Space Race with Data Science

IBM Developer
SKILLS NETWORK

OUTLINE

- Executive summary
- Introduction
- Methodology
- Results
- Conclusion
- Appendix

EXECUTIVE SUMMARY

- Data collection and data wrangling methodology related
- EDA and interactive visual analytics methodology related
- Predictive analysis methodology related
- EDA with visualization result
- EDA with SQL result
- Interactive map with Folium result
- Plotly Dash dashboard result
- Predictive analysis (classification) results
- Conclusion
- Innovative insights

INTRODUCTION

- Project Background
  - In this capstone, we will predict if the Falcon 9 first stage will land successfully. 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 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 SpaceX for a rocket launch. This course allows learners to apply data science techniques to real-world problems, providing a portfolio-worthy project that demonstrates their ability to solve complex data challenges.

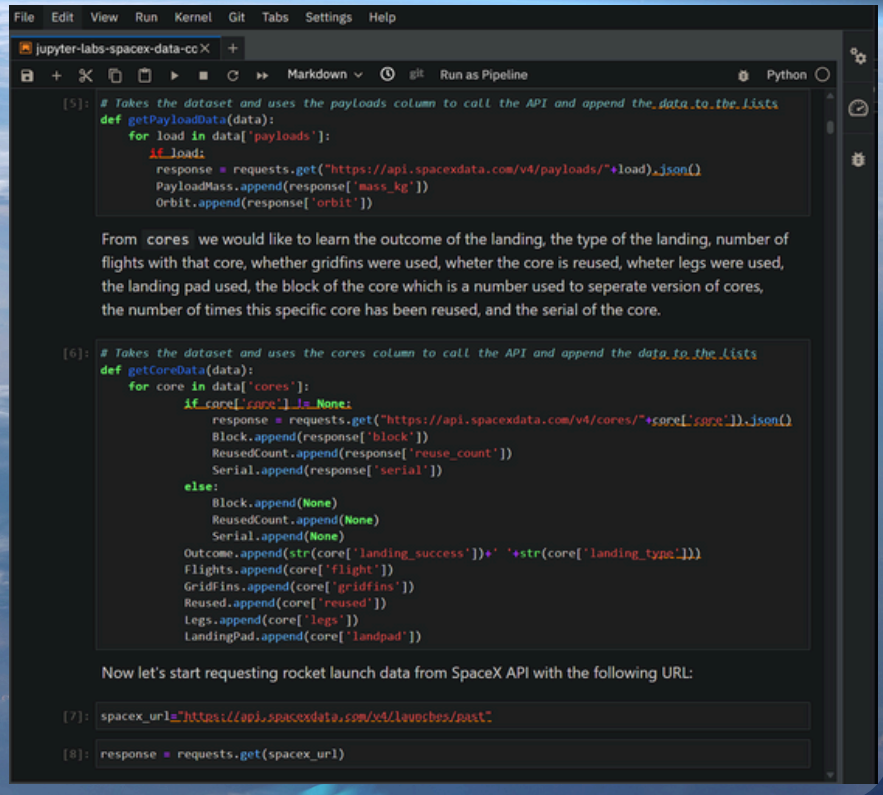
METHODOLOGY

DATA COLLECTION AND DATA WRANGLING METHODOLOGY

- Data Collection
  - Identify the dataset and related data.
  - Requesting API from SpaceX.
  - Ensuring the data format, structure, and data collected are accurate.
-
- Data Wrangling
  - Exploratory Data Analysis
  - Determine Training Labels

DATA COLLECTION AND DATA WRANGLING METHODOLOGY

- Data Collection
  - Identify the dataset and related data.
  - Requesting API from SpaceX.



