IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

Kirill
Kryzhanovskiy

Outline

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

Introduction

Project background and context

In this project, we predicted if the Falcon 9 first stage will land successfully. SpaceX company advertises its Falcon 9 rocket, whose launch costs significantly less compared to other providers. The key to this achievement is reusing the first stage of the racket. So, if we can predict if the first stage will land successfully or not, we can estimate the cost of the lauch correctly.

Problems you want to find answers

- Determining the factors affecting the first stage’s landing success rate
- Finding insights in the launches data, provided by SpaceX
- Applying Machine Learning algorithms to predict if the first stage of the rocket will land successfully

Section 1

Methodology

Methodology

Executive Summary

- Data collection methodology:
  - Request to the SpaceX API
- Perform data wrangling
  - Python (Pandas and NumPy libraries)
- Perform exploratory data analysis (EDA) using visualization and SQL
- Perform interactive visual analytics using Folium and Plotly Dash
- Perform predictive analysis using classification models
  - Preparing the data, split the data to train and test sets, apply GridSearch to determine optimal parameters, train the model with train data, evaluate the effectiveness of the model on the test data

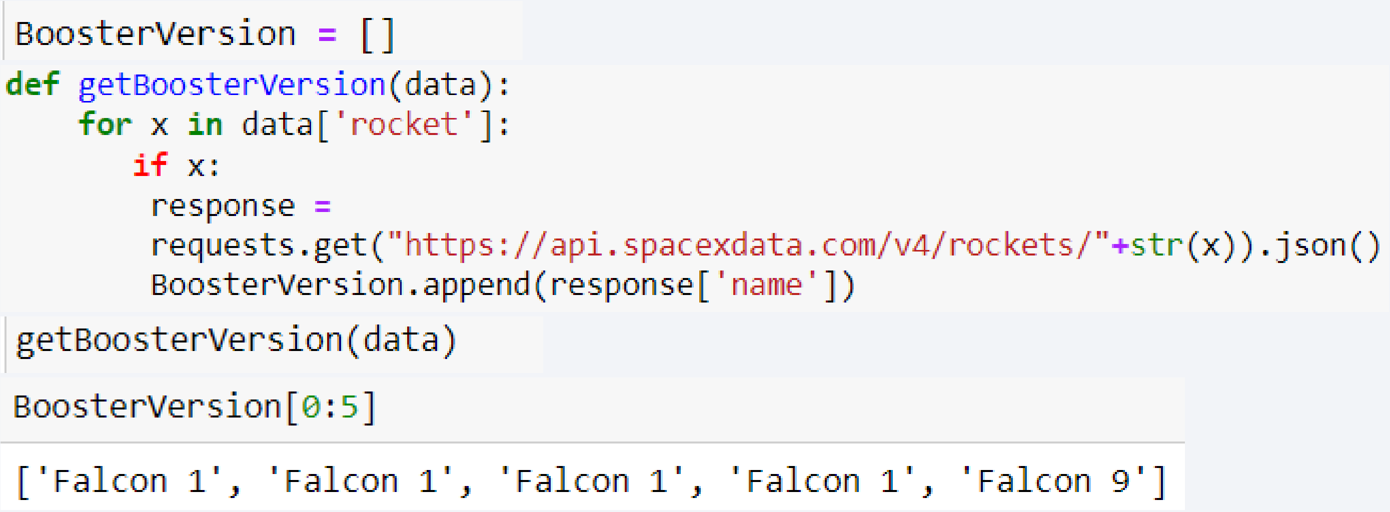
Data Collection

The data was collected using Python (Jypiter Notebooks) as follows:

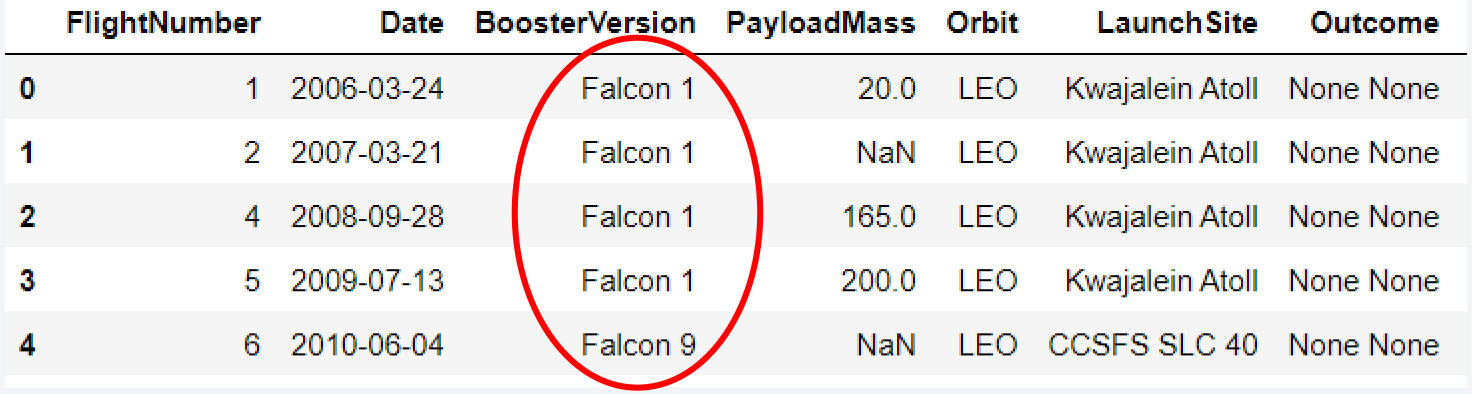
1) Request rocket launch data using request.get(spacex_url)
2) From the Json response note the ID’s of launches
3) Use API and collected IDs to get information about launches
4) Convert the obtained data to Pandas DataFrame
5) Filter the data to keep only Falcon 9 launches, substitute null values with the mean in Payload Mass column

Data Collection - SpaceX API

Process to extract the booster name using API (same for other parameters)



```
BoosterVersion = []
def getBoosterVersion(data):
    for x in data['rocket']:
       if x:
        response =
        requests.get("https://api.spacexdata.com/v4/rockets/"+str(x)).json()
        BoosterVersion.append(response['name'])
getBoosterVersion(data)
BoosterVersion[0:5]
['Falcon 1', 'Falcon 1', 'Falcon 1', 'Falcon 1', 'Falcon 9']
```



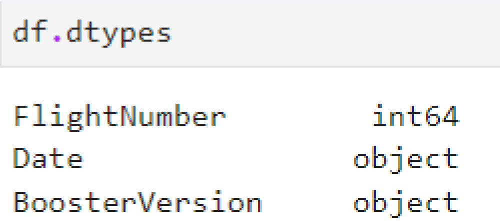
	FlightNumber	Date	BoosterVersion	PayloadMass	Orbit	LaunchSite	Outcome
0	1	2006-03-24	Falcon 1	20.0	LEO	Kwajalein Atoll	None None
1	2	2007-03-21	Falcon 1	NaN	LEO	Kwajalein Atoll	None None
2	4	2008-09-28	Falcon 1	165.0	LEO	Kwajalein Atoll	None None
3	5	2009-07-13	Falcon 1	200.0	LEO	Kwajalein Atoll	None None
4	6	2010-06-04	Falcon 9	NaN	LEO	CCSFS SLC 40	None None

LINK

to the Jupyter notebook with Data Collection process

Data Wrangling

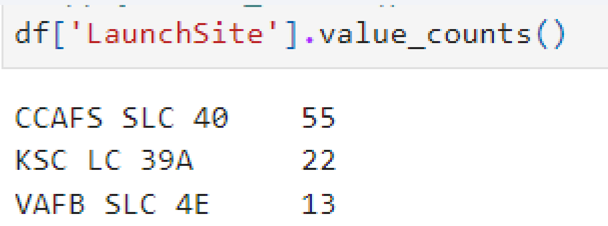
1) Checking the correctness of types of data using dtypes



```
df.dtypes
```

```
FlightNumber        int64
Date               object
BoosterVersion     object
```

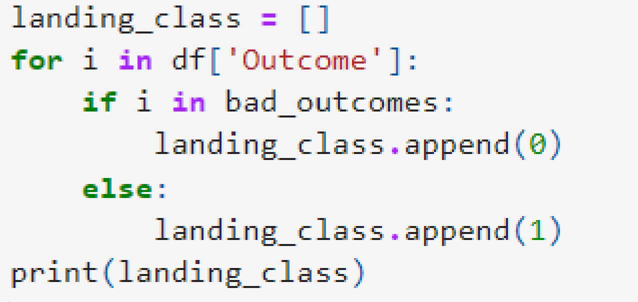
2) Checking for the unwanted content in categorical variables using value_counts()



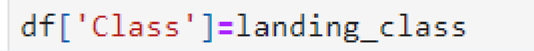
```
df['LaunchSite'].value_counts()
```

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

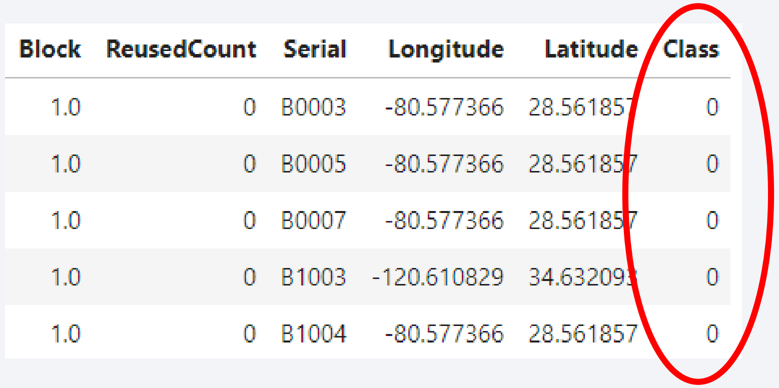
3) Adding target parameter – landing_class (1 – success, 0 – failure)



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



```
df['Class']=landing_class
```

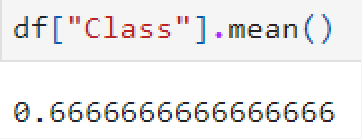


Block	ReusedCount	Serial	Longitude	Latitude	Class
1.0	0	B0003	-80.577366	28.561857	0
1.0	0	B0005	-80.577366	28.561857	0
1.0	0	B0007	-80.577366	28.561857	0
1.0	0	B1003	-120.610829	34.632093	0
1.0	0	B1004	-80.577366	28.561857	0

