



IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

Luca
05/04/2022



Outline

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

Executive Summary

- Summary of methodologies
 - Web scraping
 - Sql
 - Pandas
- Data visualization whit folium and plotly express
- Machine learning predict
- Summary of all results
 - Interactive visualization
 - Predictive analisys

Introduction

- Project background and context
- space x called us as data science to predict what are the failure factors of a launch and to be able to make space x lose less rockets and money
- Problems you want to find answers
- we will try to understand the rocket's chances of success and their survivability based on a different altitude and predict the chances of success or failure of the rocket

Section 1

Methodology

Methodology

Executive Summary

- Data collection methodology:
 - Using API and web scraping
- Perform data wrangling
 - Hone hot encoding to be used in machine learning
- Perform exploratory data analysis (EDA) using visualization and SQL
- Perform interactive visual analytics using Folium and Plotly Dash
- Perform predictive analysis using classification models
 - How to build, tune, evaluate classification models

Data Collection

- Describe how data sets were collected.
- You need to present your data collection process use key phrases and flowcharts

Json

Pandas

Beatifoulsoup

requests

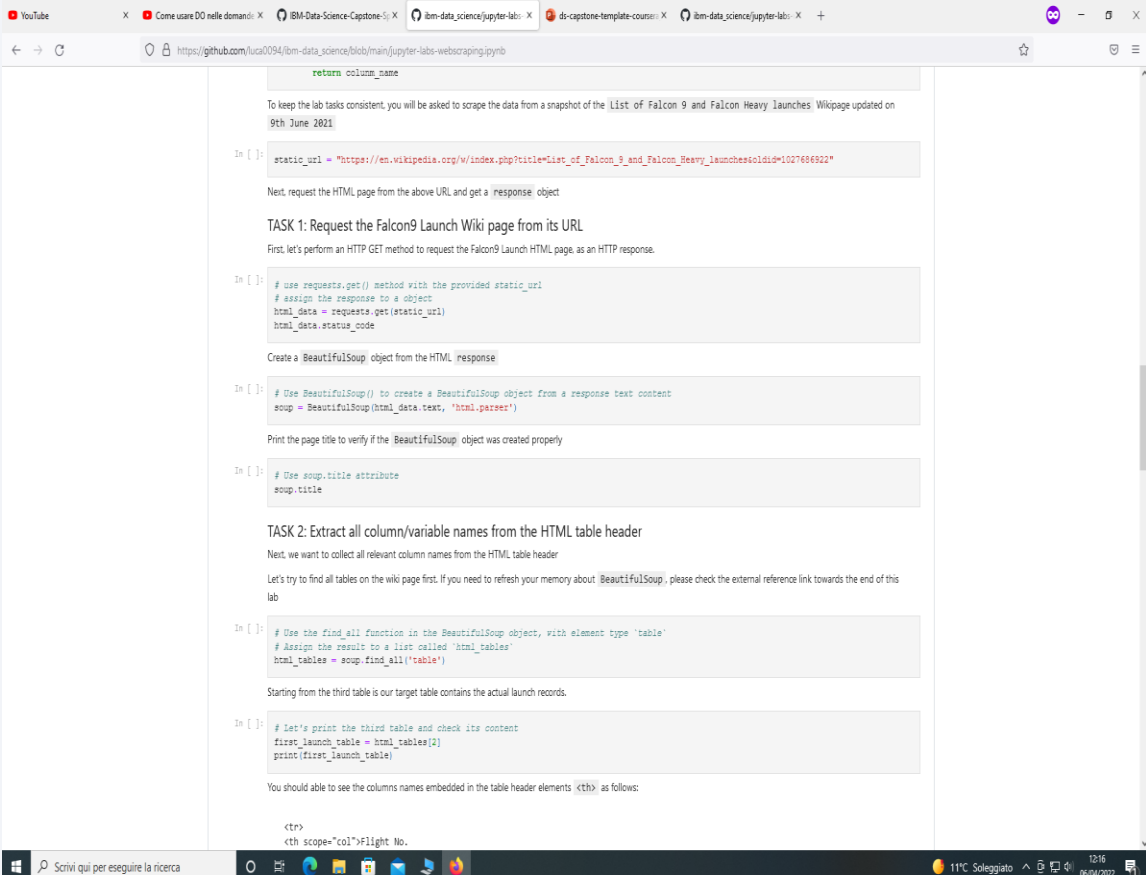
Data Collection – SpaceX API

- Present your data collection with SpaceX REST calls using key phrases and flowcharts
- Add the GitHub URL of the completed SpaceX API calls notebook (must include completed code cell and outcome cell), as an external reference and peer-review purpose

Place your flowchart of SpaceX API calls here

Data Collection - Scraping

- we use the request library to access the api spacex
- Add the GitHub URL of the completed web scraping notebook, https://github.com/luca0094/ibm-data_science/blob/683d634ba d1ce2c1324d3e93c06d579a69 daf24b/jupyter-labs-webscraping.ipynb



The screenshot shows a Jupyter Notebook interface with a browser window at the top displaying the GitHub URL: https://github.com/luca0094/ibm-data_science/blob/main/jupyter-labs-webscraping.ipynb. The notebook content includes:

