IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

Viviana Stephania Muñoz
May 25, 2025

Outline

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

Executive Summary

- Summary of methodologies
- Summary of all results

Introduction

- Project background and context
- Problems you want to find answers

Section 1

Methodology

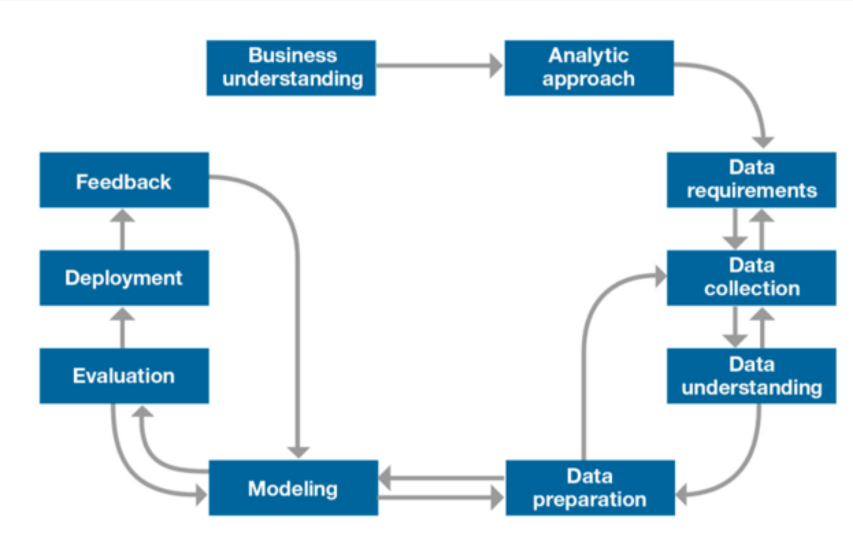
Methodology

Executive Summary

- Data collection methodology:
  - The data was collected from SpaceX API, rocket launch data, which provides SpaceX launches. The dataset includes information about launch dates, payload mass, success/failure status, launch sites, among others. Data was obtained via API download and web scraping, covering the period from 2010 to 2020.
- Perform data wrangling
  - Was created new features about rocket characteristics, imputed missing values, just include data of “Falcon 9”
- Perform exploratory data analysis (EDA) using visualization and SQL
- Perform interactive visual analytics using Folium and Plotly Dash
- Perform predictive analysis using classification models
  - The data was cleaned and split. Classification models (Decision Tree, KNN, SVM) were built and tuned with GridSearchCV. Models were evaluated and validated using accuracy and cross-validation.

Data Collection

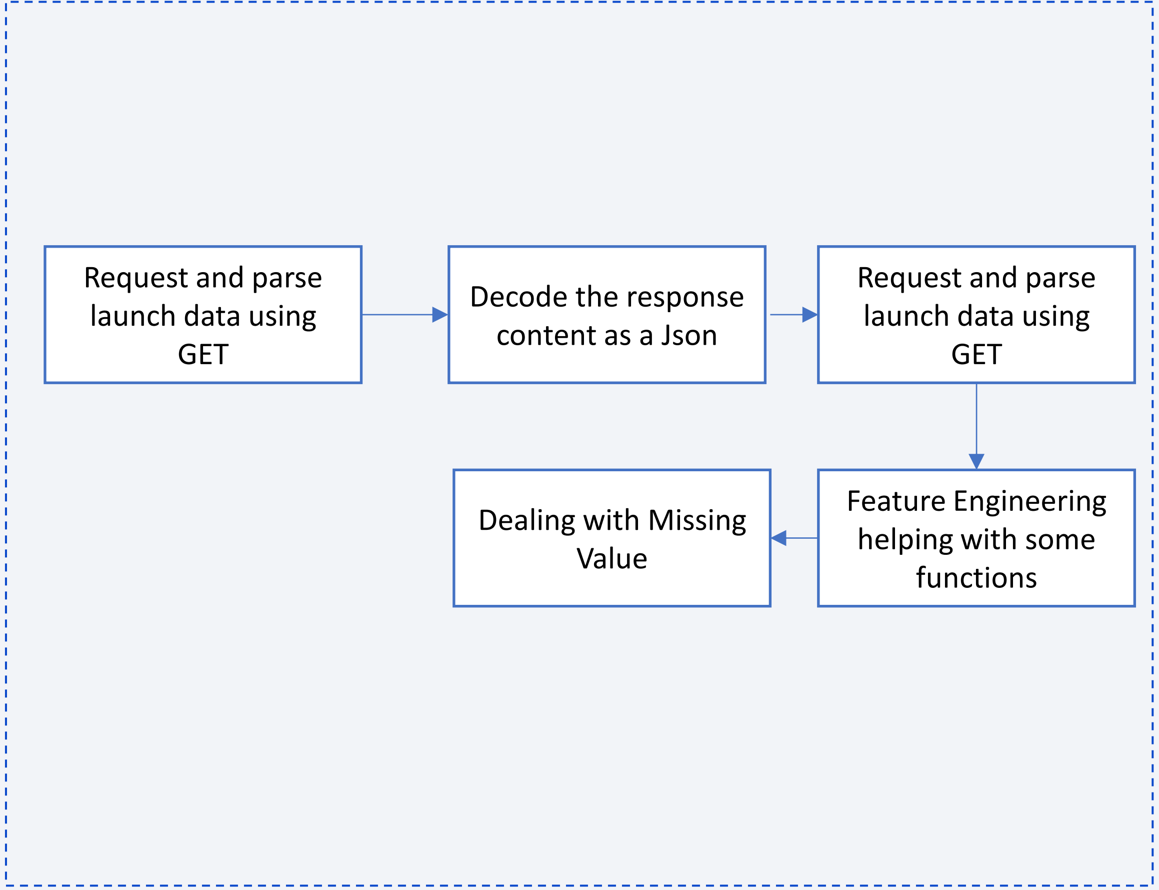
- The data was collected from SpaceX API, rocket launch data, which provides SpaceX launches. The dataset includes information about launch dates, payload mass, success/failure status, launch sites, among others. Data was obtained via API download and web scraping, covering the period from 2010 to 2020.
- John Rollins data science methodology was used.



Data Collection – SpaceX API

- Data collection SpaceX REST flow.

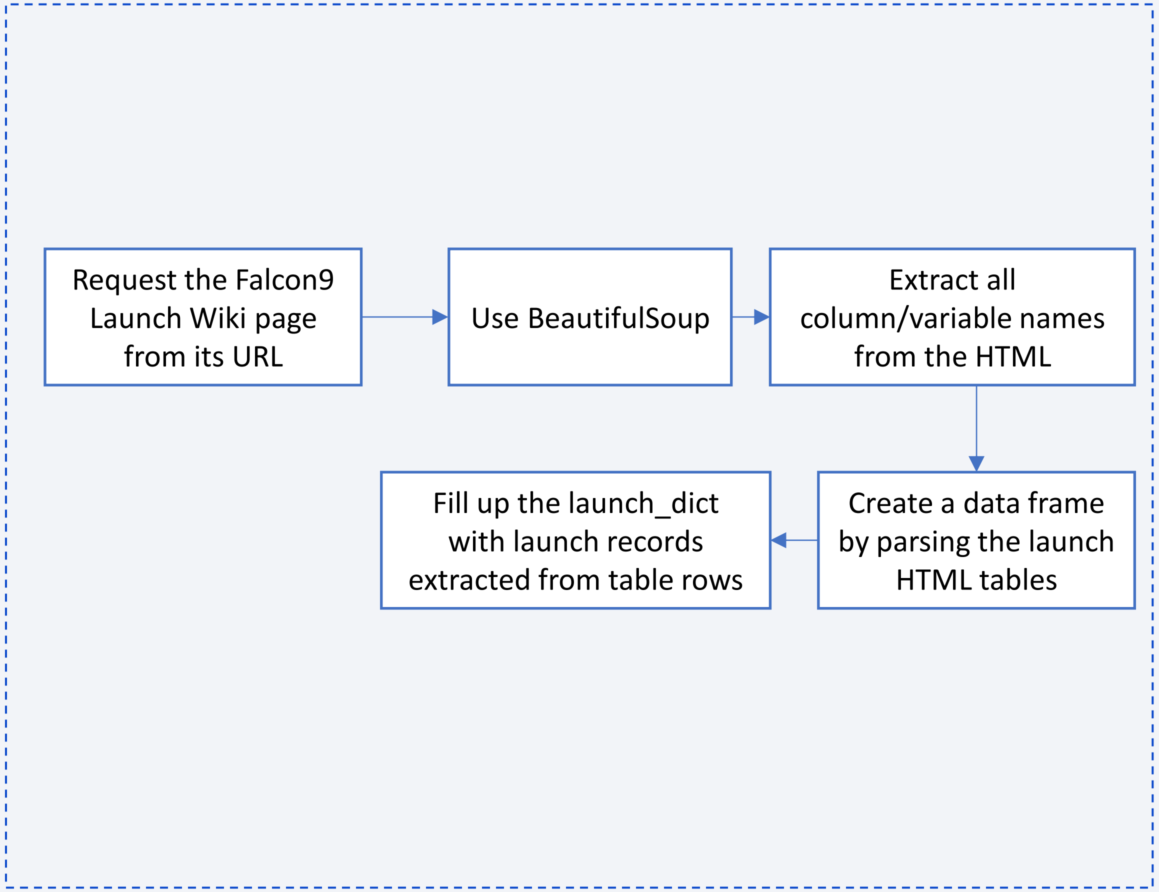
- https://github.com/ivi-bot/DataScienceCoursera/blob/main/jupyter-labs-spacex-data-collection-api.ipynb



Data Collection - Scraping

- Web scraping flow.

- https://github.com/ivi-bot/DataScienceCoursera/blob/main/jupyter-labs-webscraping.ipynb



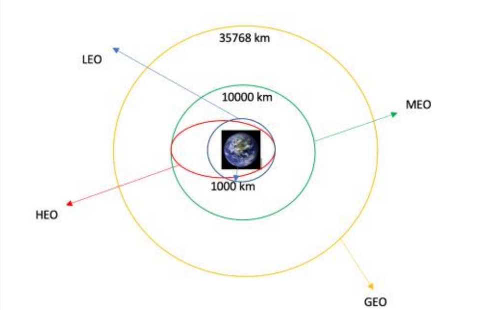
Data Wrangling

Initial raw data often contained inconsistencies, missing values. The following wrangling steps were performed:

