title='Accuracy' )
```

More details you can see in the Jupyter Notebook "Machine Learning Prediction"



Section 2

Insights drawn from EDA

Results

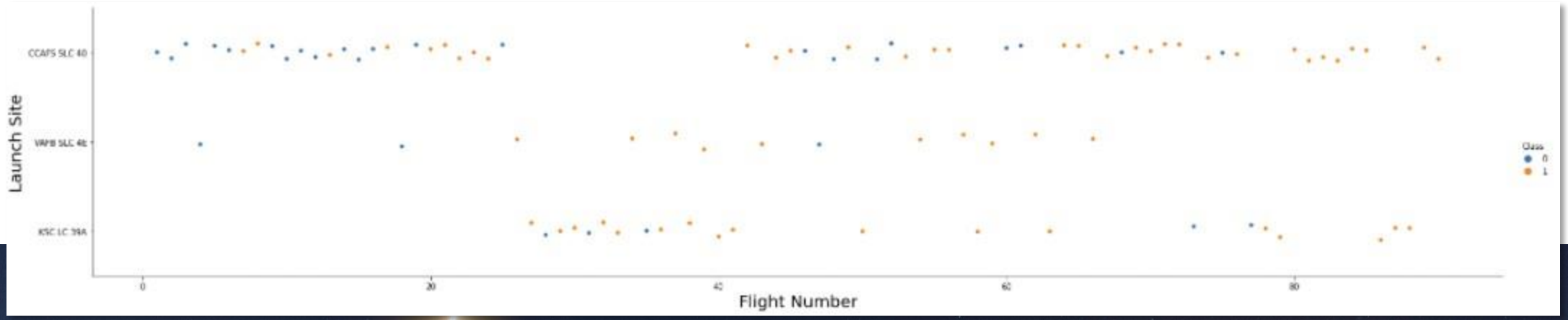
Exploratory data analysis results

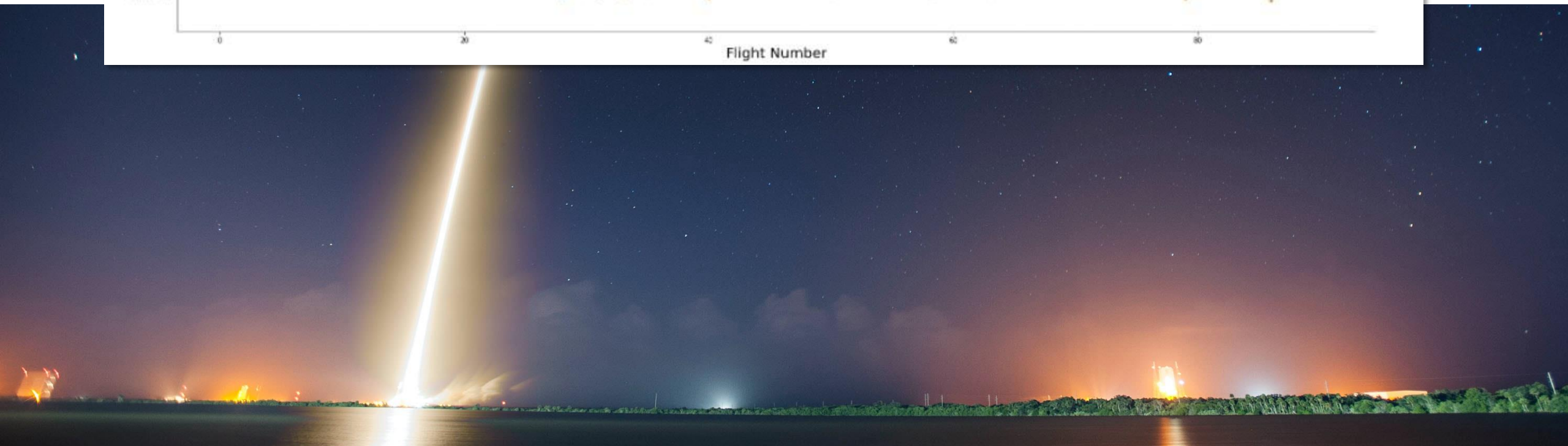
Interactive analytics demo in screenshots

Predictive analysis results

Flight Number vs. Launch Site

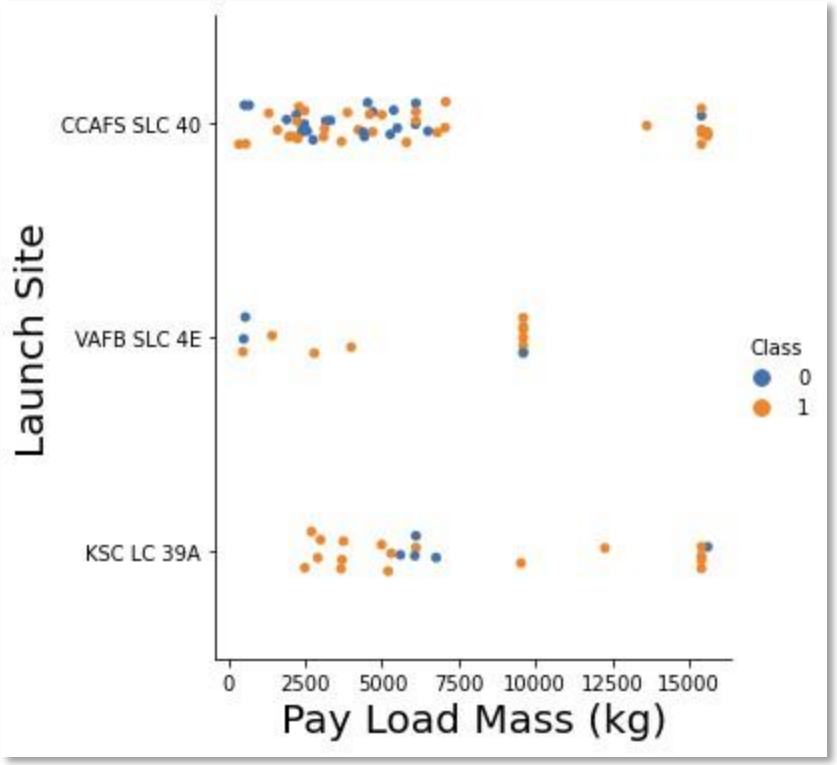
With more flight numbers (after 40) higher the success rate for the Rocket is increasing





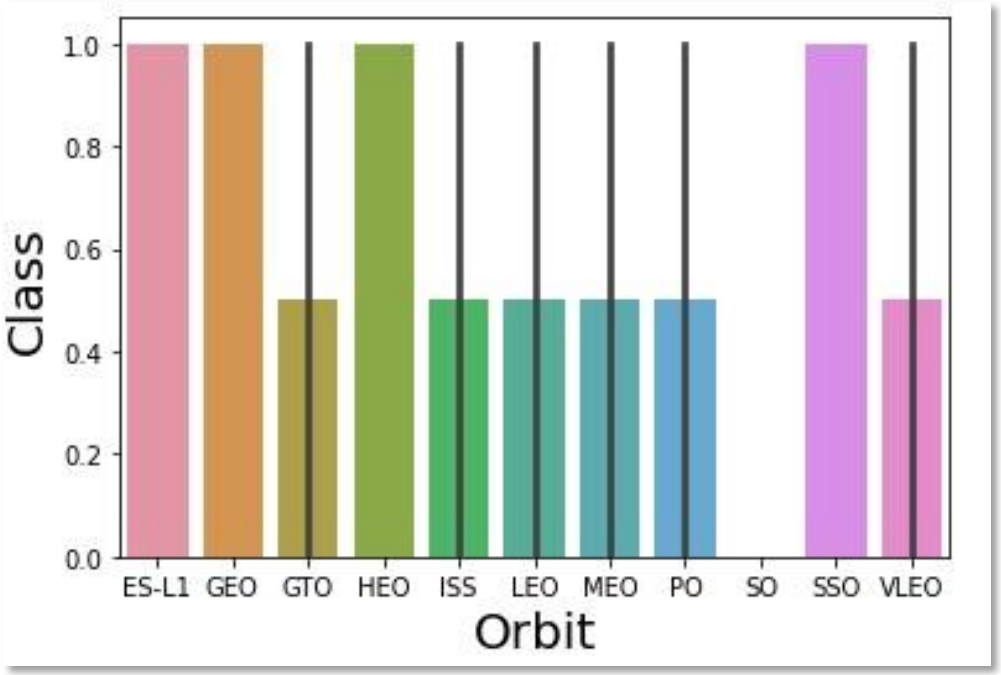
Payload vs. Launch Site

The greater the payload mass (greater than 8000) lower the success rate for the Rocket



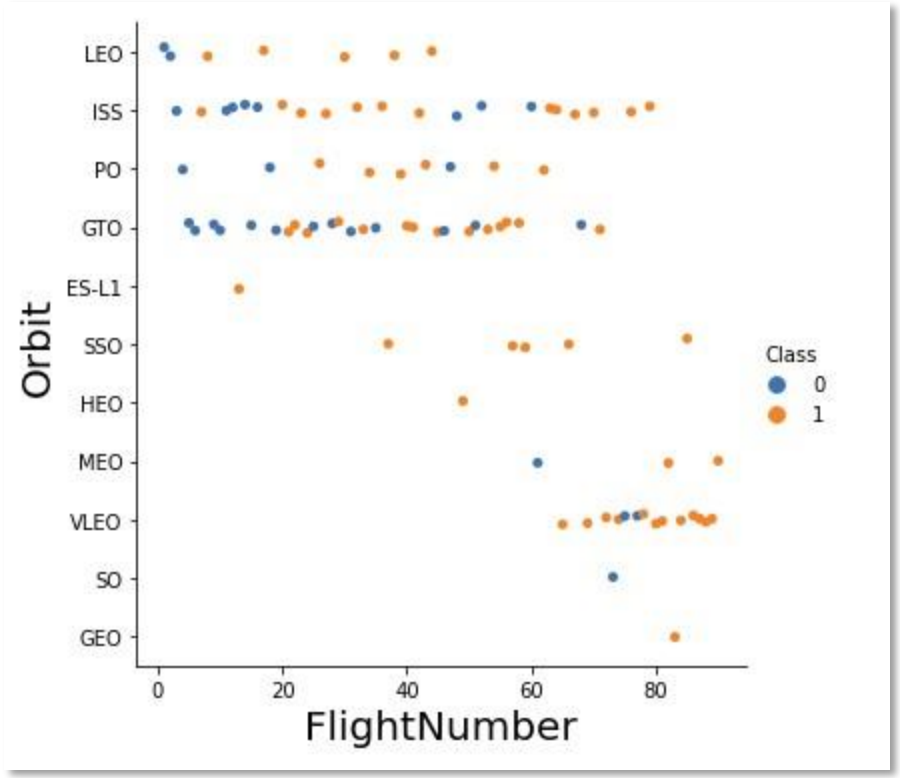
Success Rate vs. Orbit Type

ES-L1, GEO, HEO, SSO orbits has highest Sucess rates



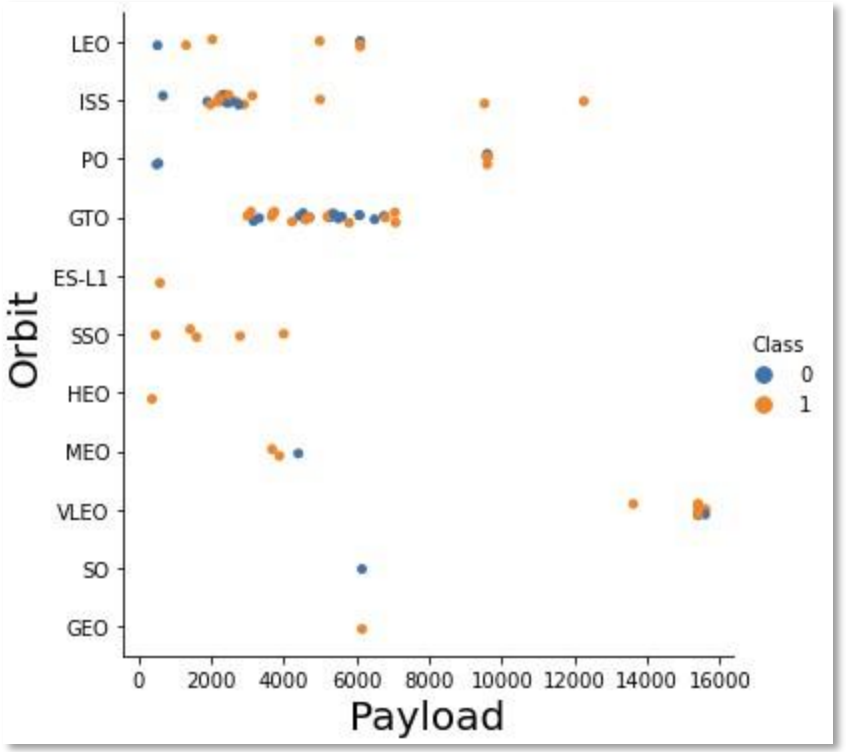
Flight Number vs. Orbit Type

The LEO orbit the Success appears related to the number of flights; on the other hand, there seems to be no relationship between flight number when in GTO orbit



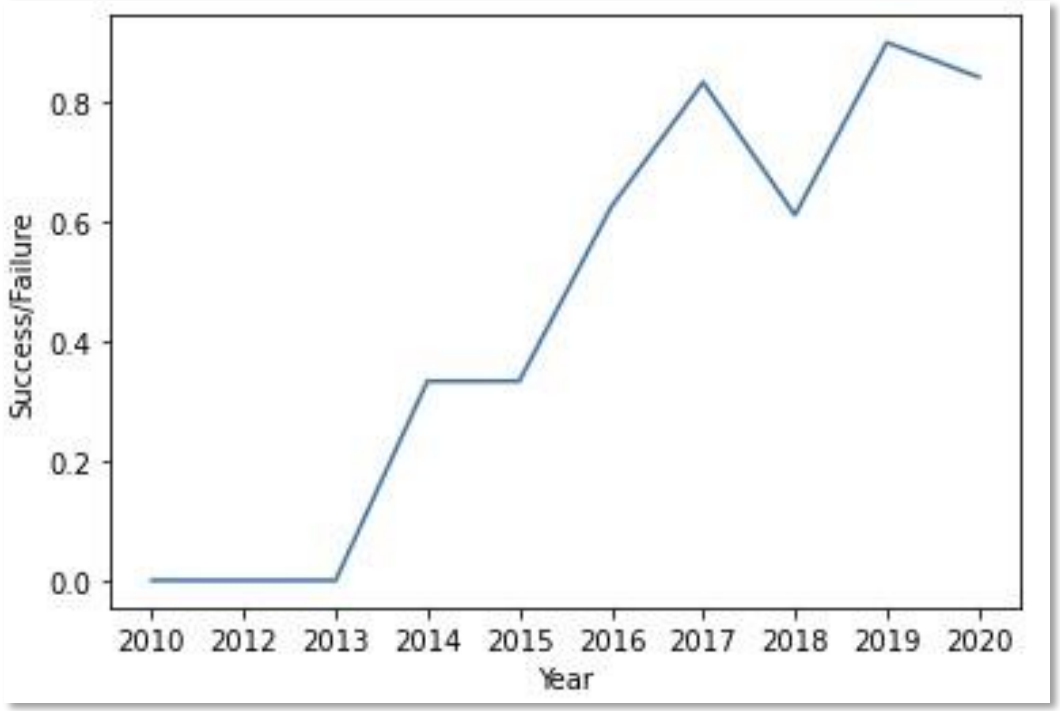
Payload vs. Orbit Type

With heavy payloads the successful landing or positive landing rate are more for Polar, LEO and ISS orbits

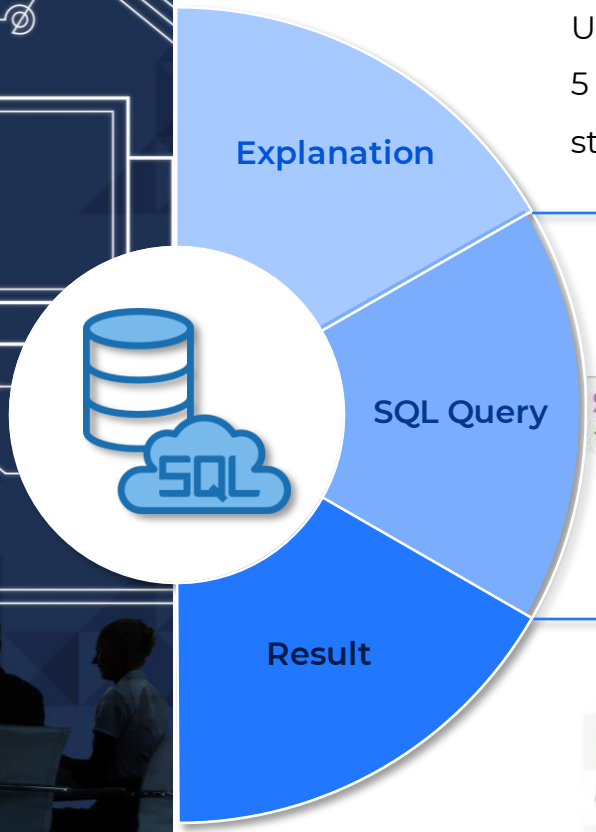


Launch Success Yearly Trend

The success rate since 2013 kept increasing till 2020



