IBM Capstone Prjct – Landing of first Falcon 9 Rocket

Marvin

19/04/2022

https://github.com/Marvin2108/IBM_DataScience_Coursera.git

IBM Developer

SKILLS NETWORK

Table of Contents

1. Executive Summary
2. Introduction
3. Methodology
4. Results
    1. Visualization – Charts
    2. Dashboard
5. Conclusion

EXECUTIVE SUMMARY

- Goal: Prediction of successful landing of SpaceX first Falcon 9 rocket
- Used different typical Data Science methodologies such as
  - Web Scraping
  - Data Wrangling,
  - Exploratory Data Analysis with SQL
  - Train different ML models and evaluate them
- Finally, got a correct classification of 94 %

INTRODUCTION

- SpaceX advertises Falcon 9 rocket launches on its website, with a cost of 62 million dollars; other providers cost upward of 165 million dollars each, much of the savings is because SpaceX can reuse the first stage.
- Therefore if the likelihood of the first stage rocket landing can be successfully predicted, the cost of a launch can be predicted. This information can be used if an alternate company wants to bid against SpaceX for a rocket launch

METHODOLOGY

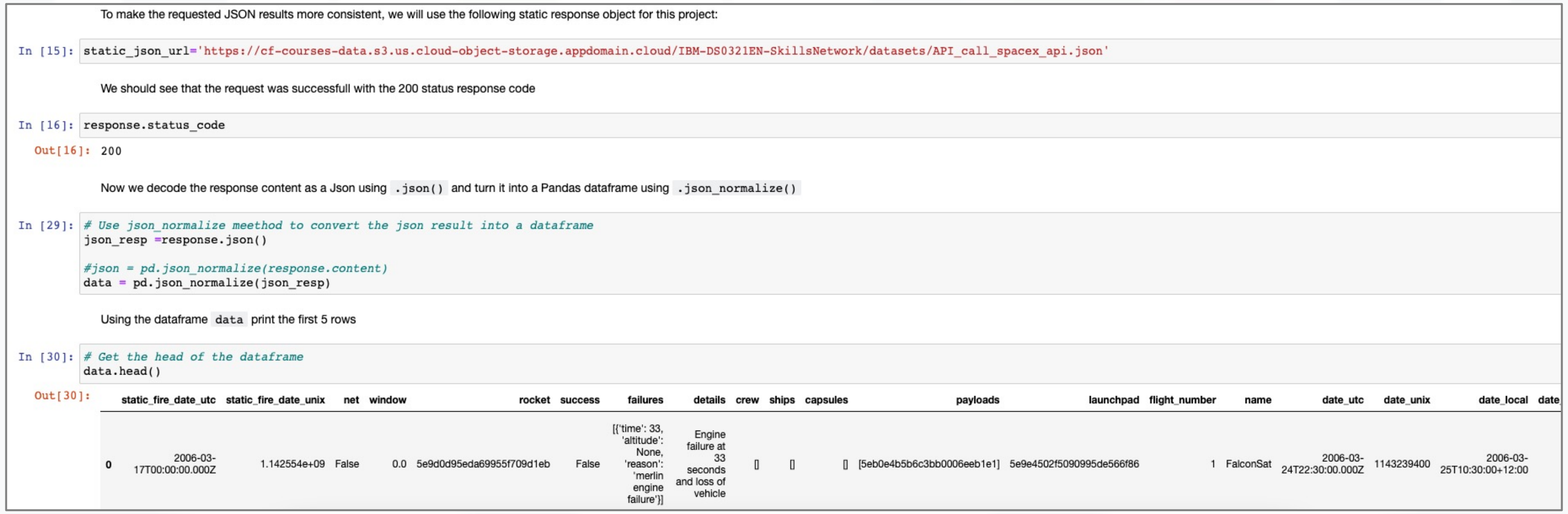
- Data Collection
  - Data was collected using SpaceX API and web scraping from Wikipedia
- Data Wrangling
  - Performing One-hot-encoding
- Exploratory Data Analysis (EDA) using SQL
- Performing predicitive analysis using classification models
  - Build, train and evaluate classification models
  - Logistic regression, SVM, Decision Trees, kNN

Data Collection

- The Data was collected using the SpaceX-API
- We decoded the response content of the API-request as a JSON and then turn it into a pd-DataFrame using .json_normalize()
  - We then cleaned the data, checked for missing values and filled missing values
- Further, we used web scraping with BeautifulSoup
  - There, we extracted the launch records from HTML-tables and converted it to pandas DataFrame

Data Collection / results

- Transformed the response from the get-request to pandas Dataframe using json_normalize()-function
- Link to the notebook: https://github.com/Marvin2108/IBM_DataScience_Coursera/blob/master/Data%20Collection%20API%20Lab.ipynb



To make the requested JSON results more consistent, we will use the following static response object for this project:

```
In [15]: static_json_url='https://cf-courses-data.s3.us.cloud-object-storage.appdomain.cloud/IBM-DS0321EN-SkillsNetwork/datasets/API_call_spacex_api.json'
```

We should see that the request was successfull with the 200 status response code

```
In [16]: response.status_code
Out[16]: 200
```