- A comment: "To keep the lab tasks consistent, you will be asked to scrape the data from a snapshot of the List of Falcon 9 and Falcon Heavy launches. Wikipage updated on 9th June, 2021."
- A code cell with the following Python code:

```
return column_name

static_url = "https://en.wikipedia.org/w/index.php?title=List_of_Falcon_9_and_Falcon_Heavy_launches&oldid=1027696922"

# use requests.get() method with the provided static_url
# assign the response to a object
html_data = requests.get(static_url)
html_data.status_code

# Use BeautifulSoup() to create a BeautifulSoup object from a response text content
soup = BeautifulSoup(html_data.text, "html.parser")

# Use soup.title attribute
soup.title
```
- Task 1: Request the Falcon9 Launch Wiki page from its URL. The code cell shows the same requests.get() call as above.
- Task 2: Extract all column/variable names from the HTML table header. The code cell shows the following Python code:

```
# Use the find_all function in the BeautifulSoup object, with element type 'table'
# Assign the result to a list called 'html_tables'
html_tables = soup.find_all('table')

# Let's print the third table and check its content
first_launch_table = html_tables[2]
print(first_launch_table)
```

The bottom of the screenshot shows the Windows taskbar with the search bar, task view button, and several open applications including Edge, File Explorer, and the Jupyter Notebook.

Data Wrangling

- Describe how data were processed
- data analysis and choice of features
- we calculate the numbers of launches and their orbits
- Transform the file into a csv file
- Add the GitHub URL of your completed data wrangling related notebooks, as an external reference and peer-review purpose
- https://github.com/luca0094/ibm-data_science/blob/835488a1b0d592d8dfc2e1a2e39c2d3c717844ba/ibm_data_wrangling.ipynb

EDA with Data Visualization

- we relate the various phases of the launches through graphs
- Add the GitHub URL of your completed EDA with data visualization notebook, as an external reference and peer-review purpose
- https://github.com/luca0094/ibm-data_science/blob/683d634bad1ce2c1324d3e93c06d579a69daf24b/Data%20Visualization%20Casptone.ipynb

EDA with SQL

- we read the database in sql
- and we extract the information we need
- Add the GitHub URL of your completed EDA with SQL notebook, as an external reference and peer-review purpose
- https://github.com/luca0094/ibm-data_science/blob/683d634bad1ce2c1324d3e93c06d579a69daf24b/jupyter-labs-eda-sql-coursera.ipynb

Build an Interactive Map with Folium

- we use colors to assign a failed or successful roll
- we calculate the distance between the launch zones and the main roads and coasts
- we have marked all launch sites on the map
- Add the GitHub URL of your completed interactive map with Folium map, as an external reference and peer-review purpose
- https://github.com/luca0094/ibm-data_science/blob/683d634bad1ce2c1324d3e93c06d579a69daf24b/lab_jupyter_launch_site_location.ipynb

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
- Add the GitHub URL of your completed Plotly Dash lab, as an external reference and peer-review purpose

Predictive Analysis (Classification)

- we use `GridSearchCv` to find the best parameters to build the model
- found that the best model to use is `DecisionTreeClassifier`
- we build the model using the `Scikit-learn` python library and evaluate its accuracy
- Add the GitHub URL of your completed predictive analysis lab, as an external reference and peer-review purpose
- https://github.com/luca0094/ibm-data_science/blob/683d634bad1ce2c1324d3e93c06d579a69daf24b/SpaceX_Machine%20Learning%20Prediction_Part_5.ipynb

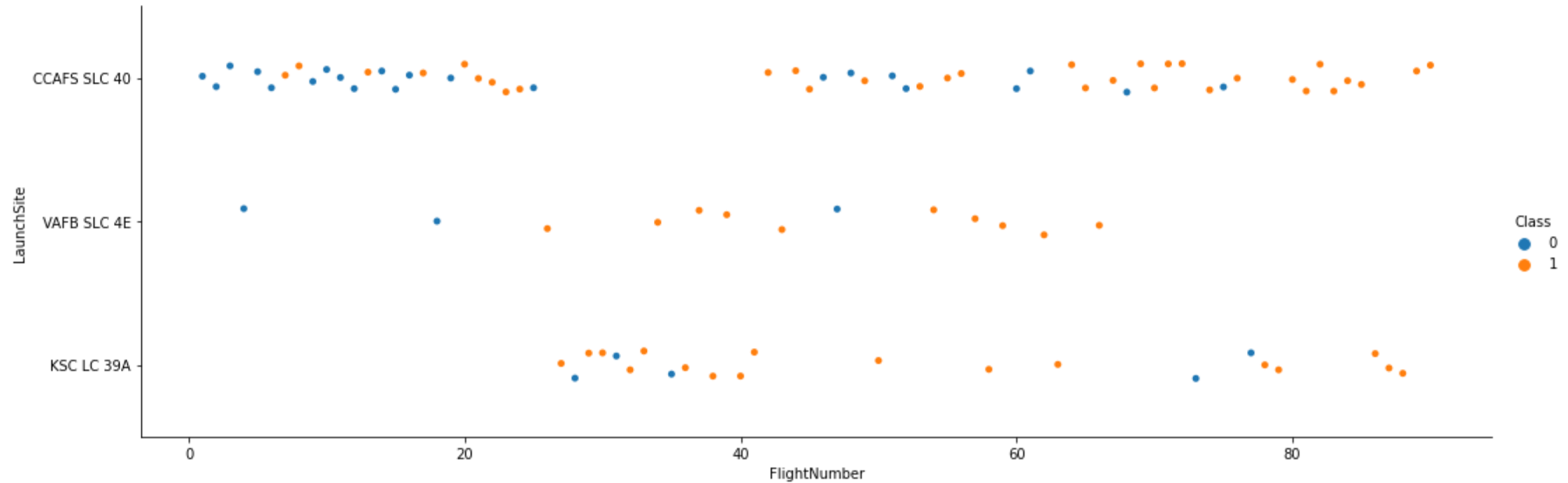
Results

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

The background of the slide is an abstract composition. It features a dark blue field on the left side, which transitions into a complex pattern of diagonal streaks in shades of blue, red, and teal on the right. These streaks have a textured, almost woven appearance. Overlaid on this pattern is a faint, light blue grid that recedes into the distance, creating a sense of depth and perspective.