All Launch Site Names



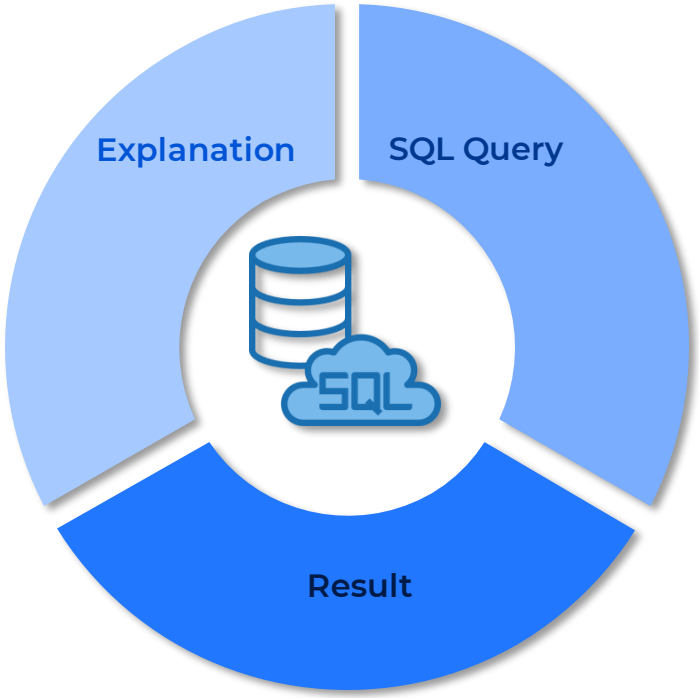
Using ‘distinct’ and ‘from’ in the query we pull 5 records where launch sites begin with the string 'CCA' from table SPACEXDATASET

```
%sql select distinct(LAUNCH_SITE) \
from SPACEXDATASET;
```

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

Launch Site Names Begin with 'CCA'

Using ‘from’, ‘where’, ‘like’ and ‘limit 5’ in the query we fetch 5 records where launch sites begin with the string 'CCA'




```
%sql select * from SPACEXDATASET \
where LAUNCH_SITE like 'CCA%' limit 5;
```

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

Using ‘sum’, ‘from’ and ‘where’ in the query we get the total payload mass carried by boosters launched by NASA (CRS) from table SPACEXDATASET

Explanation

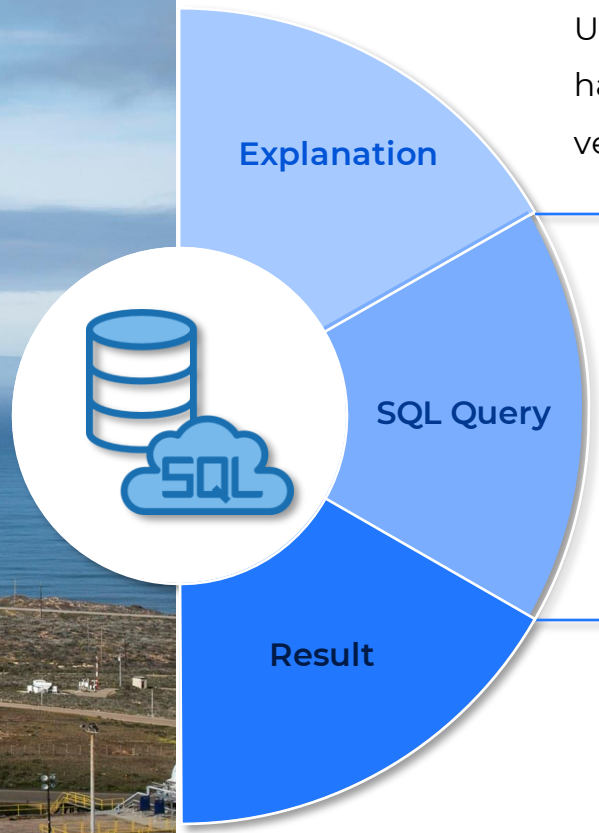
SQL Query

```
%sql select sum(PAYLOAD_MASS__KG_) \
from SPACEXDATASET \
where CUSTOMER = 'NASA (CRS)';
```

Result

```
1
45596
```

Average Payload Mass by F9 v1.1



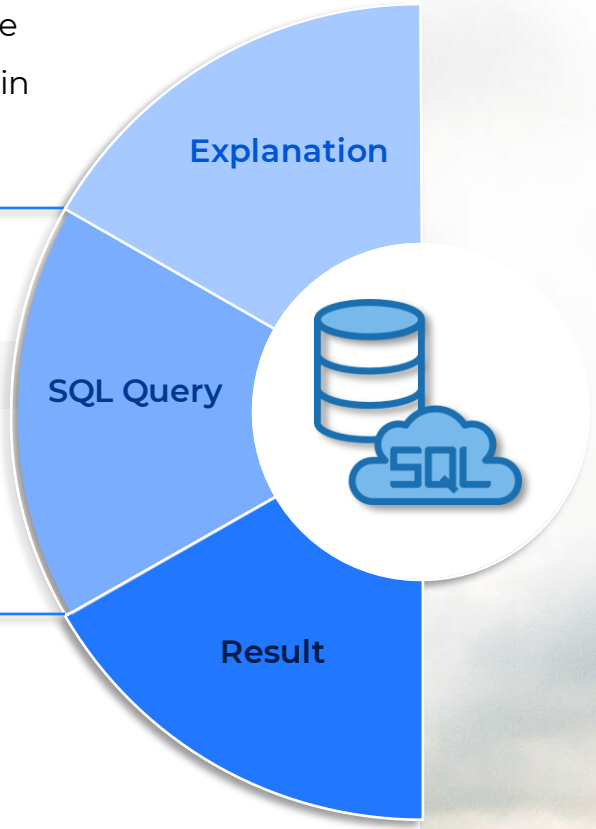
Using ‘avg’, ‘from’ and ‘where’ in the query we have average payload mass carried by booster version F9 v1.1 from table SPACEXDATASET

```
%sql select avg(PAYLOAD_MASS__KG_) \
from SPACEXDATASET \
where BOOSTER_VERSION = 'F9 v1.1';
```

1
2928

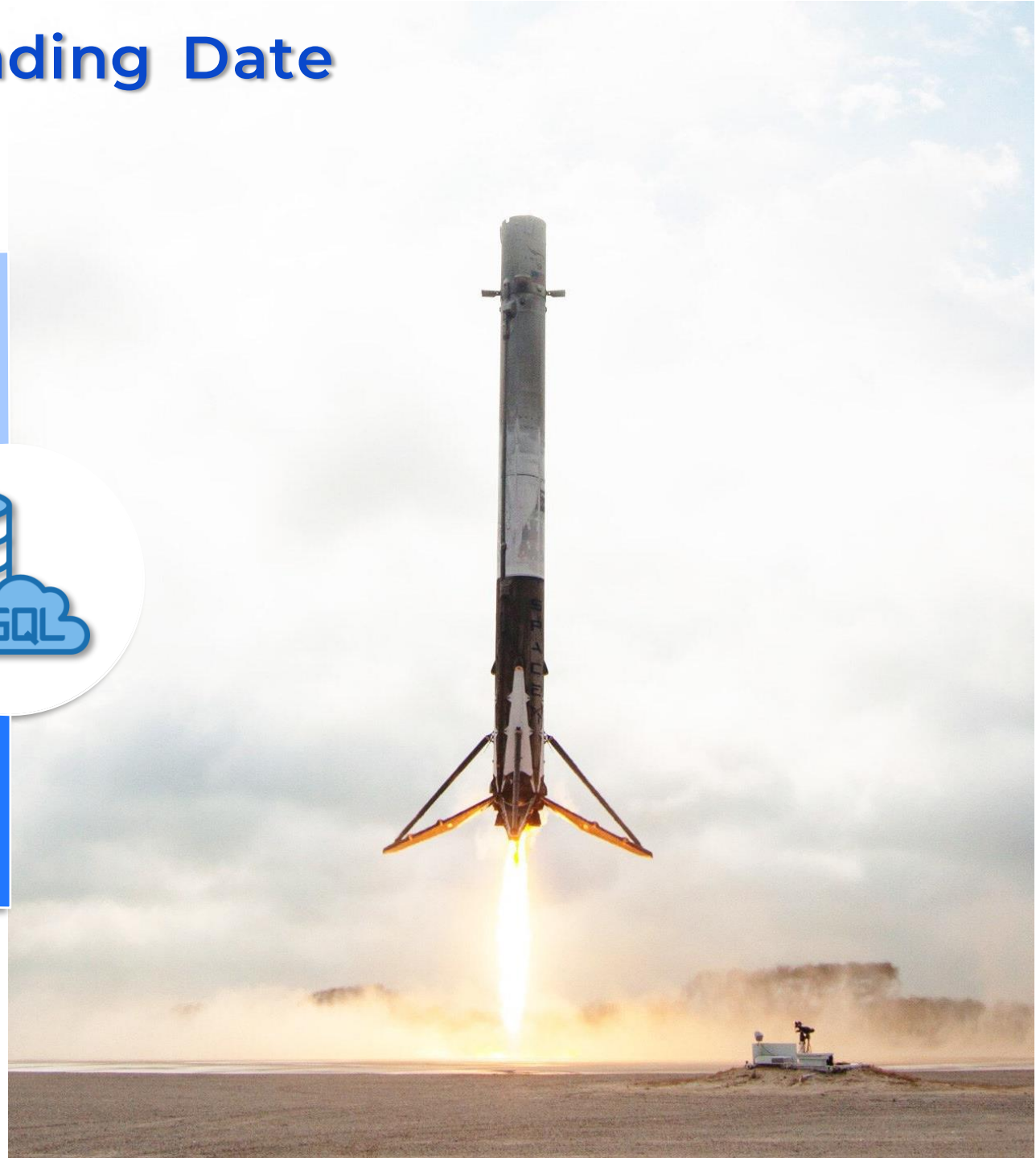