LINK

to the Jupyter notebook with Data Wrangling process

Success rate



```
df["Class"].mean()
```

```
0.6666666666666666
```

EDA with Data Visualization

In order to visualize the effect that different parameters have on the Falcon 9 landing success, the following visualizations were plotted using the Seaborn and matplotlib libraries in Python:

1) **Scatter Plots** to show how the landing success depends on: Launch site, Payload mass, Orbit

2) **Bar chart** to show the success rate for different orbits

3) **Line chart** to show how the success rate of rocket landing changed with time

LINK

to the Jupyter notebook with Exploratory Data Analysis using Vizualizations

EDA with SQL

SQL queries were written to find out the following:

- Display the names of the unique launch sites in the space mission
- Display 5 records where launch sites begin with the string 'CCA'
- Display the total payload mass carried by boosters launched by NASA (CRS)
- Display average payload mass carried by booster version F9 v1.1
- List the date when the first successful landing outcome in ground pad was acheived.
- List the names of the boosters which have success in drone ship and have payload mass greater than 4000 but less than 6000
- List the total number of successful and failure mission outcomes
- List the names of the booster_versions which have carried the maximum payload mass. Use a subquery
- List the failed landing_outcomes in drone ship, their booster versions, and launch site names for in year 2015
- 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

LINK

to the Jupyter notebook with Exploratory Data Analysis with SQL

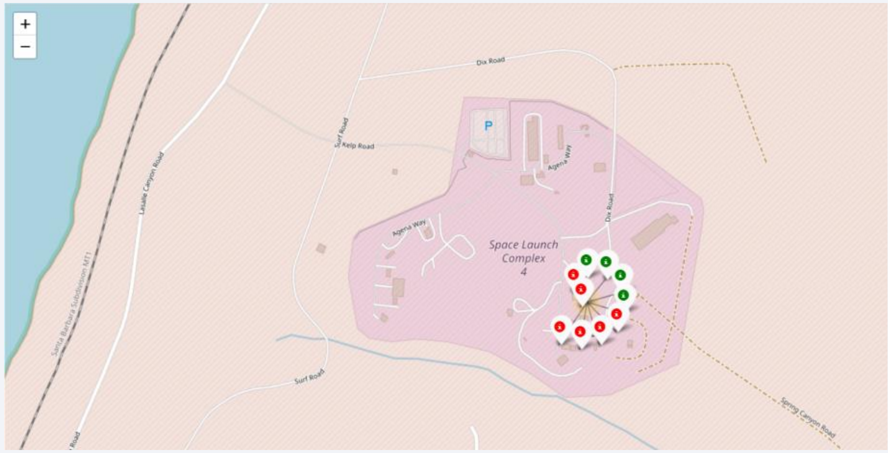
Build an Interactive Map with Folium

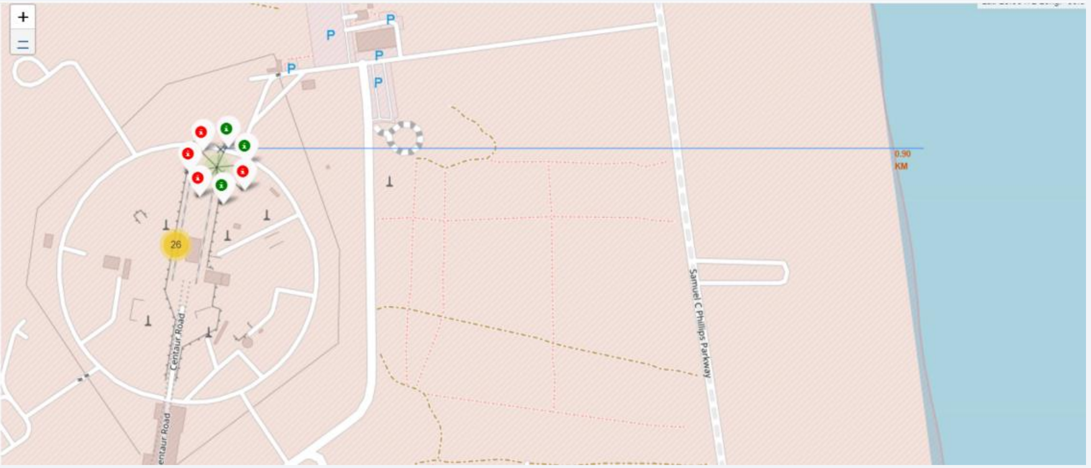
In order to study the locations of Falcon 9 Launch sites, Folium library in Python was applied:

- Markers were used to show the locations of Launch Sites and their names
- Marker clusters were used to show how many successful and not successful attempts were done fro each Launch Site
- Lines were applied to show the distances to the closest coastlines, highways, etc.

LINK

to the Jupyter notebook with Interactive Map in Folium





Predictive Analysis (Classification)

- Different classification algorithms (Logistic Regression, Decision Tree, K Nearest Neighbors, Support Vector Machine) were implemented to predict if the first stage of Falcon 9 will land successfully.
- Implemented libraries in python: Pandas, Scikit-Learn, Numpy
- The GridSearch was implemented to find out the parameters for each algorithm which allow to maximize the accuracy of the model
- The data collected and prepared in the earlier steps was split to train and test sets
- Each algorithm was trained using the train data and its efficiency of prediction was evaluated using the test data

LINK

to the Jupyter notebook with
Predictive Analysis (classification)

Results

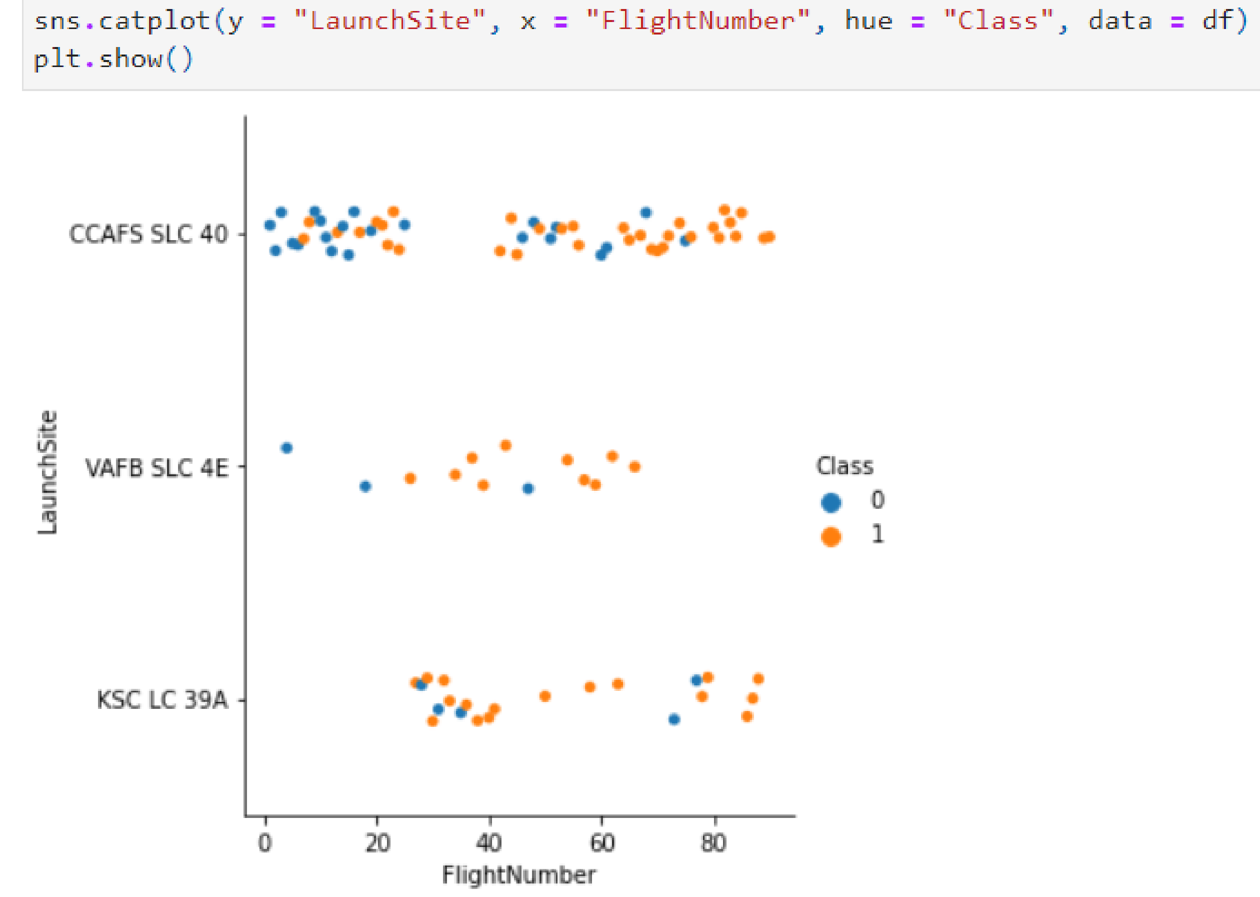
- Exploratory data analysis results
- Interactive analytics demo in screenshots
- Predictive analysis results

Section 2

Insights drawn from EDA

Flight Number vs. Launch Site

- Scatter plot of Flight Number vs. Launch Site



```
sns.catplot(y = "LaunchSite", x = "FlightNumber", hue = "Class", data = df)
plt.show()
```

