IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

Udo Riegler

02.02.2024

Outline

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

Executive Summary

- Summary of used methodologies:
  - We collected data from various public source as.
    After raw data has been collected, the data quality were improved by performing data wrangling.
  - Then we started exploring the processed data. We found some really interesting real-world datasets together.
    We performed SQL quiring as we query the data and gather insights.
  - We got further insights into the data by applying some basic statistical analysis and data visualization. In this way we saw directly how variables are related to each other.
  - Finally we drill down into finer levels of detail by splitting the data into groups defined by categorical variables or factors in the processed data.
- Summary of all results

Introduction

- Space X 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 Space X can reuse the first stage.
- Therefore if we can determine if the first stage will land, we can determine the cost of a launch. This information can be used if an alternate company wants to bid against space X for a rocket launch.
- In this lab we create a machine learning pipeline to predict if the first stage will land given the data from the preceding labs.
- As a potential competitor of Space X, the main task is to determine the price of each launch and if Space X will be able to reuse the first stage and reduce cost of each launch, using this special feature.

Section 1

Methodology

Methodology

Executive Summary

- Data collection methodology:
  - The used data set was collected from a public website (“SpaceX REST API”, given url())
- Perform data wrangling

  1: Request and parse the SpaceX launch data using the GET request

  2: Filter the data to only include ”Falcon 9” launches
- Perform exploratory data analysis (EDA) using visualization and SQL
- Perform interactive visual analytics using Folium and Plotly Dash
- Perform predictive analysis using classification models
  - We build, tuned and evaluated classification models

2: Filter the dataframe to only include `Falcon 9` launches

Data Collection

- Describe how data sets were collected from the posted “SpaceX REST API”, with given url().

SpaceX REST API

```
url="https://api.spacexdata.com/v4/launches/past"

response =requests.get(url)

response.json()
```

```
response.json()
```


[{'fairings': {'reused': False,
  'recovery_attempt': False,
  'recovered': False,
  'ships': []},
 'links': {'patch': {'small': 'https://images2.i
   'large': 'https://images2.imgbox.com/40/e3/Gy
  'reddit': {'campaign': None,
   'launch': None,
   'media': None,
   'recovery': None},
  'flickr': {'small': [], 'original': []},
  'presskit': None,
  'webcast': 'https://www.youtube.com/watch?v=0a_
  'youtube_id': '0a_00nJ_Y88',

Data collection process use key phrases and flowcharts:
in “**Lab 1: Collecting the data”:**

0: Import Libraries and Define Auxiliary Functions

1: Request and parse the SpaceX launch data using the GET request

2: Filter the dataframe to only include ”`Falcon 9`” launches

We will import the following libraries into the lab

```
# Requests allows us to make HTTP requests which we will use to get data from an API
import requests
# Pandas is a software library written for the Python programming language for data manipulation and analysis.
import pandas as pd
# NumPy is a library for the Python programming language, adding support for large, multi-dimensional arrays and matrices, along with a large collection of high-level mathe
import numpy as np
# Datetime is a library that allows us to represent dates
import datetime

# Setting this option will print all collumns of a dataframe
pd.set_option('display.max_columns', None)
# Setting this option will print all of the data in a feature
pd.set_option('display.max_colwidth', None)
```

Below we will define a series of helper functions that will help us use the API to extract information using identification numbers in the launch data.

From the `rocket` column we would like to learn the booster name.

```
# Takes the dataset and uses the rocket column to call the API and append the data to the list
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'])
```

From the `launchpad` we would like to know the name of the launch site being used, the logitude, and the latitude.

```
# Takes the dataset and uses the launchpad column to call the API and append the data to the list
def getLaunchSite(data):
    for x in data['launchpad']:
       if x:
         response = requests.get("https://api.spacexdata.com/v4/launchpads/"+str(x)).json()
         Longitude.append(response['longitude'])
         Latitude.append(response['latitude'])
         LaunchSite.append(response['name'])
```

From the `payload` we would like to learn the mass of the payload and the orbit that it is going to.

```
# Takes the dataset and uses the payloads column to call the API and append the data to the lists
def getPayloadData(data):
    for load in data['payloads']:
       if load:
        response = requests.get("https://api.spacexdata.com/v4/payloads/"+load).json()
        PayloadMass.append(response['mass_kg'])
        Orbit.append(response['orbit'])
```

Would you like to receive official Jupyter news?

	FlightNumber	Date	BoosterVersion	PayloadMass	Orbit	LaunchSite	Outcome	Flights	GridFins	Reused	Legs	LandingPad	Block	ReusedCount	Serial
4	6	2010-06-04	Falcon 9	NaN	LEO	CCSFS SLC 40	None None	1	False	False	False	None	1.0	0	B0003
5	8	2012-05-22	Falcon 9	525.0	LEO	CCSFS SLC 40	None None	1	False	False	False	None	1.0	0	B0005
6	10	2013-03-01	Falcon 9	677.0	ISS	CCSFS SLC 40	None None	1	False	False	False	None	1.0	0	B0007
7	11	2013-09-29	Falcon 9	500.0	PO	VAFB SLC 4E	False Ocean	1	False	False	False	None	1.0	0	B1003
8	12	2013-12-03	Falcon 9	3170.0	GTO	CCSFS SLC 40	None None	1	False	False	False	None	1.0	0	B1004

Data Collection – SpaceX API

```
url="https://api.spacexdata.com/v4/launches/past"
```

- Data collection with SpaceX REST calls using key phrases and flowcharts:

→ Import Libraries and Define Auxiliary Functions

→ Request and parse the SpaceX launch data using the GET request

→ Filter the dataframe to only include ”`Falcon 9`” launches

- GitHub URL of the completed SpaceX API calls notebook:
  https://github.com/UdoRiegler/IBM_10_Capstone/blob/main/jupyter-labs-spacex-data-collection-api_solved.ipynb

Finally lets construct our dataset using the data we have obtained. We we combine the columns into a dictionary.

```
In [21]: launch_dict = {'FlightNumber': list(data_f['flight_number']),
'Date': list(data_f['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}
```

Then, we need to create a Pandas data frame from the dictionary launch_dict.

```
In [22]: # Create a data from launch_dict
df_F = pd.DataFrame(launch_dict)
df_F.head()
```

	FlightNumber	Date	BoosterVersion	PayloadMass	Orbit	LaunchSite	Outcome	Flights	GridFins	Reused	Legs	LandingPad	Block	Reus
4	1	2010-06-04	Falcon 9	NaN	LEO	CCSFS SLC 40	None None	1	False	False	False	None	1.0	
5	2	2012-05-22	Falcon 9	525.0	LEO	CCSFS SLC 40	None None	1	False	False	False	None	1.0	
6	3	2013-03-01	Falcon 9	677.0	ISS	CCSFS SLC 40	None None	1	False	False	False	None	1.0	
7	4	2013-09-29	Falcon 9	500.0	PO	VAFB SLC 4E	False Ocean	1	False	False	False	None	1.0	
8	5	2013-12-03	Falcon 9	3170.0	GTO	CCSFS SLC 40	None None	1	False	False	False	None	1.0	
...	...	...	...	...	...	...	...	...	...	...	...	...	...	
89	86	2020-09-03	Falcon 9	15600.0	VLEO	KSC LC 39A	True ASDS	2	True	True	True	5e9e3032383ecb6bb234e7ca	5.0	
90	87	2020-10-06	Falcon 9	15600.0	VLEO	KSC LC 39A	True ASDS	3	True	True	True	5e9e3032383ecb6bb234e7ca	5.0	
91	88	2020-10-18	Falcon 9	15600.0	VLEO	KSC LC 39A	True ASDS	6	True	True	True	5e9e3032383ecb6bb234e7ca	5.0	
92	89	2020-10-24	Falcon 9	15600.0	VLEO	CCSFS SLC 40	True ASDS	3	True	True	True	5e9e3033383ecbb9e534e7cc	5.0	
93	90	2020-11-05	Falcon 9	3681.0	MEO	CCSFS SLC 40	True ASDS	1	True	False	True	5e9e3032383ecb6bb234e7ca	5.0	

90 rows × 17 columns

Data Collection - Scraping

- Web scraping process

→ Import Libraries and Define Auxiliary Function

→ Request and parse the SpaceX launch data using the GET requ

→ Filter the data frame to only include ”`Falcon 9`” launches

Now let's start requesting rocket launch data from SpaceX API with the following URL:

```
spacex_url="https://api.spacexdata.com/v4/launches/past"
```

```
response = requests.get(spacex_url)
```

Check the content of the response

```
print(response.content)
```

Task 1: Request and parse the SpaceX launch data using the GET request

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

```
static_json_url='https://cf-courses-data.s3.us.cloud-object-storage.appdomain.cloud/IBM-DS0321EN-SkillsNetwork/datasets/API_ca
```

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

```
response.status_code
```

```
200
```

- GitHub URL of the completed web scraping notebook, as an external reference and peer-review purpose: https://github.com/UdoRiegler/IBM_10_Capstone/blob/main/labs-jupyter-spacex-Data%20wrangling_solved2.ipynb

Data Wrangling

- Objectives:
  - Perform exploratory Data Analysis and determine Training Labels
  - Exploratory Data Analysis
  - Determine Training Labels
- Data wrangling process:

  1: Calculate the number of launches on each site

  2: Calculate the number and occurrence of each orbit

  3: Calculate the number and occurence of mission outcome of the orbits

  4: Create a landing outcome label from Outcome column

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

We can use the following line of code to determine the success rate:

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

```
0.5666666666666667
```