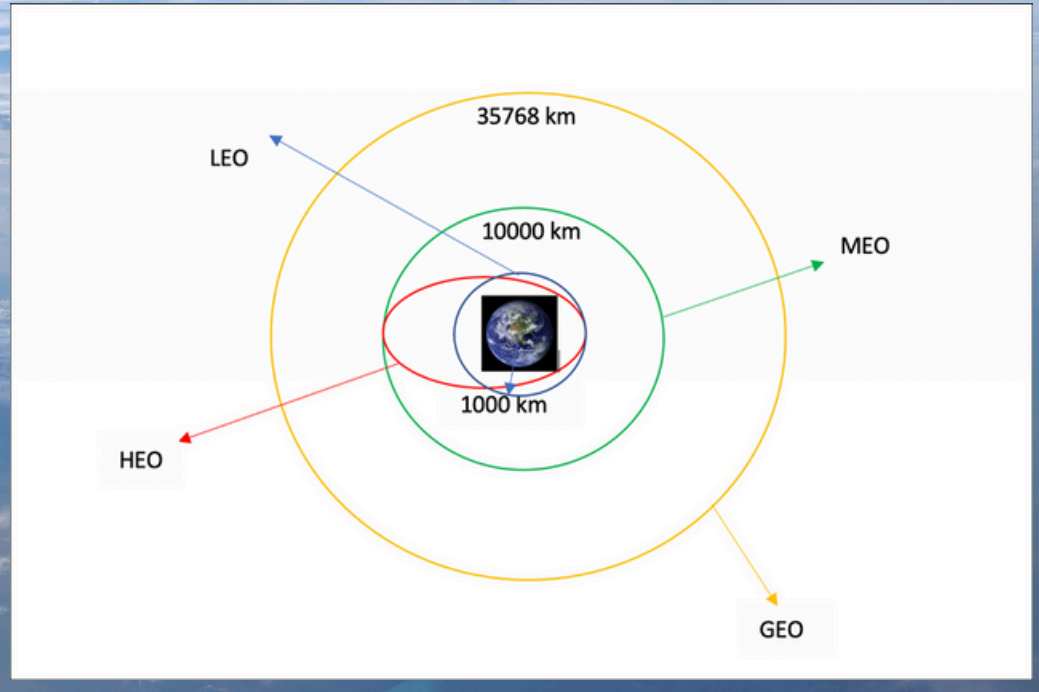
DATA COLLECTION AND DATA WRANGLING METHODOLOGY

- Data Collection
  - Ensuring the data format, structure, and data collected are accurate.

```
# Lets take a subset of our dataframe keeping only the features we want and
the flight number, and date_utc.
data = data[['rocket', 'payloads', 'launchpad', 'cores', 'flight_number',
'date_utc']]

# We will remove rows with multiple cores because those are falcon rockets
with 2 extra rocket boosters and rows that have multiple payloads in a single
rocket.
data = data[data['cores'].map(len)==1]
data = data[data['payloads'].map(len)==1]

# Since payloads and cores are lists of size 1 we will also extract the single
value in the list and replace the feature.
data['cores'] = data['cores'].map(lambda x : x[0])
data['payloads'] = data['payloads'].map(lambda x : x[0])

# We also want to convert the date_utc to a datetime datatype and then
extracting the date leaving the time
data['date'] = pd.to_datetime(data['date_utc']).dt.date

# Using the date we will restrict the dates of the launches
data = data[data['date'] <= datetime.date(2020, 11, 13)]
```

DATA COLLECTION AND DATA WRANGLING METHODOLOGY

- Data Wrangling
  - Exploratory Data Analysis
  - Determine Training Labels



EDA AND INTERACTIVE VISUAL ANALYTICS METHODOLOGY

Exploratory Data Analysis is the first step of any data science project. In the first lab, we will perform some Exploratory Data Analysis using a database. In the second lab, we will see if the data can be used to automatically determine if the Falcon 9's first stage will land.

- Predictive Analysis Methodology
- EDA with Visualization
- Complete EDA with SQL

EDA AND INTERACTIVE VISUAL ANALYTICS METHODOLOGY

- **Predictive Analysis Methodology**

Exploratory Data Analysis

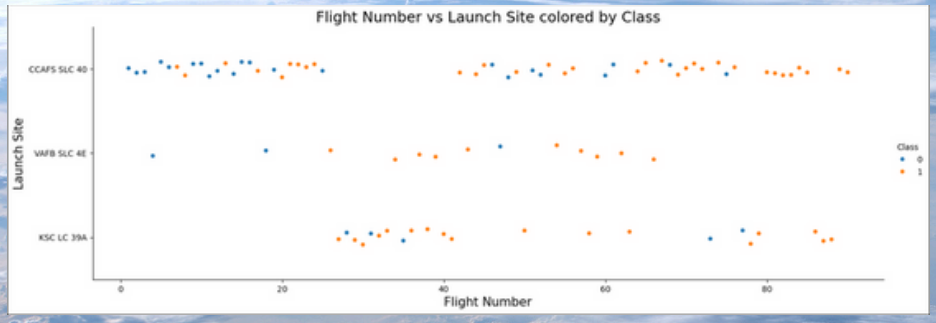
First, let's read the SpaceX dataset into a Pandas dataframe and print its summary

```
from js import fetch
import io

URL = "https://cf-courses-data.s3.us.cloud-object-storage.appdomain.cloud/IBM-DS0321EN-SkillsNetwork/datasets/dataset_part_2.csv"
resp = await fetch(URL)
dataset_part_2_csv = io.BytesIO((await resp.arrayBuffer()).to_py())
df=pd.read_csv(dataset_part_2_csv)
df.head(5)
```