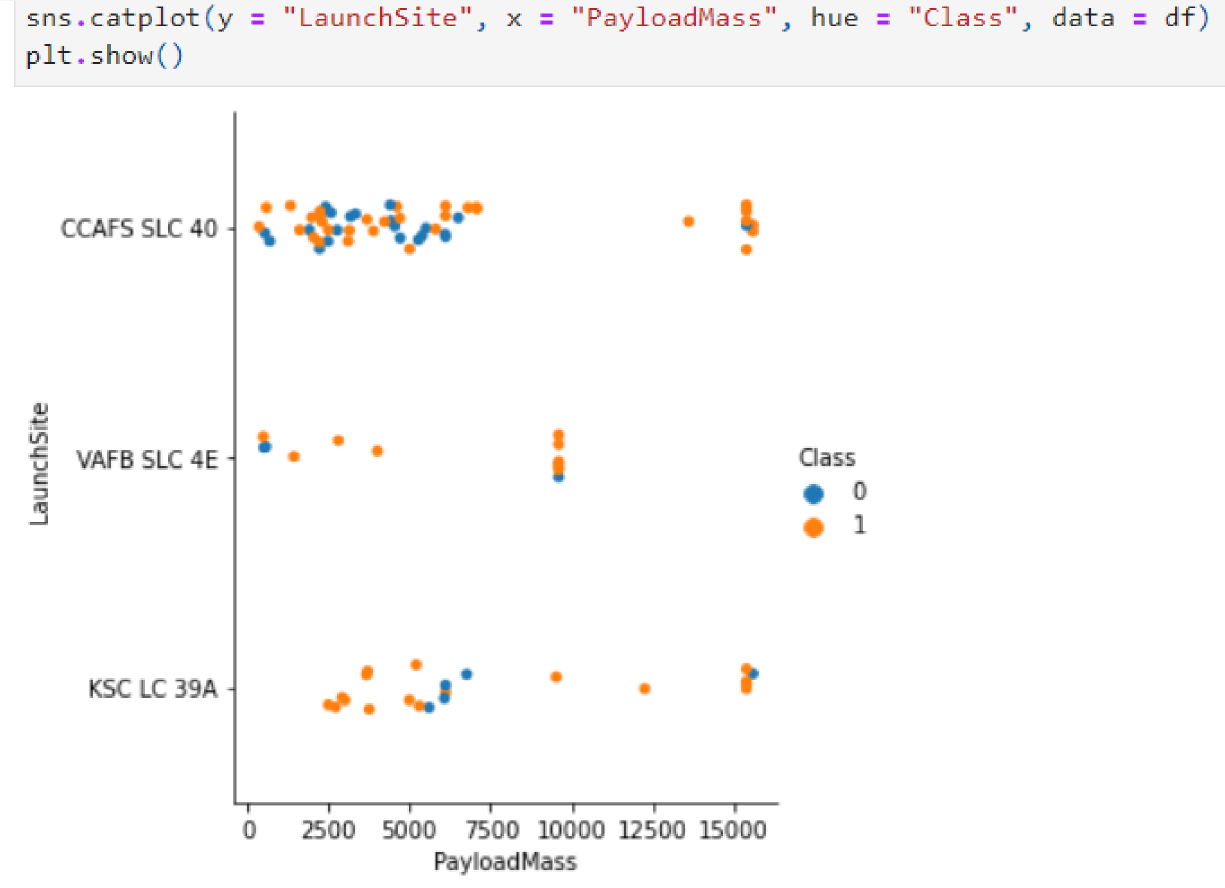
Class 0 – unsuccessful
Class 1 – successful

It can be noticed, that generally the success rate for all Launch sites increased for later launches.

This trend is the most significant for the CCAFS SLC 40 Launch Site

Payload vs. Launch Site

- Scatter plot of Payload vs. Launch Site



```
sns.catplot(y = "LaunchSite", x = "PayloadMass", hue = "Class", data = df)
plt.show()
```

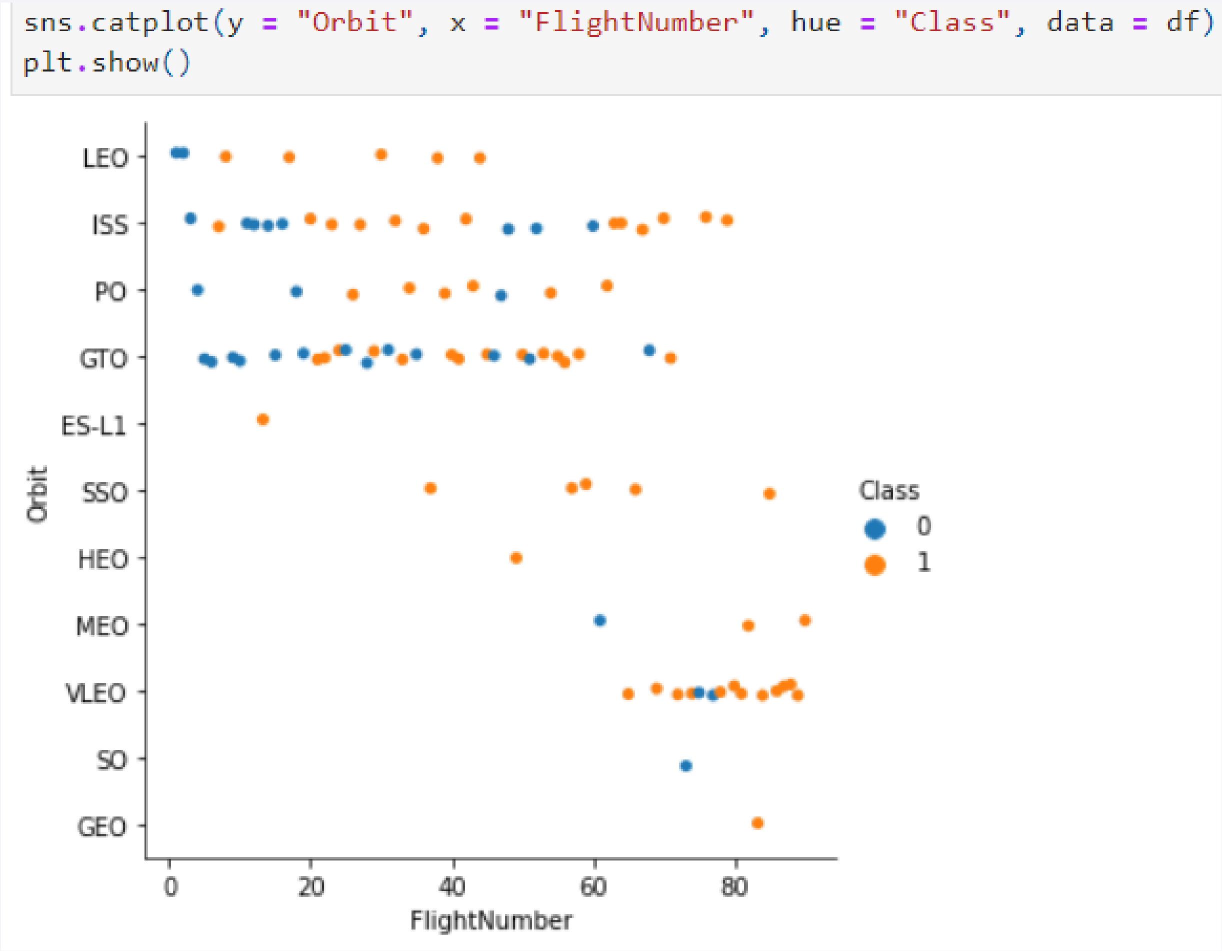
Class 0 – unsuccessful
Class 1 – successful

It can be noticed, that generally the success rate for higher payload mass is higher.

This trend is the most significant for the CCAFS SLC 40 Launch Site, where almost all launches were successful which had more than 10,000 kg payload mass

Success Rate vs. Orbit Type

- Bar chart for the success rate of each orbit type



```
sns.catplot(y = "Orbit", x = "FlightNumber", hue = "Class", data = df)
plt.show()
```

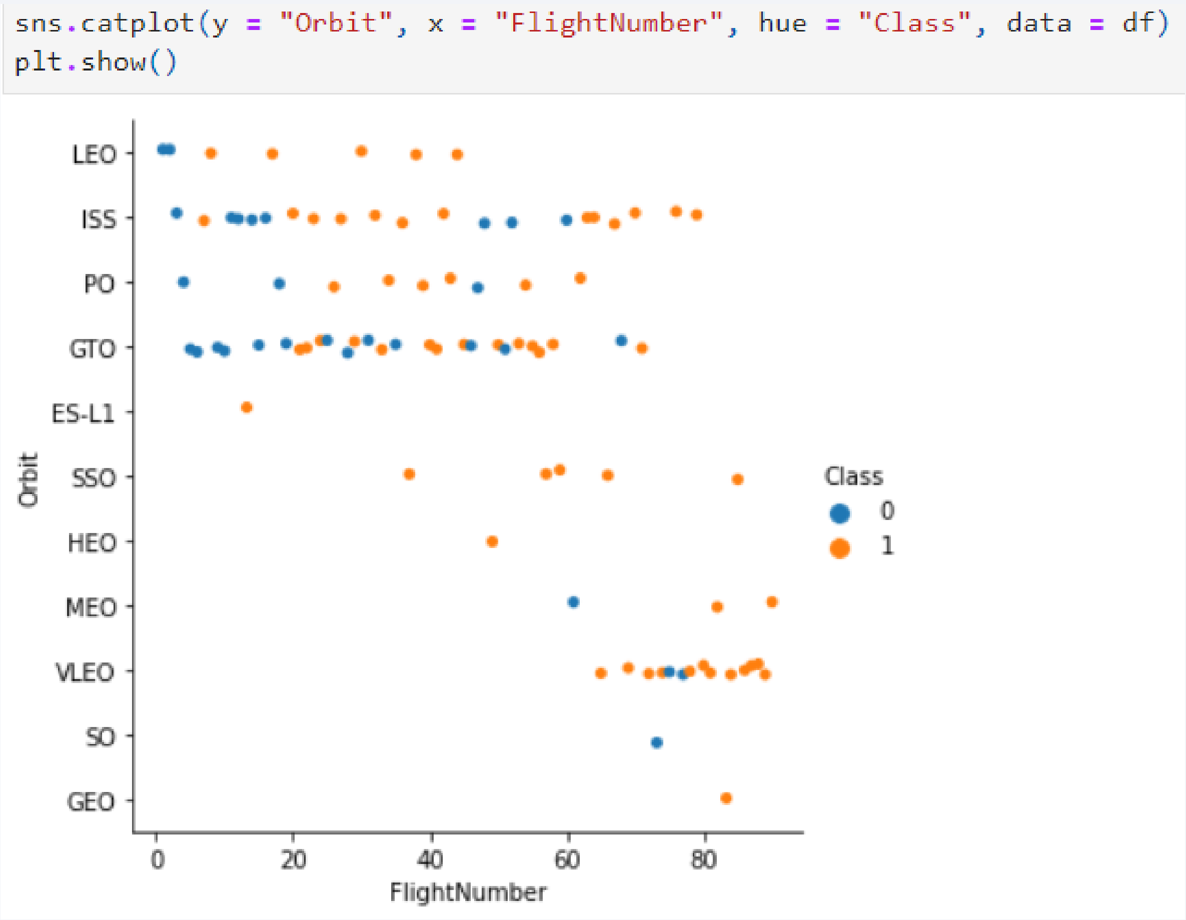
The bar chart shows that success rate was 100% for the following orbit types:

- ES-L1, GEO, HEO, SSO

The most problematic orbit with the lowest success rate was GTO (50%)

Flight Number vs. Orbit Type

- Scatter plot of Flight number vs. Orbit type



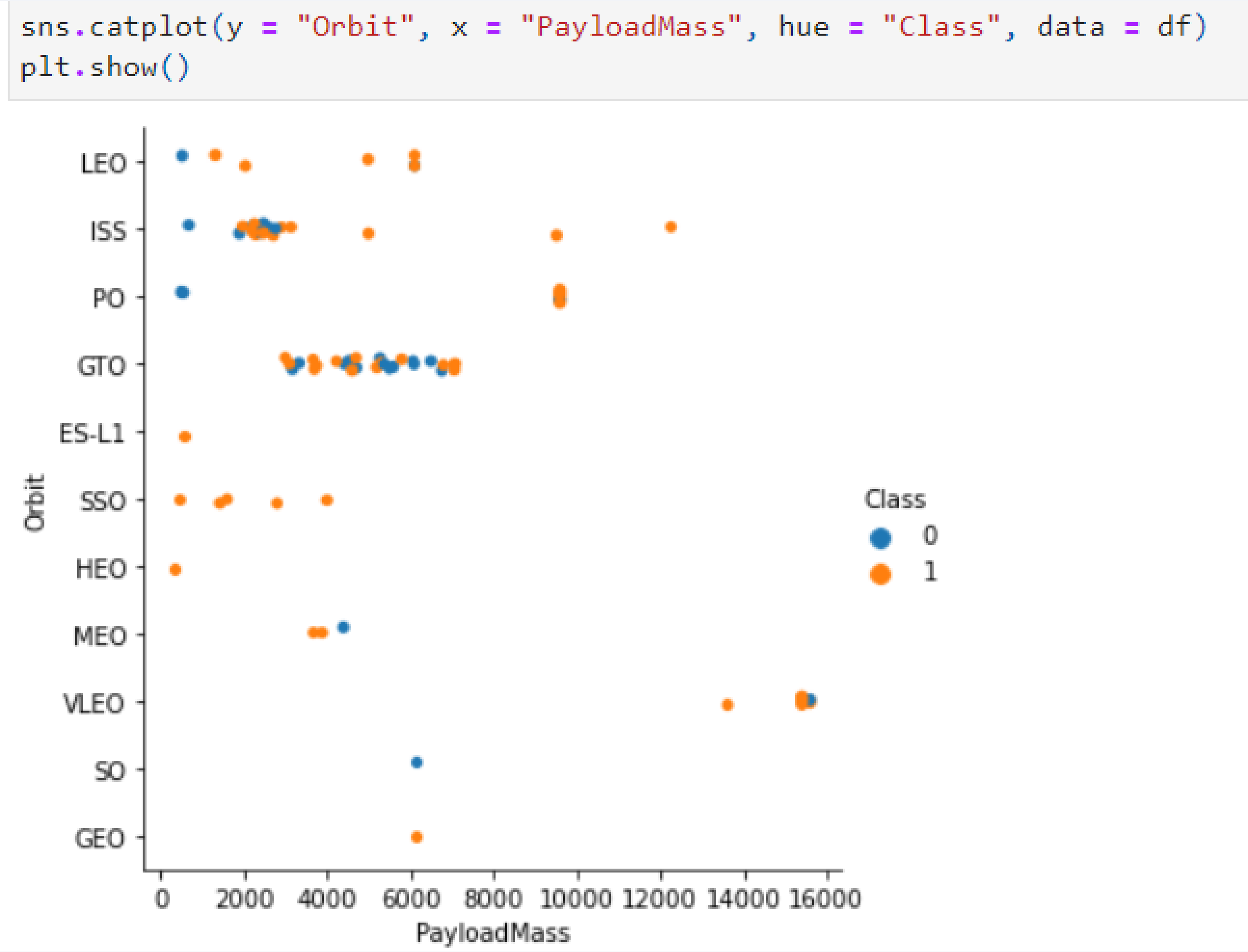
```
sns.catplot(y = "Orbit", x = "FlightNumber", hue = "Class", data = df)
plt.show()
```

Class 0 – unsuccessful
Class 1 – successful

It can be seen that for LEO, PO, VLEO orbits success rate is increasing with more attempts,
While for other orbits there is no such trend.

Payload vs. Orbit Type

- Scatter plot of payload vs. orbit type



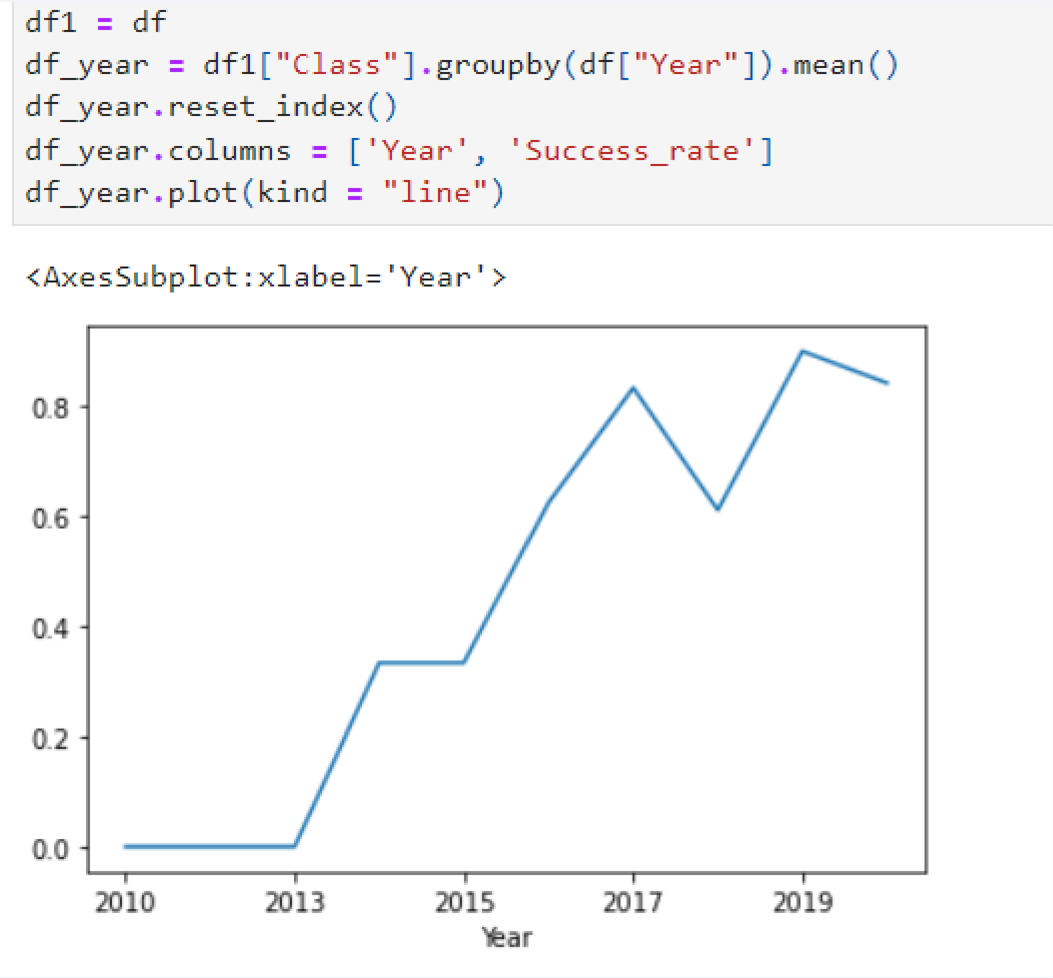
```
sns.catplot(y = "Orbit", x = "PayloadMass", hue = "Class", data = df)
plt.show()
```

Class 0 – unsuccessful
Class 1 – successful

For some orbits (LEO, ISS, PO) we can notice, that for the larger payload mass the success rate is higher, while for others there is no such trend.

Launch Success Yearly Trend

- Show a line chart of yearly average success rate



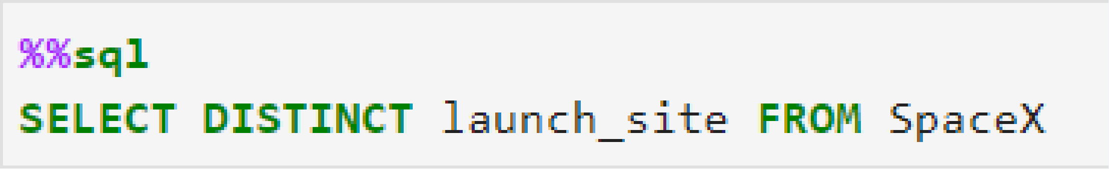
```
df1 = df
df_year = df1["Class"].groupby(df["Year"]).mean()
df_year.reset_index()
df_year.columns = ['Year', 'Success_rate']
df_year.plot(kind = "line")
```

```
<AxesSubplot:xlabel='Year'>
```

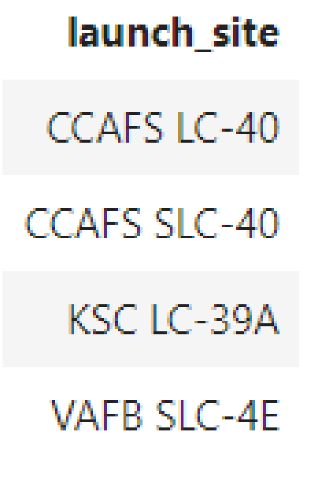
We can see the obvious trend of improvement of the success rate with time.