Now we decode the response content as a Json using `.json()` and turn it into a Pandas dataframe using `.json_normalize()`

```
In [29]: # Use json_normalize meethod to convert the json result into a dataframe
         json_resp =response.json()

         #json = pd.json_normalize(response.content)
         data = pd.json_normalize(json_resp)
```

Using the dataframe `data` print the first 5 rows

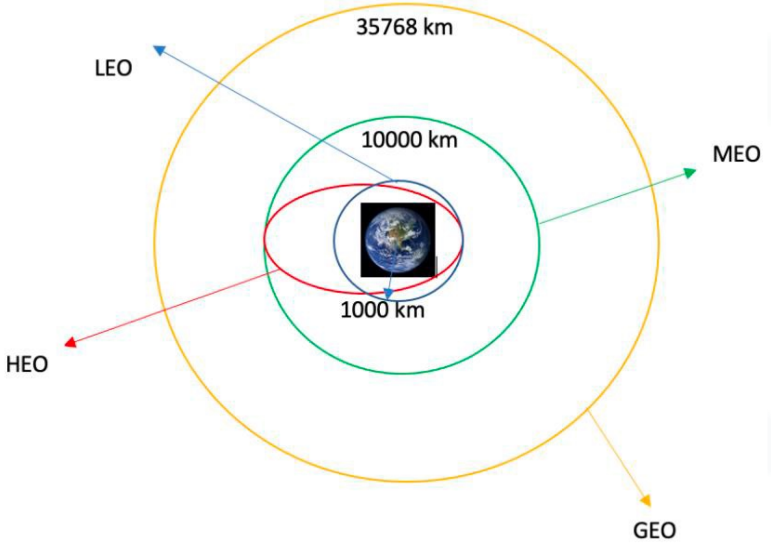
```
In [30]: # Get the head of the dataframe
         data.head()
```

Out[30]:

	static_fire_date_utc	static_fire_date_unix	net	window	rocket	success	failures	details	crew	ships	capsules	payloads	launchpad	flight_number	name	date_utc	date_unix	date_local	date
0	2006-03-17T00:00:00.000Z	1.142554e+09	False	0.0	5e9d0d95eda69955f709d1eb	False	[{'time': 33, 'altitude': None, 'reason': 'merlin engine failure'}]	Engine failure at 33 seconds and loss of vehicle	[]	[]	[]	[5eb0e4b5b6c3bb0006eeb1e1]	5e9e4502f5090995de566f86	1	FalconSat	2006-03-24T22:30:00.000Z	1143239400	2006-03-25T10:30:00+12:00	

Data Wrangling

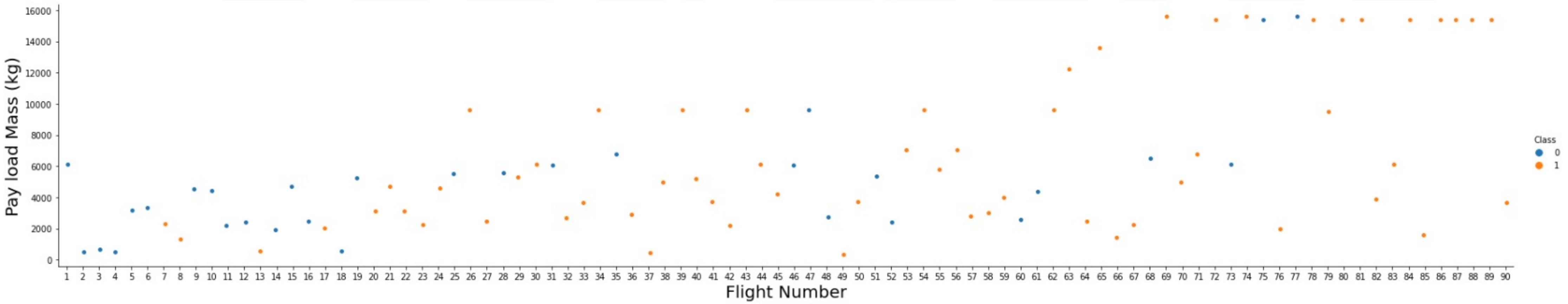
- We calculated number of launches at each site
- We created an outcome label from outcome column for the later training set and exported the result to csv



- Notebook: https://github.com/Marvin2108/IBM_DataScience_Coursera/blob/master/Data%20Wrangling.ipynb

EDA & interactive visual analytics

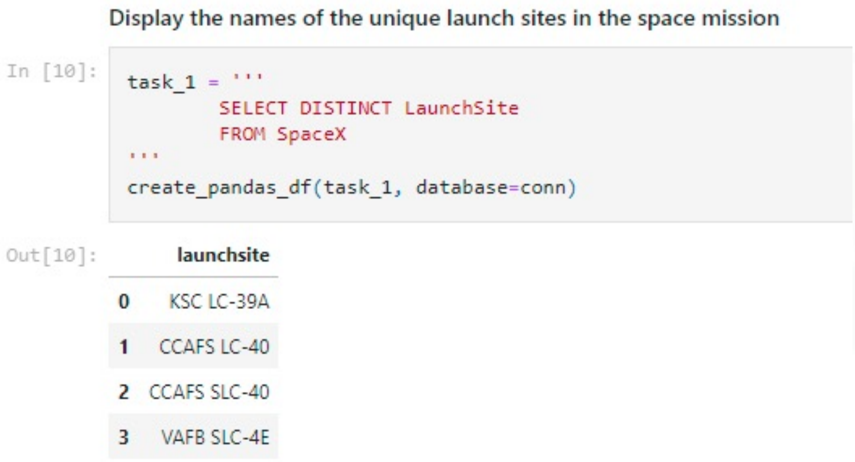
- We explored the data by visualizing the relationship between flight number and launch Site, payload and launch site, success rate of each orbit type, flight number and orbit type, the launch success yearly trend.