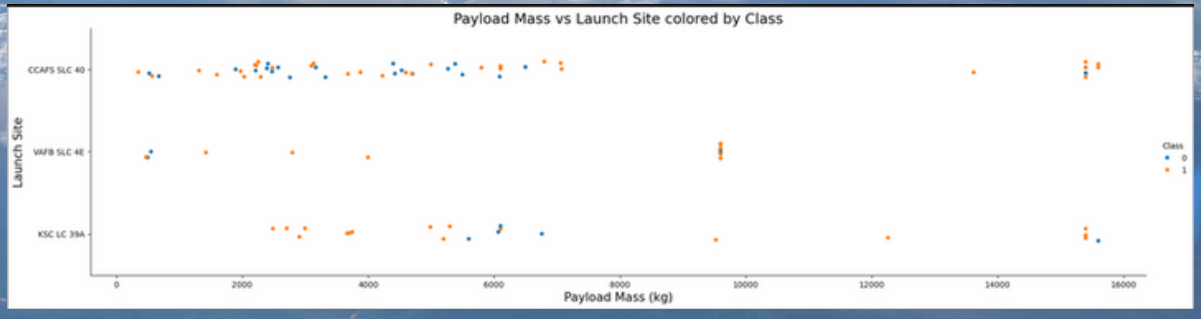
	FlightNumber	Date	BoosterVersion	PayloadMass	Orbit	LaunchSite	Outcome	Flights	GridFins	Reused	Legs	LandingPad	Block	ReusedCount	Serial	Longitude	Latitude	Class
0	1	2010-06-04	Falcon 9	6104.959412	LEO	CCAFS SLC 40	None None	1	False	False	False	NaN	1.0	0	B0003	-80.577366	28.561857	0
1	2	2012-05-22	Falcon 9	525.000000	LEO	CCAFS SLC 40	None None	1	False	False	False	NaN	1.0	0	B0005	-80.577366	28.561857	0
2	3	2013-03-01	Falcon 9	677.000000	ISS	CCAFS SLC 40	None None	1	False	False	False	NaN	1.0	0	B0007	-80.577366	28.561857	0
3	4	2013-09-29	Falcon 9	500.000000	PO	VAFB SLC 4E	False Ocean	1	False	False	False	NaN	1.0	0	B1003	-120.610829	34.632093	0
4	5	2013-12-03	Falcon 9	3170.000000	GTO	CCAFS SLC 40	None None	1	False	False	False	NaN	1.0	0	B1004	-80.577366	28.561857	0

EDA AND INTERACTIVE VISUAL ANALYTICS METHODOLOGY

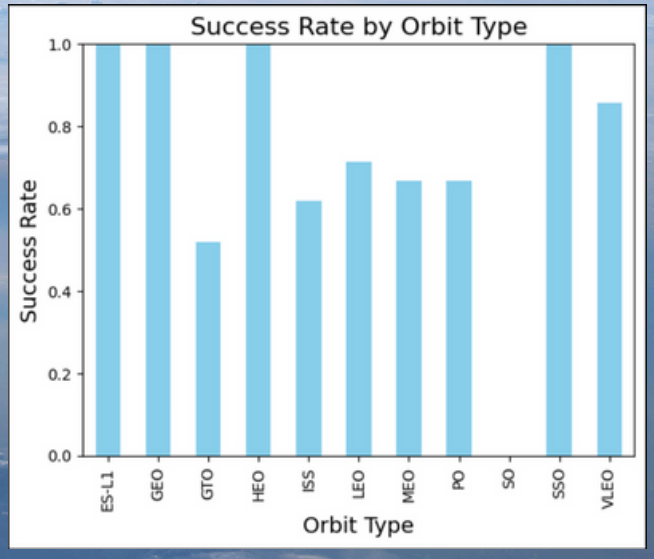
- **EDA with Visualization**
  - Visualize the relationship between Flight Number and Launch Site



  - Visualize the relationship between Payload Mass and Launch Site

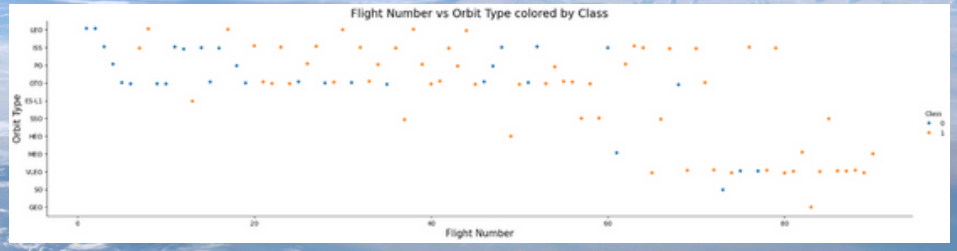


  - Visualize the relationship between success rate of each orbit type

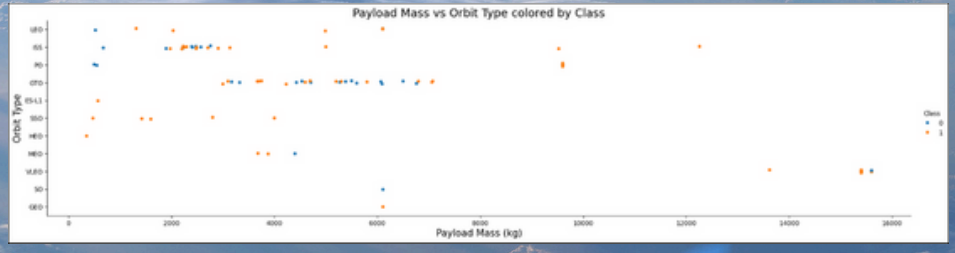


EDA AND INTERACTIVE VISUAL ANALYTICS METHODOLOGY

- EDA with Visualization
  - Visualize the relationship between Flight Number and Orbit type