TASK 1: Calculate the number of launches on each site

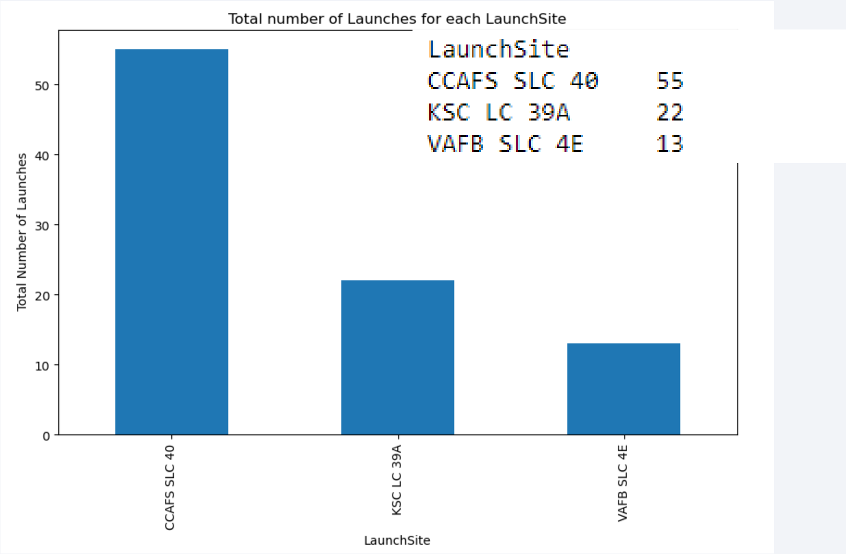
The data contains several Space X launch facilities: Cape Canaveral Space Launch Complex 40 **VAFB SLC 4E** , Vandenberg Air Force Base Space Launch Complex 4E **(SLC-4E)**, Kennedy Space Center Launch Complex 39A **KSC LC 39A** .The location of each Launch Is placed in the column `LaunchSite`

- Add the GitHub URL of your completed data wrangling related notebooks, as an external reference and peer-review purpose:
  https://github.com/UdoRiegler/IBM_10_Capstone/blob/main/Python_01_Collecting_Wrangling_Data.ipynb

EDA with Data Visualization

TASK 1: Visualize the relationship between Flight Number and Launch Site

- The plotted charts show the total success launches by site.



- GitHub URL of completed EDA with data visualization notebook for peer-review purpose: https://github.com/UdoRiegler/IBM_10_Capstone/blob/main/labs-jupyter-spacex-Data%20wrangling_solved3.ipynb

EDA with SQL

- Summarizing the performed SQL queries:
  - The names of the unique launch sites
  - The first 5 records where launch sites begin with `CCA`
  - The total payload mass carried by booster version F9 v1.1
  - The average payload mass carried by booster version F9 v1.1
  - The date of the first successful landing on a ground pad
  - The names of boosters which have successfully landed on drone ship and had payload mass greater than 4000 but less than 6000
  - Calculate the total number of successful and failure mission outcomes
  - List the names of the booster which have carried the maximum payload mass
  - List the failed landing outcomes in drone ship, their booster versions, and launch site names for in year 2015

Build an Interactive Map with Folium

- Summarize what map objects such as markers, circles, lines, etc. you created and added to a folium map
- Explain why you added those objects
- GitHub URL of completed EDA with data visualization notebook for peer-review purpose: https://github.com/UdoRiegler/IBM_10_Capstone/blob/main/

Build a Dashboard with Plotly Dash

- Summarize what plots/graphs and interactions you have added to a dashboard
- Explain why you added those plots and interactions
- GitHub URL of your completed Plotly Dash lab, as an external reference and peer-review purpose: https://github.com/UdoRiegler/IBM_10_Capstone/blob/main/jupyter-labs-eda-sql-coursera_sqllite_solve.ipynb

Predictive Analysis (Classification)

- Summarize how you built, evaluated, improved, and found the best performing classification model
- You need present your model development process using key phrases and flowchart
- GitHub URL of your completed predictive analysis lab, as an external reference and peer-review purpose: https://github.com/UdoRiegler/IBM_10_Capstone/blob/main/SpaceX_Machine_Learning_Prediction_Part_5.jupyterlite_solved.ipynb

Results

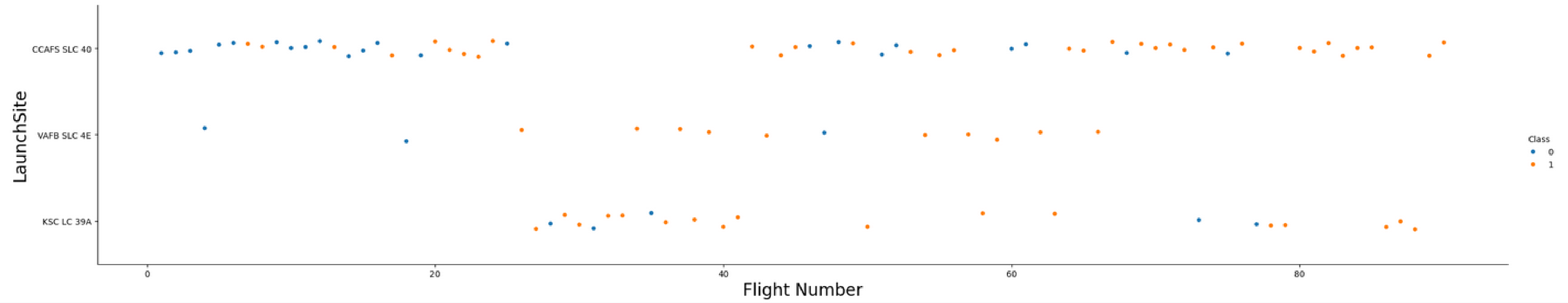
- Exploratory data analysis results
- Interactive analytics demo in screenshots
- Predictive analysis results

Section 2

Insights drawn from EDA

Flight Number vs. Launch Site

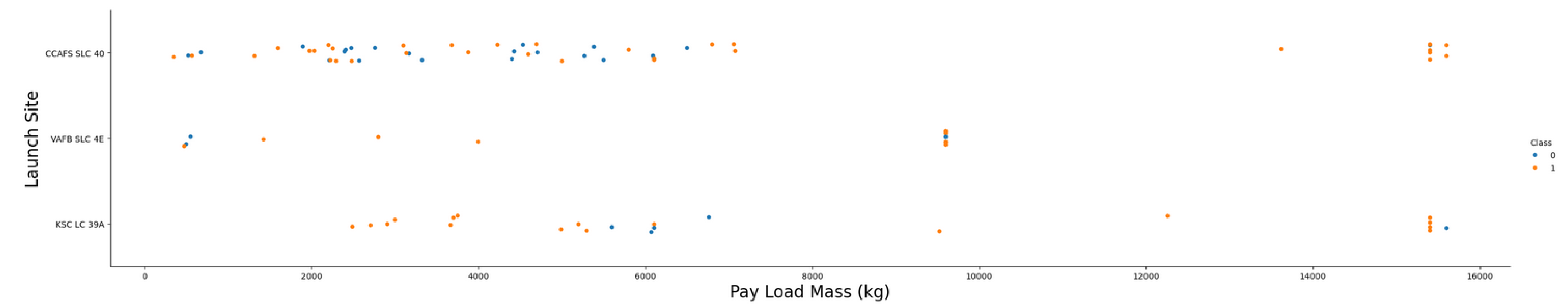
- Scatter plot of Flight Number vs. Launch Site



- The scatter plot shows the flight numbers of all launches, classified as “Success” (orange dots) or “failure” (blue dots), subdivided by the different launch sites (“CCAFS SLC 40”, “VAFB SLC-4E” and “KSC LC 39A”).
  Result: We see that different launch sites have different success rates. “CCAFS LC-40”, has a success rate of 60 %, while “KSC LC-39A” and “VAFB SLC 4E” has a success rate of 77%.

Payload vs. Launch Site

- Show a scatter plot of Payload vs. Launch Site

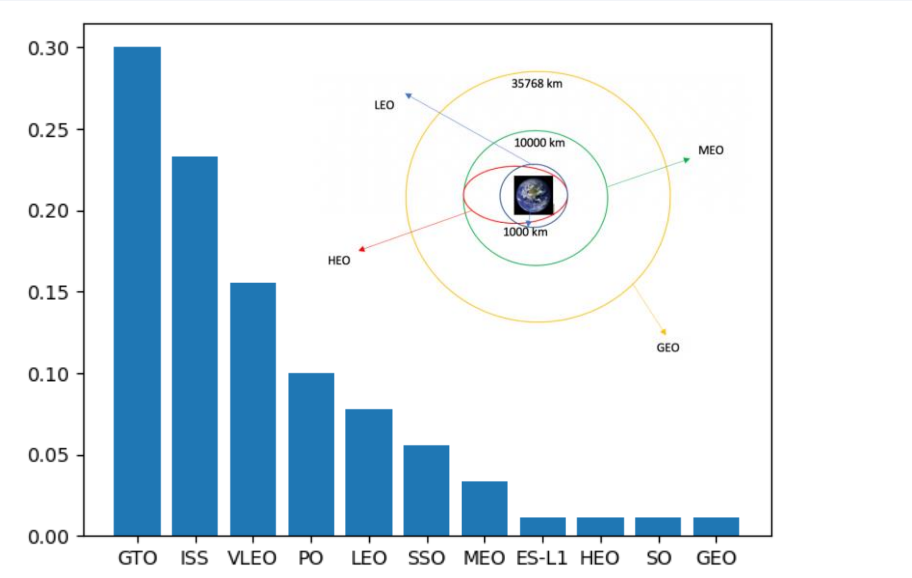


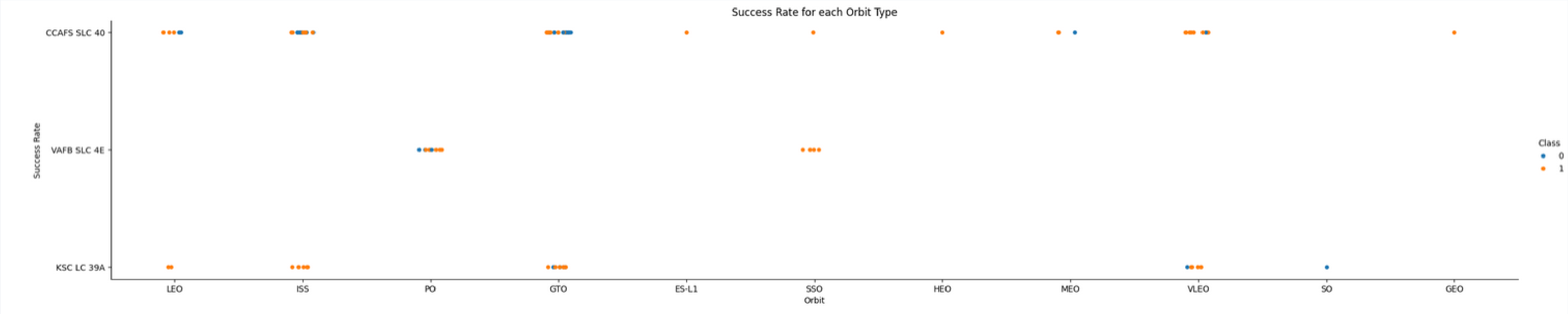
- **Result:** We see that different launch sites have different payload masses. This affects the success rate. “CCAFS LC-40”, has a success rate of 60 %, while “KSC LC-39A” and “VAFB SLC 4E” has a success rate of 77%. The highest payload mass of 16500 kg shows the highest success rate.