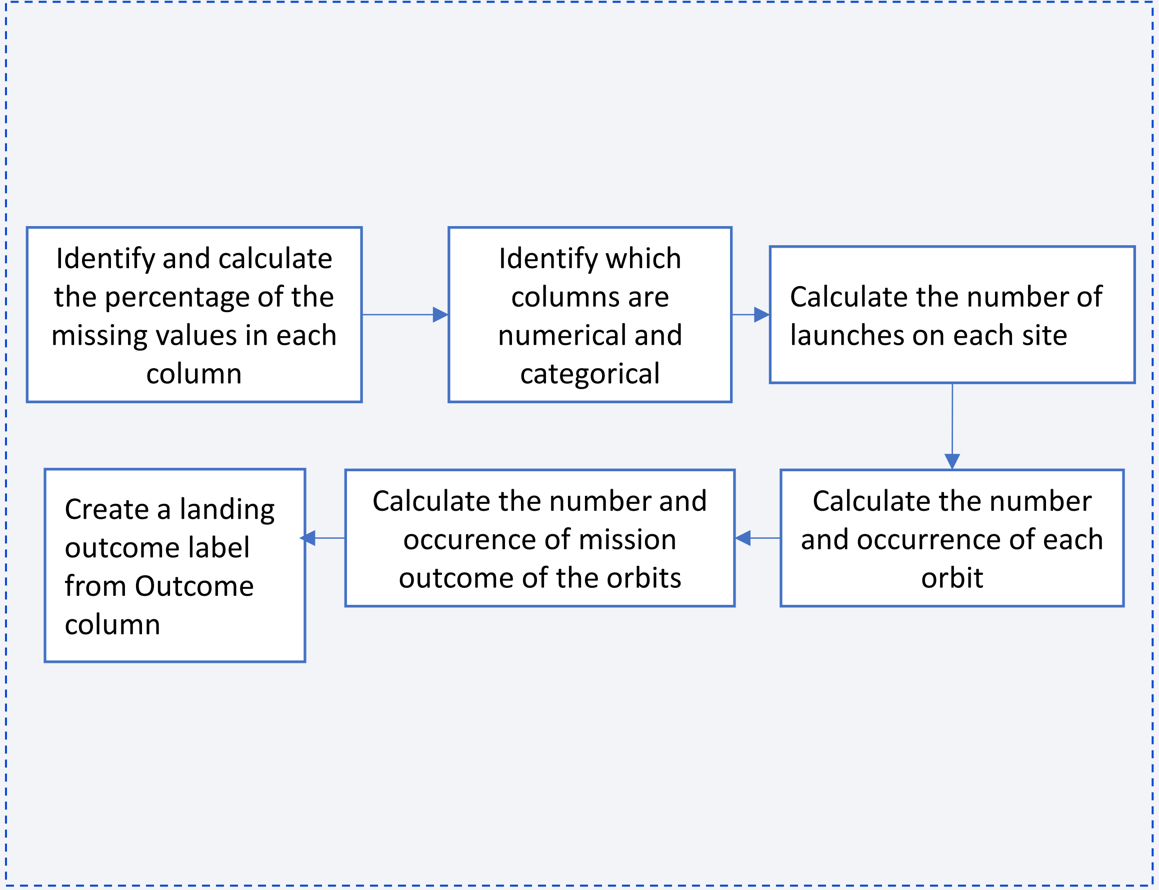
Cleaning: Imputed missing values in key columns.

Filtering: Selected only relevant rows for analysis, such as launches from specific sites (Falcon 9)

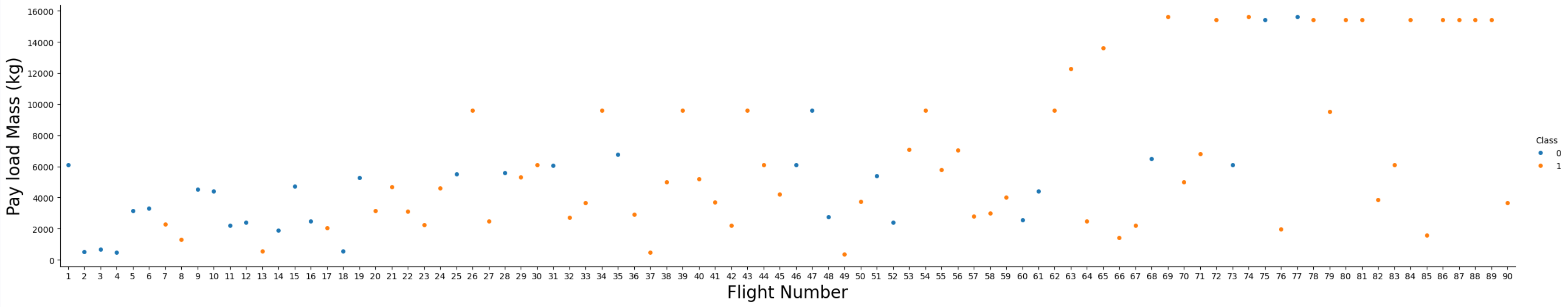
Feature Engineering: Created new columns

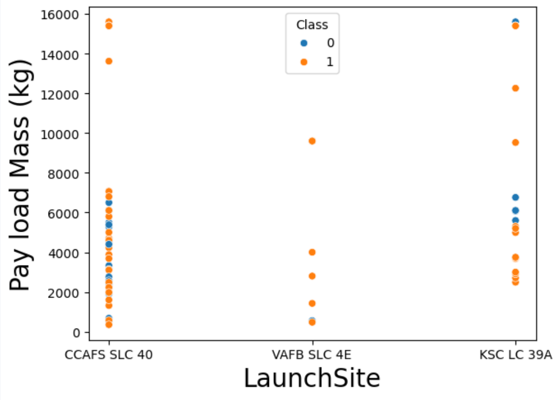


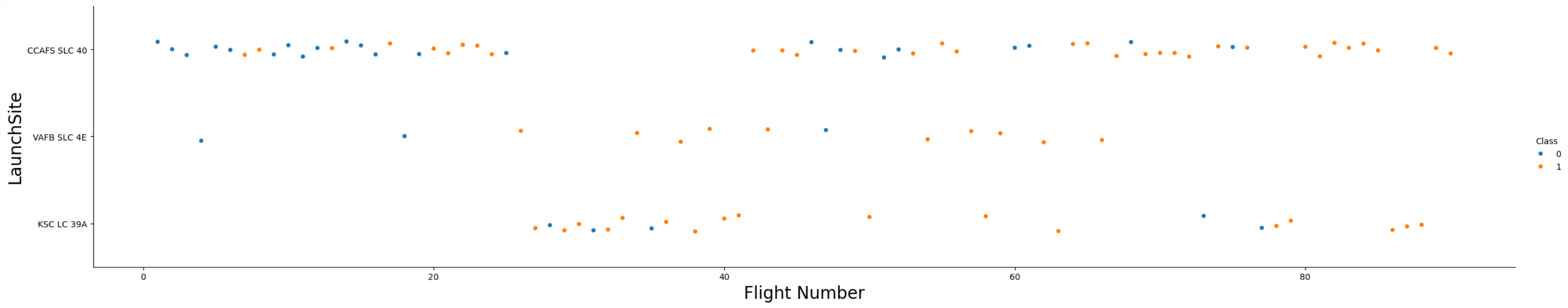
- https://github.com/ivi-bot/DataScienceCoursera/blob/main/labs-jupyter-spacex-Data%20wrangling.ipynb

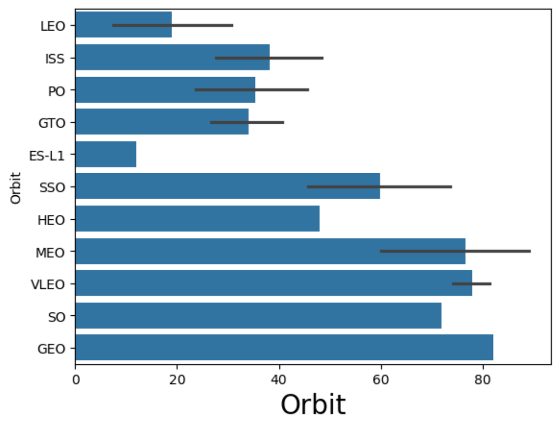


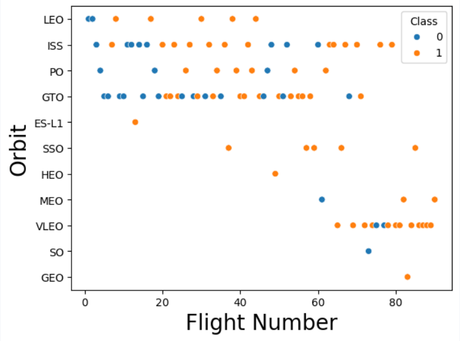
EDA with Data Visualization

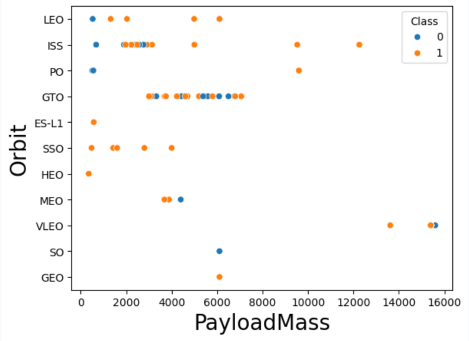


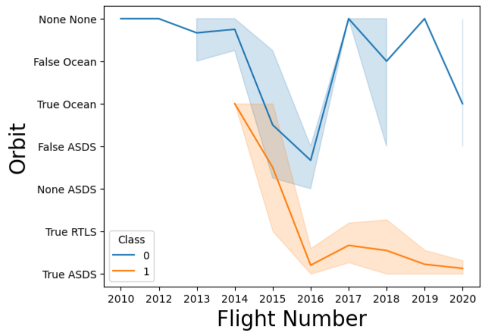










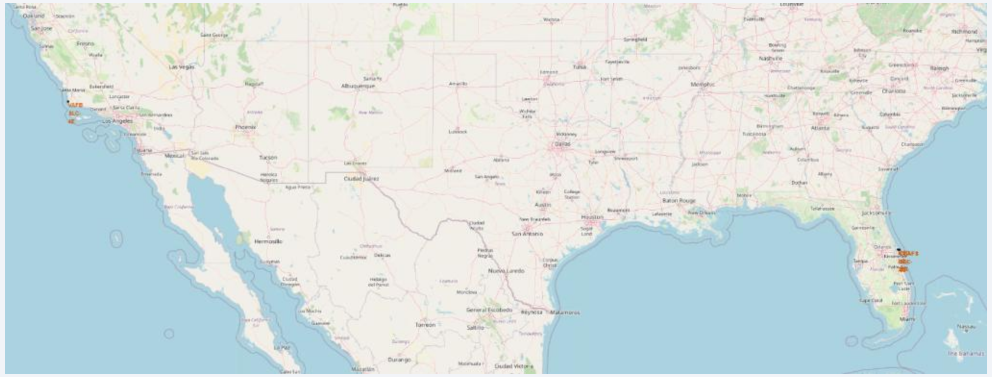


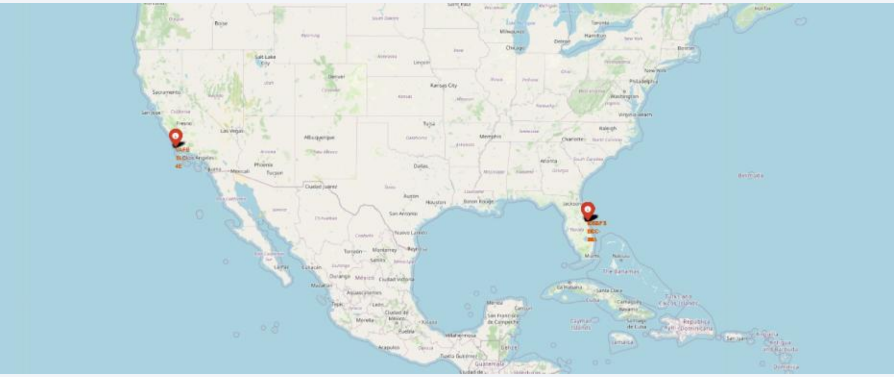
https://github.com/ivi-bot/DataScienceCoursera/blob/main/edadataviz.ipynb