EDA with SQL – results

- We created a DB2 table with IBM Cloud and saved the csv-dataset generated before
- Then, we did some EDA with sqlalchemy in the notebook after we connected to the DB2 instance

EDA with SQL – All unique launch sites



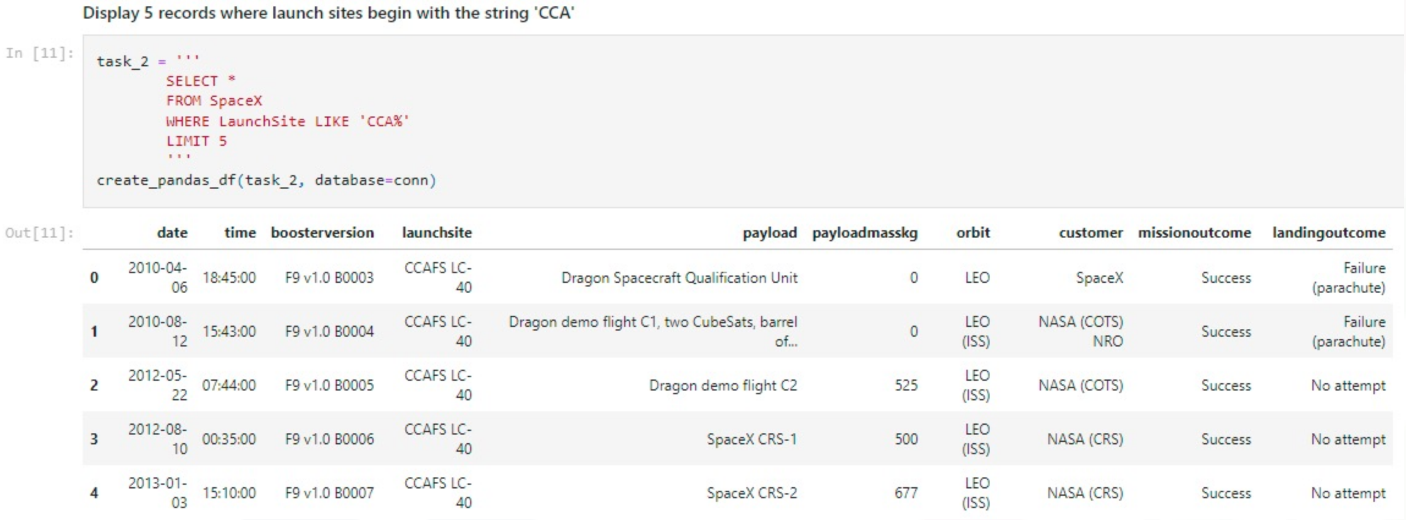
Display the names of the unique launch sites in the space mission

```
In [10]: task_1 = '''
                 SELECT DISTINCT LaunchSite
                 FROM SpaceX
         '''
         create_pandas_df(task_1, database=conn)
```

Out[10]:

	launchsite
0	KSC LC-39A
1	CCAFS LC-40
2	CCAFS SLC-40
3	VAFB SLC-4E

EDA with SQL – Launch site beginning with CCA



Display 5 records where launch sites begin with the string 'CCA'

```
In [11]: task_2 = '''
                 SELECT *
                 FROM SpaceX
                 WHERE LaunchSite LIKE 'CCA%'
                 LIMIT 5
                 '''
         create_pandas_df(task_2, database=conn)
```