First Successful Ground Landing Date

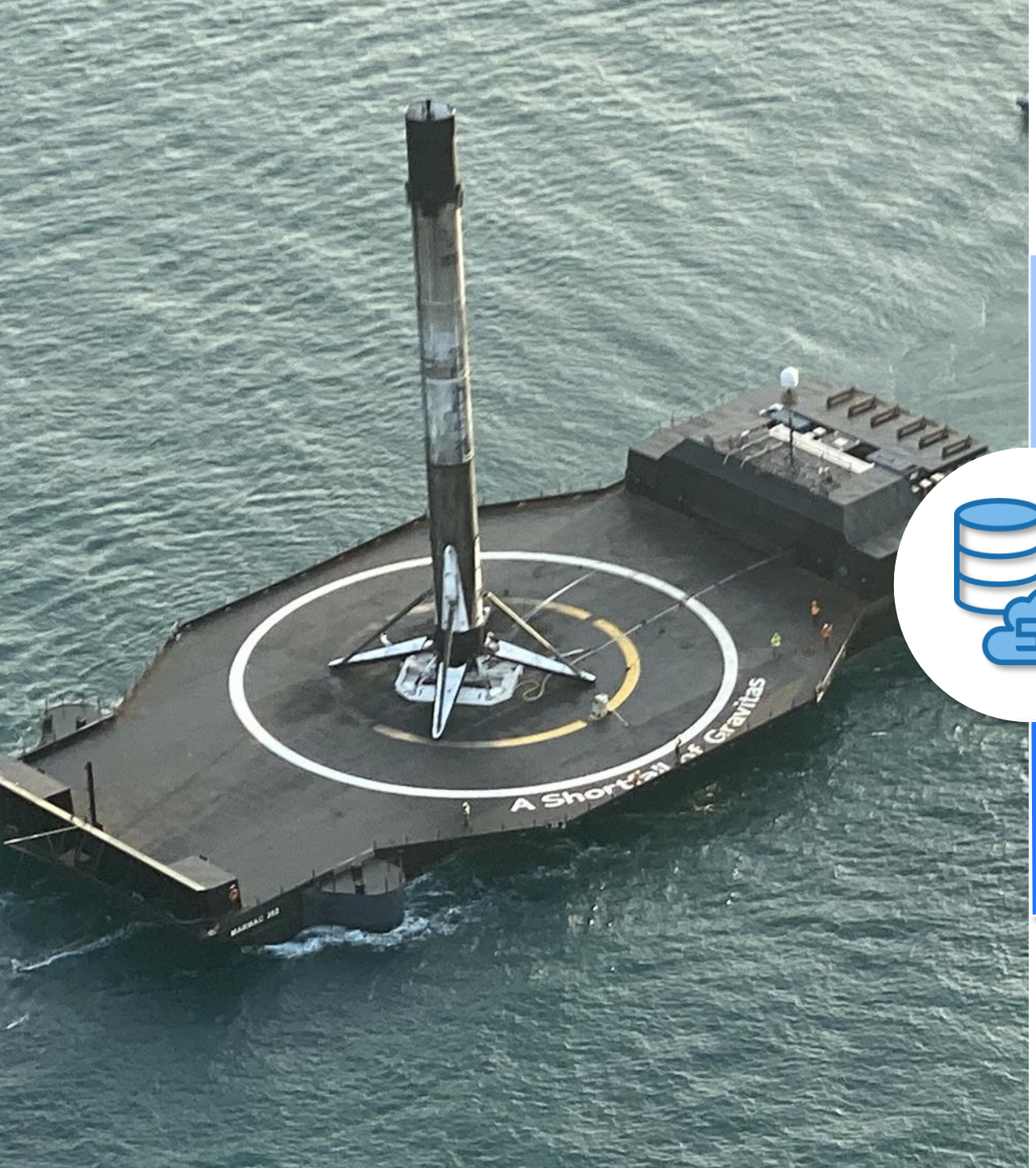
Using ‘min’, ‘from’ and ‘where’ in the query we derive from table SPACEXDATASET the date when the first successful landing outcome in ground pad was acheived




```
%sql select min(DATE) from SPACEXDATASET \
where Landing__Outcome = 'Success (ground pad)';
```

1
2015-12-22





Successful Drone Ship Landing

Explanation

Using ‘from’, ‘where’ ‘and … and’ in the query we derive from table SPACEXDATASET the names of the boosters which have success in drone ship and have payload mass greater than 4000 but less than 6000

SQL Query

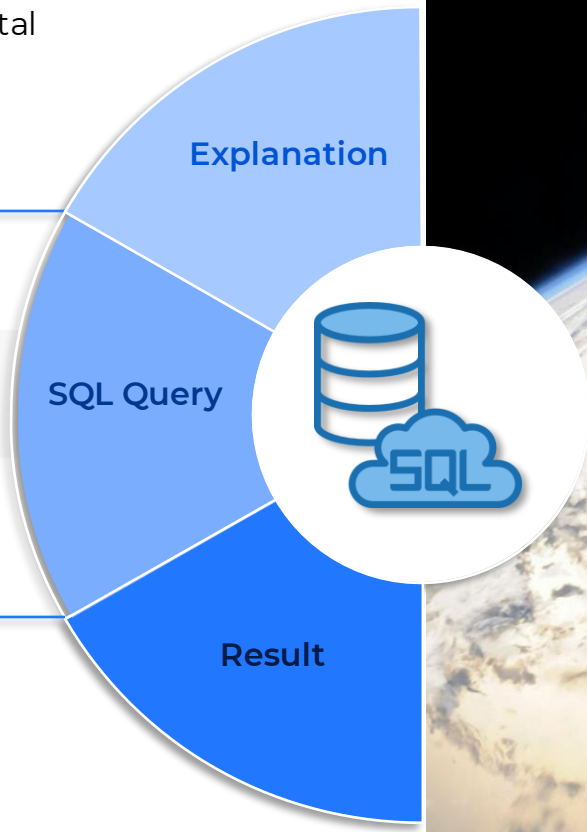
```
%sql select BOOSTER_VERSION from SPACEXDATASET \
where Landing__Outcome = 'Success (drone ship)' \
and PAYLOAD_MASS__KG_ > 4000 \
and PAYLOAD_MASS__KG_ < 6000;
```

Result

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

Mission Outcomes

Using ‘count’, ‘from’, and ‘where’ in the query we pull from table SPACEXDATASET the total number of successful and failure mission outcomes




```
%sql select count(MISSION_OUTCOME) \
from SPACEXDATASET \
where MISSION_OUTCOME = 'Success' \
or MISSION_OUTCOME = 'Failure (in flight)';
```

1
100

Maximum booster payload

Explanation

SQL Query

Result

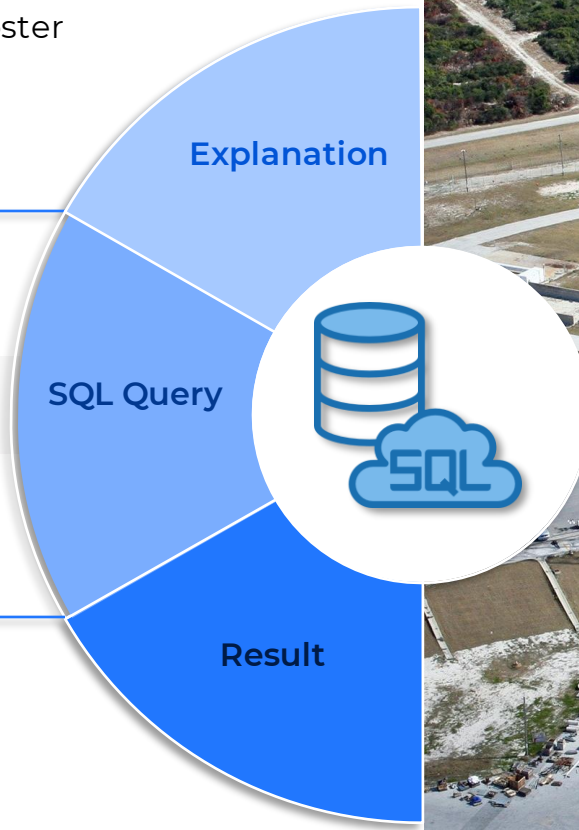