All Launch Site Names

It can be seen, that 4 different launch sites were used for Falcon 9 launches



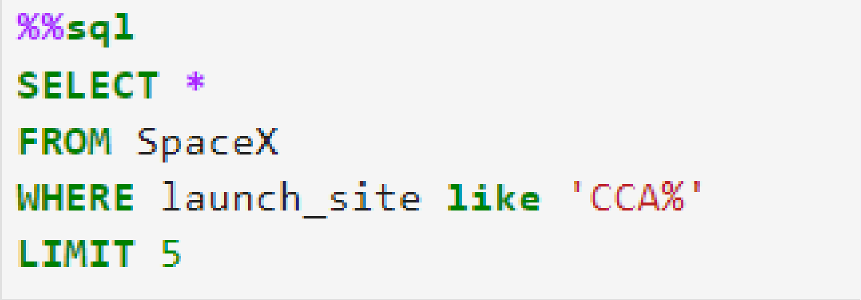
```
%%sql
SELECT DISTINCT launch_site FROM SpaceX
```



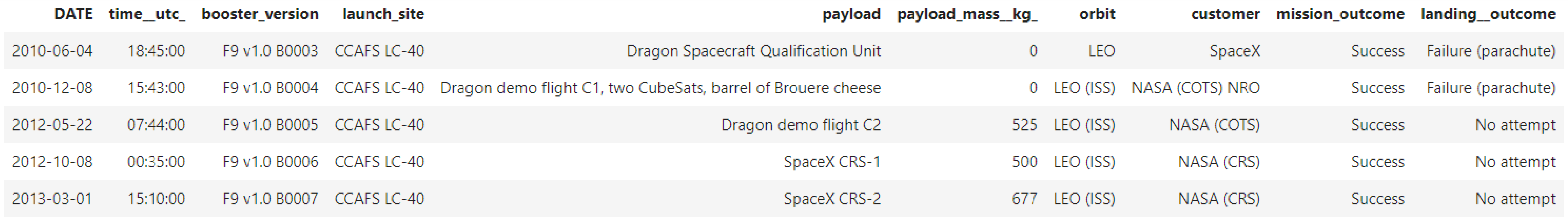
launch_site
CCAFS LC-40
CCAFS SLC-40
KSC LC-39A
VAFB SLC-4E

Launch Site Names Begin with 'CCA'

The records for the launch sites starting from ‘CCA’



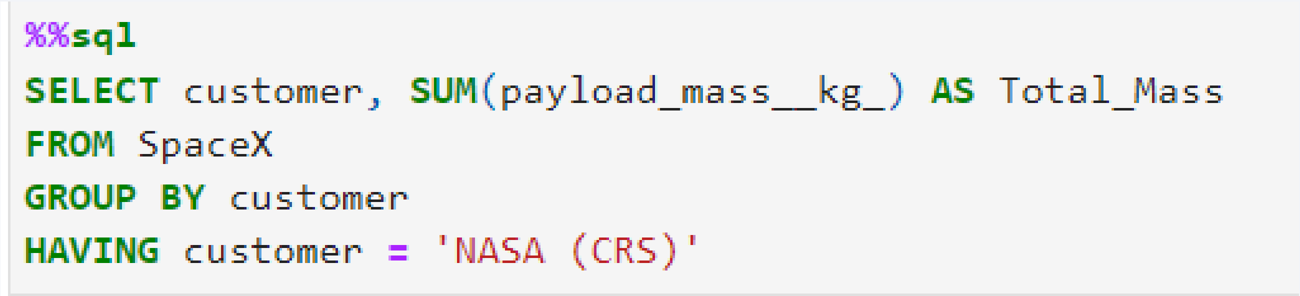
```
%%sql
SELECT *
FROM SpaceX
WHERE launch_site like 'CCA%'
LIMIT 5
```



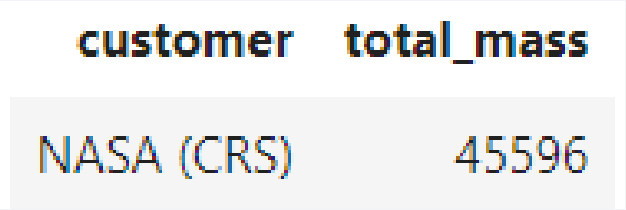
DATE	time__utc_	booster_version	launch_site	payload	payload_mass__kg_	orbit	customer	mission_outcome	landing__outcome
2010-06-04	18:45:00	F9 v1.0 B0003	CCAFS LC-40	Dragon Spacecraft Qualification Unit	0	LEO	SpaceX	Success	Failure (parachute)
2010-12-08	15:43:00	F9 v1.0 B0004	CCAFS LC-40	Dragon demo flight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

This is the total payload mass carried by NASA (CRS) boosters



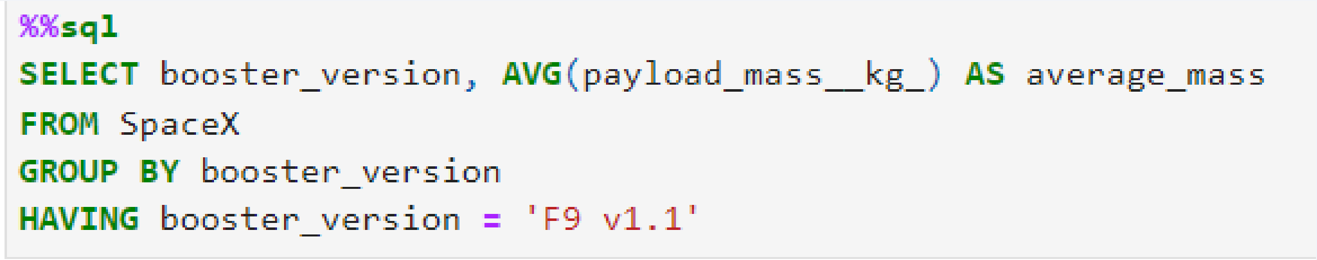
```
%%sql
SELECT customer, SUM(payload_mass__kg_) AS Total_Mass
FROM SpaceX
GROUP BY customer
HAVING customer = 'NASA (CRS)'
```



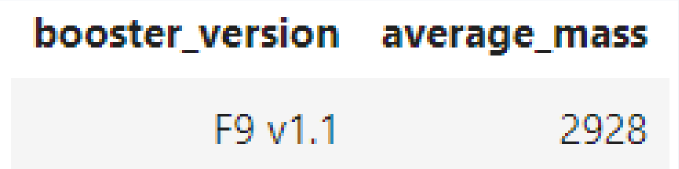
customer	total_mass
NASA (CRS)	45596

Average Payload Mass by F9 v1.1

The average payload mass carried by booster version F9 v1



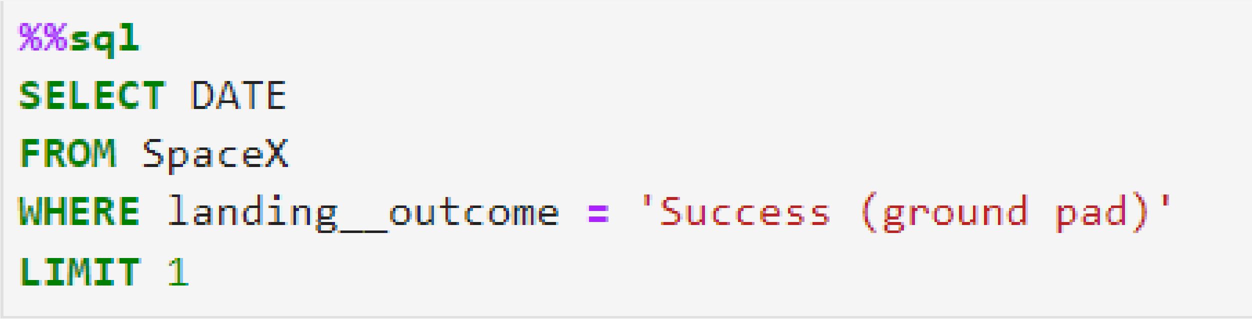
```
%%sql
SELECT booster_version, AVG(payload_mass__kg_) AS average_mass
FROM SpaceX
GROUP BY booster_version
HAVING booster_version = 'F9 v1.1'
```



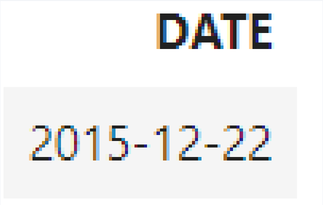
booster_version	average_mass
F9 v1.1	2928

First Successful Ground Landing Date

- Dates of the first successful landing outcome on ground pad



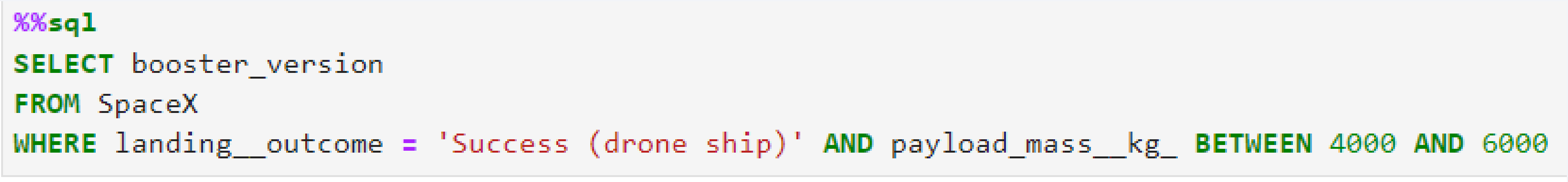
```
%%sql
SELECT DATE
FROM SpaceX
WHERE landing__outcome = 'Success (ground pad)'
LIMIT 1
```



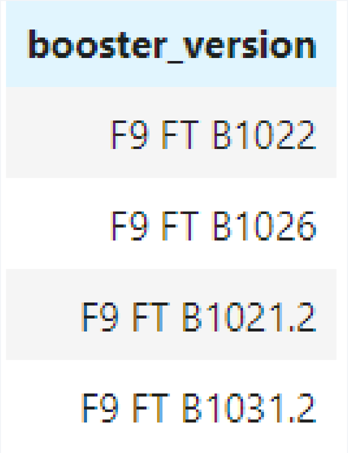
DATE
2015-12-22

Successful Drone Ship Landing with Payload between 4000 and 6000

- Names of boosters which have successfully landed on drone ship and had payload mass greater than 4000 but less than 6000