Section 2

Insights drawn from EDA

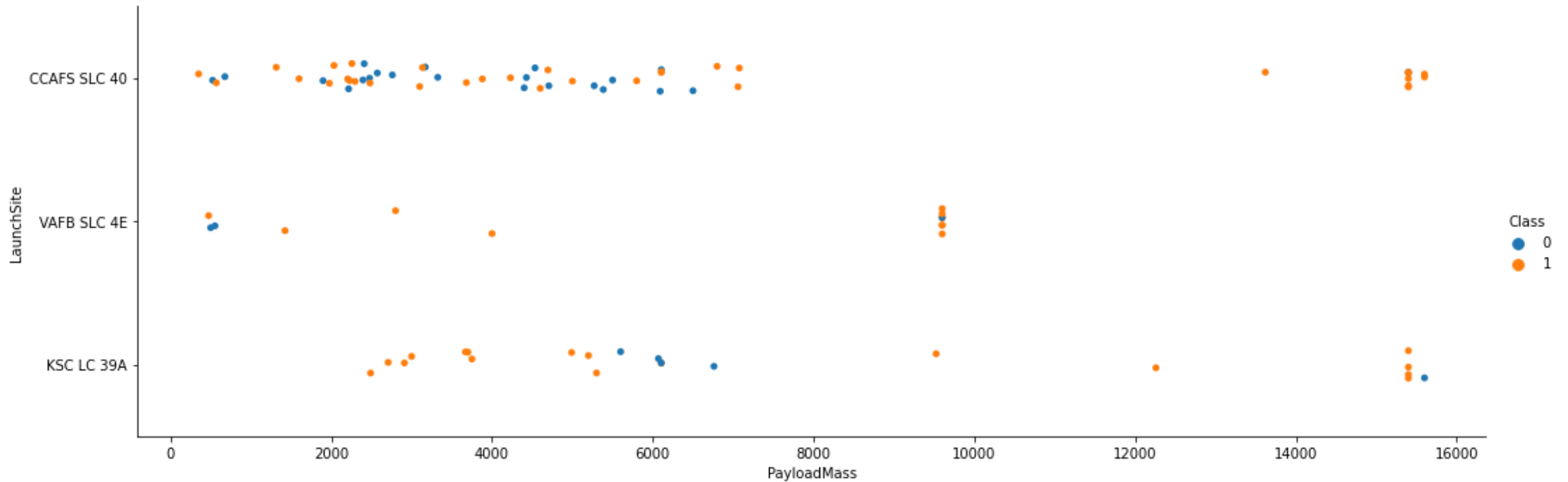


Flight Number vs. Launch Site

- Show a scatter plot of Flight Number vs. Launch Site
- Show the screenshot of the scatter plot with explanations