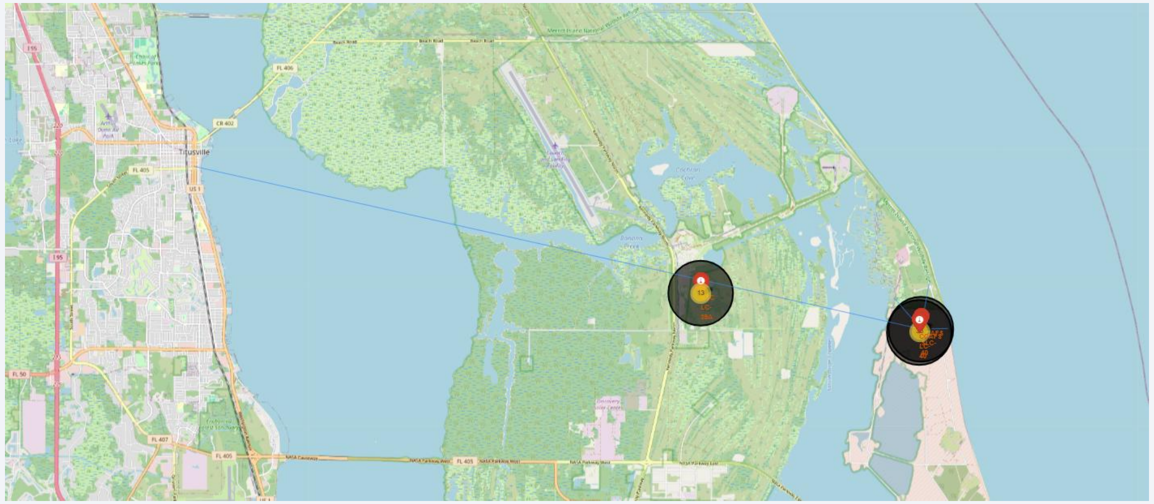
EDA with SQL

- SELECT DISTINCT Launch_Site FROM SPACEXTABLE;
- SELECT * FROM SPACEXTABLE WHERE Launch_Site LIKE "CCA%" LIMIT 5;
- SELECT BOOSTER_VERSION,PAYLOAD_MASS__KG_ FROM SPACEXTABLE WHERE CUSTOMER="NASA (CRS)" ;
- SELECT AVG(PAYLOAD_MASS__KG_) FROM SPACEXTABLE WHERE BOOSTER_VERSION= "F9 v1.1" GROUP BY BOOSTER_VERSION;
- SELECT MIN(DATE) FROM SPACEXTABLE WHERE Landing_Outcome="Success (ground pad)" ;
- SELECT Booster_Version FROM SPACEXTABLE WHERE Landing_Outcome="Success (drone ship)" AND PAYLOAD_MASS__KG_ BETWEEN 4000 AND 6000 ;
- SELECT Landing_Outcome, COUNT(Landing_Outcome) FROM SPACEXTABLE WHERE Landing_Outcome LIKE "Success%" GROUP BY Landing_Outcome;
- SELECT Landing_Outcome, COUNT(Landing_Outcome) FROM SPACEXTABLE WHERE Landing_Outcome not LIKE "Success%" GROUP BY Landing_Outcome;
- SELECT Booster_Version FROM SPACEXTABLE WHERE PAYLOAD_MASS__KG_= (SELECT MAX(PAYLOAD_MASS__KG_) PAYLOAD_MASS__KG_ FROM SPACEXTABLE)
- SELECT Landing_Outcome, Booster_Version,Launch_Site FROM SPACEXTABLE WHERE substr(Date, 0, 5) = '2015' AND Landing_Outcome ="Failure (drone ship)" ORDER BY substr(Date, 6, 2)
- SELECT Landing_Outcome, COUNT(*) AS Outcome FROM SPACEXTABLE WHERE Date BETWEEN '2010-06-04' AND '2017-03-20' GROUP BY Landing_Outcome ORDER BY Outcome DESC;
- https://github.com/ivi-bot/DataScienceCoursera/blob/main/jupyter-labs-eda-sql-coursera_sqllite.ipynb

Build an Interactive Map with Folium

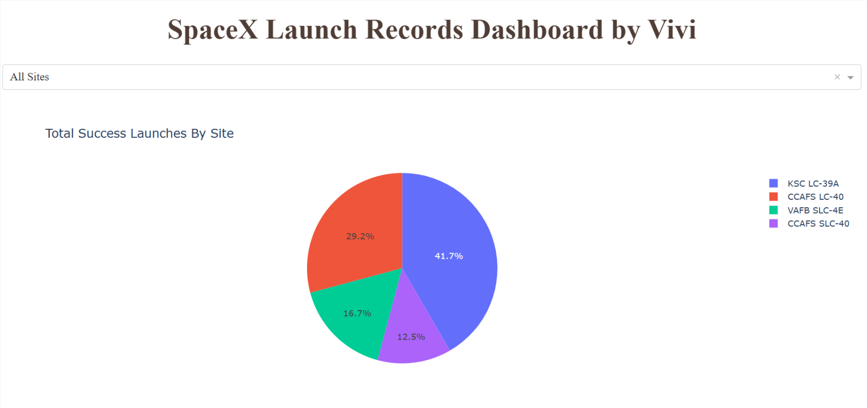


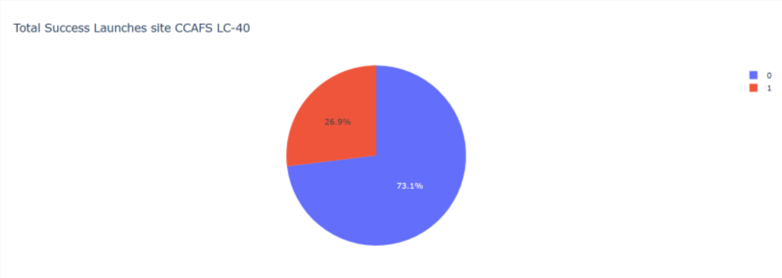


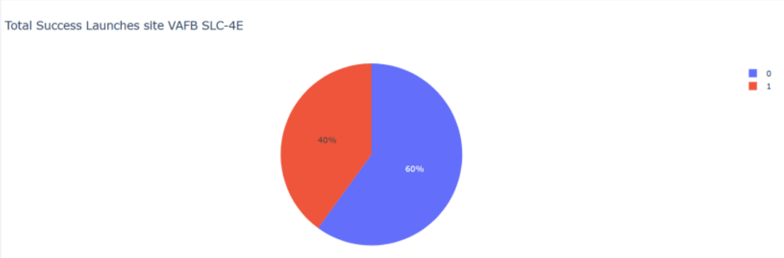


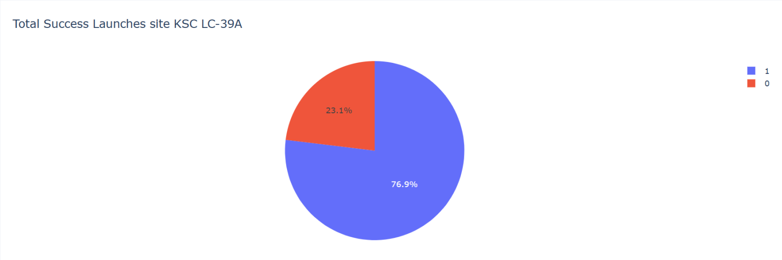
https://github.com/ivi-bot/DataScienceCoursera/blob/main/lab_jupyter_launch_site_location.ipynb

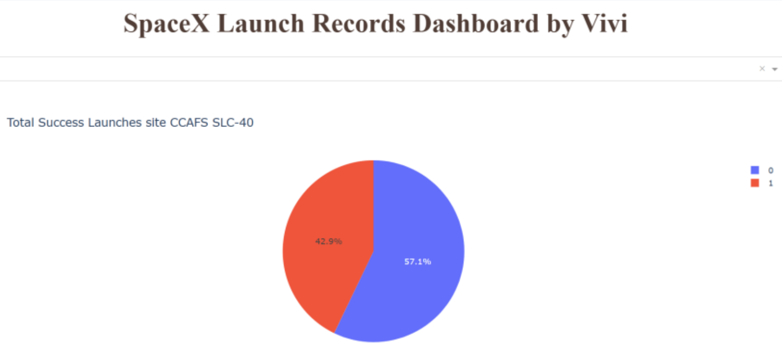
Build a Dashboard with Plotly Dash

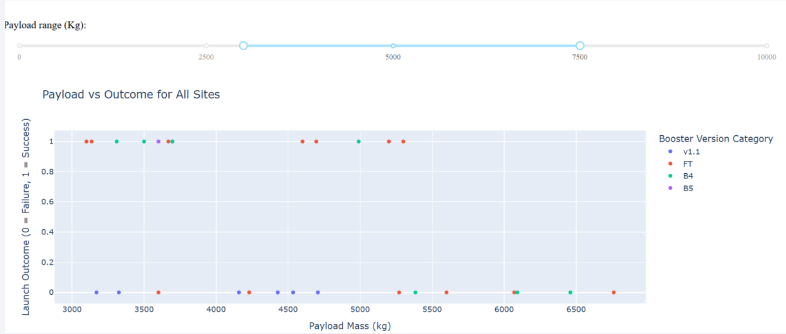








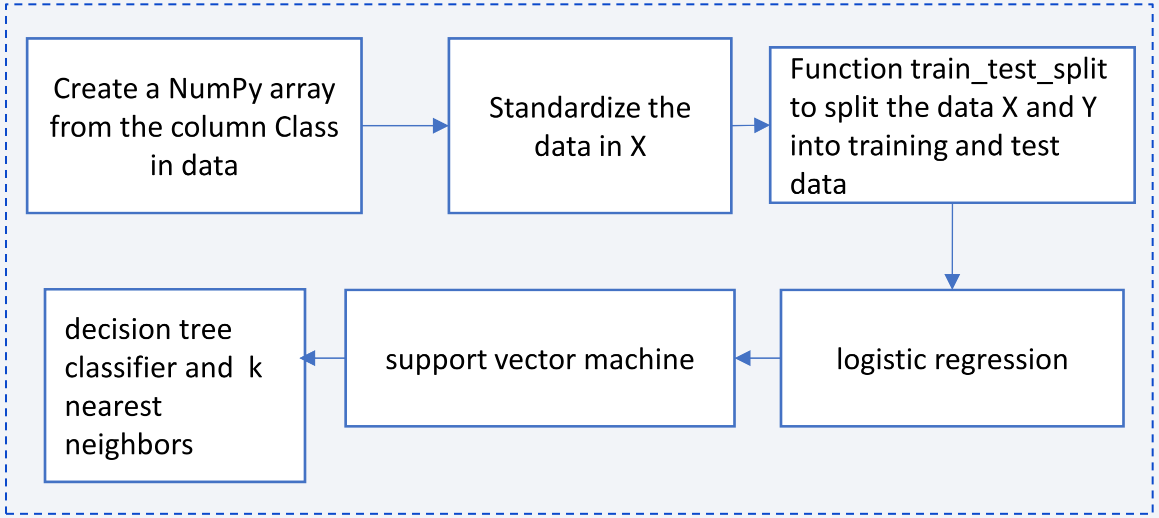


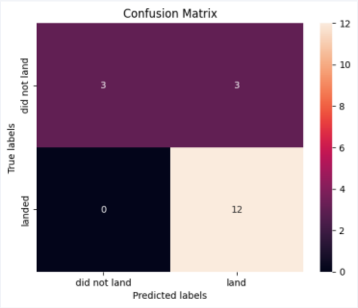


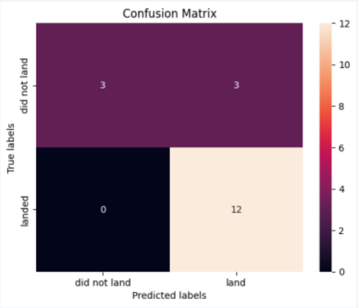
- https://github.com/ivi-bot/DataScienceCoursera/blob/main/spacex-dash-app.py

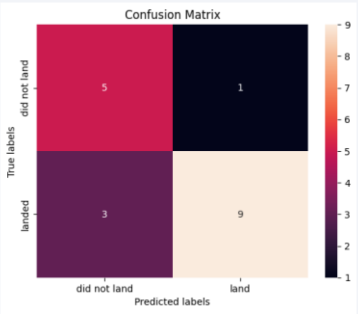
Predictive Analysis (Classification)