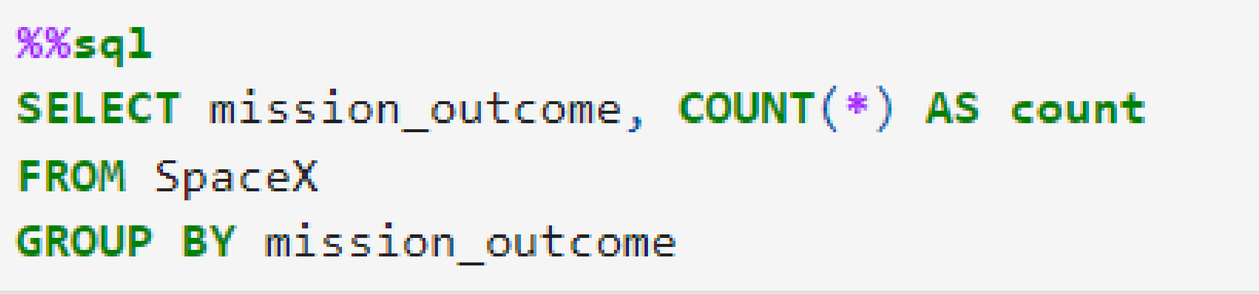
```
%%sql
SELECT booster_version
FROM SpaceX
WHERE landing__outcome = 'Success (drone ship)' AND payload_mass__kg_ BETWEEN 4000 AND 6000
```



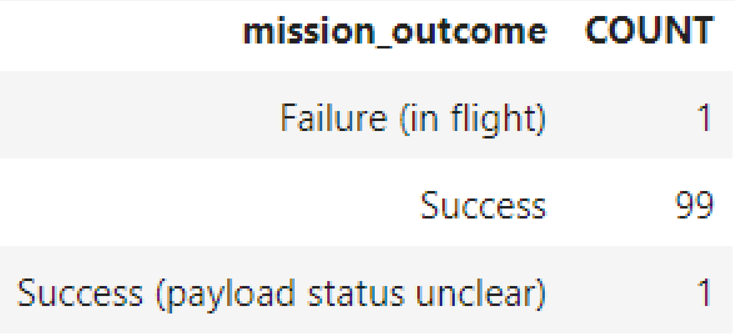
booster_version
F9 FT B1022
F9 FT B1026
F9 FT B1021.2
F9 FT B1031.2

Total Number of Successful and Failure Mission Outcomes

- Calculate the total number of successful and failure mission outcomes



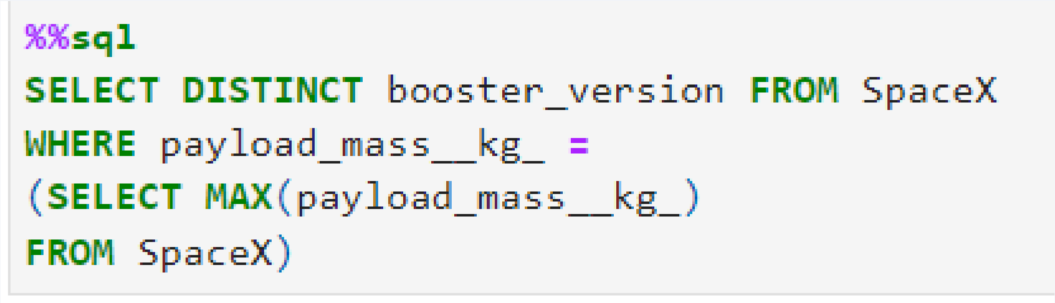
```
%%sql
SELECT mission_outcome, COUNT(*) AS count
FROM SpaceX
GROUP BY mission_outcome
```



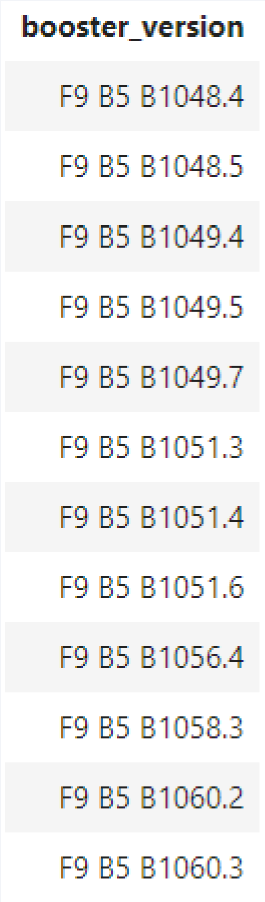
mission_outcome	COUNT
Failure (in flight)	1
Success	99
Success (payload status unclear)	1

Boosters Carried Maximum Payload

- Names of the booster which have carried the maximum payload mass



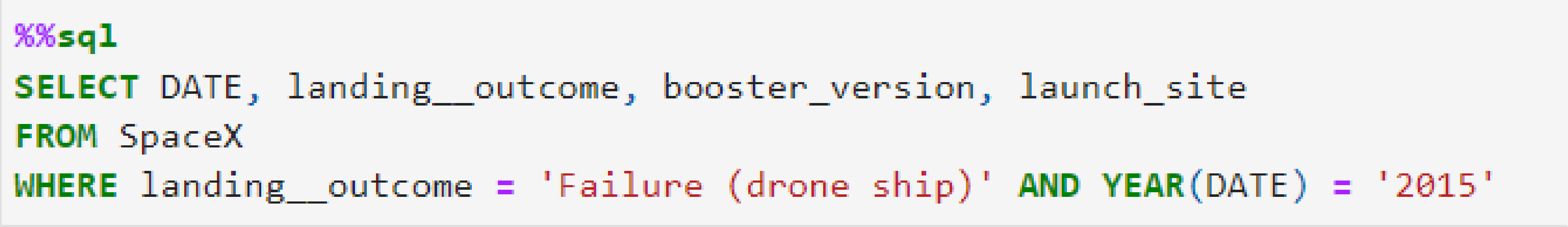
```
%%sql
SELECT DISTINCT booster_version FROM SpaceX
WHERE payload_mass__kg_ =
(SELECT MAX(payload_mass__kg_)
FROM SpaceX)
```



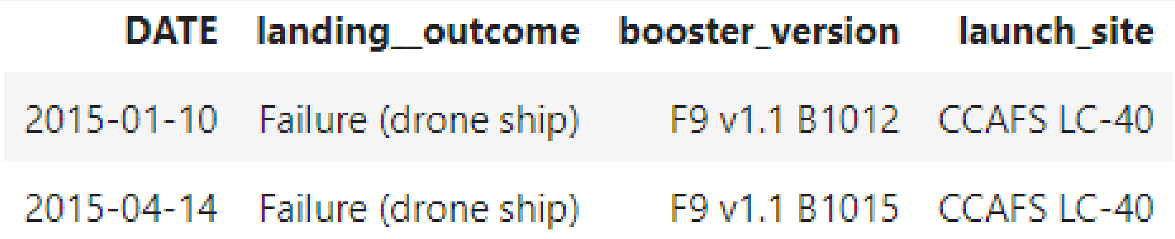
booster_version
F9 B5 B1048.4
F9 B5 B1048.5
F9 B5 B1049.4
F9 B5 B1049.5
F9 B5 B1049.7
F9 B5 B1051.3
F9 B5 B1051.4
F9 B5 B1051.6
F9 B5 B1056.4
F9 B5 B1058.3
F9 B5 B1060.2
F9 B5 B1060.3

2015 Launch Records

- The failed landing outcomes in drone ship, their booster versions, and launch site names for in year 2015



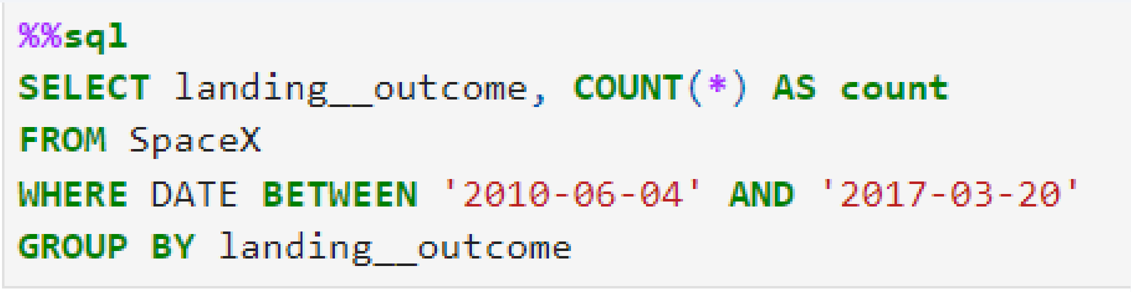
```
%%sql
SELECT DATE, landing__outcome, booster_version, launch_site
FROM SpaceX
WHERE landing__outcome = 'Failure (drone ship)' AND YEAR(DATE) = '2015'
```



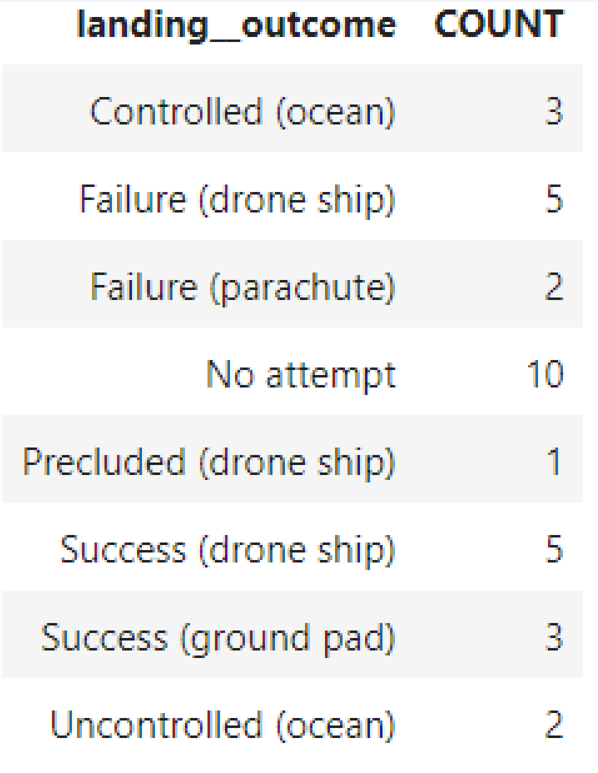
DATE	landing__outcome	booster_version	launch_site
2015-01-10	Failure (drone ship)	F9 v1.1 B1012	CCAFS LC-40
2015-04-14	Failure (drone ship)	F9 v1.1 B1015	CCAFS LC-40