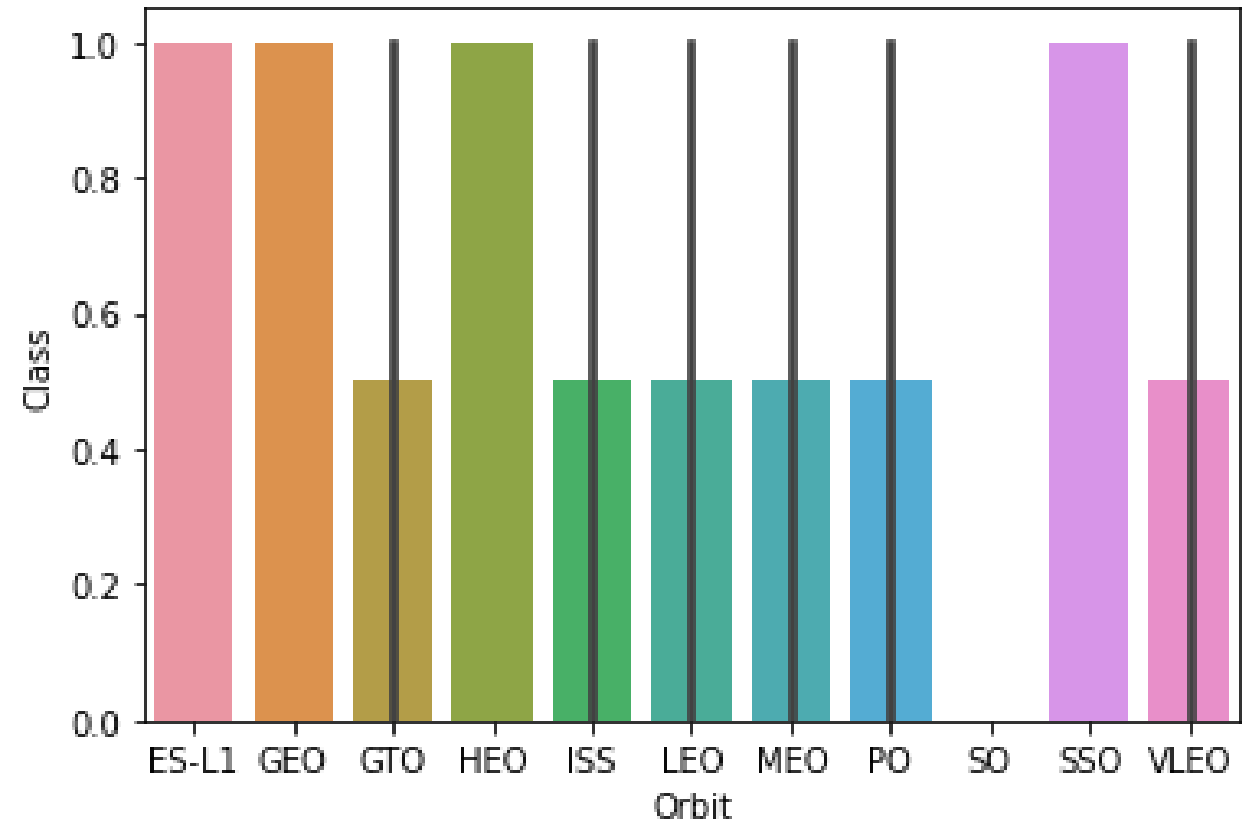
Payload vs. Launch Site

- Show a scatter plot of Payload vs. Launch Site
- Show the screenshot of the scatter plot with explanations



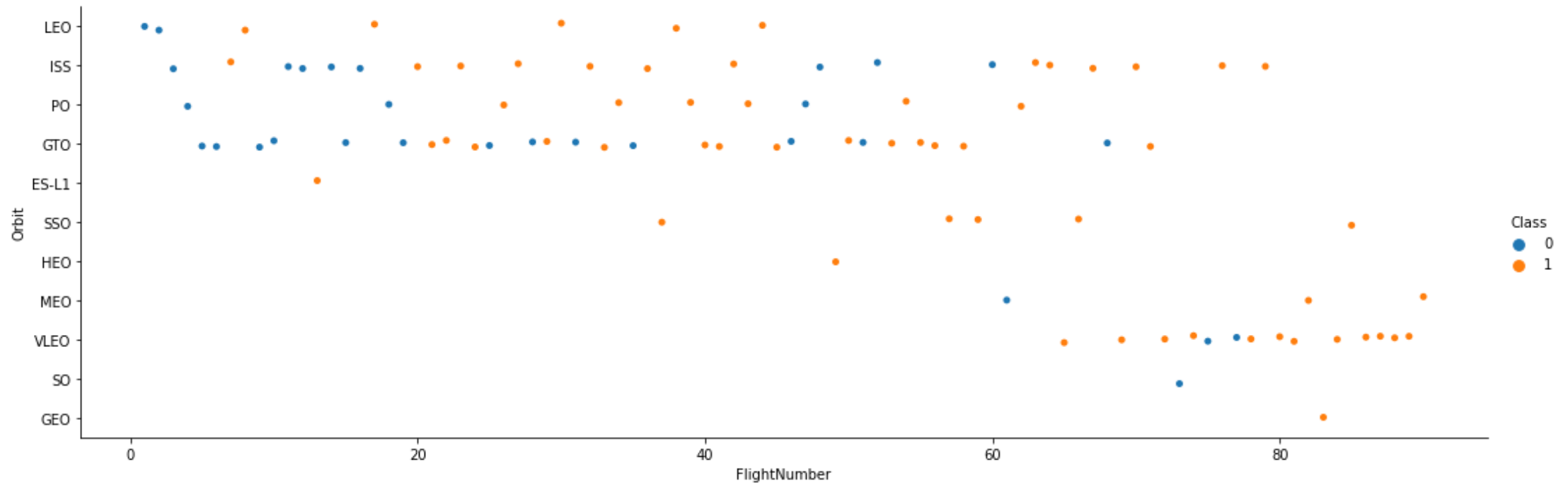
Success Rate vs. Orbit Type

- ES-L1, GEO, HEO, SSO are the best orbits
- Show the screenshot of the scatter plot with explanations



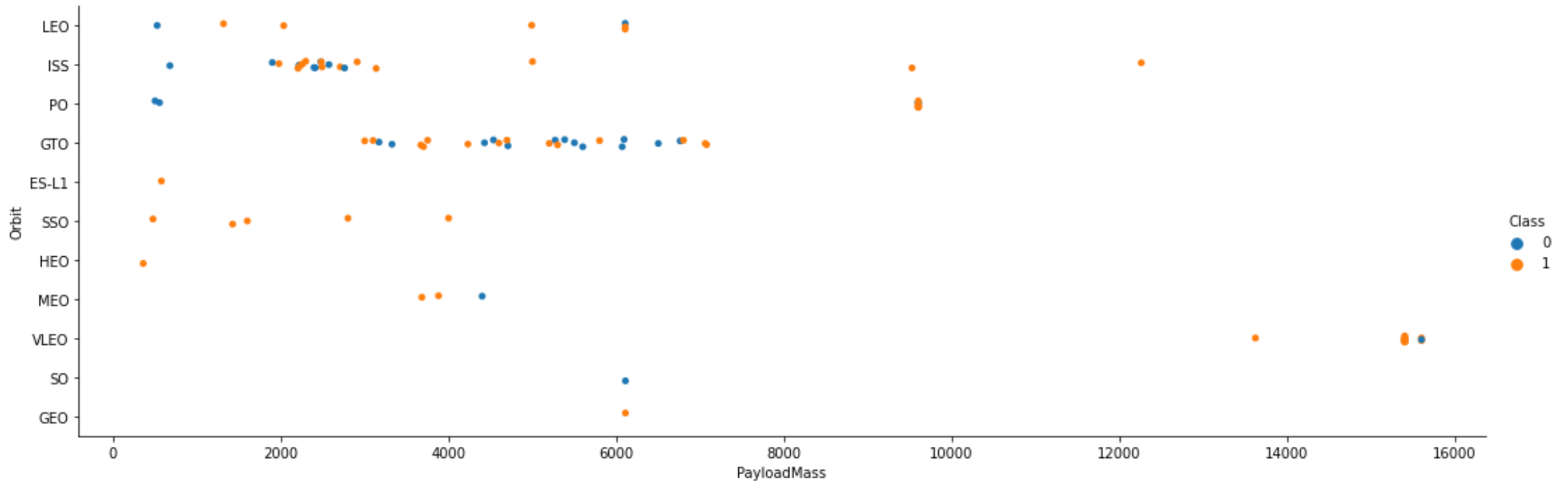
Flight Number vs. Orbit Type

- Show a scatter point of Flight number vs. Orbit type
- Show the screenshot of the scatter plot with explanations