Using subquery, ‘from’ and ‘max’ in the query we fetch from table SPACEXDATASET the names of the BOOSTER_VERSION which have carried the maximum payload mass

```
%sql select BOOSTER_VERSION from SPACEXDATASET \
where PAYLOAD_MASS__KG_ = (select \
                           max(PAYLOAD_MASS__KG_) \
                           from SPACEXDATASET);
```

booster_version	
F9 B5 B1048.4	F9 B5 B1049.5
F9 B5 B1049.4	F9 B5 B1060.2
F9 B5 B1051.3	F9 B5 B1058.3
F9 B5 B1056.4	F9 B5 B1051.6
F9 B5 B1048.5	F9 B5 B1060.3
F9 B5 B1051.4	F9 B5 B1049.7

2015 Launch Records

Using ‘from’ and ‘where’ in the query we have from table SPACEXDATASET the failed landing_outcomes in drone ship, their booster versions, and launch site names for in year 2015




```
%sql select MONTH(DATE),MISSION_OUTCOME,\
BOOSTER_VERSION,LAUNCH_SITE from SPACEXDATASET \
where EXTRACT(YEAR FROM DATE)='2015';
```

1	mission_outcome	booster_version	launch_site
1	Success	F9 v1.1 B1012	CCAFS LC-40
2	Success	F9 v1.1 B1013	CCAFS LC-40
3	Success	F9 v1.1 B1014	CCAFS LC-40
4	Success	F9 v1.1 B1015	CCAFS LC-40
4	Success	F9 v1.1 B1016	CCAFS LC-40
6	Failure (in flight)	F9 v1.1 B1018	CCAFS LC-40
12	Success	F9 FT B1019	CCAFS LC-40

Rank Landing Outcomes

Explanation

Using ‘from’ and ‘where’ in the query we get Rank the count of landing outcomes (such as Failure (drone ship) or Success (ground pad)) between the date 2010-06-04 and 2017-03-20, in descending order from table SPACEXDATASET