  - Visualize the relationship between Payload and Orbit type



  - Visualize the Launch success yearly trend

	FlightNumber	Date	BoosterVersion	PayloadMass	Orbit	LaunchSite	Outcome	Flights	GridFins	Reused	Legs	LandingPad	Block	ReusedCount	Serial	Longitude	Latitude	Class
0	1	2010	Falcon 9	6104.959412	LEO	CCAFS SLC 40	None None	1	False	False	False	NaN	1.0	0	B0003	-80.577366	28.561857	0
1	2	2012	Falcon 9	525.000000	LEO	CCAFS SLC 40	None None	1	False	False	False	NaN	1.0	0	B0005	-80.577366	28.561857	0
2	3	2013	Falcon 9	677.000000	ISS	CCAFS SLC 40	None None	1	False	False	False	NaN	1.0	0	B0007	-80.577366	28.561857	0
3	4	2013	Falcon 9	500.000000	PO	VAFB SLC 4E	False Ocean	1	False	False	False	NaN	1.0	0	B1003	-120.610829	34.632093	0
4	5	2013	Falcon 9	3170.000000	GTO	CCAFS SLC 40	None None	1	False	False	False	NaN	1.0	0	B1004	-80.577366	28.561857	0

EDA AND INTERACTIVE VISUAL ANALYTICS METHODOLOGY

- **Complete EDA with SQL**

1. All launch site names

```
[10]: %sql select distinct Launch_Site from SPACEXTABLE;
 * sqlite:///my_data1.db
Done.
```

[10]:

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

2. Launch site names begin with `CCA`

```
%sql select * from SPACEXTABLE where Launch_Site like 'CCA%' limit 5;
 * sqlite:///my_data1.db
Done.
```

Date	Time (UTC)	Booster_Version	Launch_Site	Payload	PAYLOAD_MASS__KG_	Orbit	Customer	Mission_Outcome	Landing_Outcome
2010-06-04	18:45:00	F9 v1.0 B0003	CCAFS LC-40	Dragon Spacecraft Qualification Unit	0	LEO	SpaceX	Success	Failure (parachute)
2010-12-08	15:43:00	F9 v1.0 B0004	CCAFS LC-40	Dragon demo flight C1, two CubeSats, barrel of Brouere cheese	0	LEO (ISS)	NASA (COTS) NRO	Success	Failure (parachute)
2012-05-22	7:44:00	F9 v1.0 B0005	CCAFS LC-40	Dragon demo flight C2	525	LEO (ISS)	NASA (COTS)	Success	No attempt
2012-10-08	0:35:00	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

3. Total payload mass

```
%sql select sum(PAYLOAD_MASS__KG_) from SPACEXTABLE where Customer = 'NASA (CRS)';
 * sqlite:///my_data1.db
Done.
```

sum(PAYLOAD_MASS__KG_)
45596

4. Average payload mass by F9 v1.1

```
%sql select avg(PAYLOAD_MASS__KG_) from SPACEXTABLE where Booster_Version = 'F9 v1.1';
 * sqlite:///my_data1.db
Done.
```

avg(PAYLOAD_MASS__KG_)
2928.4

EDA AND INTERACTIVE VISUAL ANALYTICS METHODOLOGY

- **Complete EDA with SQL**

5. First successful ground landing date

```
%sql select min(Date) from SPACEXTABLE where Landing_Outcome = 'Success (ground pad)'
```

```
 * sqlite:///my_data1.db
Done.
```

min(Date)
2015-12-22

6. Successful drone ship landing with payload between 4000 and 6000

```
%sql select Booster_Version from SPACEXTABLE\
where Landing_Outcome = "Success (drone ship)" and PAYLOAD_MASS__KG_ between 4000 and 6000;
```

```
 * sqlite:///my_data1.db
Done.
```

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

7. Total number of successful and failure mission outcomes

```
%sql SELECT Mission_Outcome, COUNT(*) as count from SPACEXTABLE\
GROUP BY Mission_Outcome ORDER BY count DESC;
```

```
 * sqlite:///my_data1.db
Done.
```

Mission_Outcome	count
Success	98
Success (payload status unclear)	1
Success	1
Failure (in flight)	1

EDA AND INTERACTIVE VISUAL ANALYTICS METHODOLOGY

- **Complete EDA with SQL**

8. Boosters carried maximum payload

```
%sql select Booster_Version from SPACEXTABLE\
where PAYLOAD_MASS__KG_ = (select max(PAYLOAD_MASS__KG_) from SPACEXTABLE)
```

```
 * sqlite:///my_data1.db
Done.
```

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

9. 2015 launch records

```
%sql select substr(Date, 6, 2) as month, Landing_Outcome, Booster_Version, Launch_Site from SPACEXTABLE\
where substr(Date, 1, 4) = '2015' and Landing_Outcome = 'Failure (drone ship)'
```

```
 * sqlite:///my_data1.db
Done.
```

month	Landing_Outcome	Booster_Version	Launch_Site
01	Failure (drone ship)	F9 v1.1 B1012	CCAFS LC-40
04	Failure (drone ship)	F9 v1.1 B1015	CCAFS LC-40