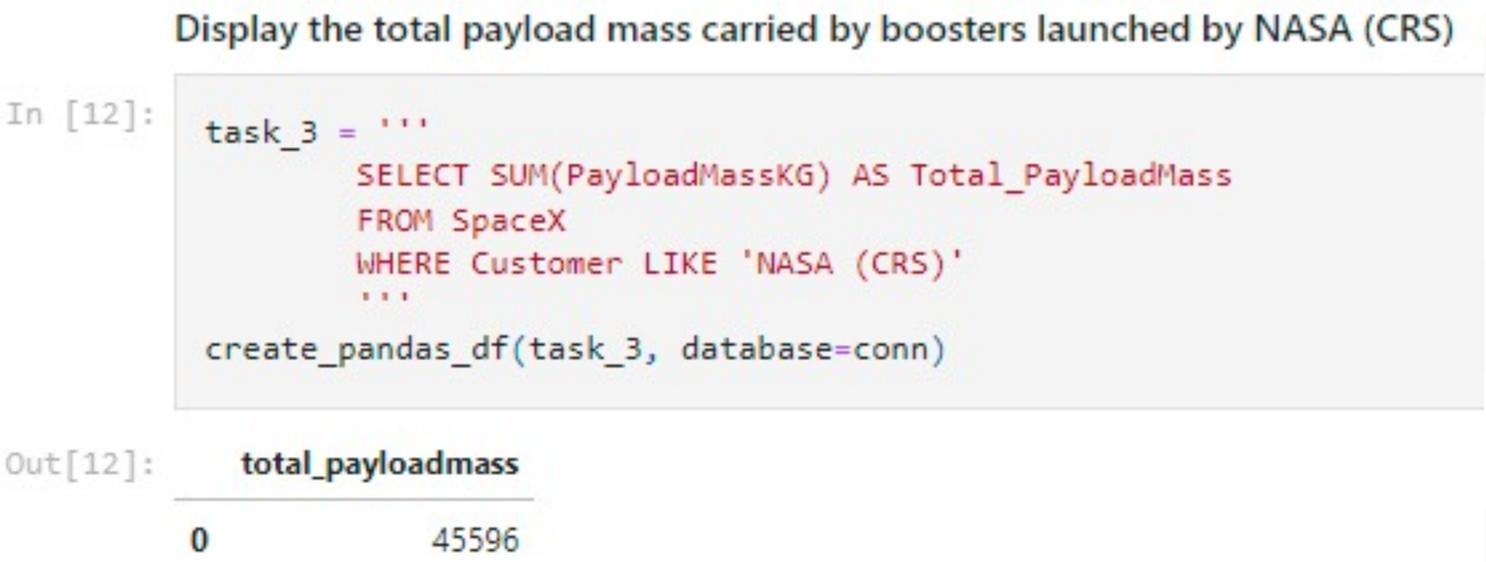
Out[11]:

	date	time	boosterversion	launchsite	payload	payloadmasskg	orbit	customer	missionoutcome	landingoutcome
0	2010-04-06	18:45:00	F9 v1.0 B0003	CCAFS LC-40	Dragon Spacecraft Qualification Unit	0	LEO	SpaceX	Success	Failure (parachute)
1	2010-08-12	15:43:00	F9 v1.0 B0004	CCAFS LC-40	Dragon demo flight C1, two CubeSats, barrel of...	0	LEO (ISS)	NASA (COTS) NRO	Success	Failure (parachute)
2	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
3	2012-08-10	00:35:00	F9 v1.0 B0006	CCAFS LC-40	SpaceX CRS-1	500	LEO (ISS)	NASA (CRS)	Success	No attempt
4	2013-01-03	15:10:00	F9 v1.0 B0007	CCAFS LC-40	SpaceX CRS-2	677	LEO (ISS)	NASA (CRS)	Success	No attempt

- Displayed only five first records

EDA with SQL – Total Payload Mass

- Calculated sum with following query



Display the total payload mass carried by boosters launched by NASA (CRS)

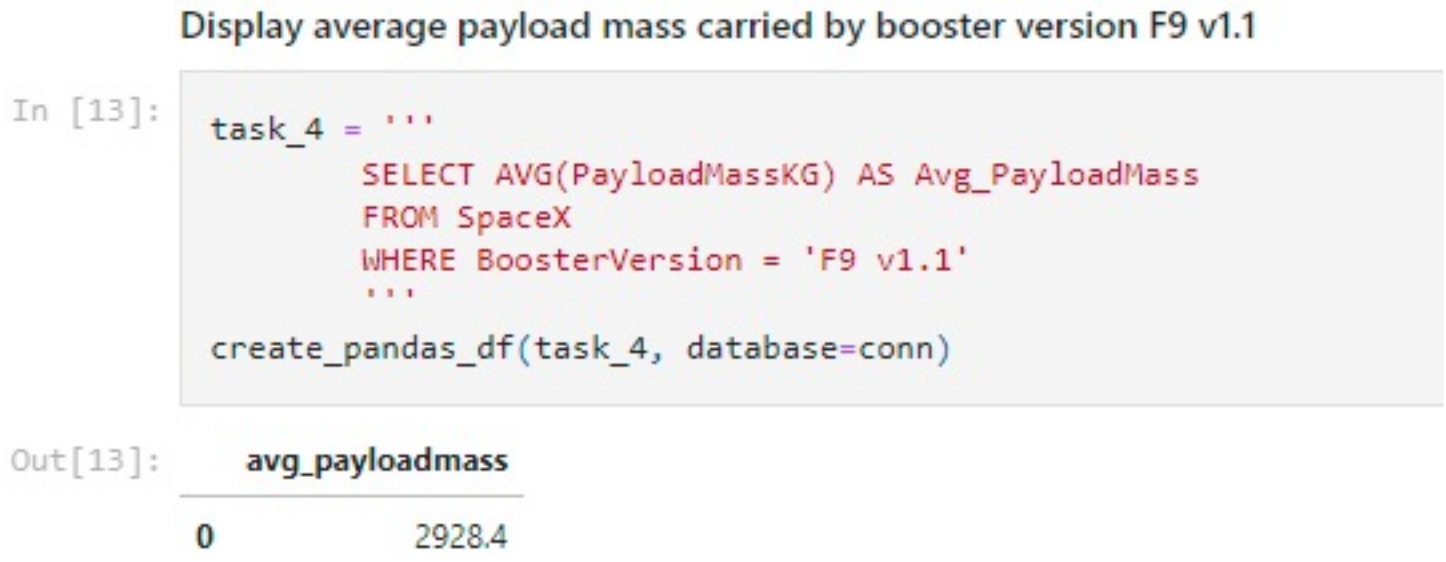
In [12]:

```
task_3 = '''
        SELECT SUM(PayloadMassKG) AS Total_PayloadMass
        FROM SpaceX
        WHERE Customer LIKE 'NASA (CRS)'
        '''
create_pandas_df(task_3, database=conn)
```