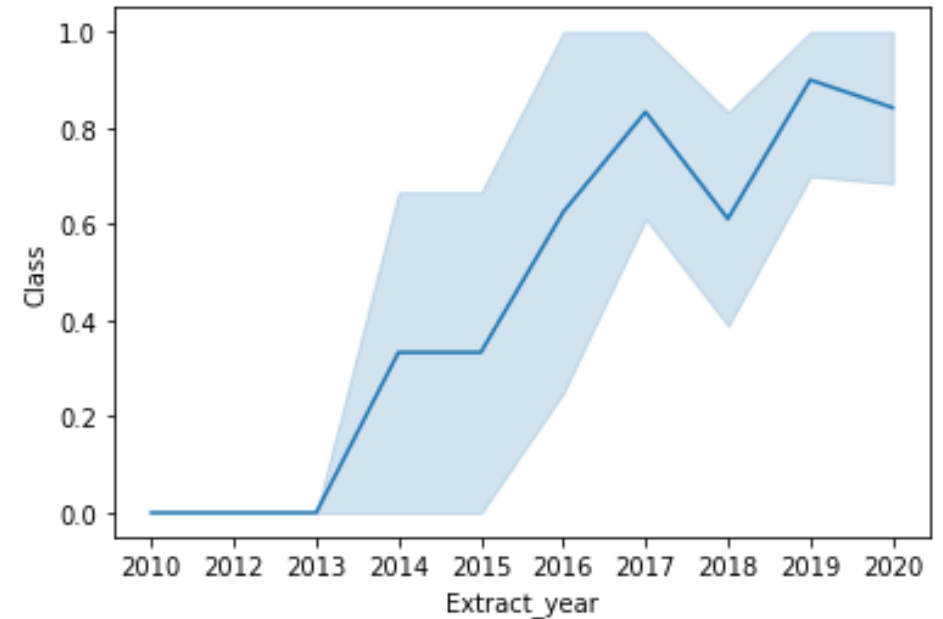
Payload vs. Orbit Type

- Show a scatter point of payload vs. orbit type
- Show the screenshot of the scatter plot with explanations



Launch Success Yearly Trend

- the success of the launches has always been increasing except in 2018 but has immediately started to grow again
- Show the screenshot of the scatter plot with explanations



All Launch Site Names

- Find the names of the unique launch sites
- Present your query result with a short explanation here



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

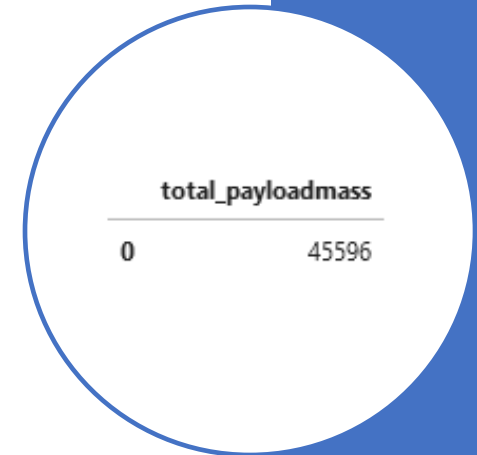
Launch Site Names Begin with 'CCA'

- Find 5 records where launch sites begin with 'CCA'
- Present your query result with a short explanation here

	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

Total Payload Mass

- Calculate the total payload carried by boosters from NASA
- Present your query result with a short explanation here



total_payloadmass	
0	45596

Average Payload Mass by F9 v1.1

27

avg_payloadmass	
0	2928.4

- Calculate the average payload mass carried by booster version F9 v1.1
- Present your query result with a short explanation here

First Successful Ground Landing Date

- Find the dates of the first successful landing outcome on ground pad
- Present your query result with a short explanation here

firstsuccessfull_landing_date	
0	2015-12-22

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
- Present your query result with a short explanation here

booster version	
0	F9 FT B1022
1	F9 FT B1026
2	F9 FT B1021.2
3	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

The total number of successful mission outcome is:

successoutcome	
0	100

The total number of failed mission outcome is:

failureoutcome	
0	1

Boosters Carried Maximum Payload

- List the names of the booster which have carried the maximum payload mass
- Present your query result with a short explanation here

	boosterversion	payloadmasskg
0	F9 B5 B1048.4	15600
1	F9 B5 B1048.5	15600
2	F9 B5 B1049.4	15600
3	F9 B5 B1049.5	15600
4	F9 B5 B1049.7	15600
5	F9 B5 B1051.3	15600
6	F9 B5 B1051.4	15600
7	F9 B5 B1051.6	15600
8	F9 B5 B1056.4	15600
9	F9 B5 B1058.3	15600
10	F9 B5 B1060.2	15600
11	F9 B5 B1060.3	15600