10. Rank success count between 2010-06-04 and 2017-03-20

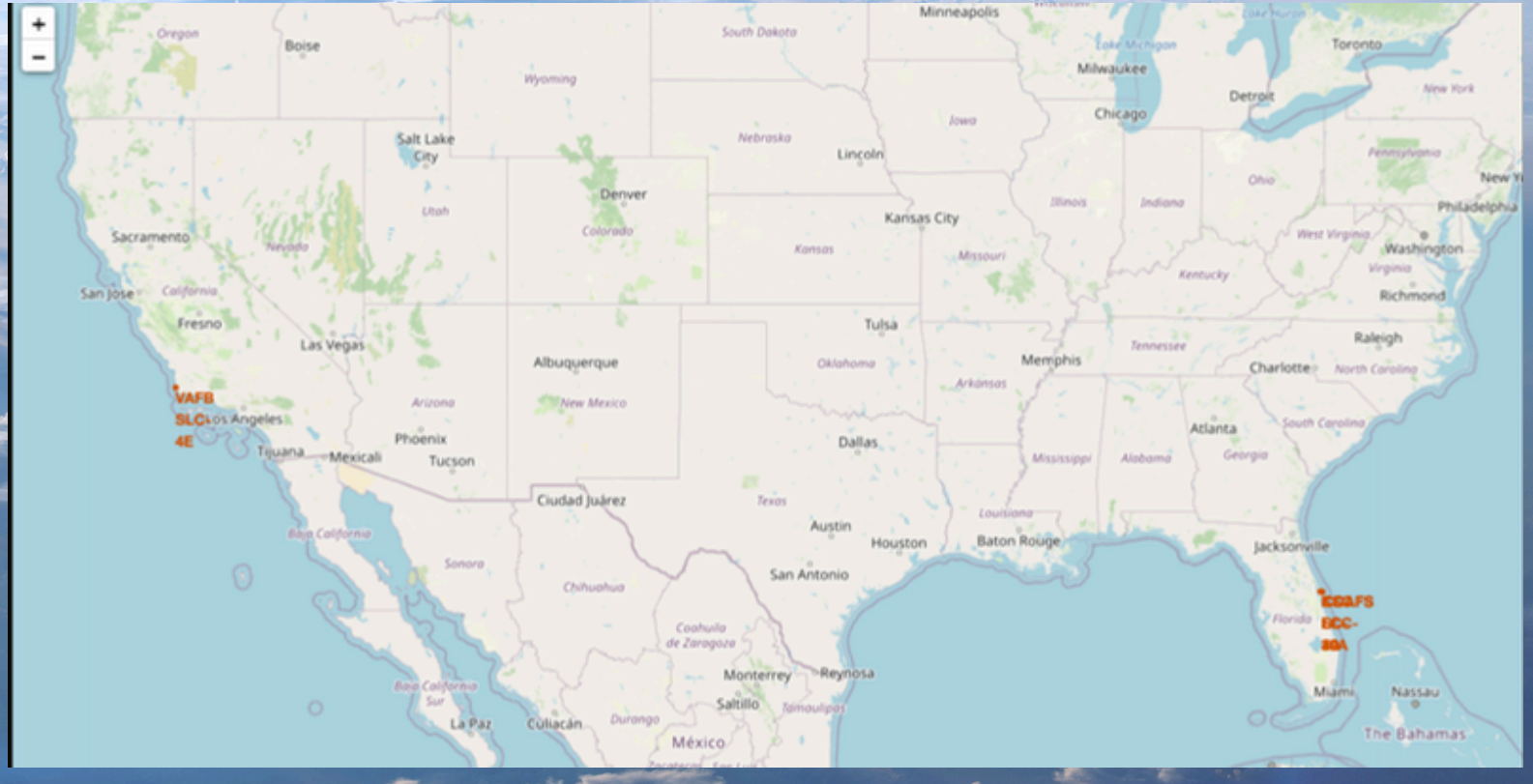
```
%sql select Landing_Outcome, count(*) as count_outcomes \
from SPACEXTABLE\
where Date between '2010-06-04' and '2017-03-20'\
group by Landing_Outcome\
order by count_outcomes Desc;
```

```
 * sqlite:///my_data1.db
Done.
```