SQL Query

```
%sql select LANDING__OUTCOME from SPACEXDATASET \
where DATE between '2010-06-04' \
and '2017-03-20' order by DATE DESC;
```

Result

landing__outcome			
	No attempt	Success (drone ship)	Controlled (ocean)
No attempt	Success (ground pad)	Failure (drone ship)	Failure (drone ship)
Uncontrolled (ocean)	Success (drone ship)	Failure (drone ship)	Uncontrolled (ocean)
No attempt	Success (drone ship)	Success (ground pad)	No attempt
No attempt	Success (ground pad)	Precluded (drone ship)	No attempt
No attempt	Failure (drone ship)	No attempt	Controlled (ocean)
Failure (parachute)	Success (drone ship)	Failure (drone ship)	Controlled (ocean)
Failure (parachute)	Success (drone ship)	No attempt	No attempt


Section 3
Launch Sites
Proximities Analysis

All launch Sites on Folium Map

We can see that the SpaceX launch sites are near to the United States of America coasts i.e., Florida and California Regions

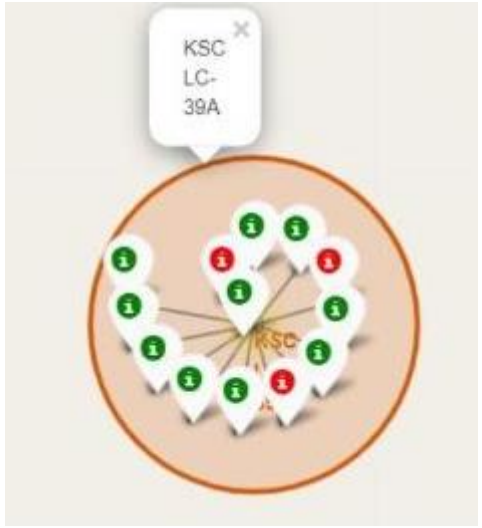
Color-coded start records

We can see on these screenshots, that the KSC LC-39A has the highest success rate

Green Markers shows us successful launches

Red Markers shows us failures launches

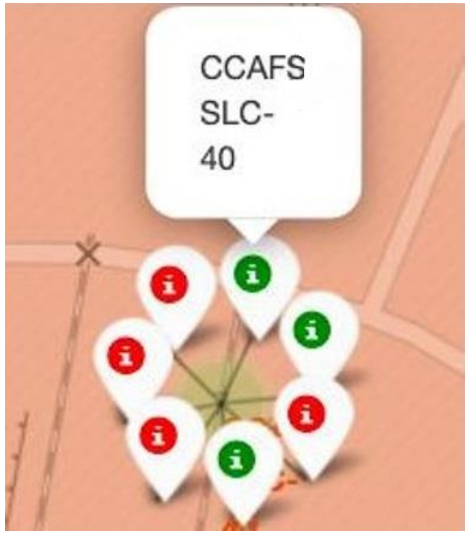
KSC LC-39A



CCAFS LC-40