2015 Launch Records

- List the failed landing_outcomes in drone ship, their booster versions, and launch site names for in year 2015
- Present your query result with a short explanation here

	boosterversion	launchsite	landingoutcome
0	F9 v1.1 B1012	CCAFS LC-40	Failure (drone ship)
1	F9 v1.1 B1015	CCAFS LC-40	Failure (drone ship)

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
- Present your query result with a short explanation here

	landingoutcome	count
0	No attempt	10
1	Success (drone ship)	6
2	Failure (drone ship)	5
3	Success (ground pad)	5
4	Controlled (ocean)	3
5	Uncontrolled (ocean)	2
6	Precluded (drone ship)	1
7	Failure (parachute)	1

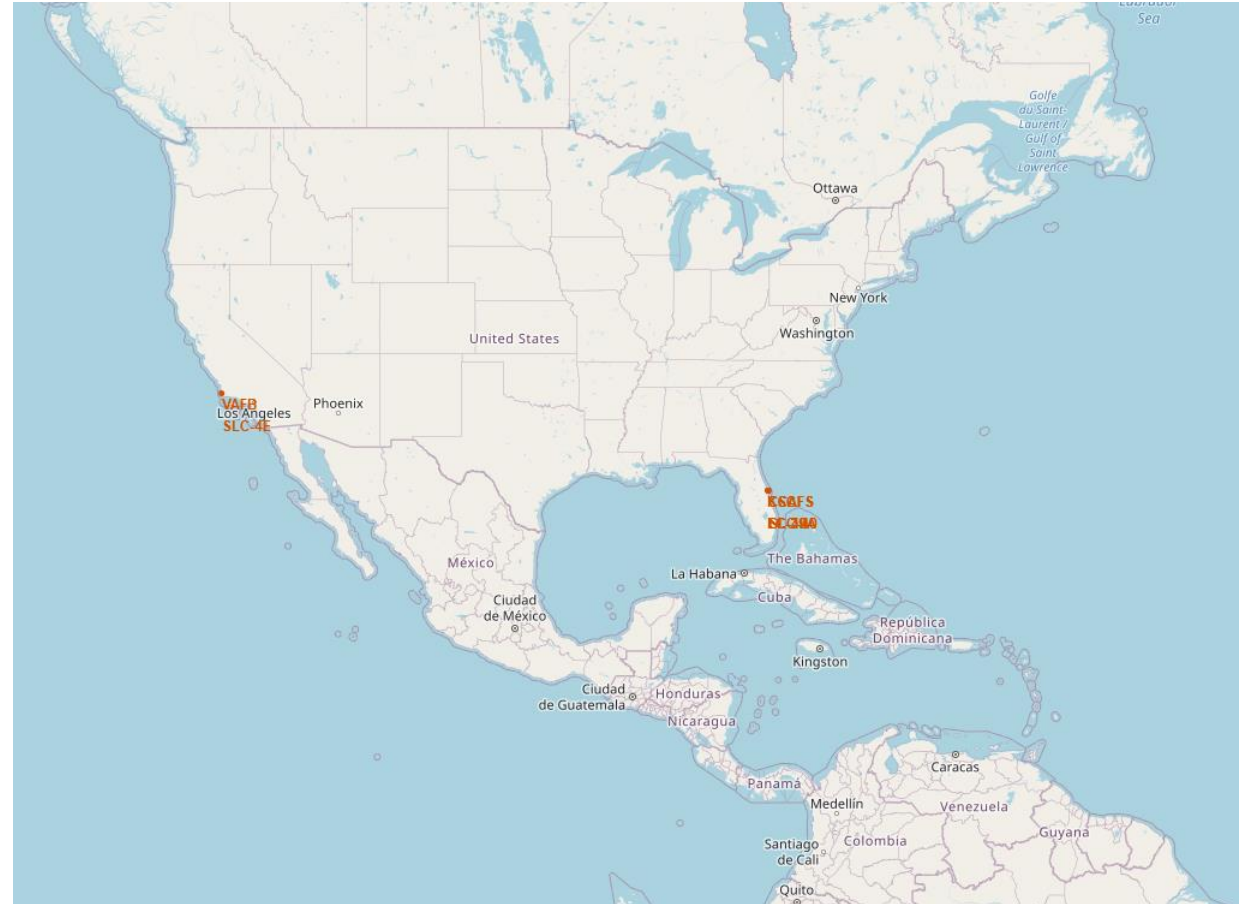
A satellite view of Earth from space, showing the curvature of the planet and city lights at night. The image is a composite of a solid blue background on the left and a satellite photograph of Earth on the right. The Earth's surface is dark, with numerous bright yellow and orange lights representing cities and urban areas. The horizon of the Earth is visible as a curved line separating the dark surface from the deep blue of space.