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

INTERACTIVE MAP WITH FOLIUM

1. All launch sites' markers on a global map

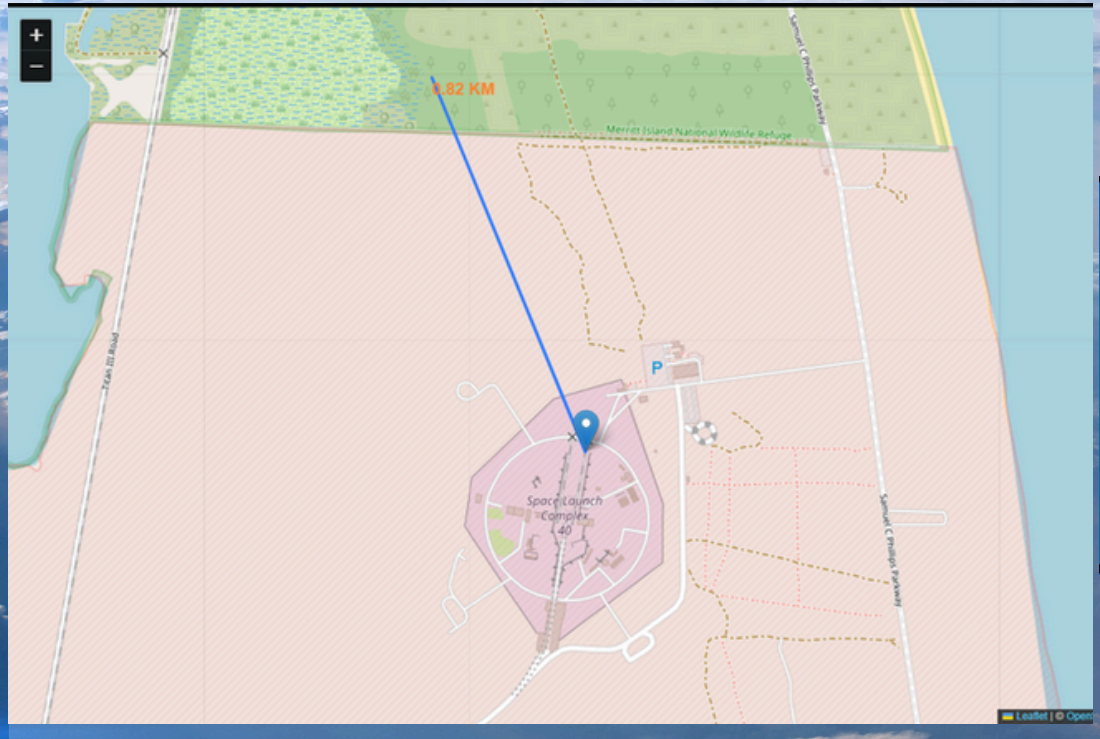


INTERACTIVE MAP WITH FOLIUM

2. All launch records per site on the map

INTERACTIVE MAP WITH FOLIUM

3. Launch sites' proximities such as railway, highway, coastline, with distance calculated and displayed



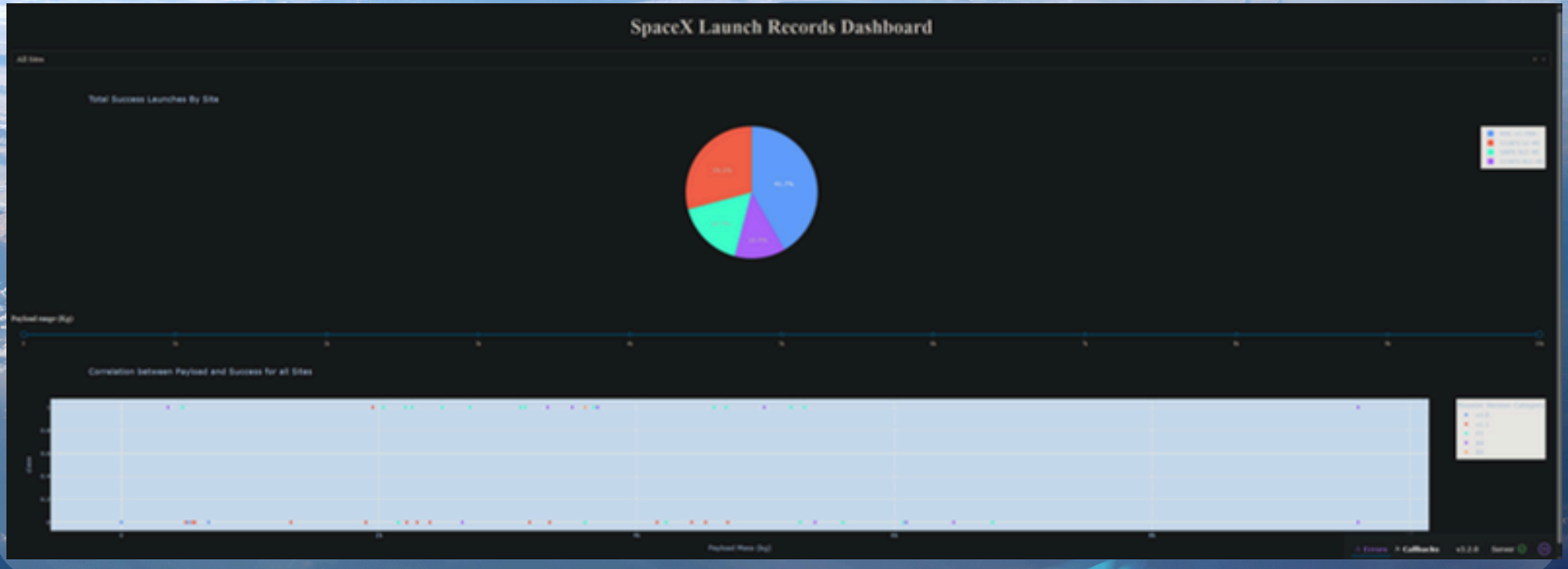
After you plot distance lines to the proximities, you can answer the following questions easily:

- Are launch sites in close proximity to railways? No
- Are launch sites in close proximity to highways? No
- Are launch sites in close proximity to coastline? Yes
- Do launch sites keep certain distance away from cities? Yes

Also please try to explain your findings.