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

- Ranked 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



```
%%sql
SELECT landing__outcome, COUNT(*) AS count
FROM SpaceX
WHERE DATE BETWEEN '2010-06-04' AND '2017-03-20'
GROUP BY landing__outcome
```



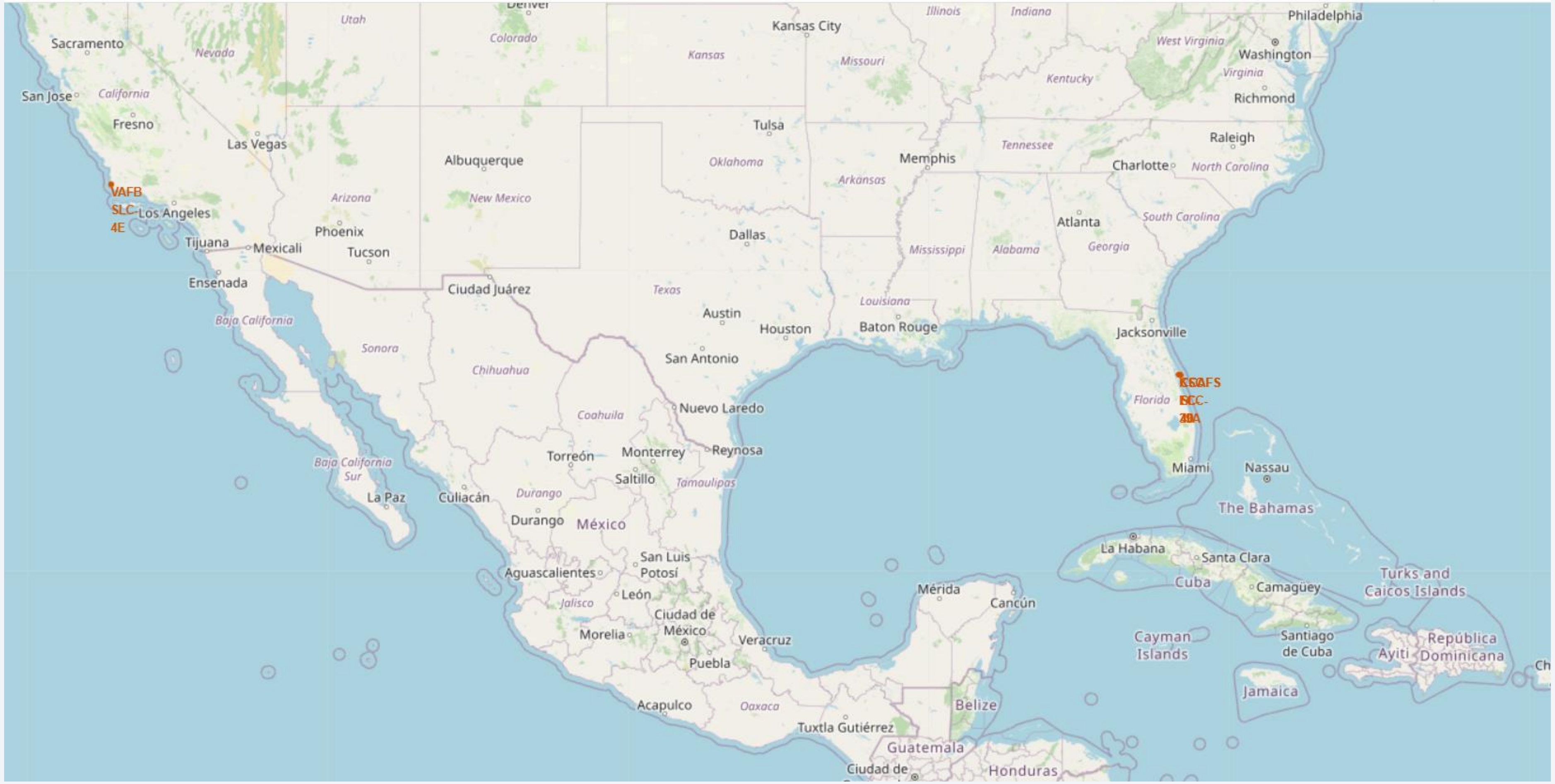
landing__outcome	COUNT
Controlled (ocean)	3
Failure (drone ship)	5
Failure (parachute)	2
No attempt	10
Precluded (drone ship)	1
Success (drone ship)	5
Success (ground pad)	3
Uncontrolled (ocean)	2

Section 3

Launch Sites Proximities Analysis

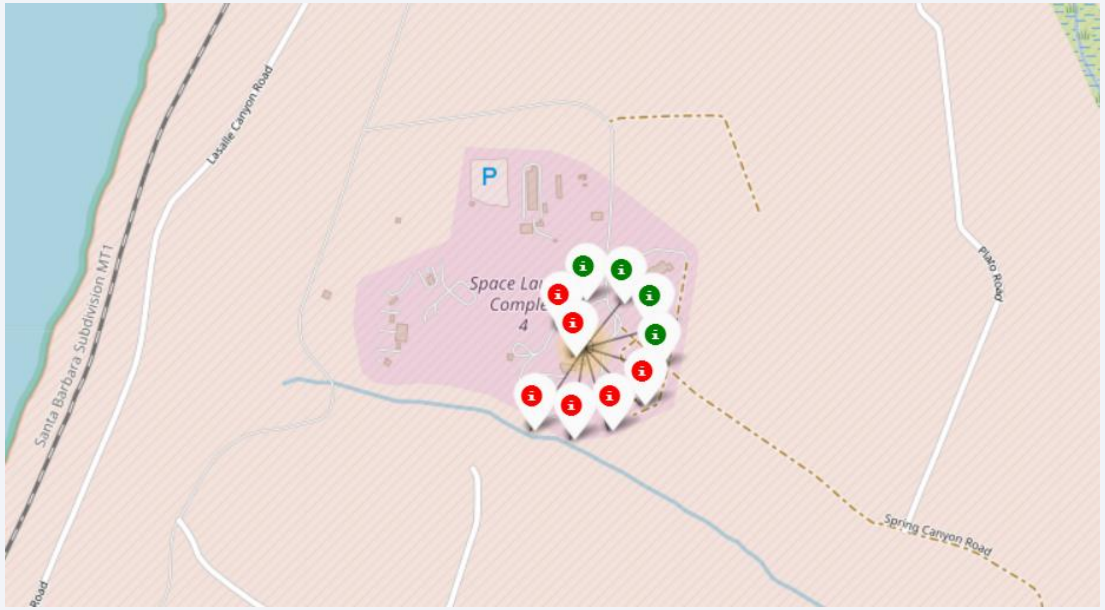
Launch Sites locations on the Folium map

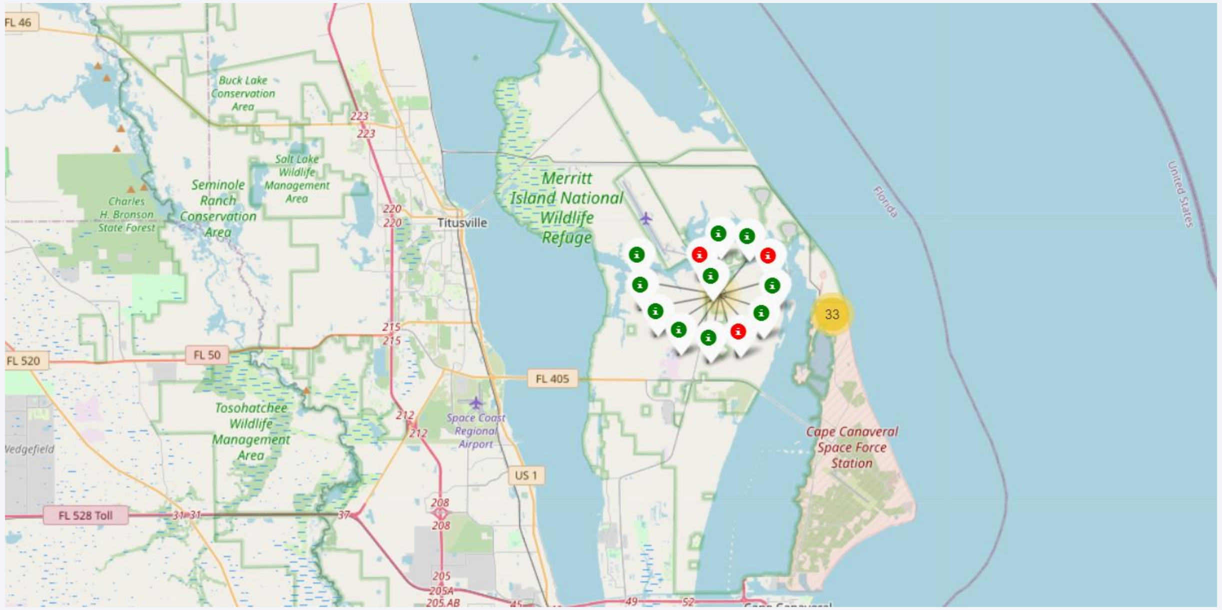
- It can be noticed that all launch locations are in the coastal regions of the United States



Launch outcomes for different launch sites

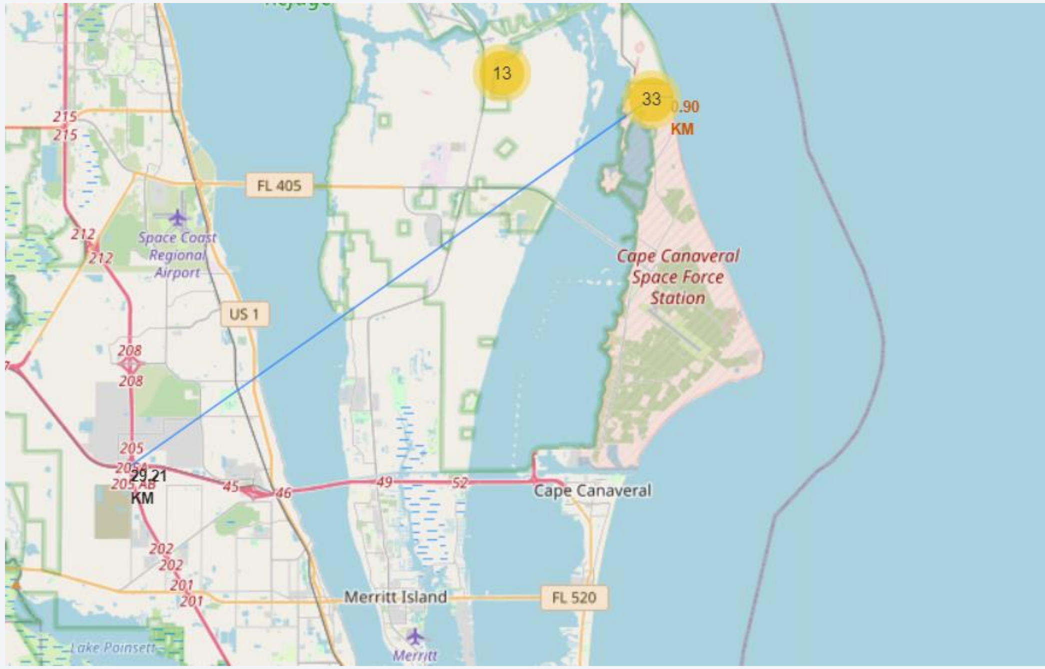
- Marker clusters can be efficiently used to locate several markers to the same place on the Folium map
- Green markers indicate successful launches and red markers indicate unsuccessful launches.