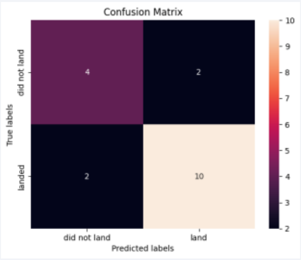
- Selected and preprocessed relevant features (including encoding categorical data)
- Built classification models: Logistic Regression, Decision Tree, KNN, SVM
- Evaluated models using cross-validation and accuracy metrics
- Tuned hyperparameters with GridSearchCV for optimization
- Compared models to identify the best performer
- Improved performance by refining features and tuning parameters
- https://github.com/ivi-bot/DataScienceCoursera/blob/main/SpaceX_Machine%20Learning%20Prediction_Part_5.ipynb





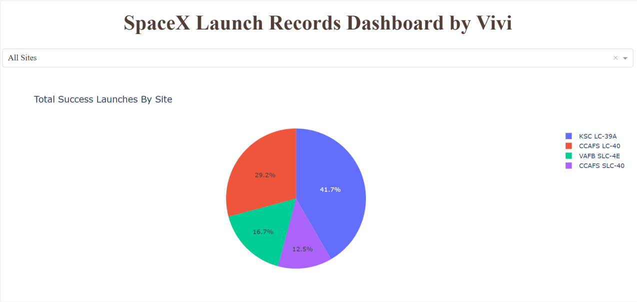


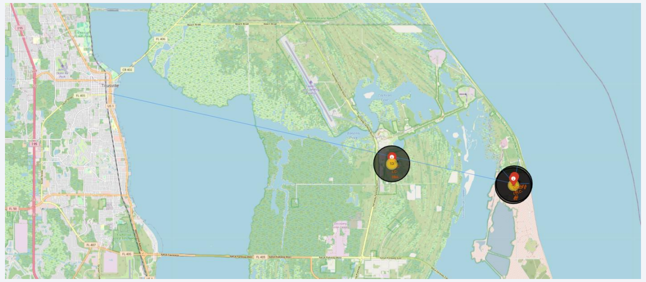




Results

- Exploratory Data Analysis (EDA) Results:
- Identified key features influencing the target variable
- Visualized distributions and relationships (e.g., payload vs. success rate)
- Detected and handled missing values and outliers
- Found correlations between variables like booster version and mission outcome
- Predictive Analysis Results:
- Built and compared several classification models (Decision Tree, KNN, SVM)
- Tuned hyperparameters to improve accuracy and reduce overfitting
- Selected the best model based on cross-validation scores
- Demonstrated model's ability to predict mission success with reasonable accuracy

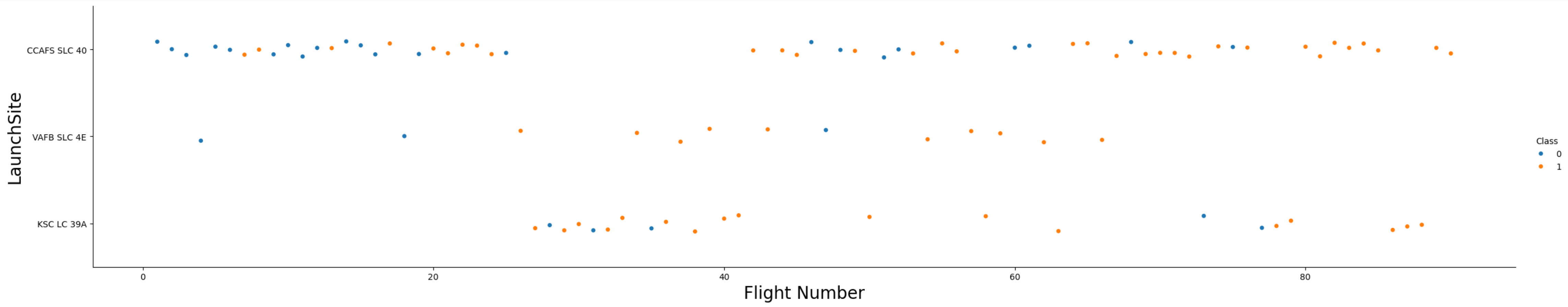




Section 2

Insights drawn from EDA

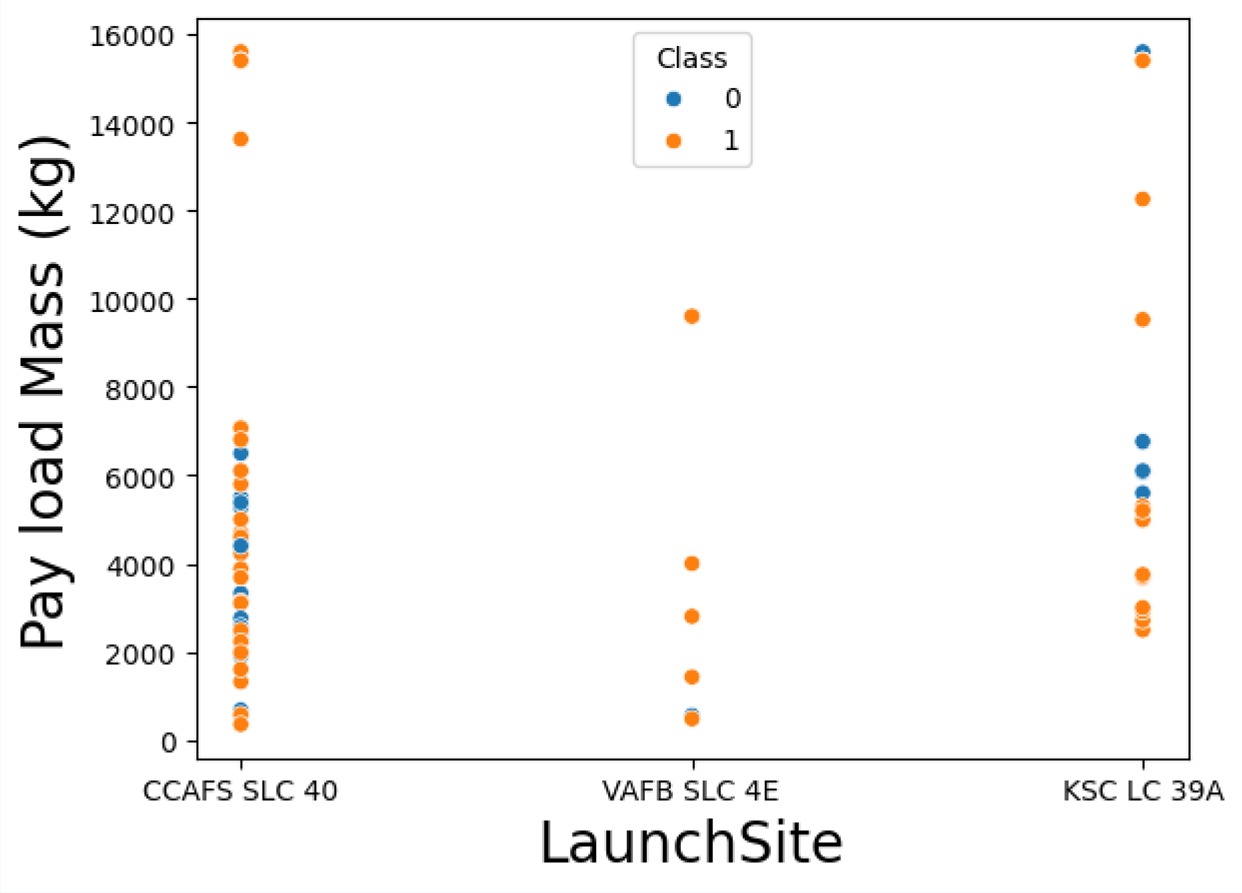
Flight Number vs. Launch Site



- Scatter plot of Flight Number vs. Launch Site
- Here, we can visualize a scatter plot showing the different launch sites and their corresponding flight numbers, highlighting the success or failure of each mission. Overall, it appears that the CCAFS SLC 40 launch site has more data points and predominantly successful missions, especially after reaching around 60 flights.

Payload vs. Launch Site

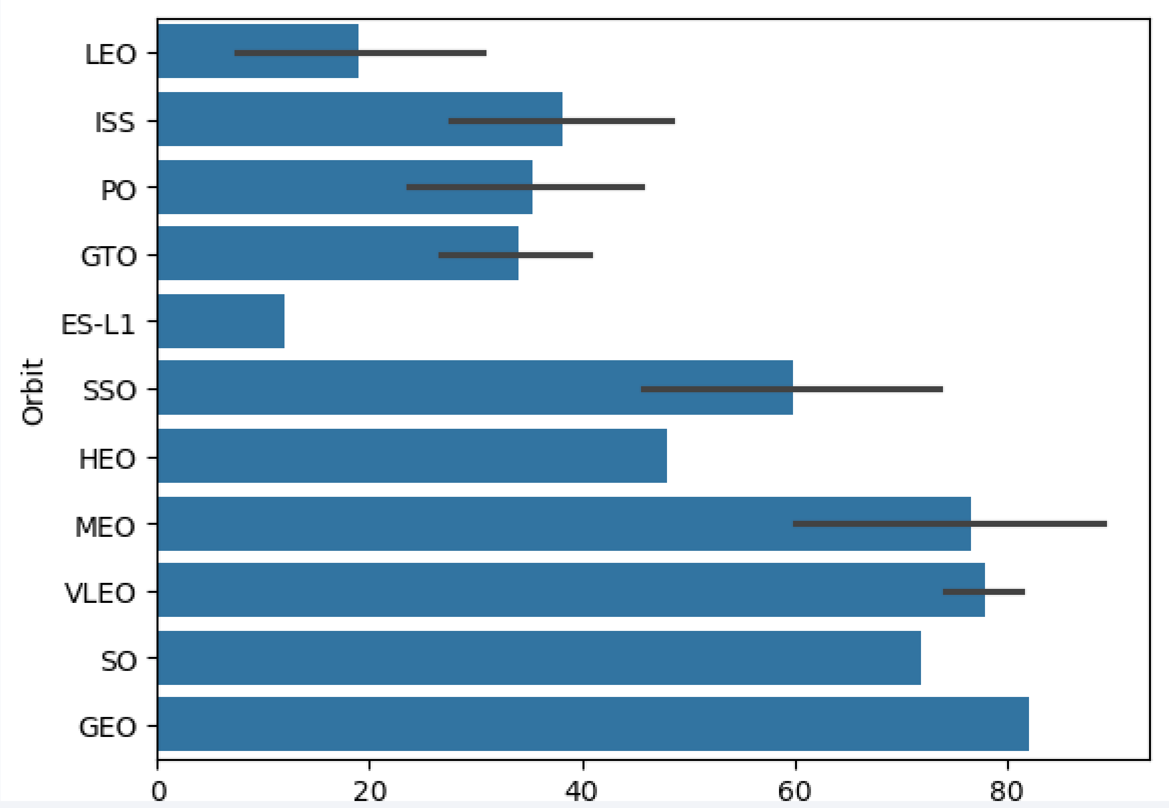
- Scatter plot of Payload vs. Launch Site
- Here, we have the different launch sites and their payload masses, showing both successful and failed missions. The most notable is CCAFS SLC 40, which has a dense concentration of successful launches with payload masses ranging from 0 to 7000 kg.