Success Rate vs. Orbit Type

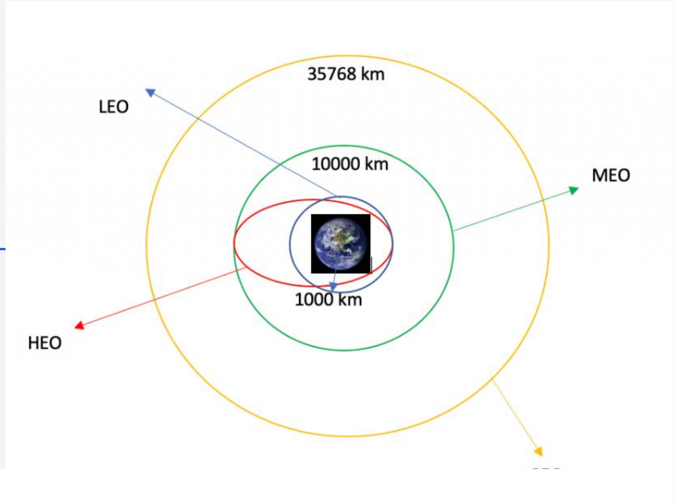
- Bar chart for the success rate of each orbit type:
  - the orbit type “GTO” shows the highest success rate of ~30% followed by “ISS” with ~22%.

- The scatter plot shows the flight numbers of all launches, classified as “Success” (orange dots)
  or “failure” (blue dots), subdivided by the different Orbit types (e.g. “LEO”, “MEO”, “ISS” or “GTO”).

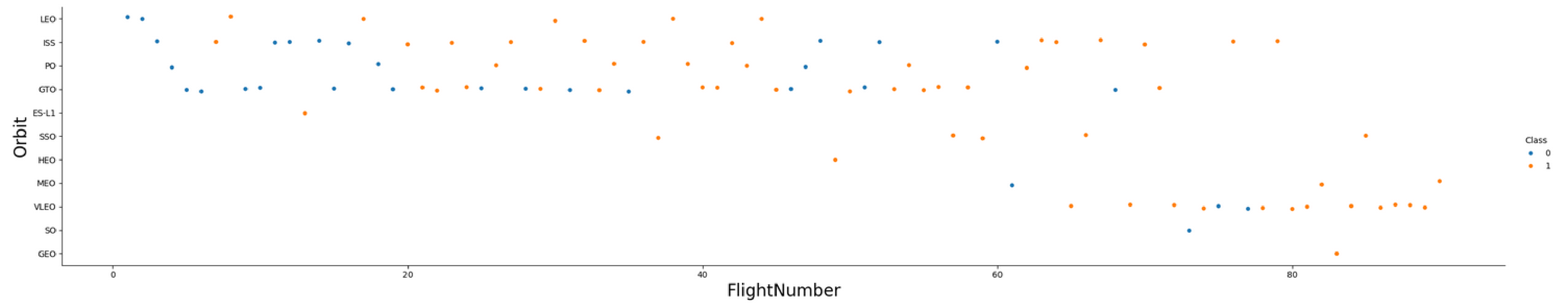




Flight Number vs. Orbit Type



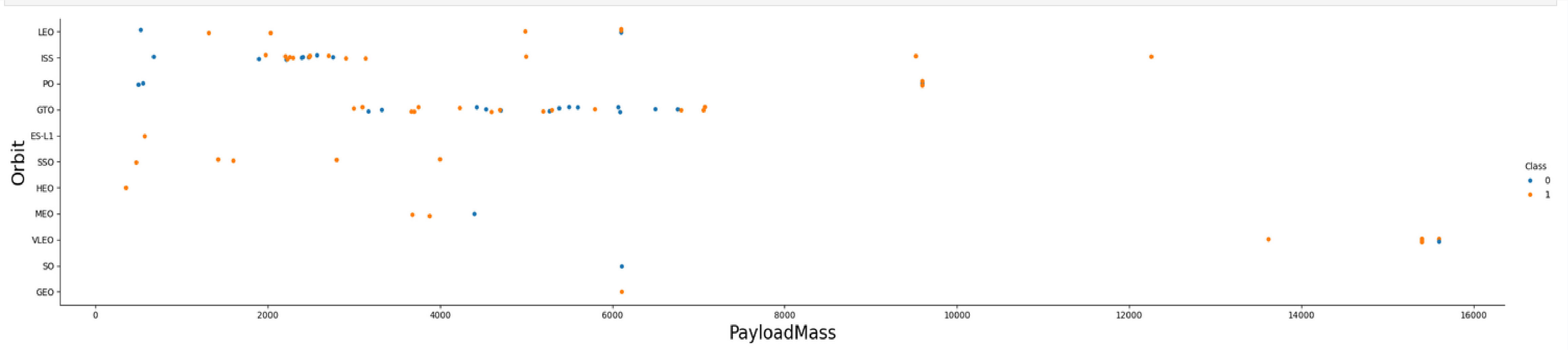
- Show a scatter point of Flight number vs. Orbit type



- The scatter plot shows the flight numbers of all launches, classified as “Success” (orange dots) or “failure” (blue dots), subdivided by the different Orbit types (e.g. “LEO”, “MEO”, “ISS” or “GTO”). **Result:** For 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

- Show a scatter point of payload vs. orbit type:

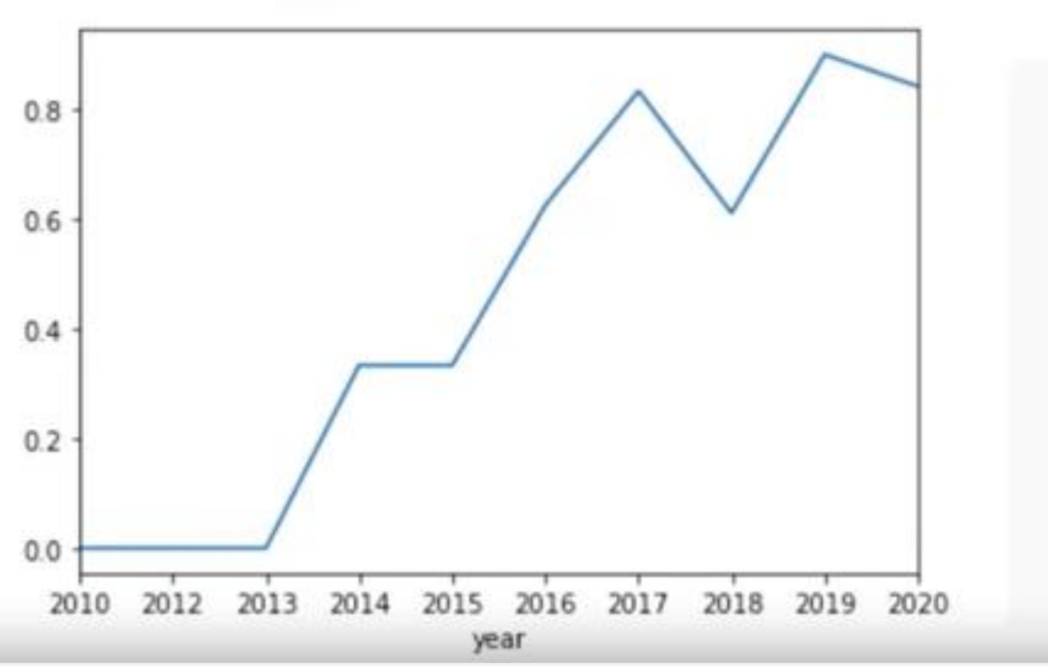


- The scatter plot shows the pay load mass [kg] of all launches, classified as “Success” (orange dots) or “failure” (blue dots), subdivided by the different Orbit types (e.g. “LEO”, “MEO”, “ISS” or “GTO”). **Result:** The higher payload masses between 9000 kg and 16500 kg show the highest success rate. With heavy payloads the successful landing or positive landing rate are more for Polar, LEO and ISS. However for GTO we cannot distinguish this well as both positive landing rate and negative landing (unsuccessful mission) are both there here.