Out[12]:

```
   total_payloadmass
0              45596
```

EDA with SQL – Average Payload Mass F9 v.1.1

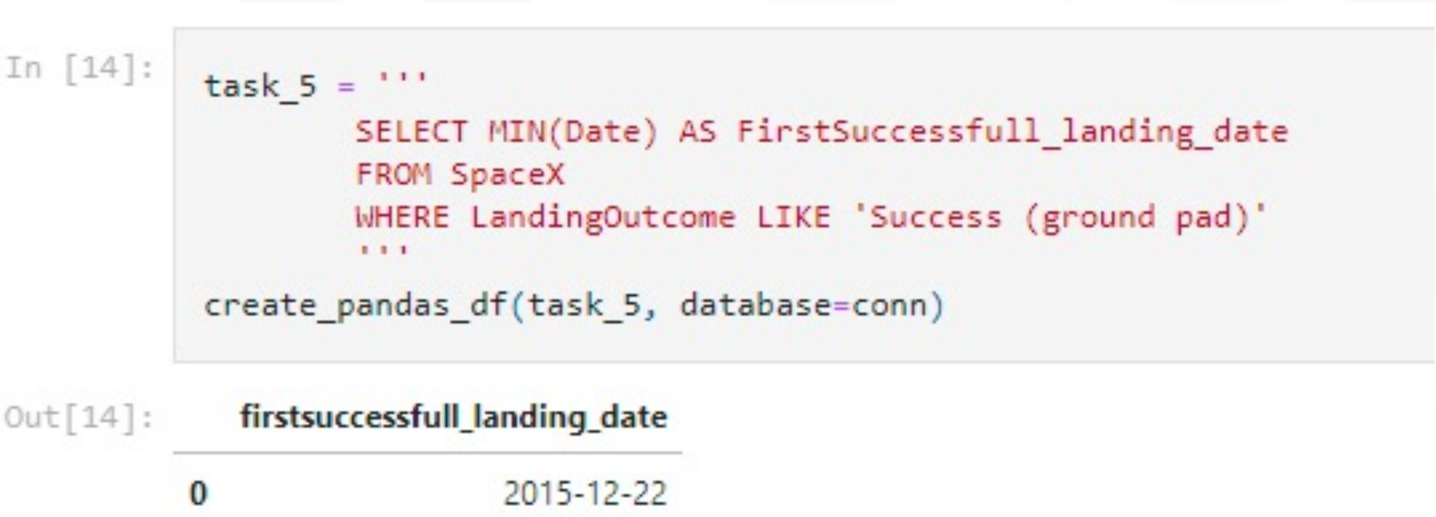


Display average payload mass carried by booster version F9 v1.1

```
In [13]:  task_4 = '''
                  SELECT AVG(PayloadMassKG) AS Avg_PayloadMass
                  FROM SpaceX
                  WHERE BoosterVersion = 'F9 v1.1'
                  '''
          create_pandas_df(task_4, database=conn)
```

```
Out[13]:     avg_payloadmass
          0           2928.4
```