Success Rate vs. Orbit Type

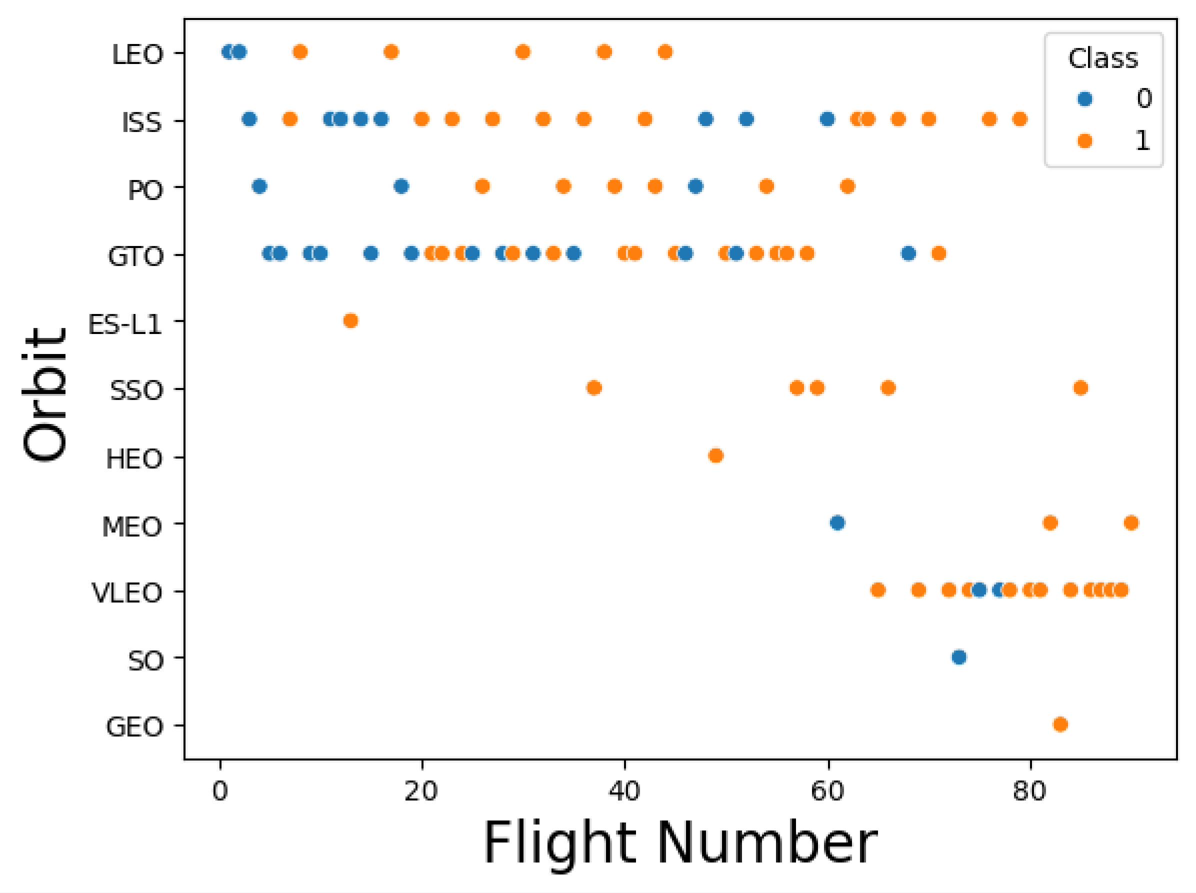
- Bar chart for the success rate of each orbit type

- This bar chart shows the success rate by orbit type, highlighting that the lowest success rate is for ES-L1, while the highest is for GEO.



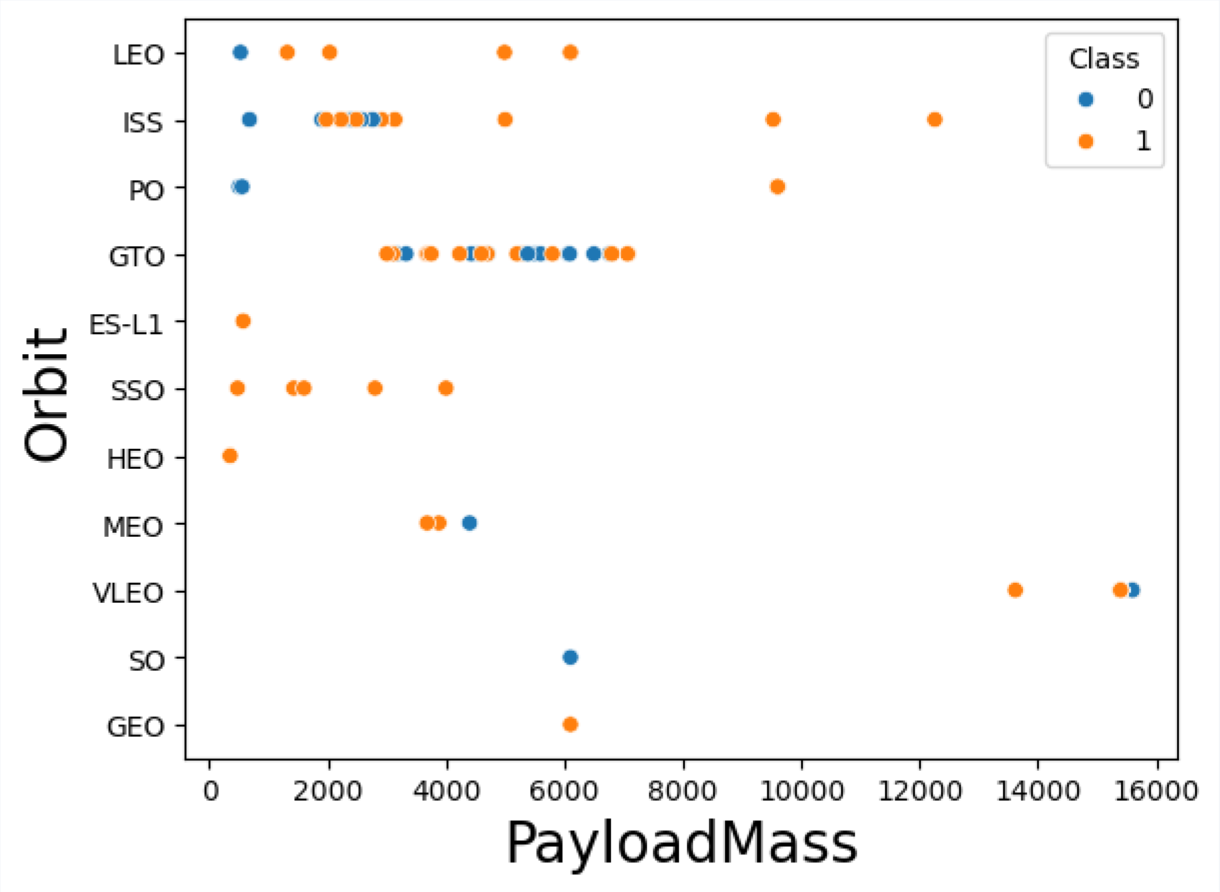
Flight Number vs. Orbit Type

- Scatter point of Flight number vs. Orbit type.
- This scatter plot shows the distribution of flight numbers across different orbit types. There is very little data for ES-L1, HEO, and GEO orbits, while LEO, ISS, PO, GTO, and VLEO have a lot of data, mostly indicating successful missions.



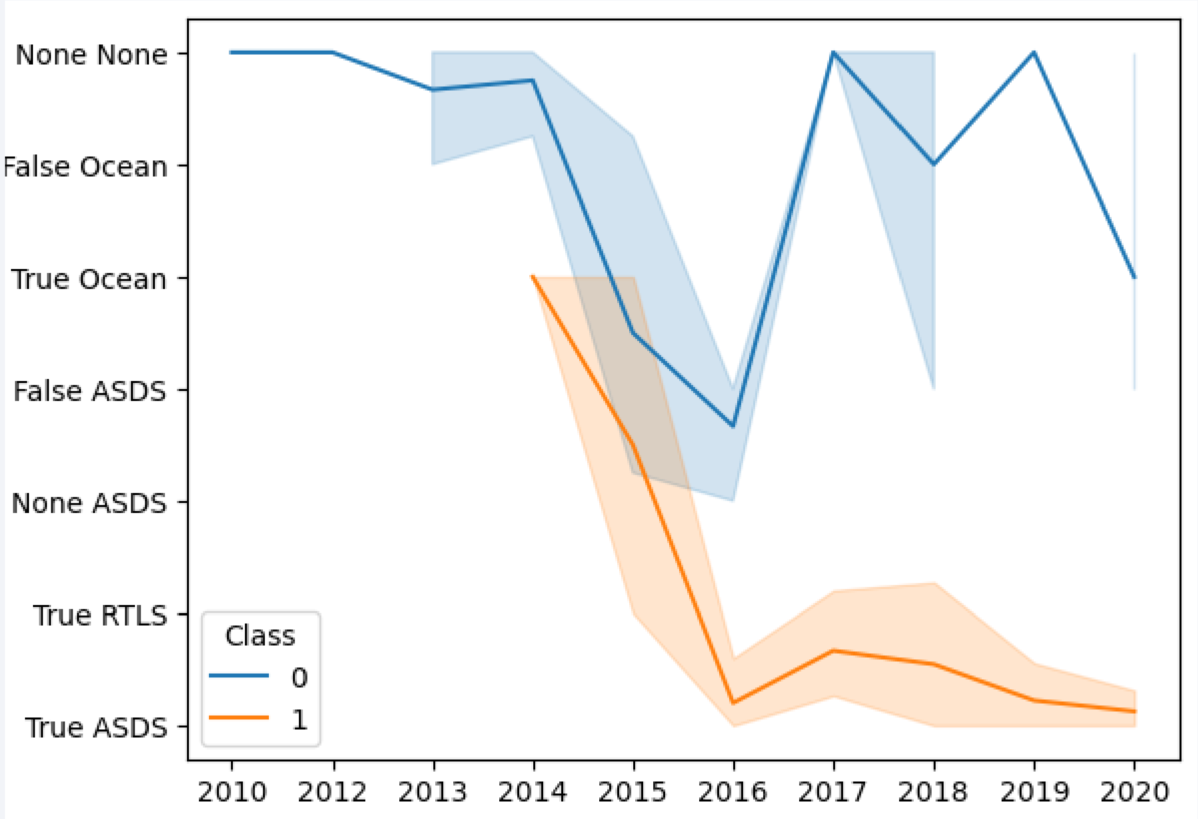
Payload vs. Orbit Type

- Scatter point of payload vs. orbit type
- Similarly, this scatter plot relates to payload mass across orbit types. Most orbits have limited data, except for ISS and GTO, which show a wide distribution of both successful and failed missions.



Launch Success Yearly Trend

- Show a line chart of yearly average success rate
- Here, the number of successful launches per year is visualized linearly.



All Launch Site Names

- The names of the unique launch sites

Task 1

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

In [12]:
```
%sql SELECT DISTINCT Launch_Site FROM SPACEXTABLE;
```

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

Out[12]:

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

Launch Site Names Begin with 'CCA'

- 5 records where launch sites begin with `CCA`

Task 2

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

In [15]:
```
%sql SELECT * FROM SPACEXTABLE WHERE Launch_Site LIKE "CCA%" LIMIT 5;
```

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

Out[15]:

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

Total Payload Mass

- The total payload carried by boosters from NASA

Task 3

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

In [18]:
```
%sql SELECT BOOSTER_VERSION,PAYLOAD_MASS__KG_ FROM SPACEXTABLE WHERE CUSTOMER="NASA (CRS)" ;
```

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

Out[18]:

Booster_Version	PAYLOAD_MASS__KG_
F9 v1.0 B0006	500
F9 v1.0 B0007	677
F9 v1.1	2296
F9 v1.1 B1010	2216
F9 v1.1 B1012	2395
F9 v1.1 B1015	1898
F9 v1.1 B1018	1952
F9 FT B1021.1	3136
F9 FT B1025.1	2257
F9 FT B1031.1	2490
F9 FT B1035.1	2708
F9 B4 B1039.1	3310
F9 FT B1035.2	2205
F9 B4 B1039.2	2647
F9 B4 B1045.2	2697
F9 B5B1050	2500
F9 B5B1056.1	2495
F9 B5 B1056.2	2268
F9 B5 B1059.2	1977
F9 B5 B1058.4	2972

Average Payload Mass by F9 v1.1

- Average payload mass carried by booster version F9 v1.1

Task 4

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

In [19]:
```
%sql SELECT AVG(PAYLOAD_MASS__KG_) FROM SPACEXTABLE WHERE BOOSTER_VERSION= "F9 v1.1" GROUP BY BOOSTER_VERSION;
```

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

Out[19]:

AVG(PAYLOAD_MASS__KG_)
2928.4

Task 5

First Successful Ground Landing Date

- Dates of the first successful landing outcome on ground pad

Task 5

List the date when the first succesful landing outcome in ground pad was acheived.