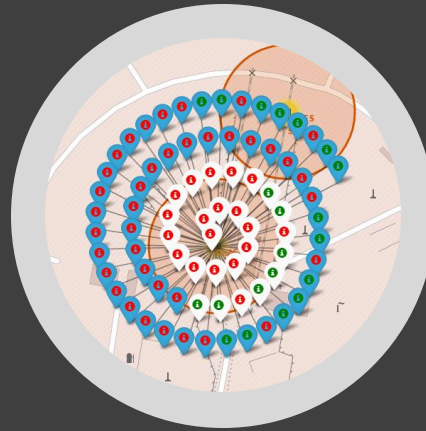
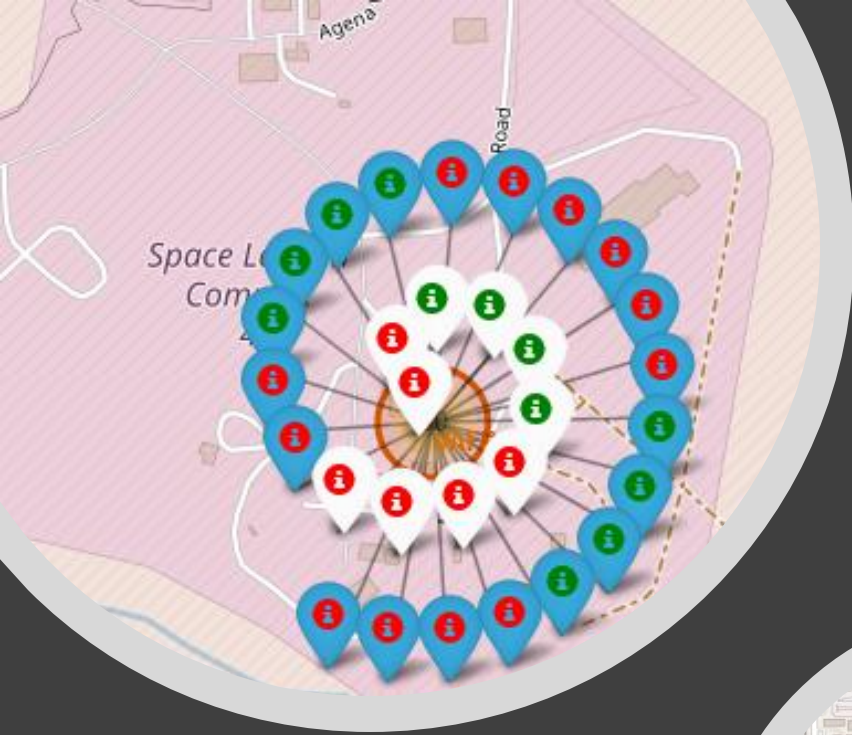
Section 3

Launch Sites Proximities Analysis

Launch site space x

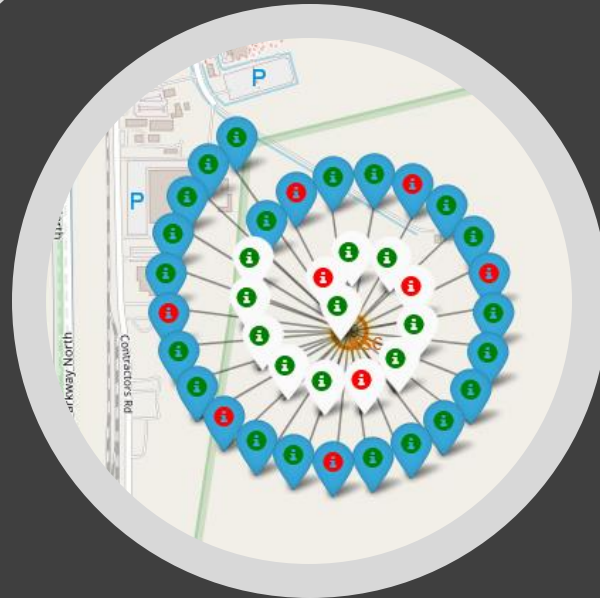
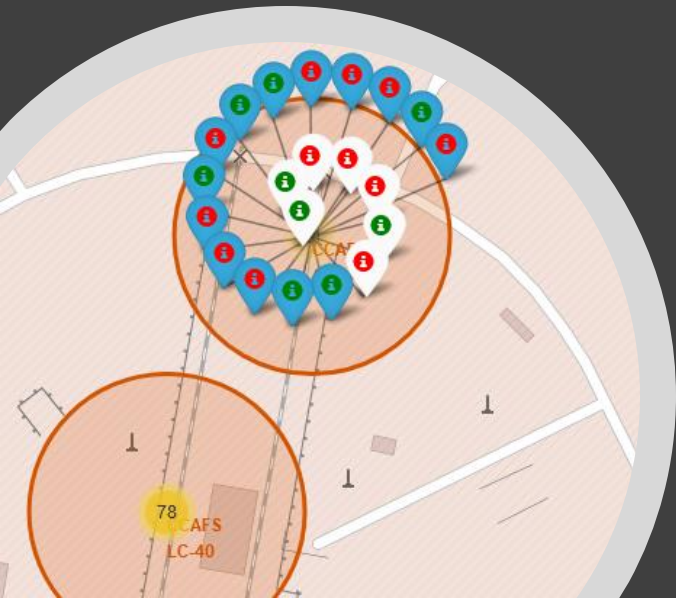
- Replace <Folium map screenshot 1> title with an appropriate title
- Explore the generated folium map and make a proper screenshot to include all launch sites' location markers on a global map
- Explain the important elements and findings on the screenshot