EDA with SQL – First successful landing date

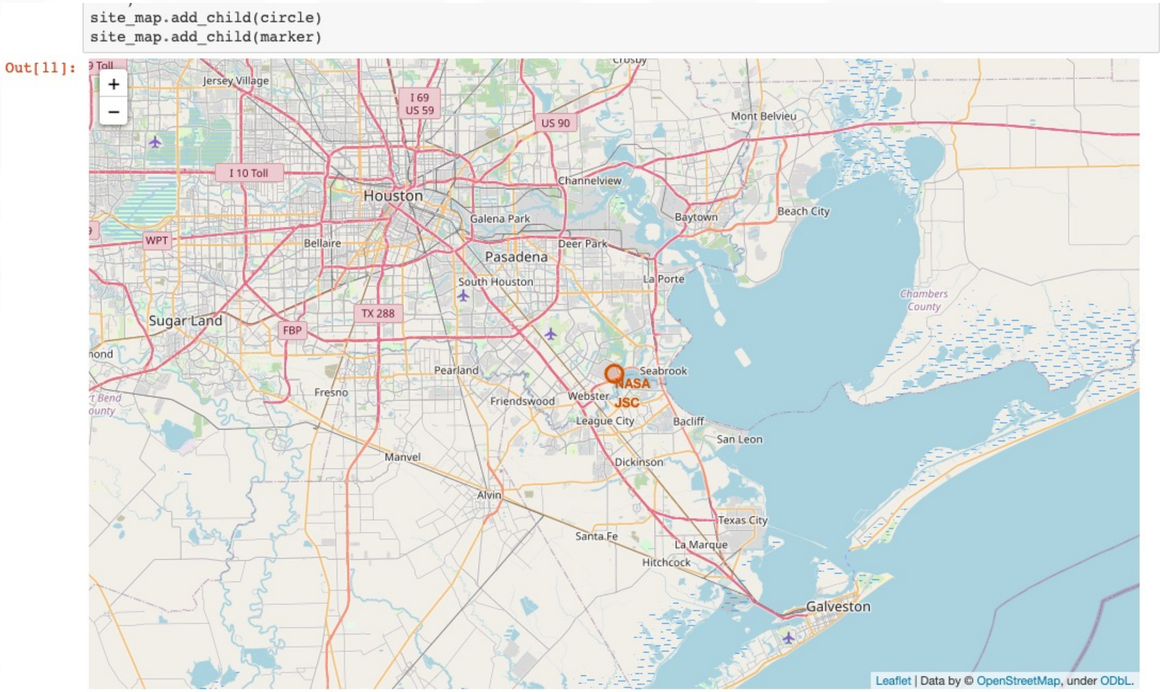


```
In [14]:
task_5 = '''
        SELECT MIN(Date) AS FirstSuccessfull_landing_date
        FROM SpaceX
        WHERE LandingOutcome LIKE 'Success (ground pad)'
        '''
create_pandas_df(task_5, database=conn)
```

```
Out[14]:
     firstsuccessfull_landing_date
  0                     2015-12-22
```

Interactive map result

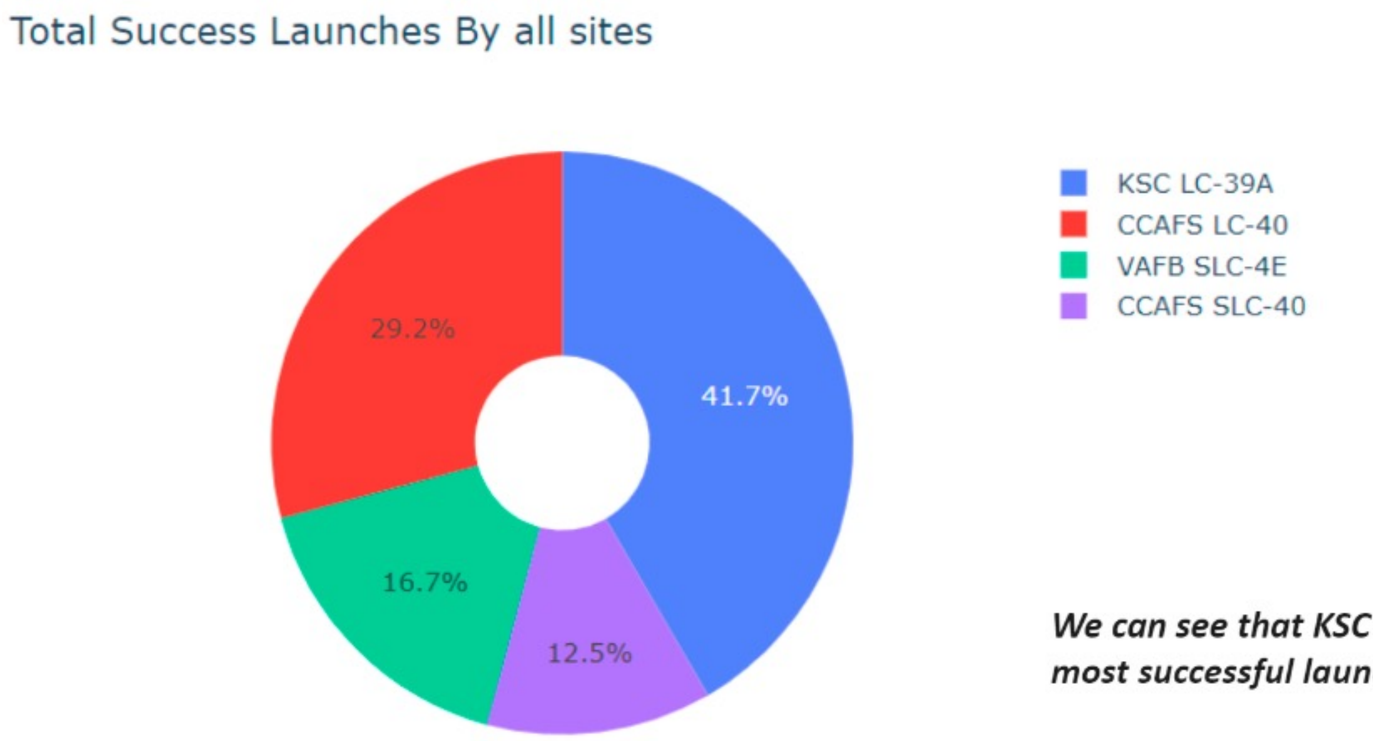
- We marked all launch sites, and added map objects such as markers, circles, lines to mark the success or failure of launches for each site on the folium map.



```
site_map.add_child(circle)
site_map.add_child(marker)
```