Hint:Use min function

In [23]:
```
%sql SELECT MIN(DATE) FROM SPACEXTABLE WHERE Landing_Outcome="Success (ground pad)" ;
```

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

Out[23]:

MIN(DATE)
2015-12-22

Successful Drone Ship Landing with Payload between 4000 and 6000

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

Task 6

List the names of the boosters which have success in drone ship and have payload mass greater than 4000 but less than 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

Total Number of Successful and Failure Mission Outcomes

- Total number of successful and failure mission outcomes

Task 7

List the total number of successful and failure mission outcomes

```
[39]
%sql SELECT Landing_Outcome, COUNT(Landing_Outcome) FROM SPACEXTABLE WHERE  Landing_Outcome LIKE "Success%" GROUP BY Landing_Outcome;
```

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

Landing_Outcome	COUNT(Landing_Outcome)
Success	38
Success (drone ship)	14
Success (ground pad)	9

```
[40]
%sql SELECT  Landing_Outcome, COUNT(Landing_Outcome) FROM SPACEXTABLE WHERE  Landing_Outcome not LIKE "Success%" GROUP BY Landing_Outcome;
```

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

Landing_Outcome	COUNT(Landing_Outcome)
Controlled (ocean)	5
Failure	3
Failure (drone ship)	5
Failure (parachute)	2
No attempt	21
No attempt	1
Precluded (drone ship)	1
Uncontrolled (ocean)	2

¿Desea inst
recomenda

Boosters Carried Maximum Payload

- The names of the booster which have carried the maximum payload mass

Task 8

List all the booster_versions that have carried the maximum payload mass. Use a subquery.

[42]

```
%sql SELECT  Booster_Version FROM SPACEXTABLE WHERE PAYLOAD_MASS__KG_= (SELECT MAX(PAYLOAD_MASS__KG_) 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

2015 Launch Records

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

Task 9

List the records which will display the month names, failure landing_outcomes in drone ship ,booster versions, launch_site for the months in year 2015.

Note: SQLLite does not support monthnames. So you need to use substr(Date, 6,2) as month to get the months and substr(Date,0,5)='2015' for year.

[48]

```
%sql SELECT Landing_Outcome, Booster_Version,Launch_Site FROM SPACEXTABLE WHERE substr(Date, 0, 5) = '2015' AND Landing_Outcome ="Failure (drone ship)" ORDER BY substr(Date, 6, 2)
```

Python

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

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

Task 10

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.