VAFB SLC-4E

CCAFS SLC-40

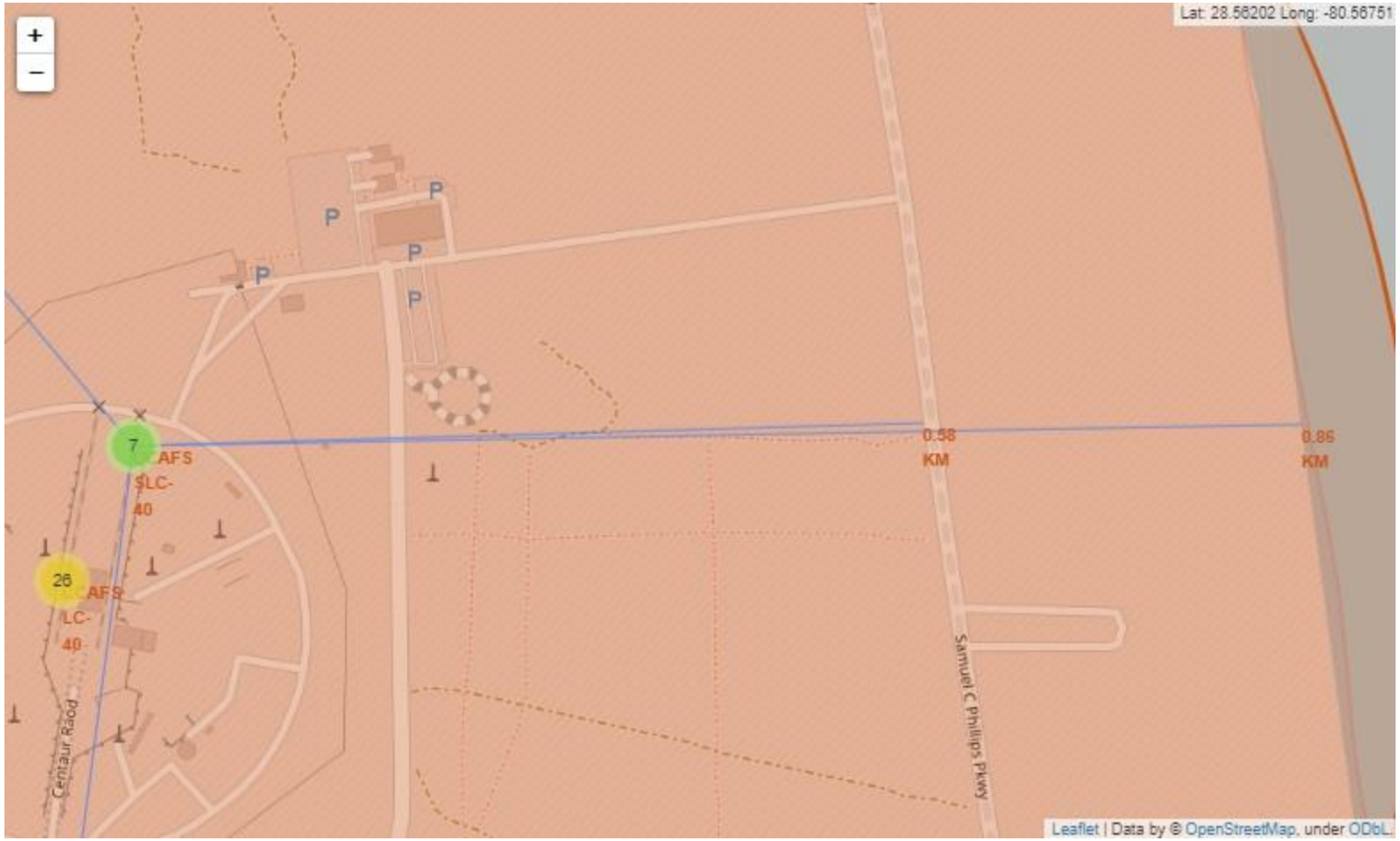


Distance from the launch Sites to the coastline

Launch sites are near the coastline

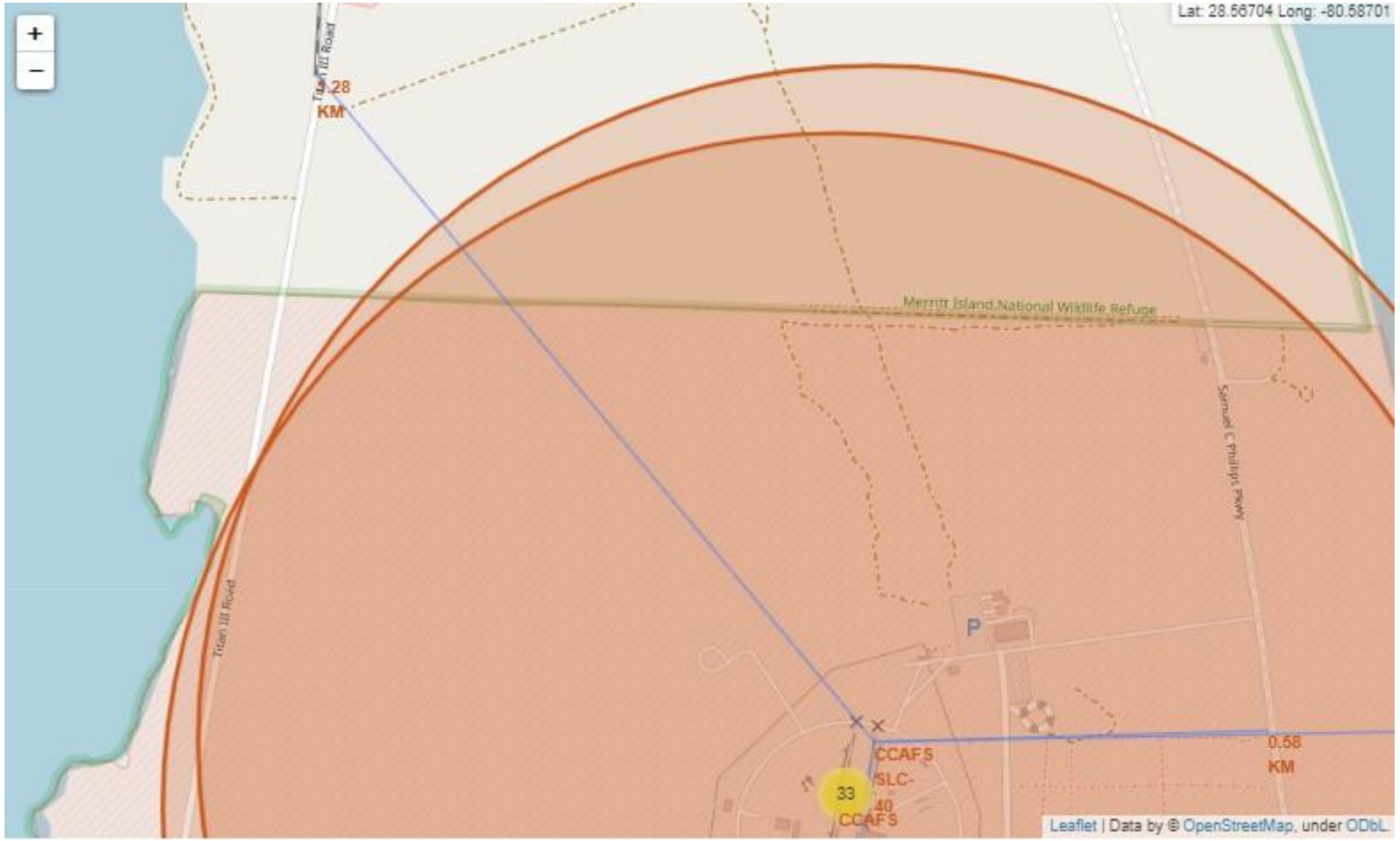
so, they can fly over the ocean during launch for two safety reasons:

- the crew can abort the launch and try to land on the water
- minimize the risk of debris falling on people and buildings



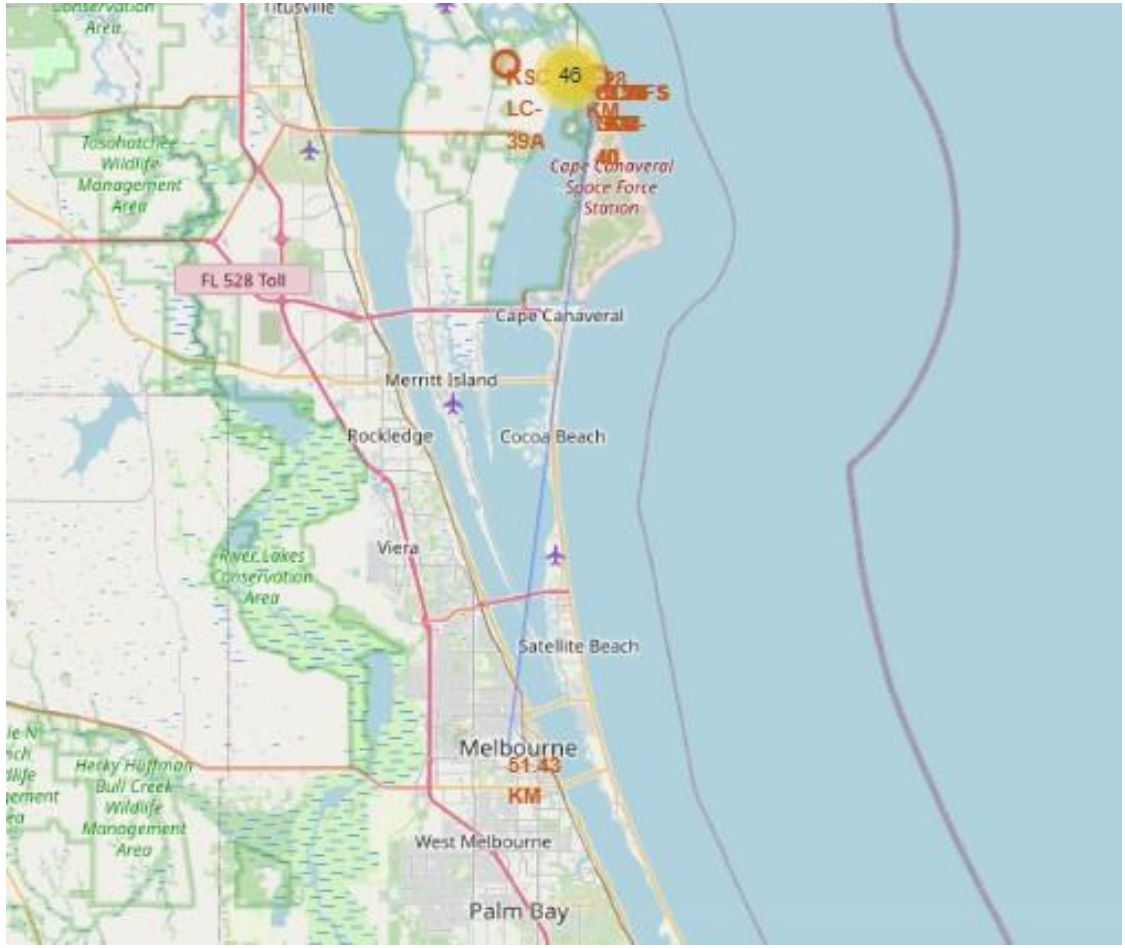
Distance from the launch Sites to the highways

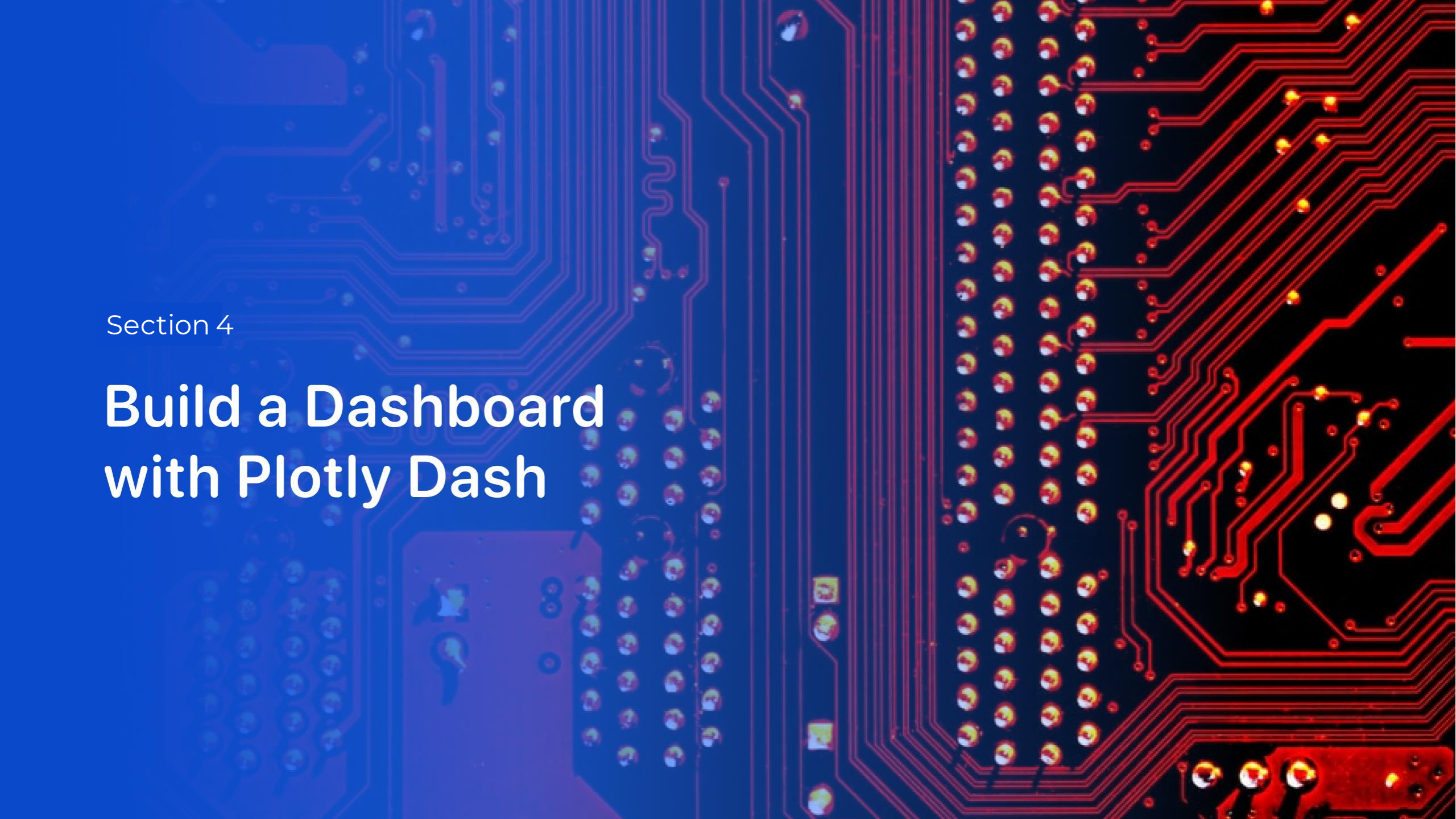
The launch sites are near highways, which makes it easy to transport the necessary people and equipment



Distance from the launch Sites to the cities

Launch sites are not near cities, but not in the cities themselves, minimizing the danger to densely populated areas



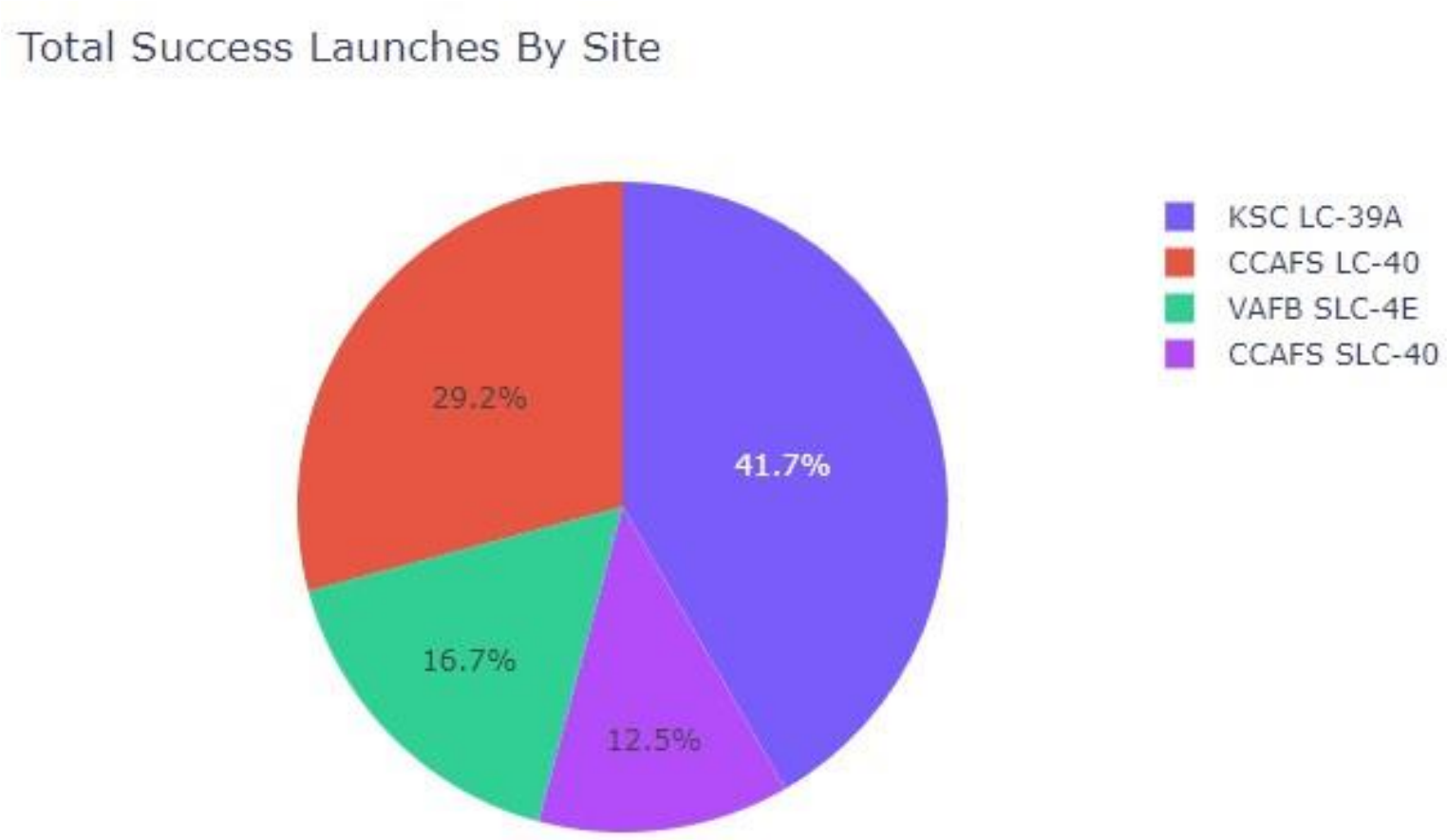


Section 4

Build a Dashboard with Plotly Dash

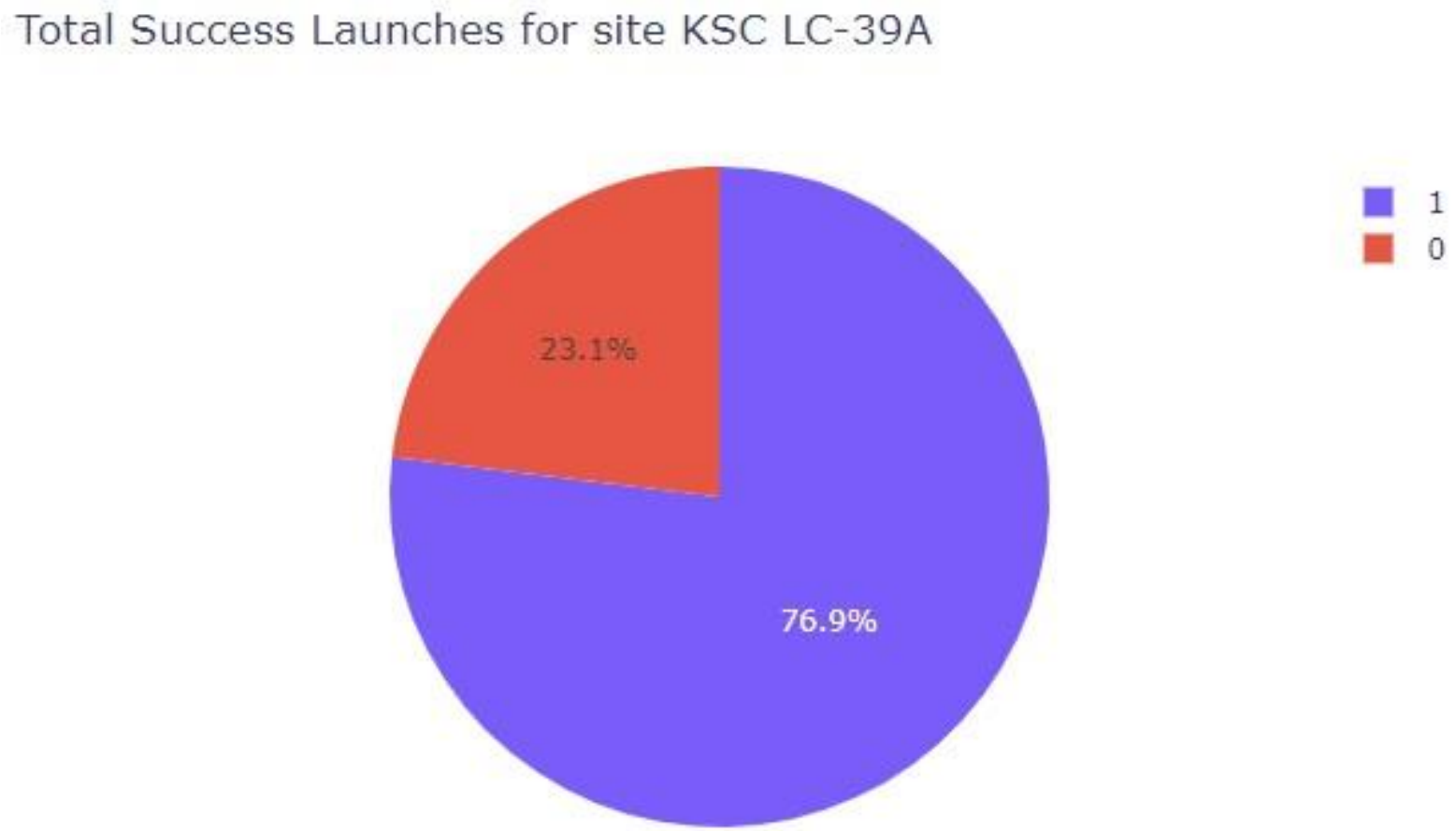
The number of successful launches for all sites

This piechart shows that the KSC LC-39A has the most successful launches from all sites - 41.7%



The launch site with highest launch success ratio

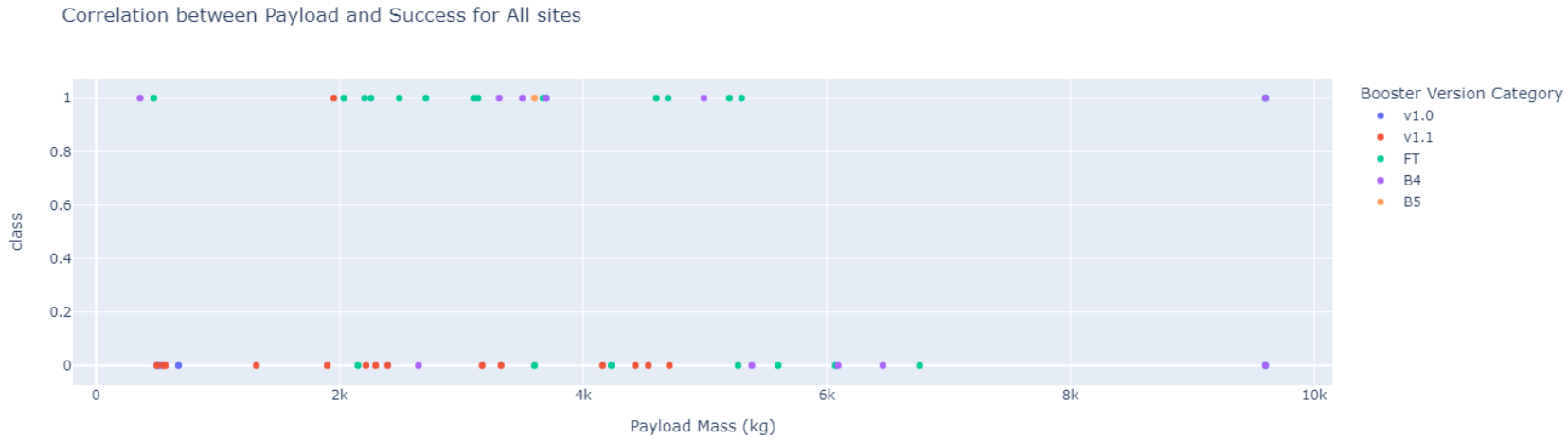
On this piechart we can see that KSC LC-39A achieved a success rate of 76.9% with a failure rate of 23.1%

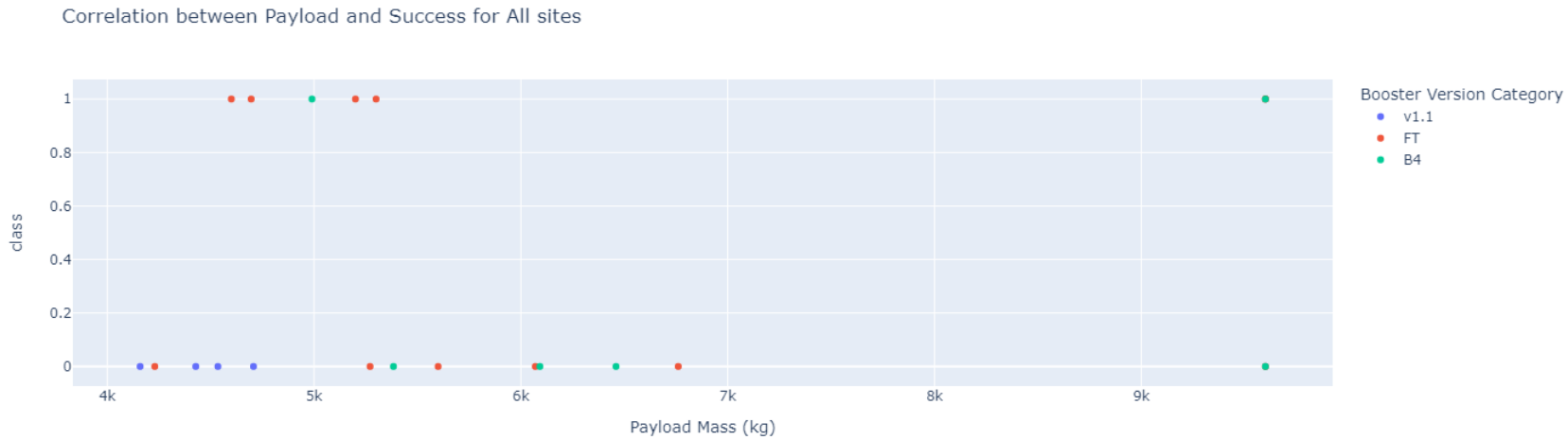