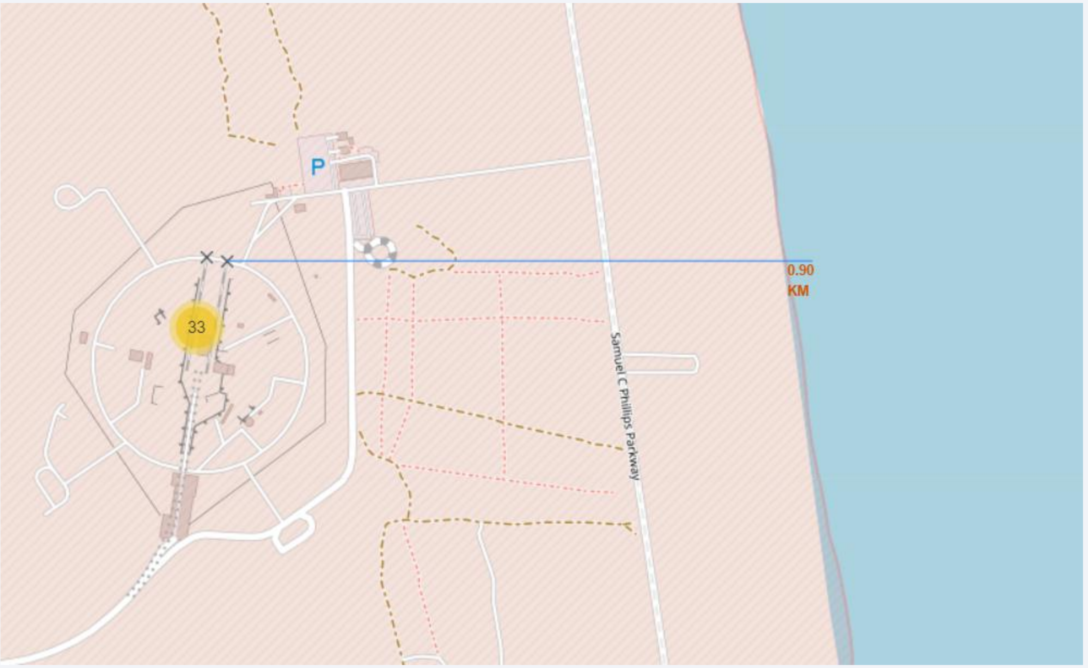



FL 46
Buck Lake Conservation Area
Salt Lake Wildlife Management Area
Seminole Ranch Conservation Area
Charles H. Bronson State Forest
Titusville
Merritt Island National Wildlife Refuge
33
FL 520
FL 50
FL 405
Tosohatchee Wildlife Management Area
Space Coast Regional Airport
US 1
FL 528 Toll
Wedgefield
Cape Canaveral Space Force Station
Florida
United States

Launch sites proximities

- Lines in Folium can be applied to show the distance from the launch sites to coasts, highways etc.



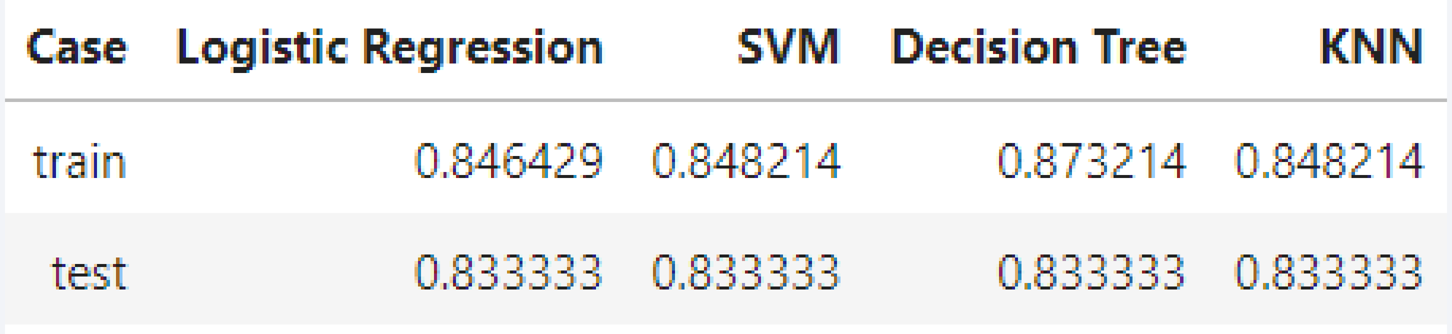


Section 5

Predictive Analysis (Classification)

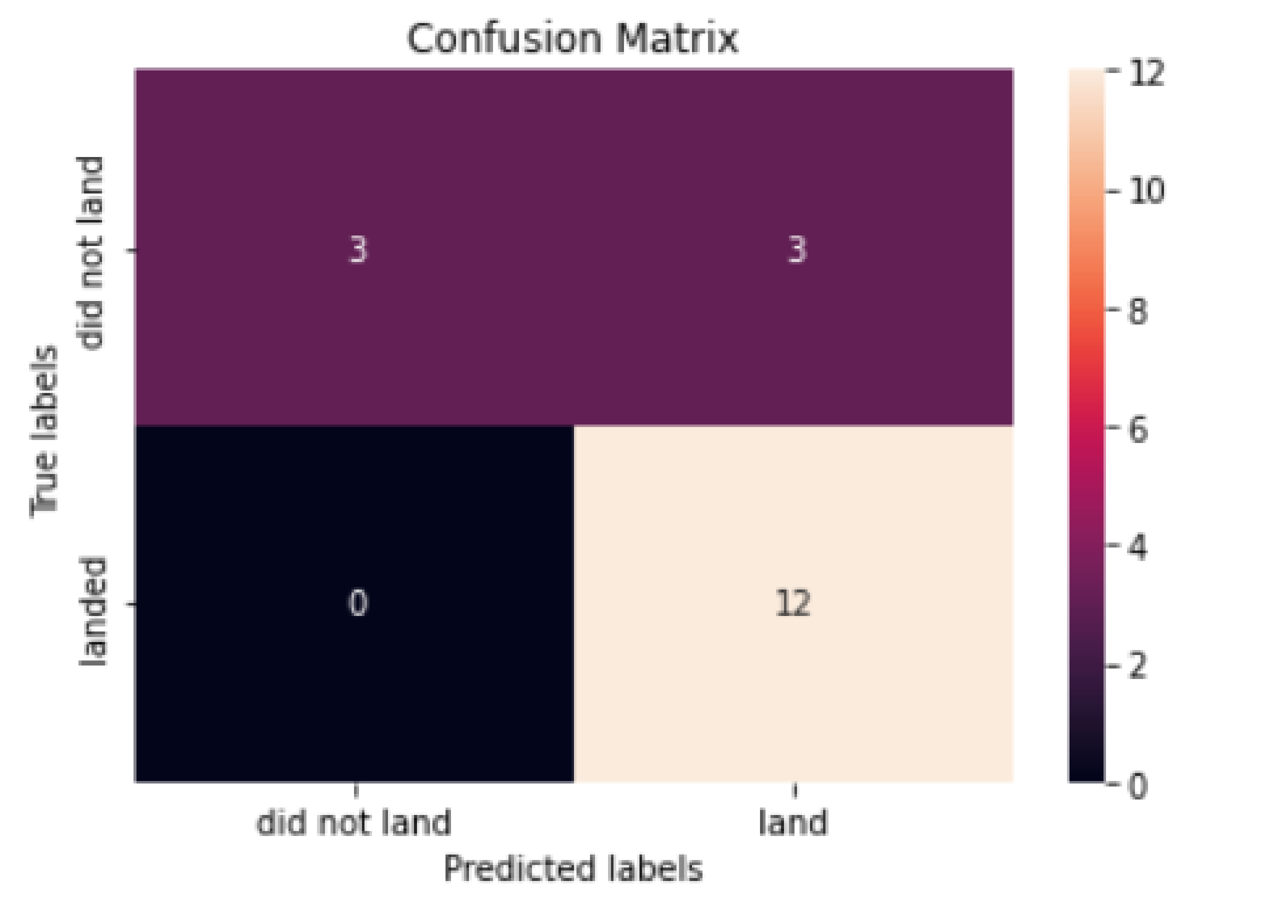
Classification Accuracy

It can be noticed, that the accuracy of all classification algorithms is approximately the same. Decision Tree classifier provided slightly higher accuracy, therefore it was chosen as an optimum prediction model.



Case	Logistic Regression	SVM	Decision Tree	KNN
train	0.846429	0.848214	0.873214	0.848214
test	0.833333	0.833333	0.833333	0.833333

Confusion Matrix



On the confusion matrix for the Decision Tree it can be noticed, that all success cases were predicted correctly for the test data. The weak place of the prediction is False Positives.

Conclusions

- The data was successfully collected from SpaceX API
- The data was cleaned and prepared for the analysis using Pandas and Numpy
- The Explarotary Data Analysis was conducted in Python using visualizations and SQL queries.
- Several alternative Machine Learning algorithms (Scikit-Learn) were suggested to predict weather the Falcon 9 rocket will land successfully
- The effectiveness of the models was compared and the most effective one was selected

Thank you!