Launch Success Yearly Trend

- The line chart of yearly average success rate is continously increasing from 0% between 2010 and 2013 to ~80% since 2019

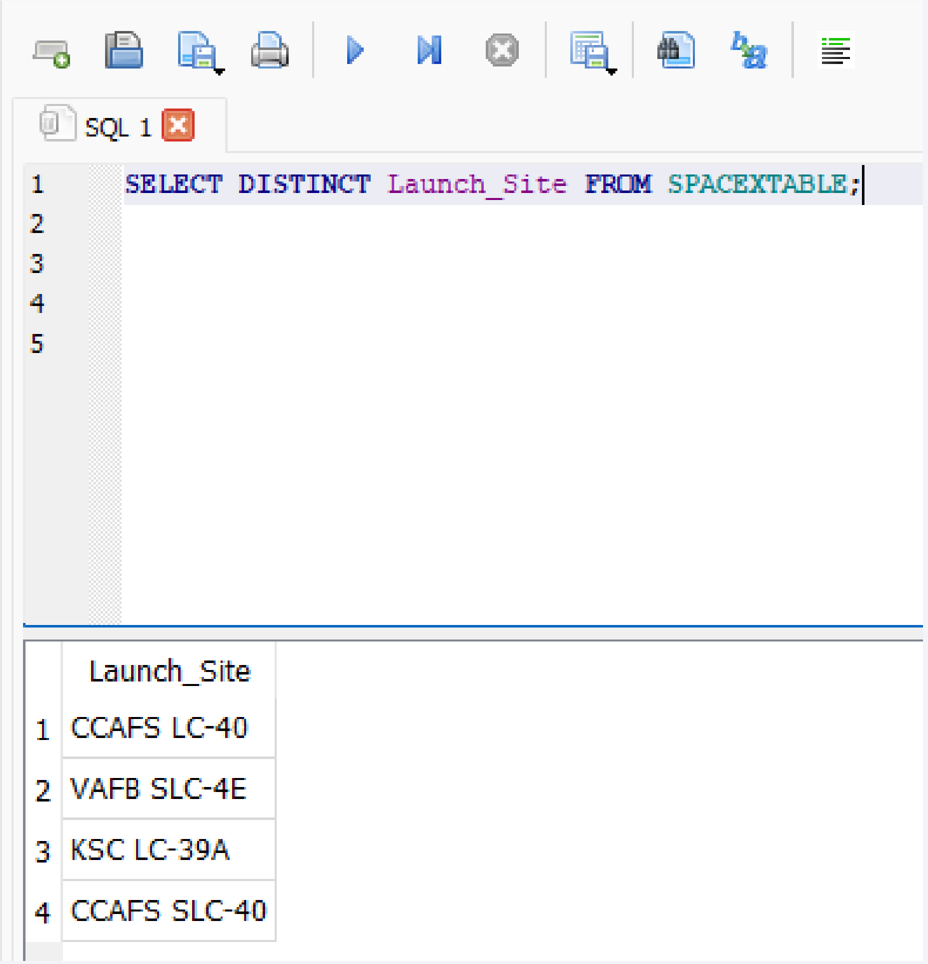
- Show the screenshot of the scatter plot with explanations



All Launch Site Names

- The names of the unique launch sites are:
  - Cape Canaveral Space Launch Complex 40: “CCAFS LC-40”
  - Vandenberg Air Force Base Space Launch Complex 4E : “VAFB SLC-4E”
  - Kennedy Space Center Launch Complex 39A KSC LC 39A : “KSC LC-39A”
  - Cape Canaveral Space Launch Complex SLC-40: “CCAFS SLC-40”

The location of each Launch Is placed in the column ‘Launch_Site’.



Launch Site Names Begin with 'CCA'

- The first 5 records where launch sites begin with `CCA` are dated:
    1. 2010/06/04
    2. 2010/12/08
    3. 2012/05/22
    4. 2012/10/08
    5. 2013/03/01
- Query result with a short explanation here

SQL 1

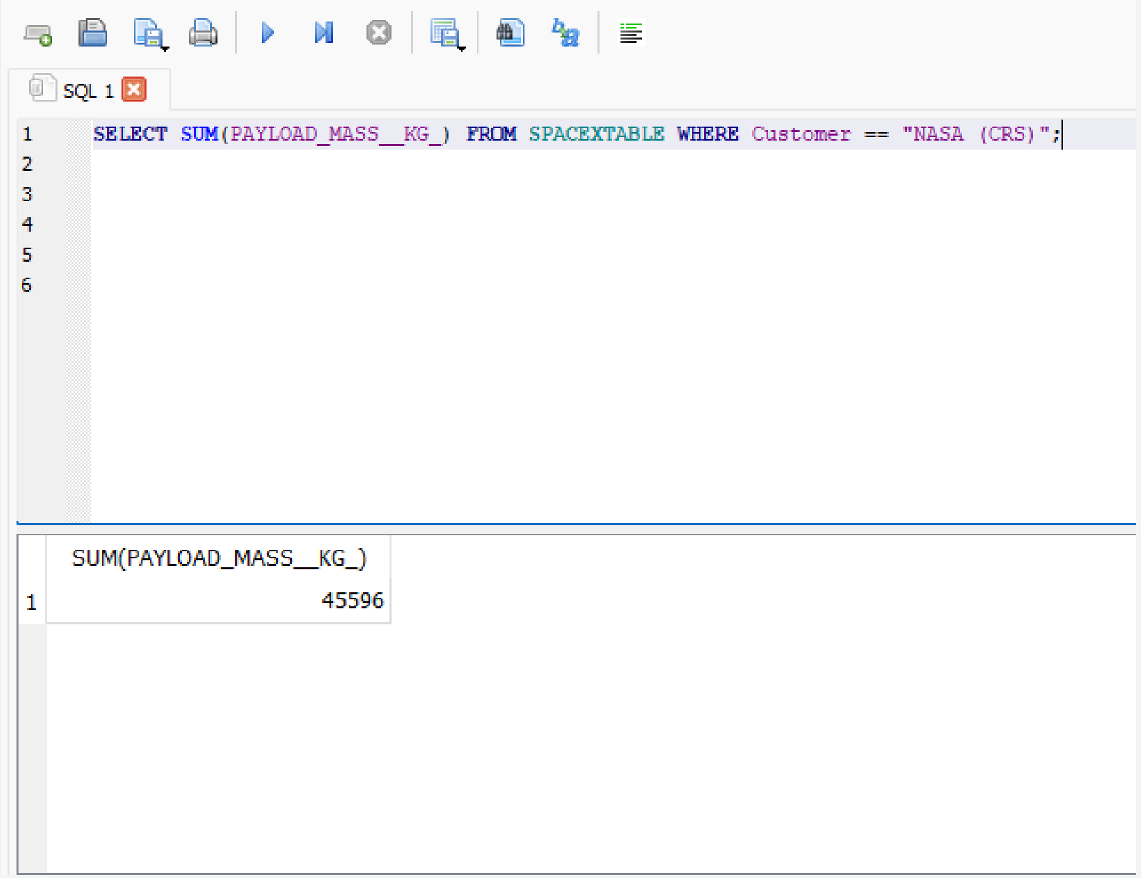
```
SELECT * FROM SPACEXTABLE WHERE Launch_Site == "CCAFS LC-40" LIMIT 5;
```

	Date	Time (UTC)	Booster_Version	Launch_Site	Payload
1	2010-06-04	18:45:00	F9 v1.0 B0003	CCAFS LC-40	Dragon Spacecraft Qualification Unit
2	2010-12-08	15:43:00	F9 v1.0 B0004	CCAFS LC-40	Dragon demo flight C1, two CubeSats, barrel of ...
3	2012-05-22	7:44:00	F9 v1.0 B0005	CCAFS LC-40	Dragon demo flight C2
4	2012-10-08	0:35:00	F9 v1.0 B0006	CCAFS LC-40	SpaceX CRS-1
5	2013-03-01	15:10:00	F9 v1.0 B0007	CCAFS LC-40	SpaceX CRS-2

Total Payload Mass

- Calculate the total payload carried by boosters from NASA

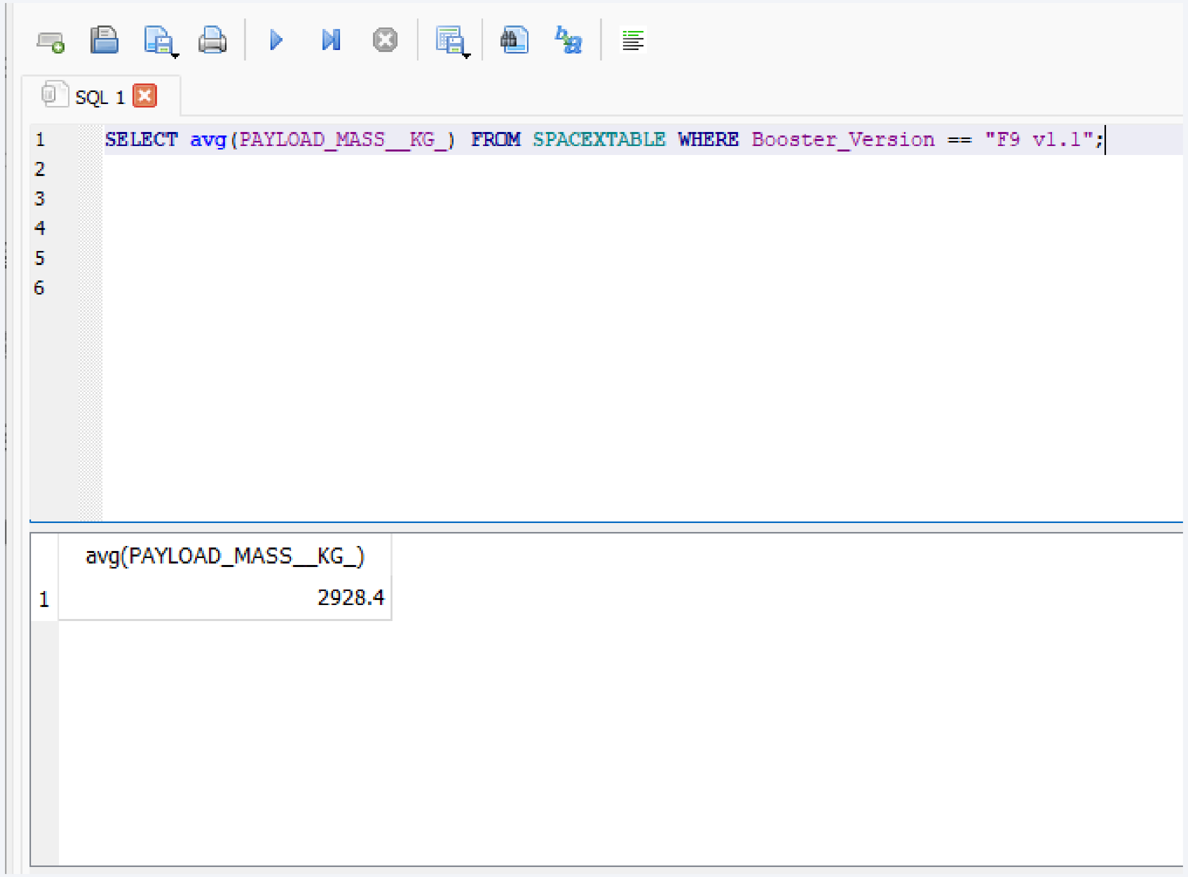
- The total payload mass carried by booster version F9 v1.1 is 45,596 kg.