Payload vs. Launch Outcome scatter plot for all sites

We can see that the success rates for weighted average payloads (1952 kg - 5300 kg) are higher than for very light payloads (<1952 kg) or heavier payloads (>5300 kg)

Also, these scatterplots show us that F9 Booster version FT has the highest launch success rate





Section 5

Predictive Analysis (Classification)

Classification Accuracy

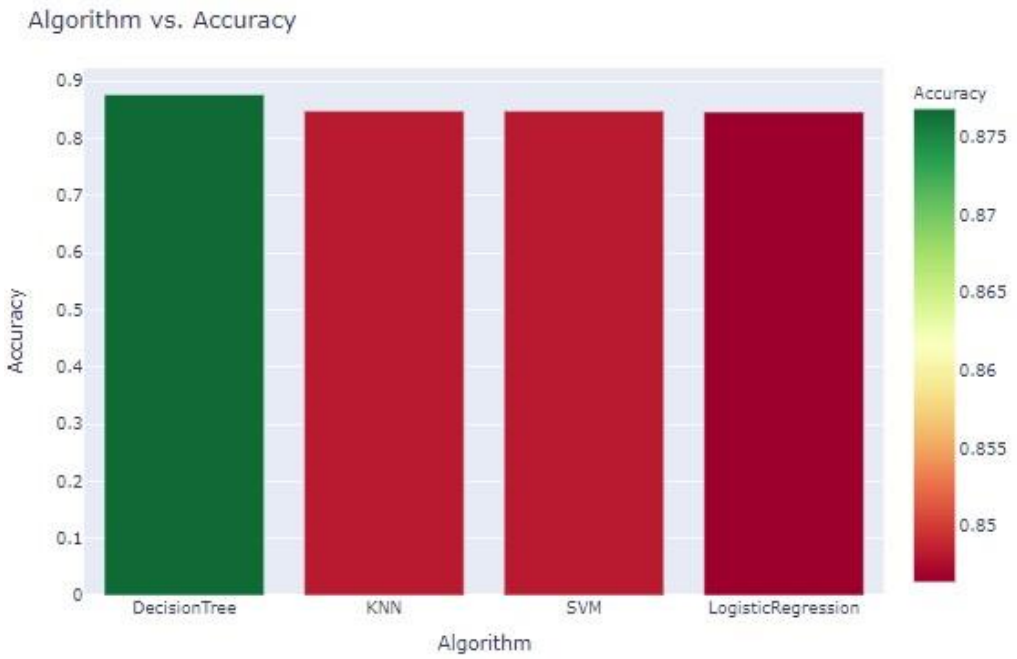
We trained four different models, each with 83% accuracy

Algorithm	Accuracy	Accuracy on Test Data	Tuned Hyperparameters
Logistic regression	0.846429	0.833333	{'C': 0.01, 'penalty': 'l2', 'solver': 'lbfgs'}
SVM	0.848214	0.833333	{'C': 1.0, 'gamma': 0.03162277660168379, 'kernel': 'sigmoid'}
Decision Tree	0.876786	0.833333	{'criterion': 'gini', 'max_depth': 4, 'max_features': 'auto', 'min_samples_leaf': 2, 'min_samples_split': 2, 'splitter': 'random'}
KNN	0.848214	0.833333	{'algorithm': 'auto', 'n_neighbors': 10, 'p': 1}

We can see that our accuracy is very close, but we have a clear winner with the best score - Decision Tree with a score of 0.876786

Let's build a histogram Algorithm vs. Accuracy

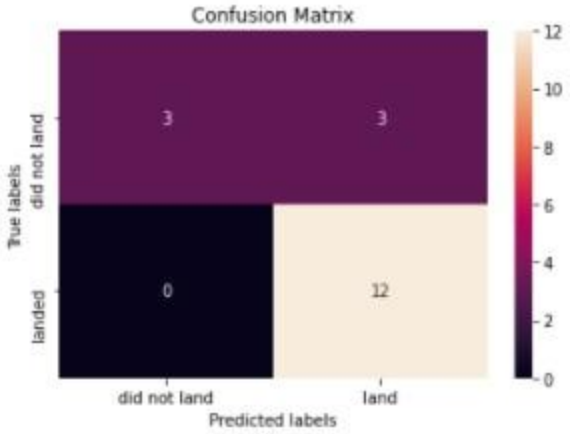
```
fig = px.bar(algo_df, x='Algorithm', y='Accuracy',
             hover_data=['Algorithm', 'Accuracy'],