Out[11]:

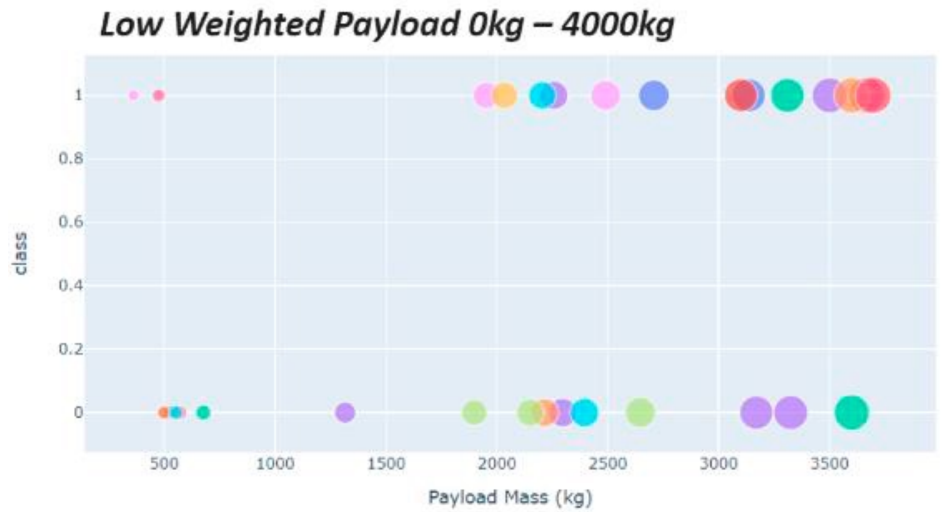
Dashboard with dash results

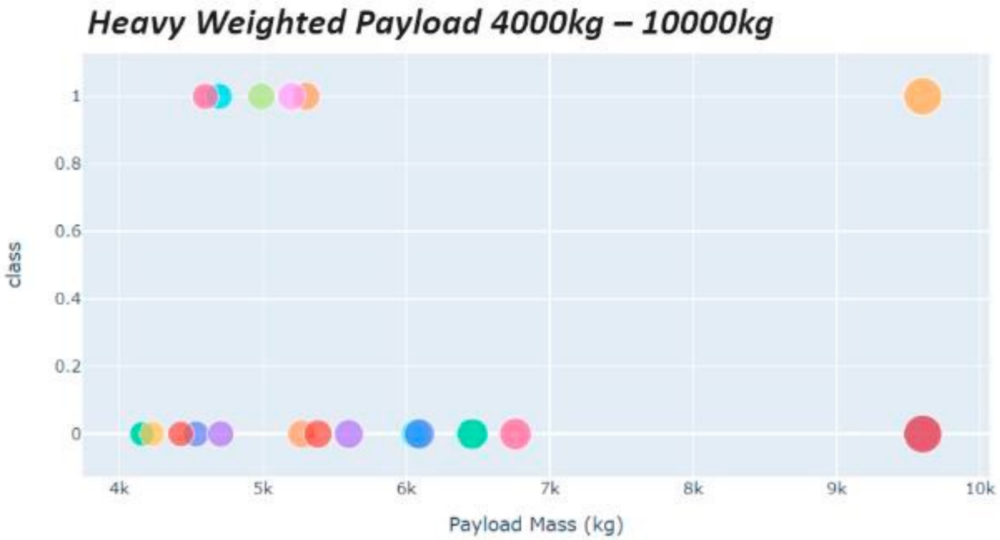


Total Success Launches By all sites

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

Dashboard with dash results



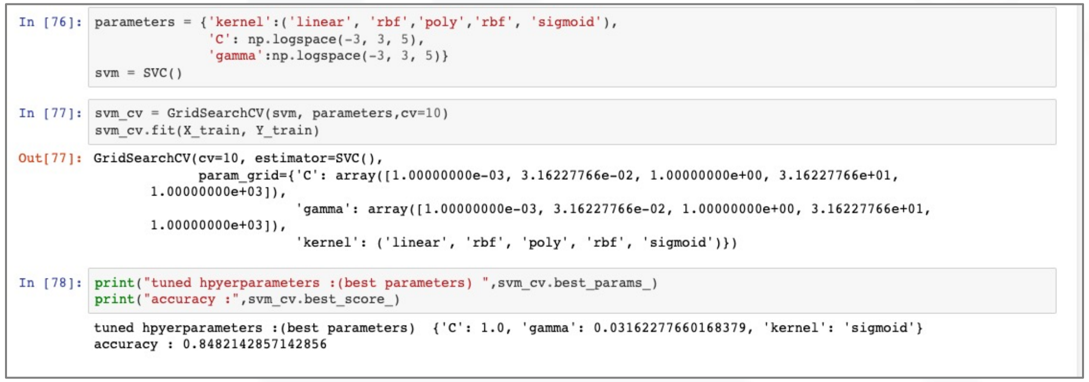


We can see the success rates for low weighted payloads is higher than the heavy weighted payloads