Average Payload Mass by F9 v1.1

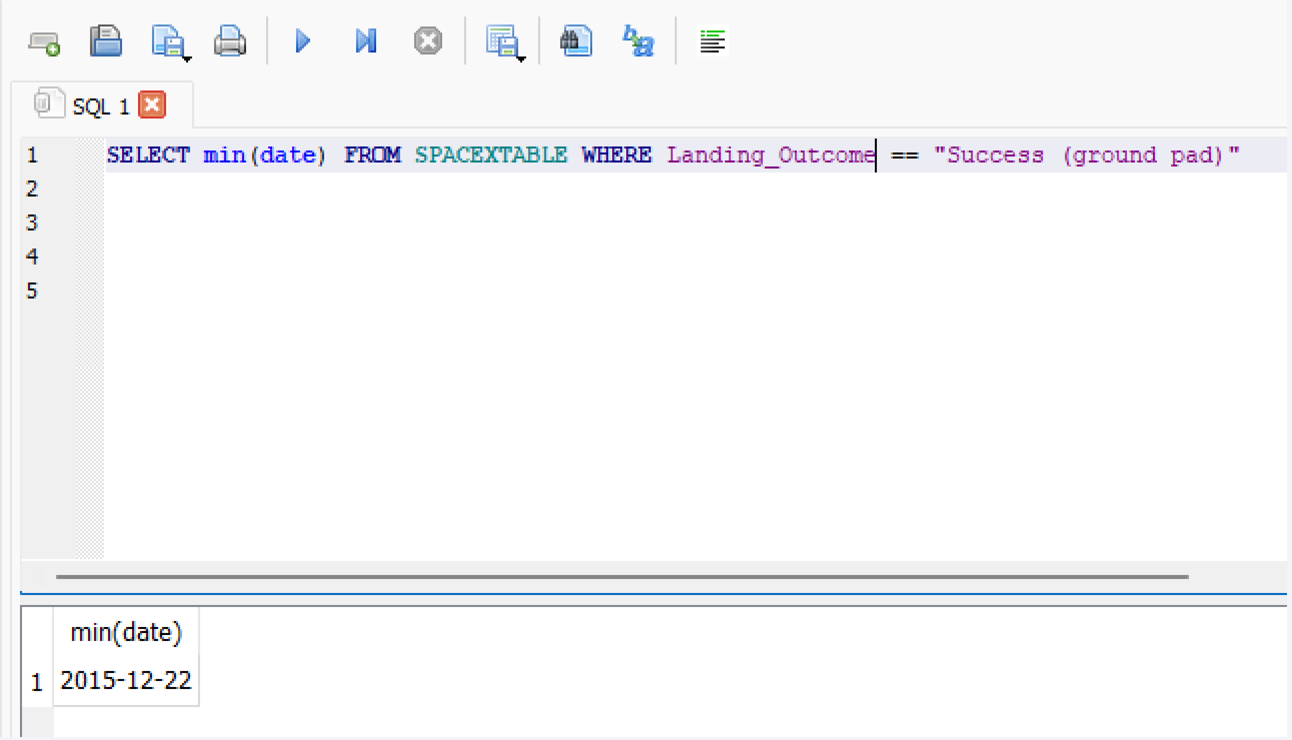
- Calculate the average payload mass carried by booster version F9 v1.1

- The average payload mass carried by booster version F9 v1.1 is 2928.4 kg.



First Successful Ground Landing Date

- Find the dates of the first successful landing outcome on ground pad:
- The date of the first successful landing on a ground pad is 12/22/2015.



Successful Drone Ship Landing with Payload between 4000 and 6000

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

- Result:
  4 booster version listed in the table were able to successfully landed on drone ship and had payload mass greater than 4000 but less than 6000.

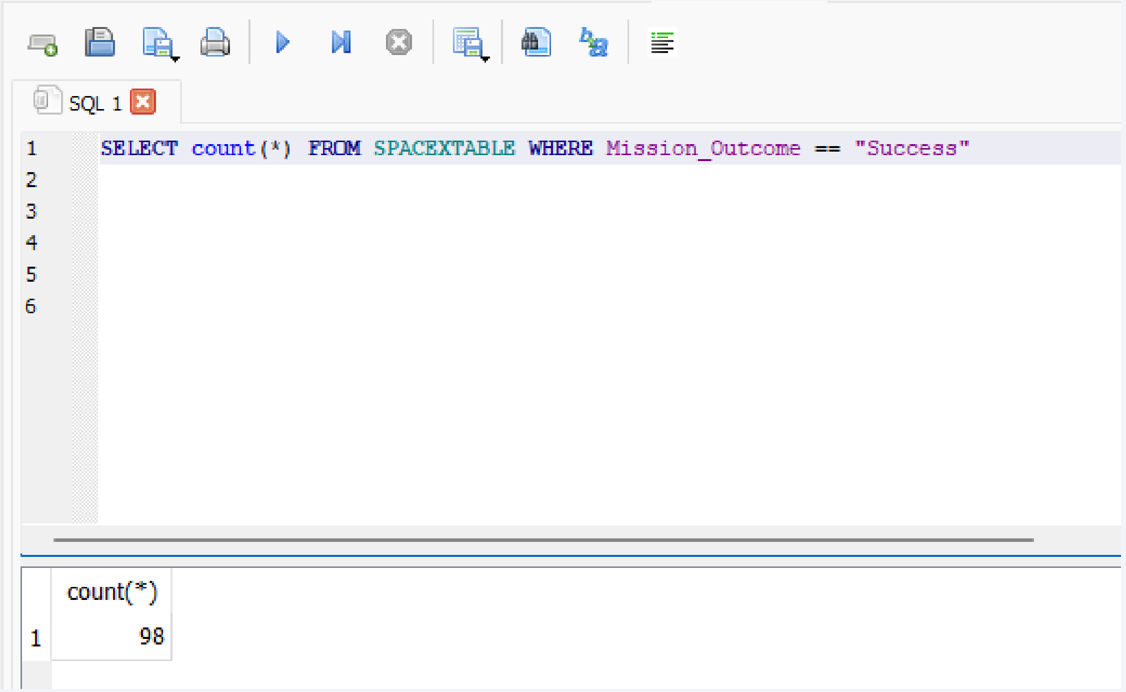
SQL 1

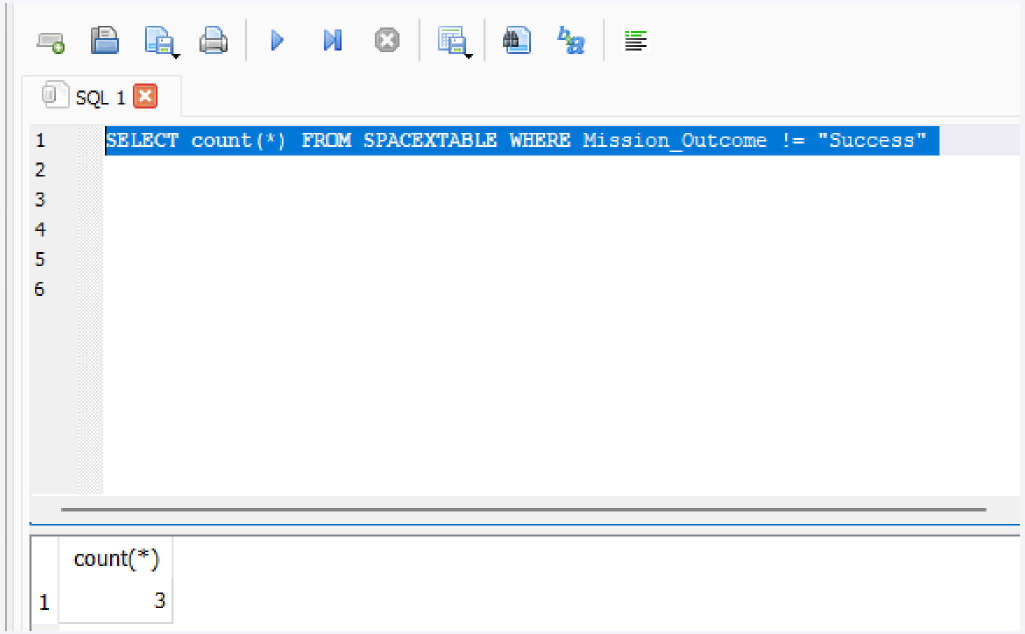
```
SELECT DISTINCT Booster_Version FROM SPACEXTABLE WHERE Landing_Outcome == "Success (drone ship)"
AND PAYLOAD_MASS__KG_ > 4000 AND PAYLOAD_MASS__KG_ < 6000
```