```
%sql SELECT Landing_Outcome, COUNT(*) AS Outcome FROM SPACEXTABLE WHERE Date BETWEEN '2010-06-04' AND '2017-03-20' GROUP BY Landing_Outcome ORDER BY Outcome DESC;
```

[49] Python

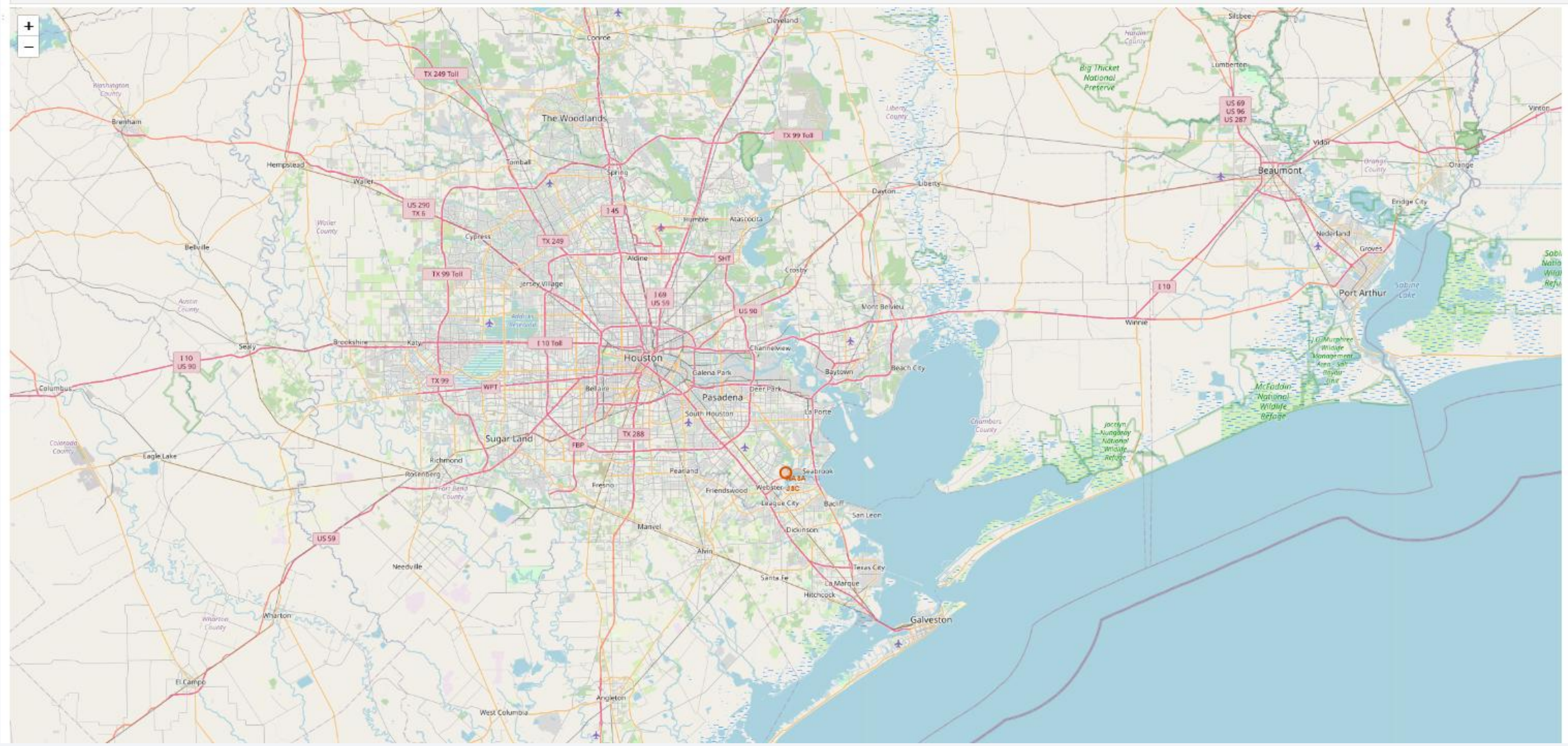
```
 * sqlite:///my_data1.db
Done.
```

Landing_Outcome	Outcome
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

Section 3

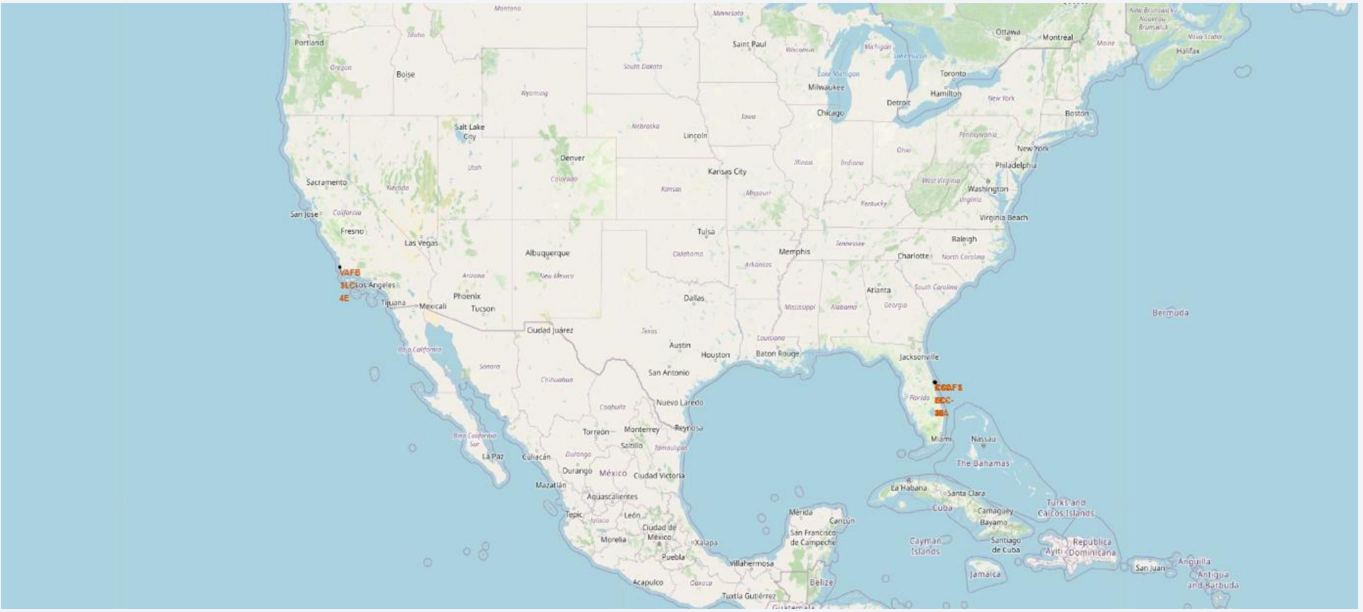
Launch Sites Proximities Analysis

NASA Johnson Space Center's

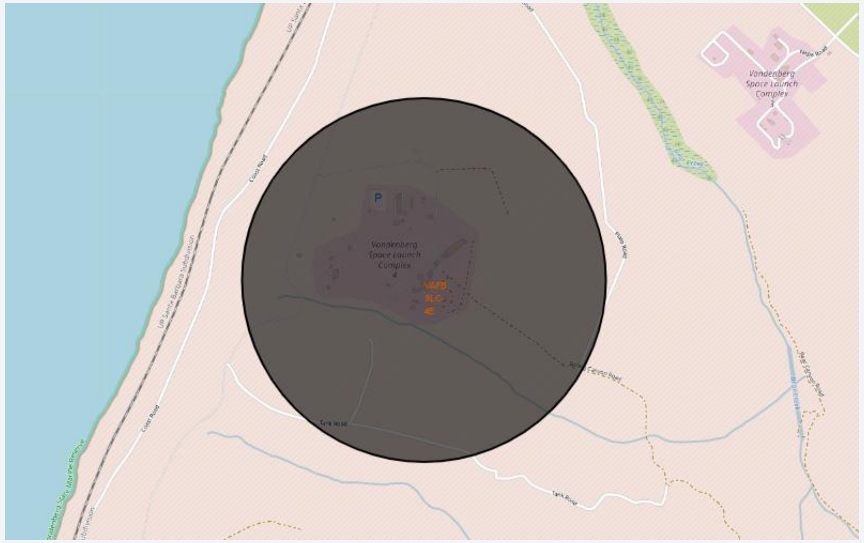


- First marker on the map.

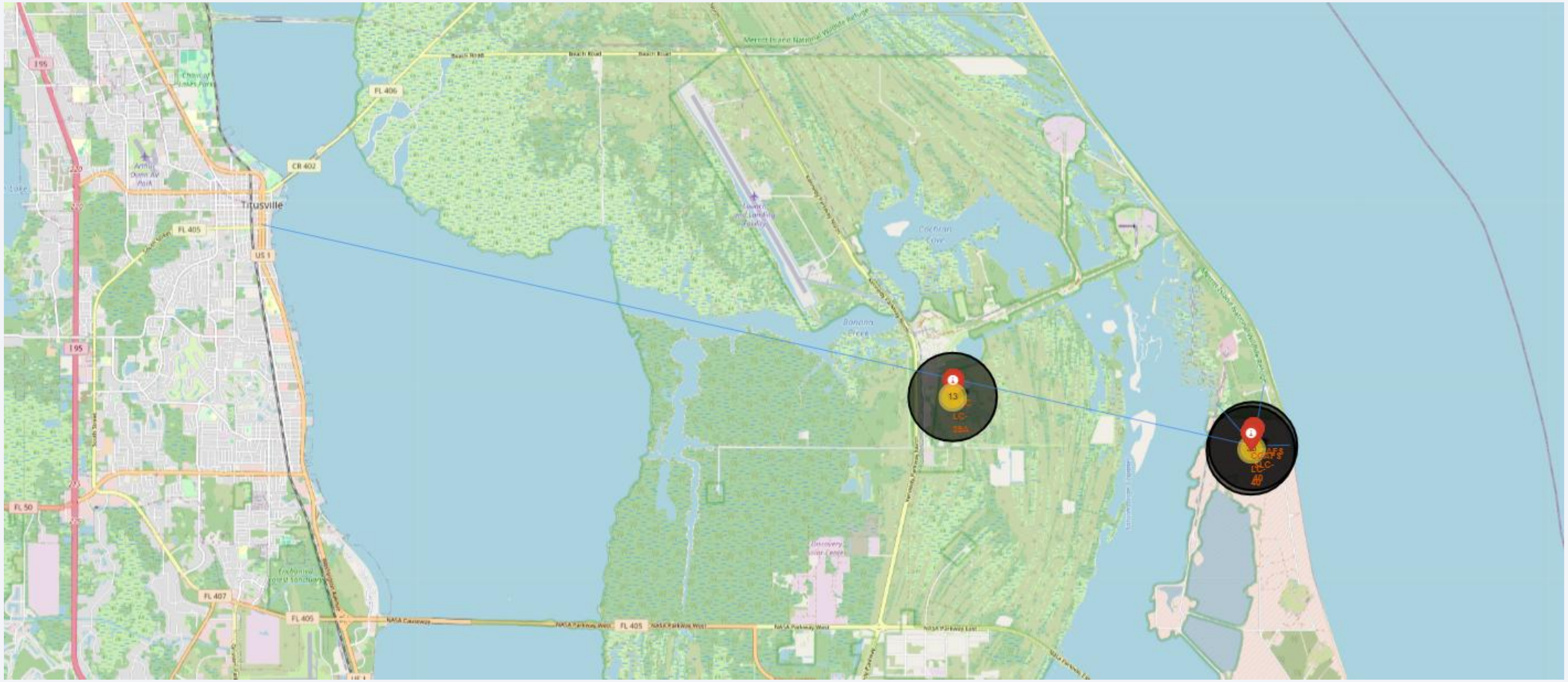
Launch Sites



- Adding circle object for each launch site.



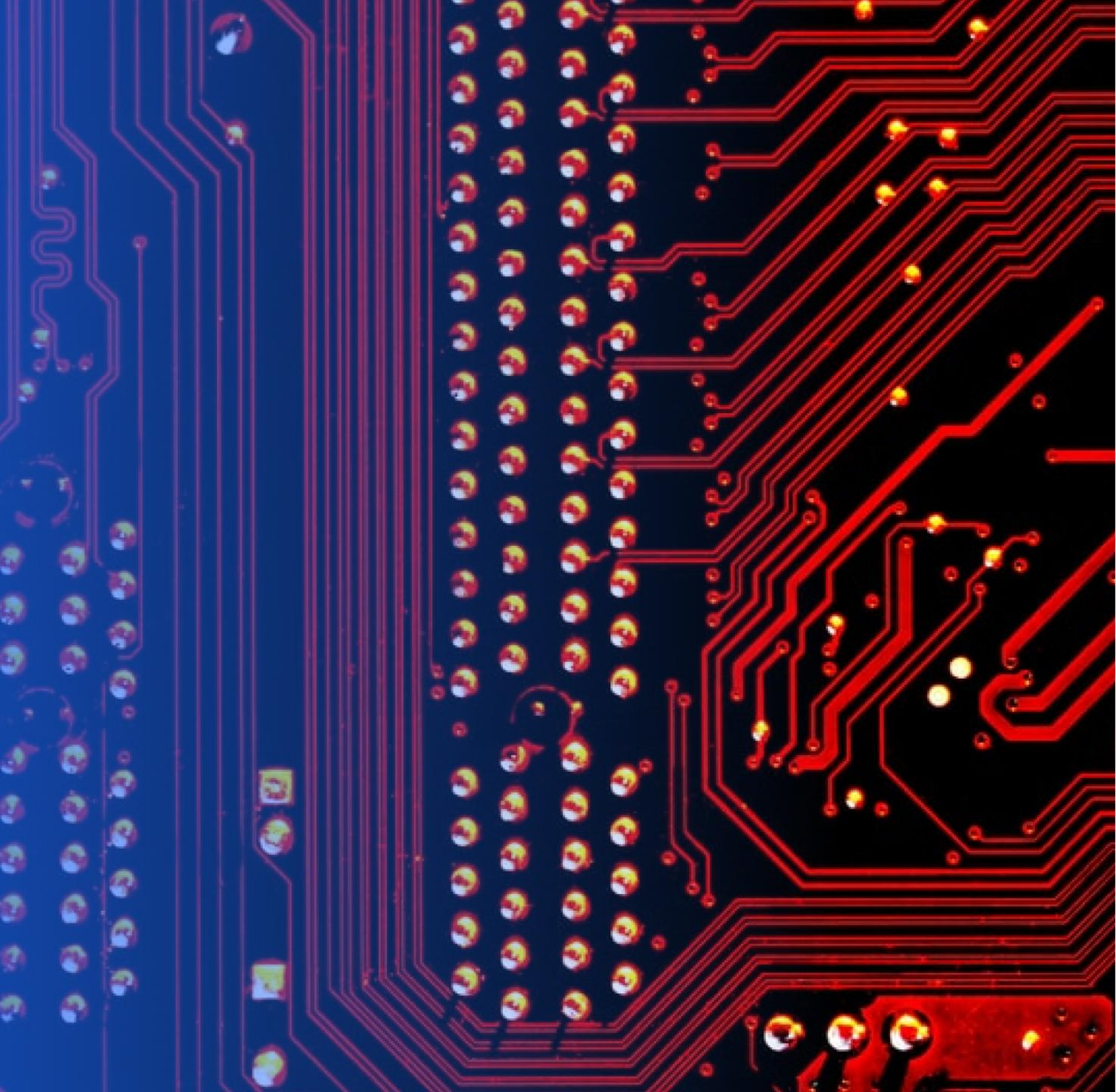
Distances



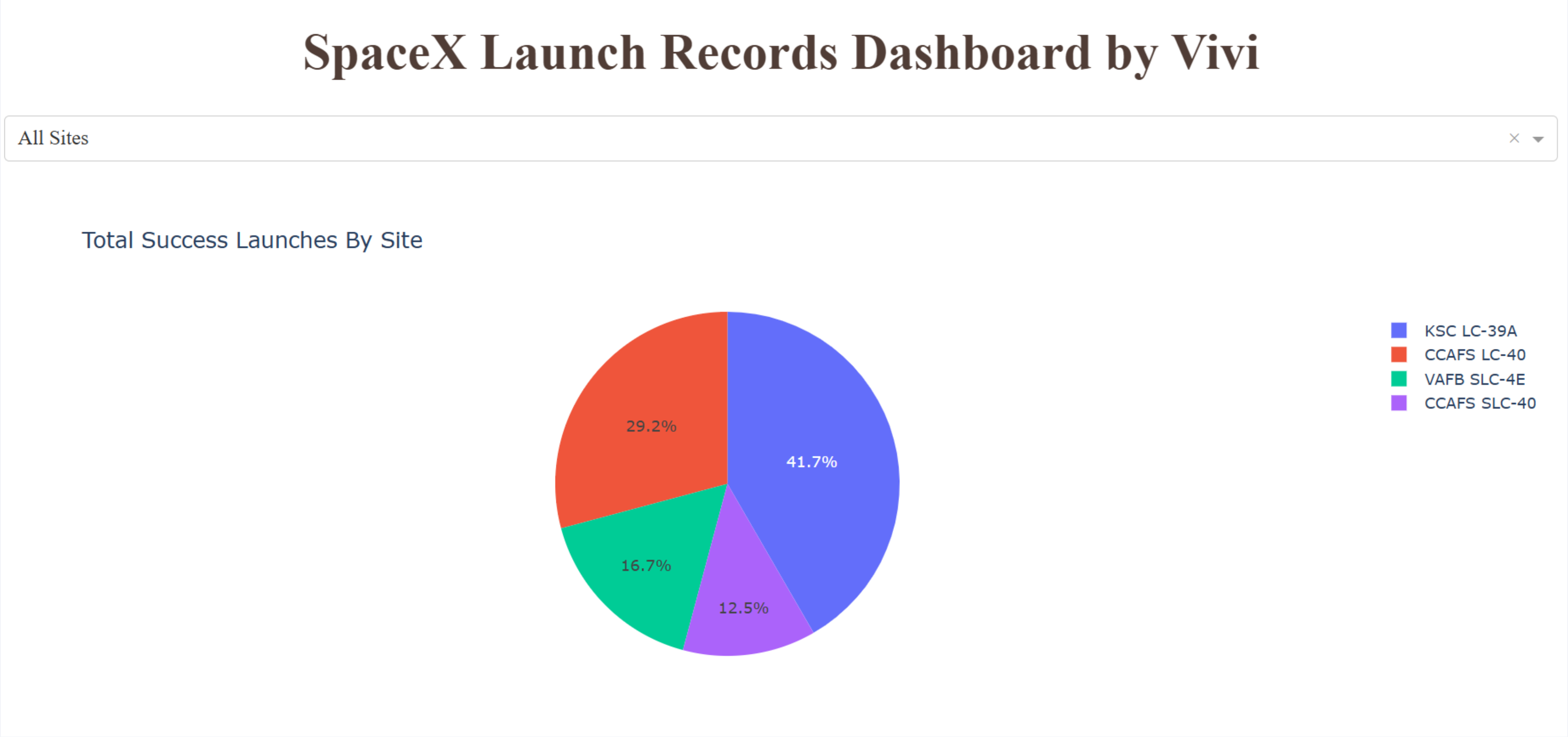
- Draw lined between a launch site to its closest city, railway and a highway

Section 4

Build a Dashboard with Plotly Dash

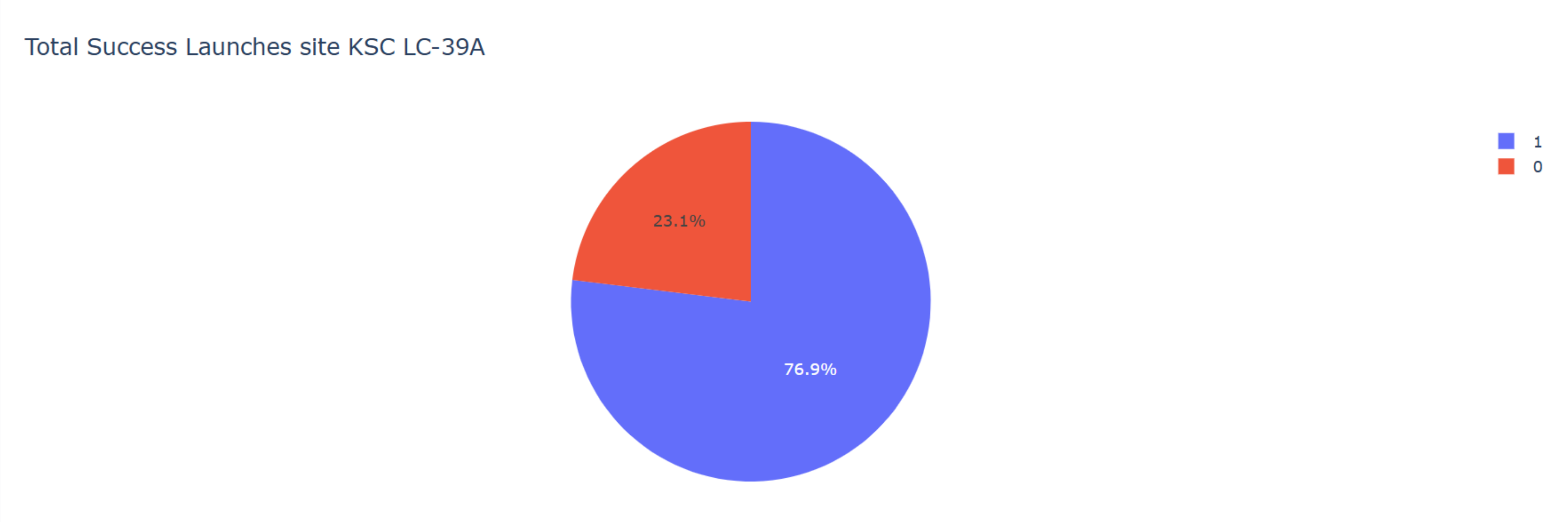


Success Launches By Site



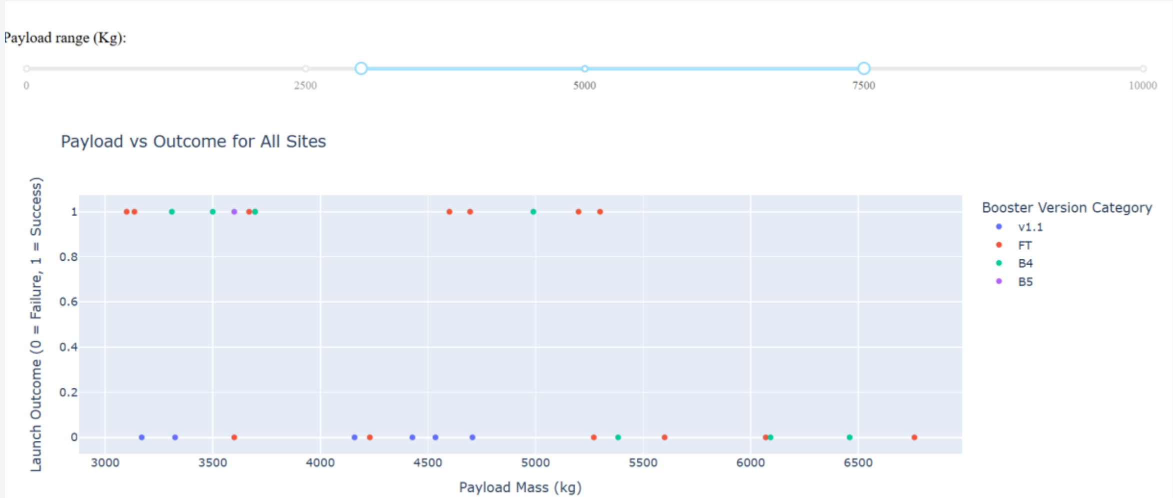
- The Launch Site with more percentage of success is the KSC LC-39A with 41,7%

KSC LC-39A



- The Launch Site with more percentage has 76,9% of success.

Payload vs Outcome for All Sites



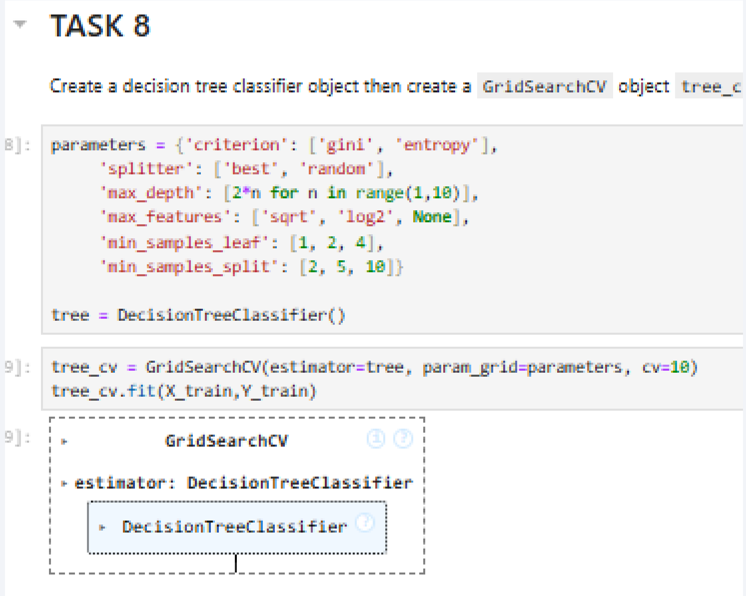
- There are different booster version categories, and the relation can be visualized just in the values of 1 and 0 setting different ranges of payload.

Section 5

Predictive Analysis (Classification)

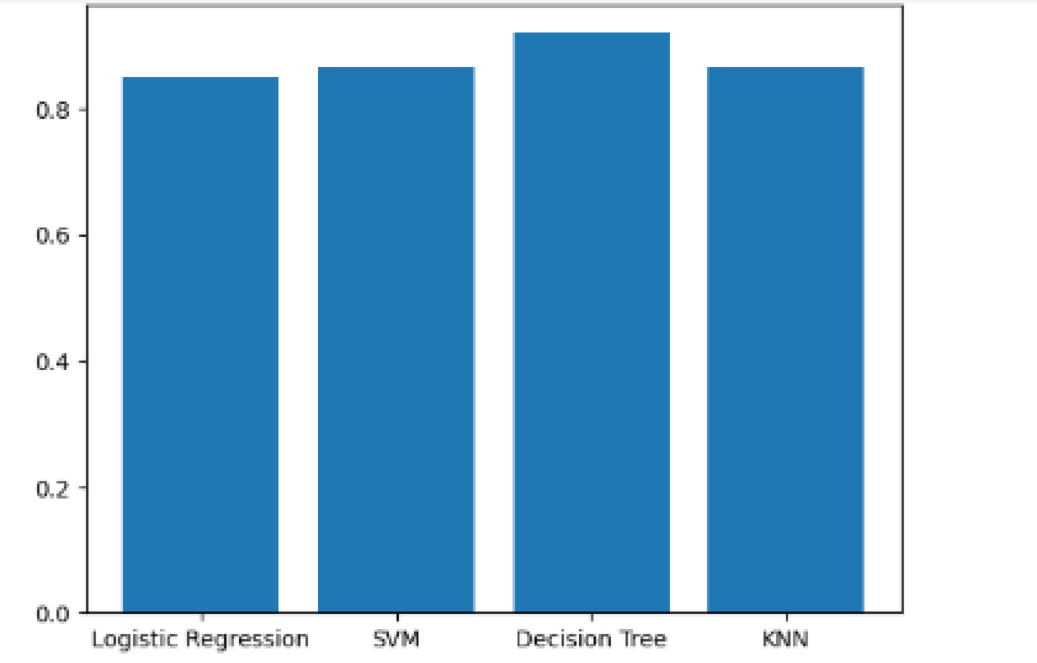
Classification Accuracy

- The best model was Decision Tree with an accuracy of 91%.



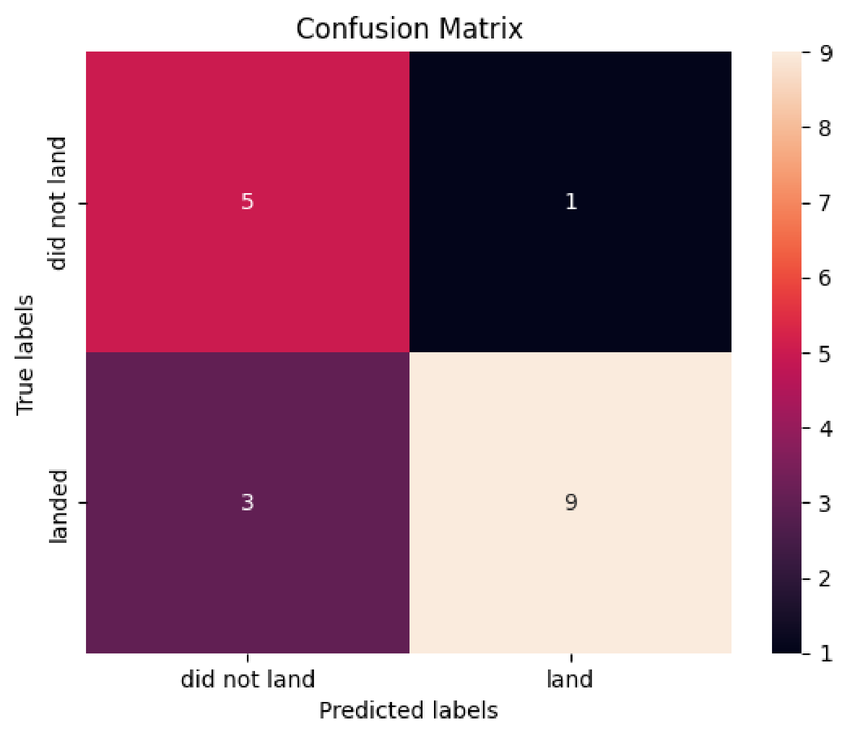