Predictive analysis results

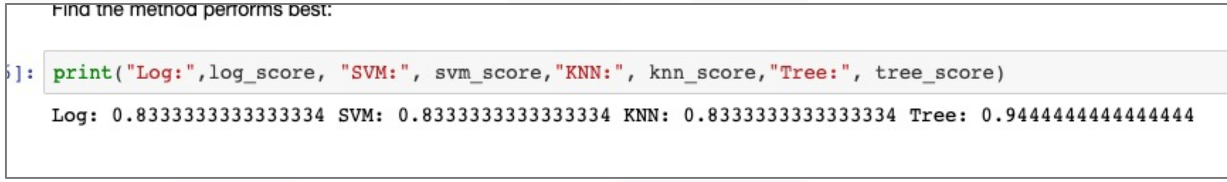
- We trained some classification models based on the train set and evaluated them based on test set
- Notebook: https://github.com/Marvin2108/IBM_DataScience_Coursera/blob/master/Predictive%20Analysis%20Falcon%209%20SpaceX.ipynb

E.g. SVM-Classifier



```
In [76]: parameters = {'kernel':('linear', 'rbf','poly','rbf', 'sigmoid'),
                       'C': np.logspace(-3, 3, 5),
                       'gamma':np.logspace(-3, 3, 5)}
         svm = SVC()

In [77]: svm_cv = GridSearchCV(svm, parameters,cv=10)
         svm_cv.fit(X_train, Y_train)

Out[77]: GridSearchCV(cv=10, estimator=SVC(),
                      param_grid={'C': array([1.00000000e-03, 3.16227766e-02, 1.00000000e+00, 3.16227766e+01,
                1.00000000e+03]),
                                  'gamma': array([1.00000000e-03, 3.16227766e-02, 1.00000000e+00, 3.16227766e+01,
                1.00000000e+03]),
                                  'kernel': ('linear', 'rbf', 'poly', 'rbf', 'sigmoid')})

In [78]: print("tuned hpyerparameters :(best parameters) ",svm_cv.best_params_)
         print("accuracy :",svm_cv.best_score_)

         tuned hpyerparameters :(best parameters)  {'C': 1.0, 'gamma': 0.03162277660168379, 'kernel': 'sigmoid'}
         accuracy : 0.8482142857142856
```

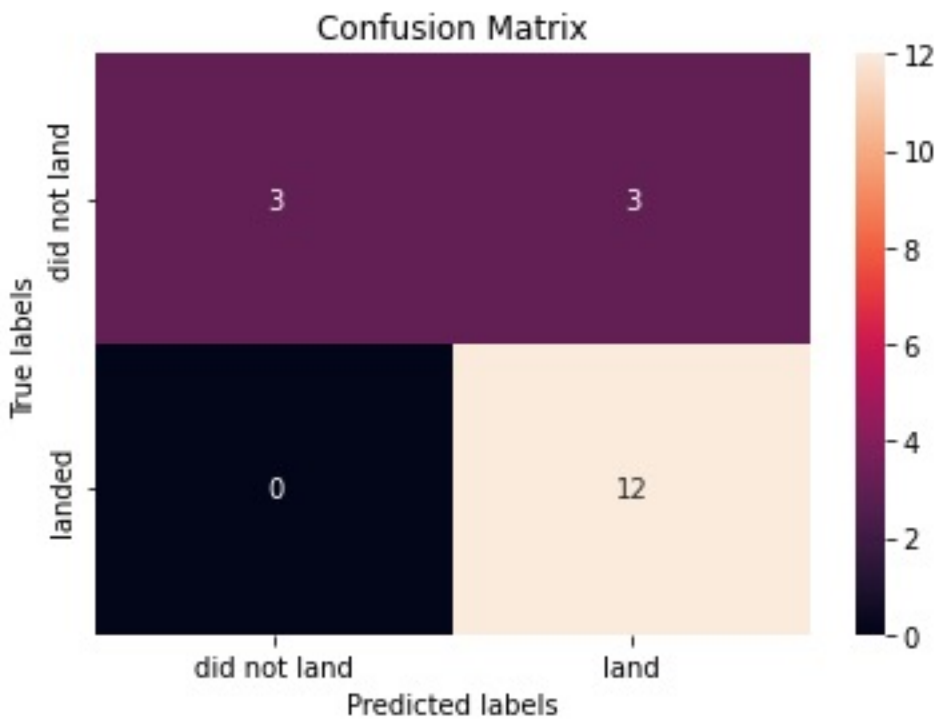
Score of all models



Find the method performs best:

```
]: print("Log:",log_score, "SVM:", svm_score,"KNN:", knn_score,"Tree:", tree_score)

   Log: 0.8333333333333334 SVM: 0.8333333333333334 KNN: 0.8333333333333334 Tree: 0.9444444444444444
```

Confusion Matrix - SVM



- This Matrix of the SVM classifier shows that it can distinguish between a successful and an unsuccessful landing

CONCLUSION

- The larger the flight amount at a launch site, the greater the success rate at a launch site.
- KSC LC-39A had the most successful launches of any sites.
- The Decision Tree is the best classifier to predict the successful landing, i.e. the costs of a Falcon 9 first Rocket