	Booster_Version
1	F9 FT B1022
2	F9 FT B1026
3	F9 FT B1021.2
4	F9 FT B1031.2

Total Number of Successful and Failure Mission Outcomes

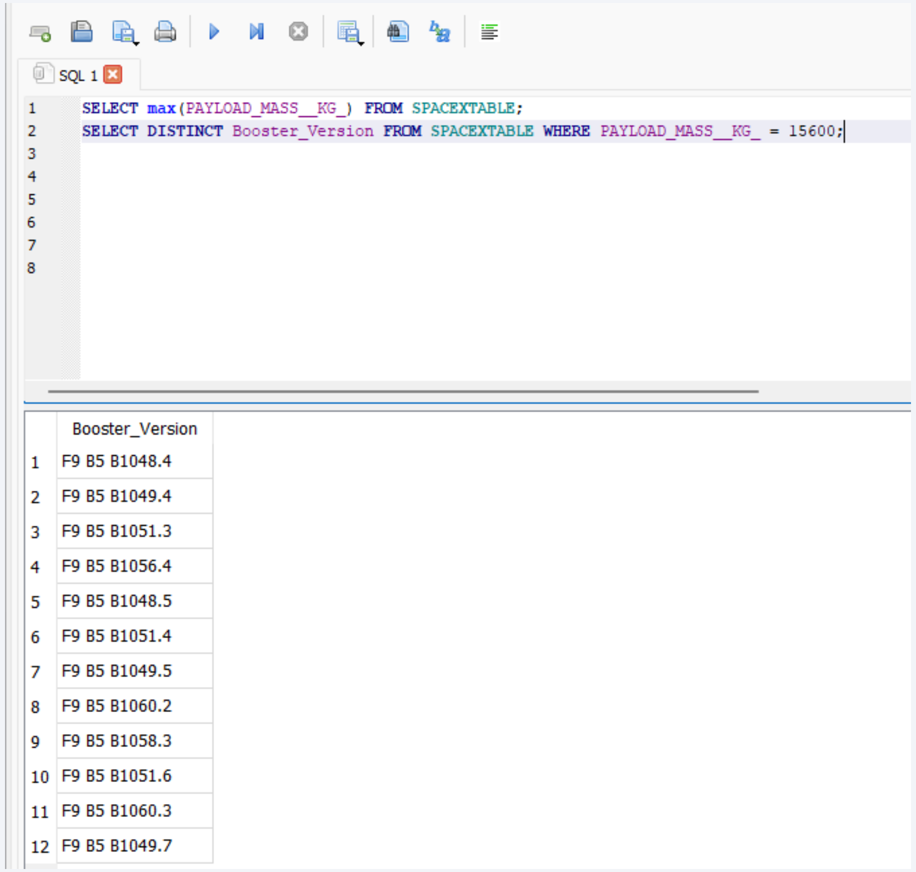
- Calculate the total number of successful and failure mission outcomes
- Present your query result with a short explanation here





Boosters Carried Maximum Payload

- List the names of the booster which have carried the maximum payload mass
- The following 12 listed booster versions have carried the maximum payload mass of 15600 kg




```sql
SELECT max(PAYLOAD_MASS__KG_) FROM SPACEXTABLE;
SELECT DISTINCT Booster_Version FROM SPACEXTABLE WHERE PAYLOAD_MASS__KG_ = 15600;
```

	Booster_Version
1	F9 B5 B1048.4
2	F9 B5 B1049.4
3	F9 B5 B1051.3
4	F9 B5 B1056.4
5	F9 B5 B1048.5
6	F9 B5 B1051.4
7	F9 B5 B1049.5
8	F9 B5 B1060.2
9	F9 B5 B1058.3
10	F9 B5 B1051.6
11	F9 B5 B1060.3
12	F9 B5 B1049.7

2015 Launch Records

- List the failed landing_outcomes in drone ship, their booster versions, and launch site names for in year 2015

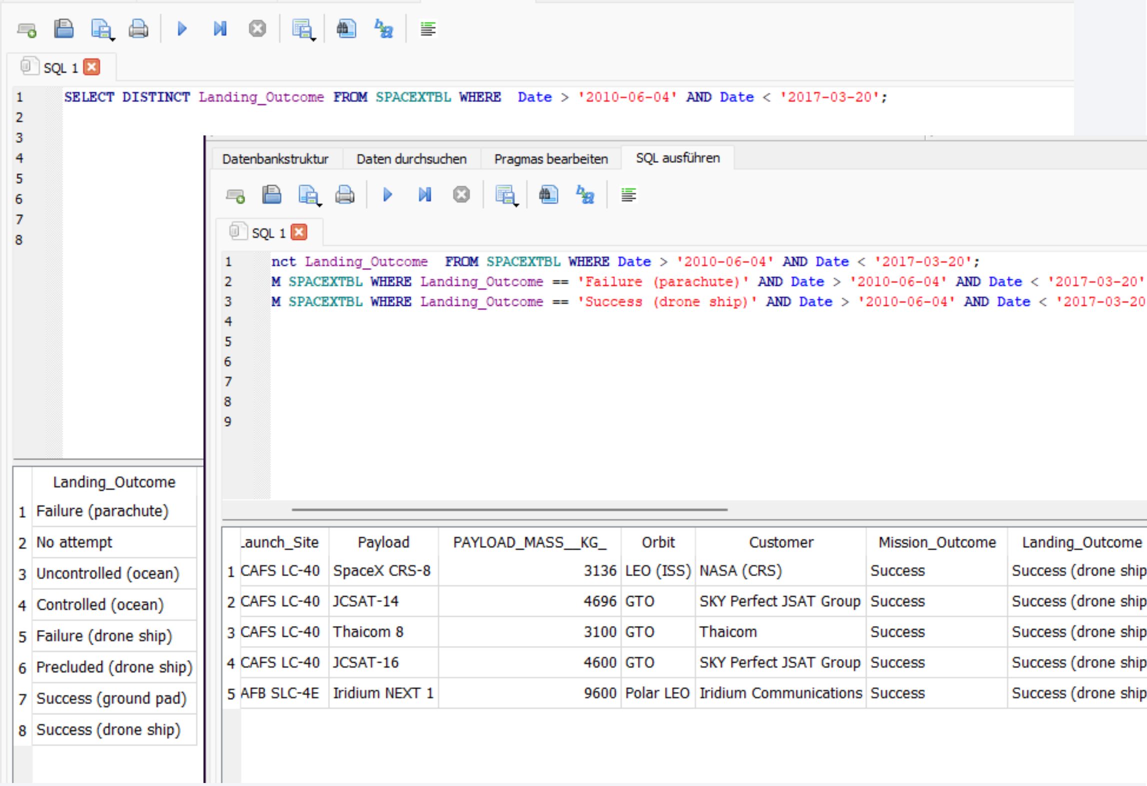
- The following five launches had 2015 a failed landing outcome in drone ship

```sql
SELECT Orbit, Booster_Version, Launch_Site FROM SPACEXTBL WHERE Landing_Outcome != "Failure (drone ship)" AND Date > '2015-01-01' AND Date < '2015-12-31';
```

	Orbit	Booster_Version	Launch_Site
1	HEO	F9 v1.1 B1013	CCAFS LC-40
2	GTO	F9 v1.1 B1014	CCAFS LC-40
3	GTO	F9 v1.1 B1016	CCAFS LC-40
4	LEO (ISS)	F9 v1.1 B1018	CCAFS LC-40
5	LEO	F9 FT B1019	CCAFS LC-40

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

- 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
- Query result: there are 8 different types of landing outcomes like: “Failure (parachute)”, “Uncontrolled (ocean)” or “Success (ground pad”.
- In the given period there were 1 launch ending in a “Failure (parachute)” and 5 successful launches with landing on a drone ship.



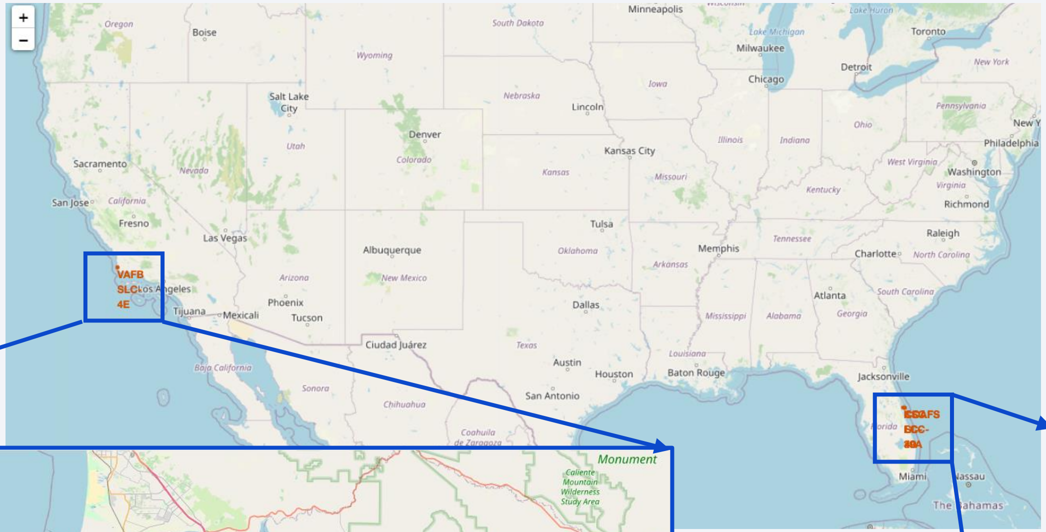
	Landing_Outcome
1	Failure (parachute)
2	No attempt
3	Uncontrolled (ocean)
4	Controlled (ocean)
5	Failure (drone ship)
6	Precluded (drone ship)
7	Success (ground pad)
8	Success (drone ship)

	.aunch_Site	Payload	PAYLOAD_MASS__KG_	Orbit	Customer	Mission_Outcome	Landing_Outcome
1	CAFS LC-40	SpaceX CRS-8	3136	LEO (ISS)	NASA (CRS)	Success	Success (drone ship
2	CAFS LC-40	JCSAT-14	4696	GTO	SKY Perfect JSAT Group	Success	Success (drone ship
3	CAFS LC-40	Thaicom 8	3100	GTO	Thaicom	Success	Success (drone ship
4	CAFS LC-40	JCSAT-16	4600	GTO	SKY Perfect JSAT Group	Success	Success (drone ship
5	AFB SLC-4E	Iridium NEXT 1	9600	Polar LEO	Iridium Communications	Success	Success (drone ship