```
parameters = {'criterion': ['gini', 'entropy'],
     'splitter': ['best', 'random'],
     'max_depth': [2*n for n in range(1,10)],
     'max_features': ['sqrt', 'log2', None],
     'min_samples_leaf': [1, 2, 4],
     'min_samples_split': [2, 5, 10]}

tree = DecisionTreeClassifier()
```

```
tree_cv = GridSearchCV(estimator=tree, param_grid=parameters, cv=10)
tree_cv.fit(X_train,Y_train)
```



Confusion Matrix

- This is the confusion matrix of the Decision Tree model. It shows strong values along the diagonal, which represent the correctly classified instances. Specifically, there are 5 true negatives and 9 true positives. The off-diagonal values, representing misclassifications, are relatively small compared to the diagonal values, indicating good overall performance.



Conclusions

- Successfully explored and visualized the SpaceX launch data using maps, scatter plots, and bar charts to identify patterns in launch success related to launch sites, payload mass, and orbit types.
- Calculated distances between launch sites and nearby landmarks, enhancing geographical understanding.
- Built and evaluated several classification models, including Decision Tree, KNN, and SVM, to predict mission outcomes.
- Tuned model parameters using GridSearchCV, selecting the best-performing model based on accuracy.
- The Decision Tree model showed strong predictive capability with high accuracy and low misclassification rates.
- Overall, the analysis provided valuable insights into factors influencing SpaceX mission success and demonstrated the effectiveness of combining exploratory data analysis with predictive modeling.

Appendix

- All the notebooks, code and images are stored on my Github repository
- https://github.com/ivi-bot/DataScienceCoursera

Thank you!