             color='Accuracy', color_continuous_scale='rdylgn')
fig.update_layout(title='Algorithm vs. Accuracy',
                  xaxis_title='Algorithm',
                  yaxis_title='Accuracy' )
```



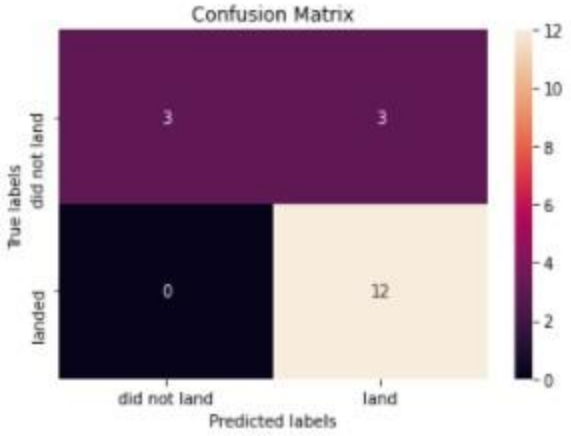
Confusion Matrix

As you can see, when plotting, we got the same confusion matrixes for all models

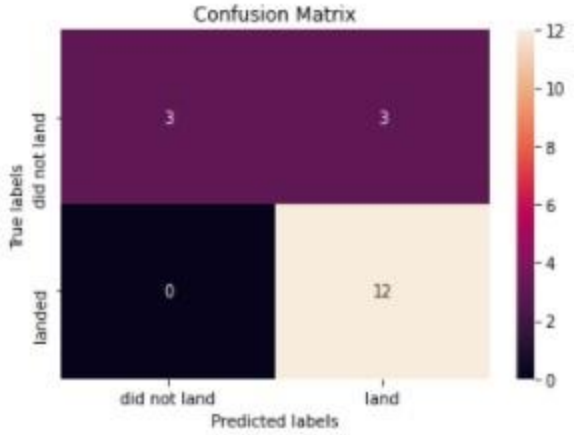
Decision Tree



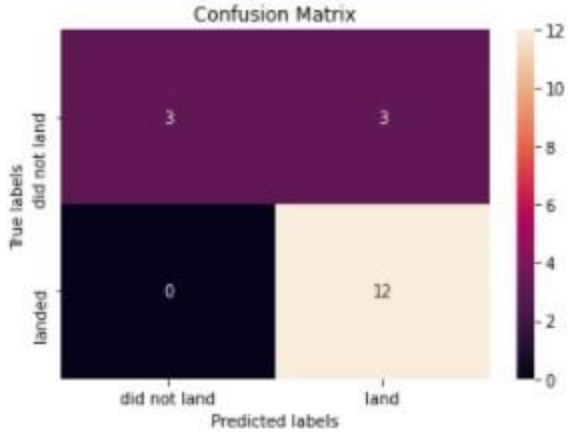
SVM



KNN



Logistic regression



Conclusions

- The greater the number of launches (more than 40), the higher the success rate for the rocket
- We see that the success rates for a weighted average payload (1952kg - 5300kg) are higher than for a very light payload (<1952kg) or a heavier payload (>5300kg)
- ES-L1, GEO, HEO, SSO orbits have the highest success rates
- In LEO orbit, success appears as a function of the number of launches; on the other hand, there seems to be no connection between the GTO flight number
- With heavy payloads, landing success or positive landing speed is greater for Polar, LEO and ISS orbits
- In addition, the proximity of the equator line, coastline, highways has a positive effect on the probability of a successful rocket launch
- SpaceX launch success rates are relatively increasing over time, and it looks like they will hit the required target soon

Thank you!