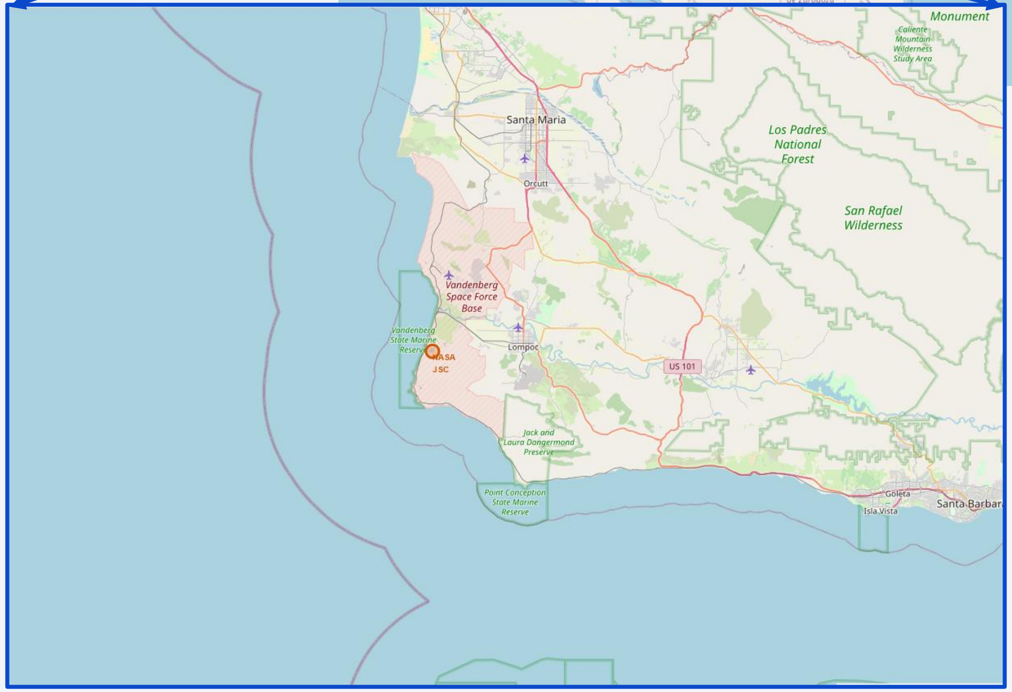
Section 3

Launch Sites Proximities Analysis

Global Position of Launching Sites (“CCAFS SLC 40”, “VAFB SLC-4E”)

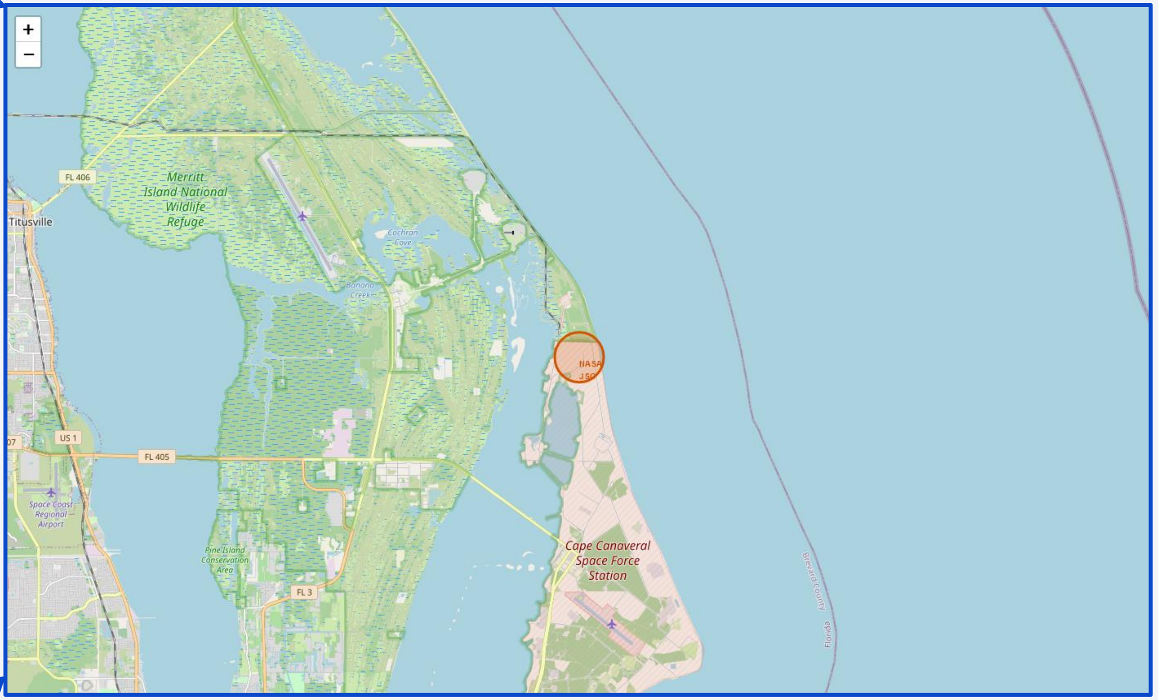


→ All launch sites are placed in proximity to the Equator line and the sea coast.

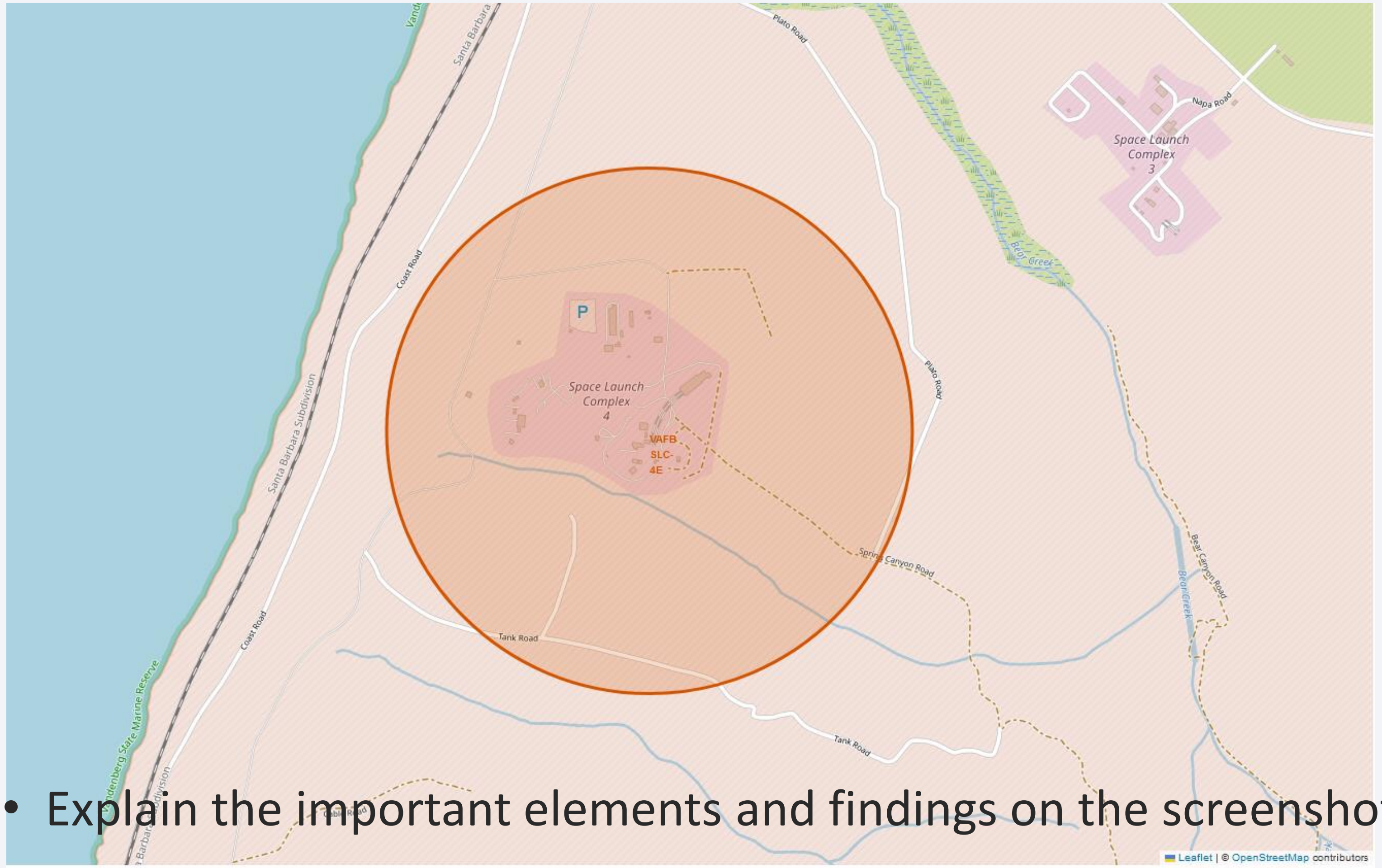


Cape Canaveral Space Launch Complex SLC-40

Kennedy



Vandenberg Air Force Base Space Launch Complex 4E : “VAFB SLC-4E”