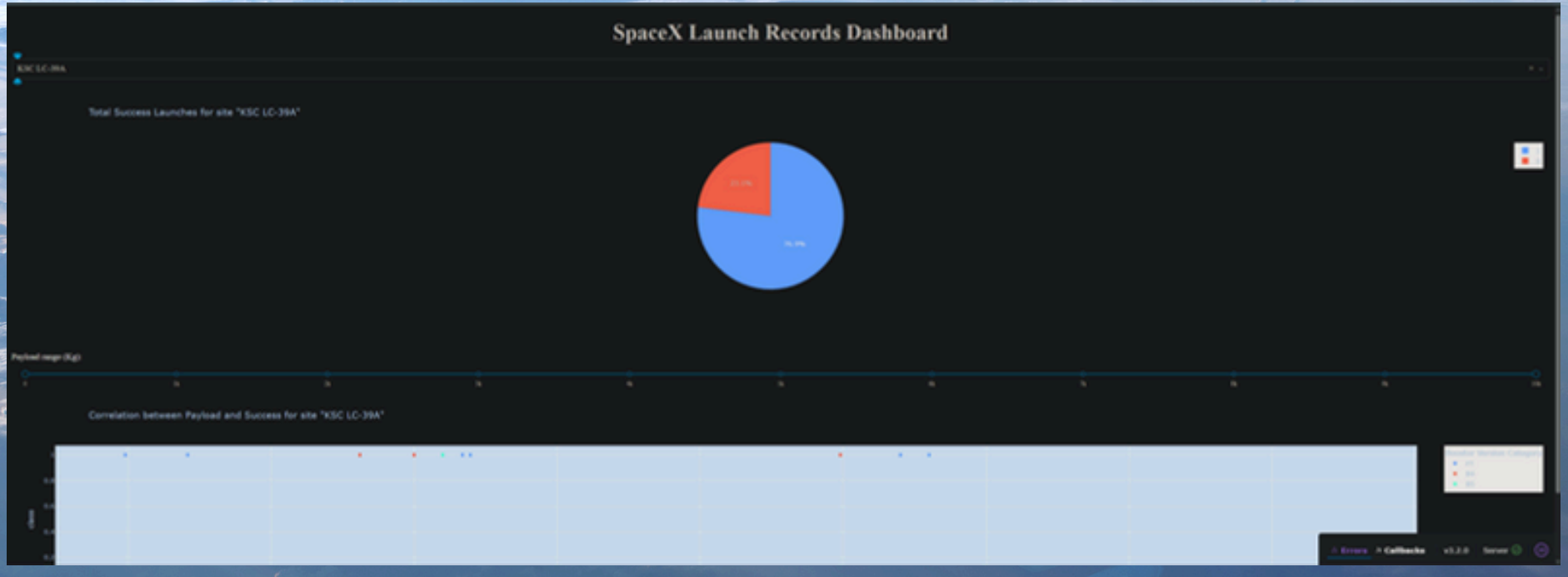
PLOTLY DASH DASHBOARD

1. Launch success count for all sites, in a piechart



PLOTLY DASH DASHBOARD

2. the piechart for the launch site with the highest launch success ratio



PLOTLY DASH DASHBOARD

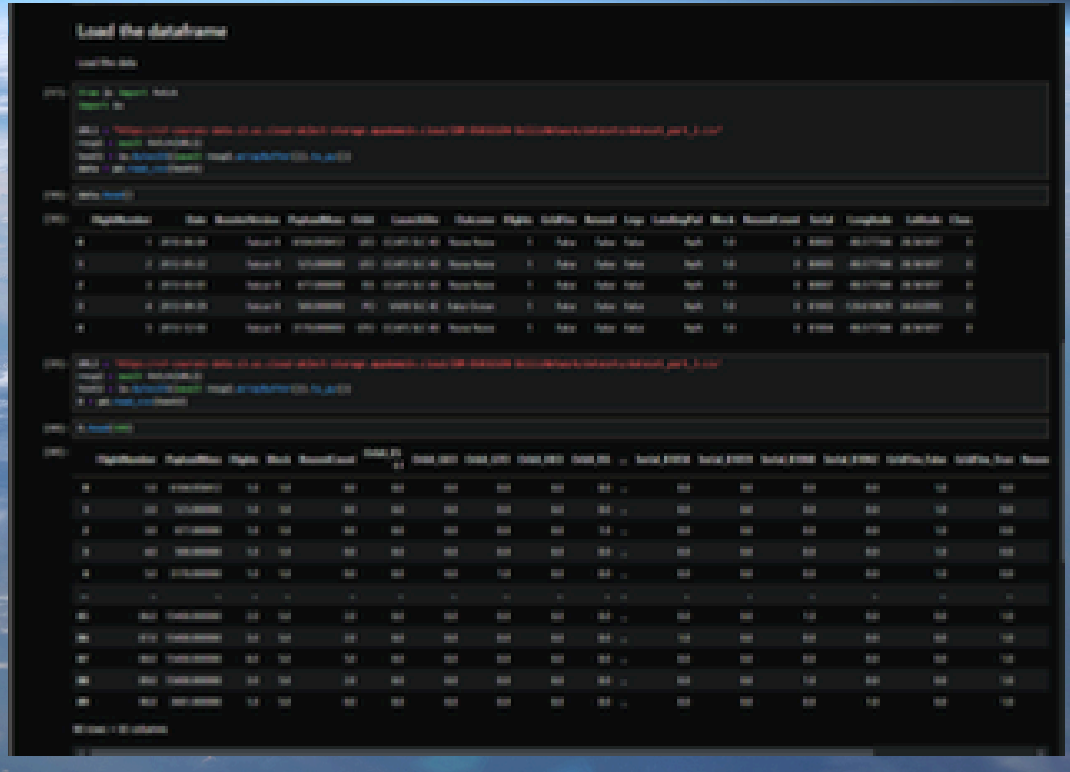
3. Payload vs. Launch outcome scatter plot for all sites