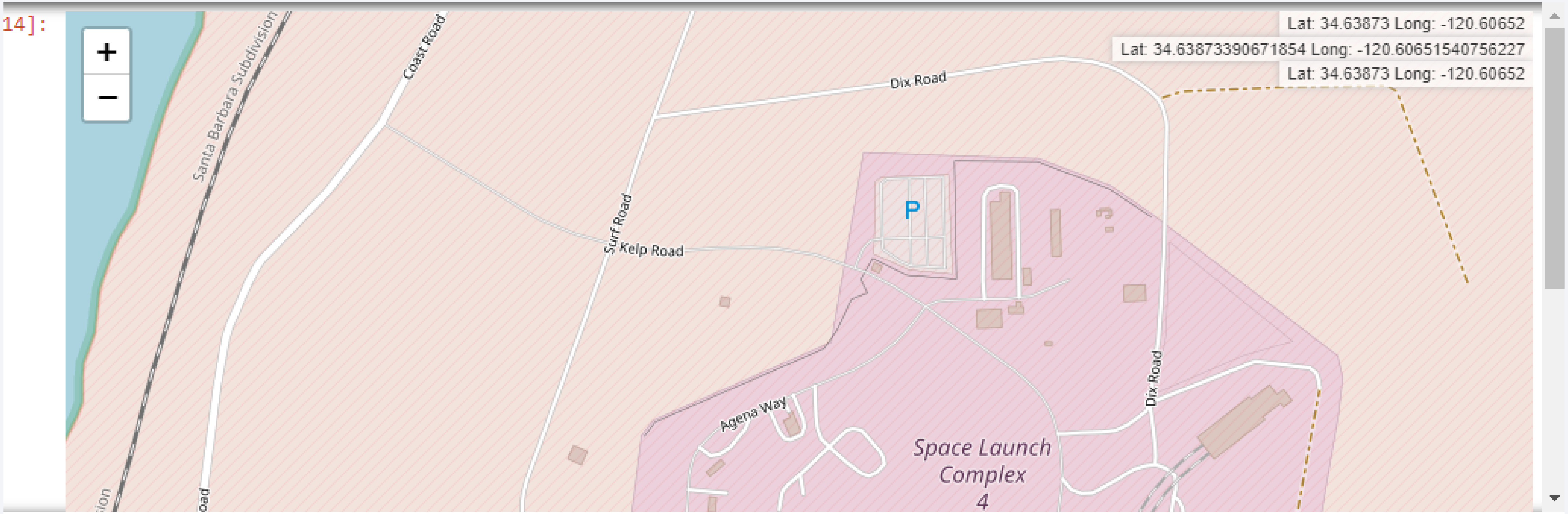


	Launch Site	Lat	Long	class
26	VAFB SLC-4E	34.632834	-120.610745	0
27	VAFB SLC-4E	34.632834	-120.610745	0
28	VAFB SLC-4E	34.632834	-120.610745	1
29	VAFB SLC-4E	34.632834	-120.610745	1
30	VAFB SLC-4E	34.632834	-120.610745	1
31	VAFB SLC-4E	34.632834	-120.610745	1
32	VAFB SLC-4E	34.632834	-120.610745	0
33	VAFB SLC-4E	34.632834	-120.610745	0
34	VAFB SLC-4E	34.632834	-120.610745	0
35	VAFB SLC-4E	34.632834	-120.610745	0

- Explain the important elements and findings on the screenshot

→ „VAFB SLC-4E launch site is located in proximity of an railway line and the sea coast.

Vandenberg Air Force Base Space Launch Complex 4E : “VAFB SLC-4E”



- The “VAFB SLC-4E” launch site is situated near railway and coastline.

Section 4

Build a Dashboard with Plotly Dash

<Dashboard Screenshot 1>

- Replace <Dashboard screenshot 1> title with an appropriate title
- Show the screenshot of launch success count for all sites, in a piechart
- Explain the important elements and findings on the screenshot

<Dashboard Screenshot 2>

- Replace <Dashboard screenshot 2> title with an appropriate title
- Show the screenshot of the piechart for the launch site with highest launch success ratio
- Explain the important elements and findings on the screenshot

<Dashboard Screenshot 3>

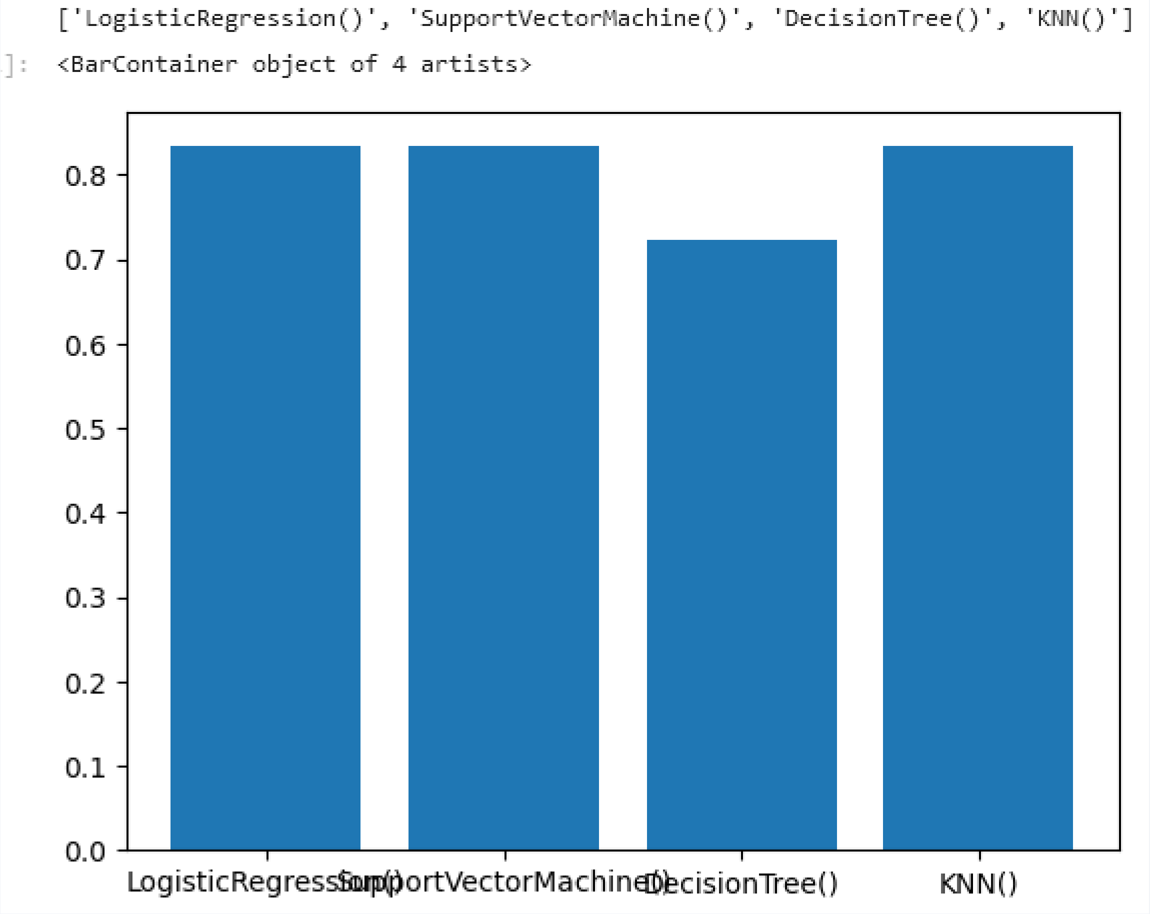
- Replace <Dashboard screenshot 3> title with an appropriate title
- Show screenshots of Payload vs. Launch Outcome scatter plot for all sites, with different payload selected in the range slider
- Explain the important elements and findings on the screenshot, such as which payload range or booster version have the largest success rate, etc.

Section 5

Predictive Analysis (Classification)

Classification Accuracy

- The built model accuracy for all built classification models is shown in the following bar chart. Logistic Regression and KNN are my favorite models, having the highest classification accuracy ~0.833.



https://github.com/UdoRiegler/IBM_10_Capstone/blob/main/SpaceX_Machine_Learning_Prediction_Part_5.jupyterlite_solved.ipynb

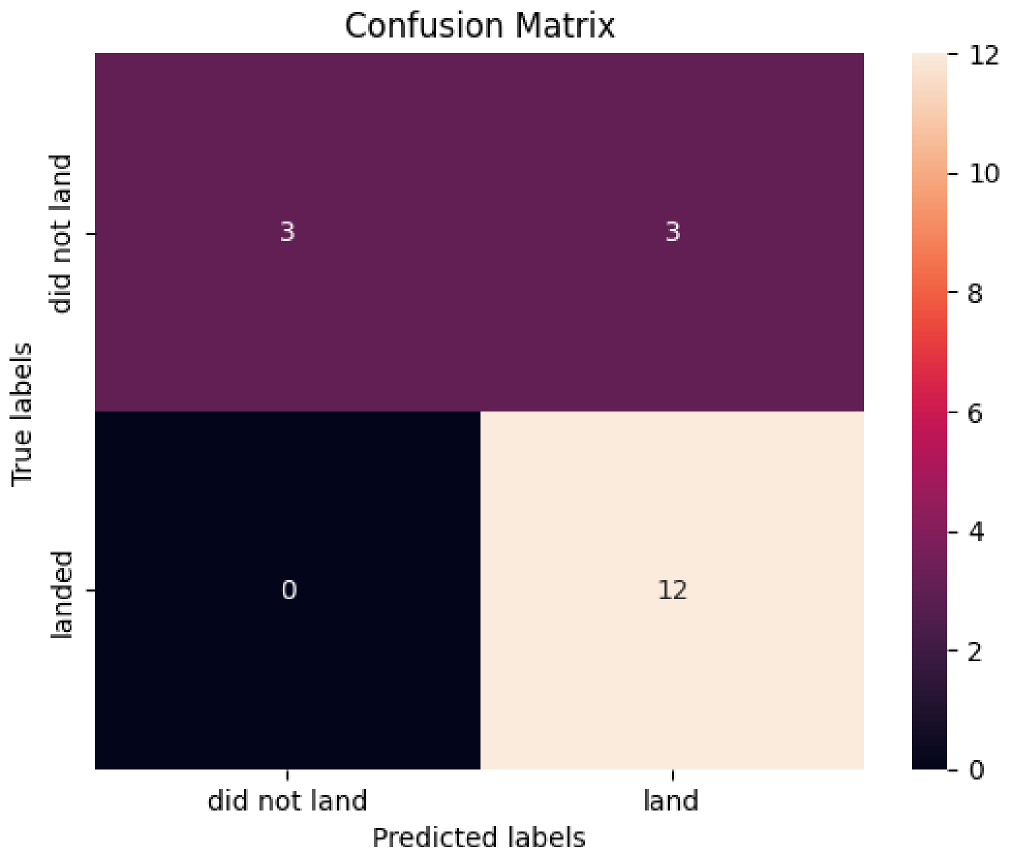
Confusion Matrix

- The confusion matrix of the best

performing model is shown here.

Examining the confusion matrix, we see that logistic regression and KNN can distinguish between the different classes. We see that the major problem is false positives.

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



Conclusions

- The success rate recovering the first stage of SpaceX is stable > 70% since 2019
- This implies that every competitor like “SpaceY” must be able to over launchings at a competitive price level to SpaceX.
- The significant cost saving due to reuse of the first stage will soon be state of the art
- SpaceY should focus on the needed technology to offer similar features or to go beyond the level of SpaceX.
- This facts should be know by the management of SpaceY before investing in testing capabilities, building and different samples of the launching system and all related parts.

Appendix

- Include any relevant assets like Python code snippets, SQL queries, charts, Notebook outputs, or data sets that you may have created during this project

Thank you!