Low weighted payload between 0kg - 6000 kg

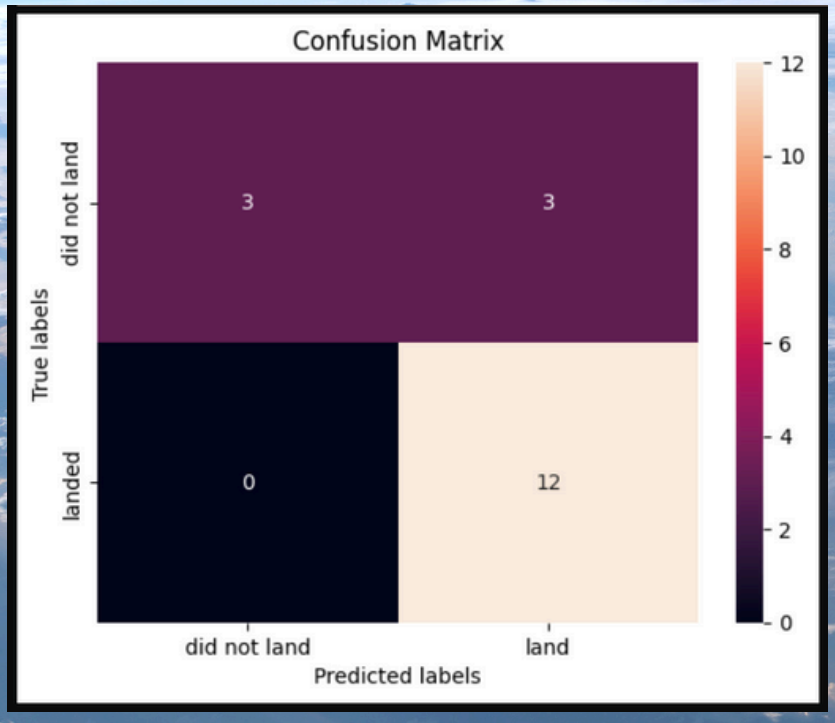
Heavy weighted payload between 6000kg - 10000 kg

PREDICTIVE ANALYSIS (CLASSIFICATION)

In this lab, we build a machine learning pipeline to predict if the first stage of the Falcon 9 lands successfully. This will include: Preprocessing, allowing us to standardize our data, and Train_test_split, allowing us to split our data into training and testing data, We will train the model and perform Grid Search, allowing us to find the hyperparameters that allow a given algorithm to perform best.



PREDICTIVE ANALYSIS (CLASSIFICATION)



Find the method performs best:

```
models = {'KNeighbors':grid_search_knn.best_score_,
              'DecisionTree':tree_cv.best_score_,
              'LogisticRegression':logreg_cv.best_score_,
              'SupportVector': svm_cv.best_score_}

bestalgorithm = max(models, key=models.get)
print('Best model is', bestalgorithm,'with a score of', models[bestalgorithm])
if bestalgorithm == 'DecisionTree':
    print('Best params is :', tree_cv.best_params_)
if bestalgorithm == 'KNeighbors':
    print('Best params is :', knn_cv.best_params_)
if bestalgorithm == 'LogisticRegression':
    print('Best params is :', logreg_cv.best_params_)
if bestalgorithm == 'SupportVector':
    print('Best params is :', svm_cv.best_params_)
```

```
Best model is DecisionTree with a score of 0.8732142857142856
Best params is : {'criterion': 'gini', 'max_depth': 6, 'max_features': 'sqrt',
'min_samples_leaf': 2, 'min_samples_split': 5, 'splitter': 'random'}
```

CONCLUSION

Base on this course, we can conclude that:

- Data understanding and collection are crucial in every data science project. Without clean, structured data, even the best machine learning models would not perform effectively.
- Building accurate models requires careful consideration of the features involved and the choice of algorithms.
- Model evaluation is as important as building the model itself. The effectiveness of a model should be tested rigorously, and the parameters should be tuned to ensure the model performs optimally.
- Deployment is the stage where a model is made usable for the target audience or system. While building and evaluating the model is important, deployment ensures that the results of the model can be applied to real-world data and scenarios.
- The ultimate goal of data science is not only to build a high-performing model but to derive actionable insights that can benefit real-world decision-making. Communicating these insights effectively is just as important as the technical implementation of the model.

INNOVATIVE INSIGHTS

The IBM Applied Data Science Capstone course offers opportunities to innovate by using real-time data for faster decision-making through predictive models. Automated hyperparameter tuning and AI tools can streamline the process of optimizing models and make machine learning accessible to non-experts. By improving model transparency with explainable AI, we can build trust in critical sectors like healthcare and finance. Real-time data pipelines and cross-industry collaborations can enhance predictions by combining data from different fields. AI-powered tools for data cleaning and wrangling can save time and improve model quality. Lastly, focusing on ethics and bias detection in models ensures fairer and more reliable AI systems, making data science more socially responsible.