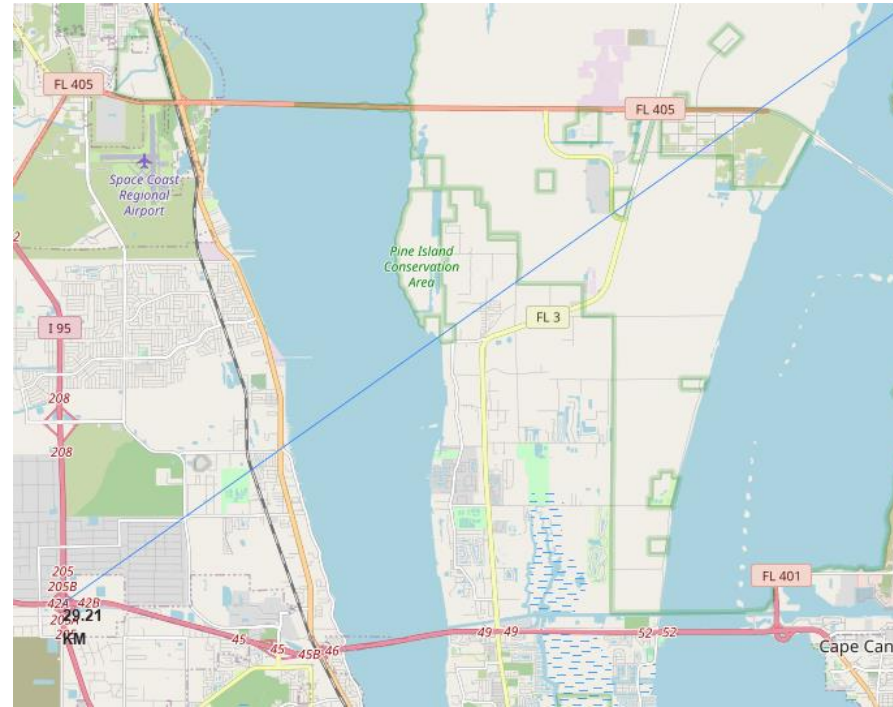




color-labeled launch outcomes on the map

- Replace <Folium map screenshot 2> title with an appropriate title
- Explore the folium map and make a proper screenshot to show the color-labeled launch outcomes on the map
- Explain the important elements and findings on the screenshot





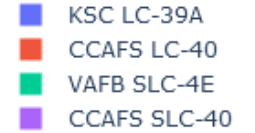
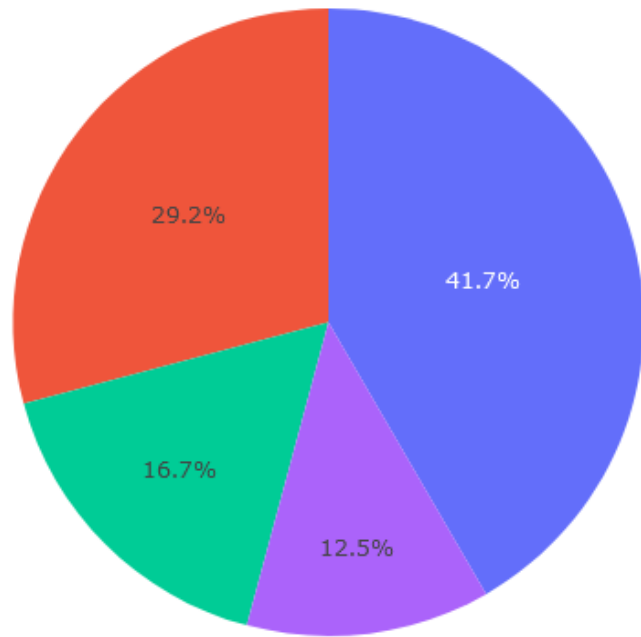
launch site to its
proximities such
as railway

- Replace <Folium map screenshot 3> title with an appropriate title
- Explore the generated folium map and show the screenshot of a selected launch site to its proximities such as railway, highway, coastline, with distance calculated and displayed
- Explain the important elements and findings on the screenshot



Section 4

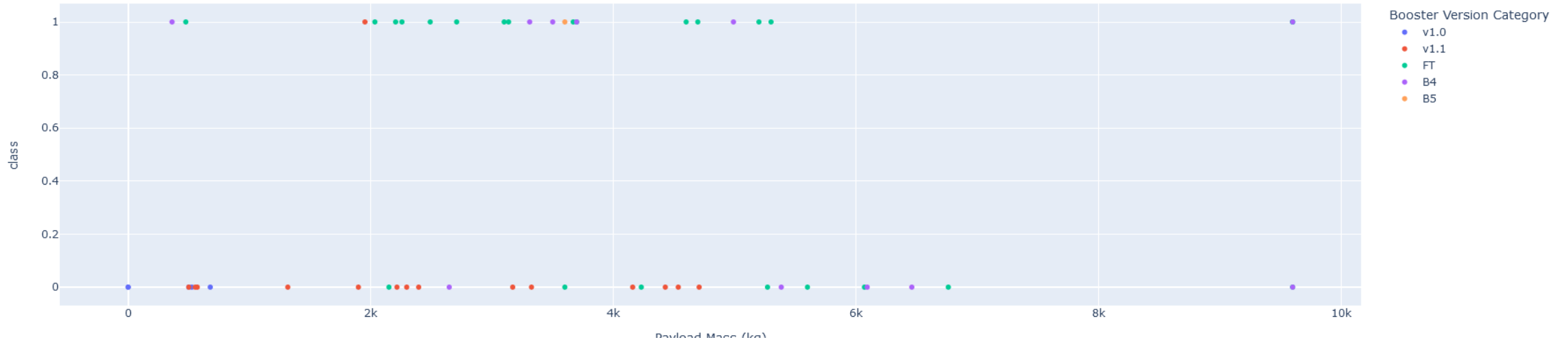
Build a Dashboard with Plotly Dash



Launch success for all site

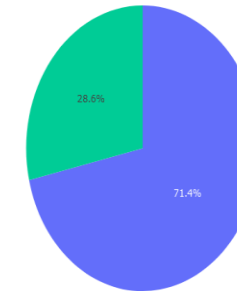
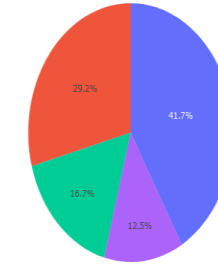
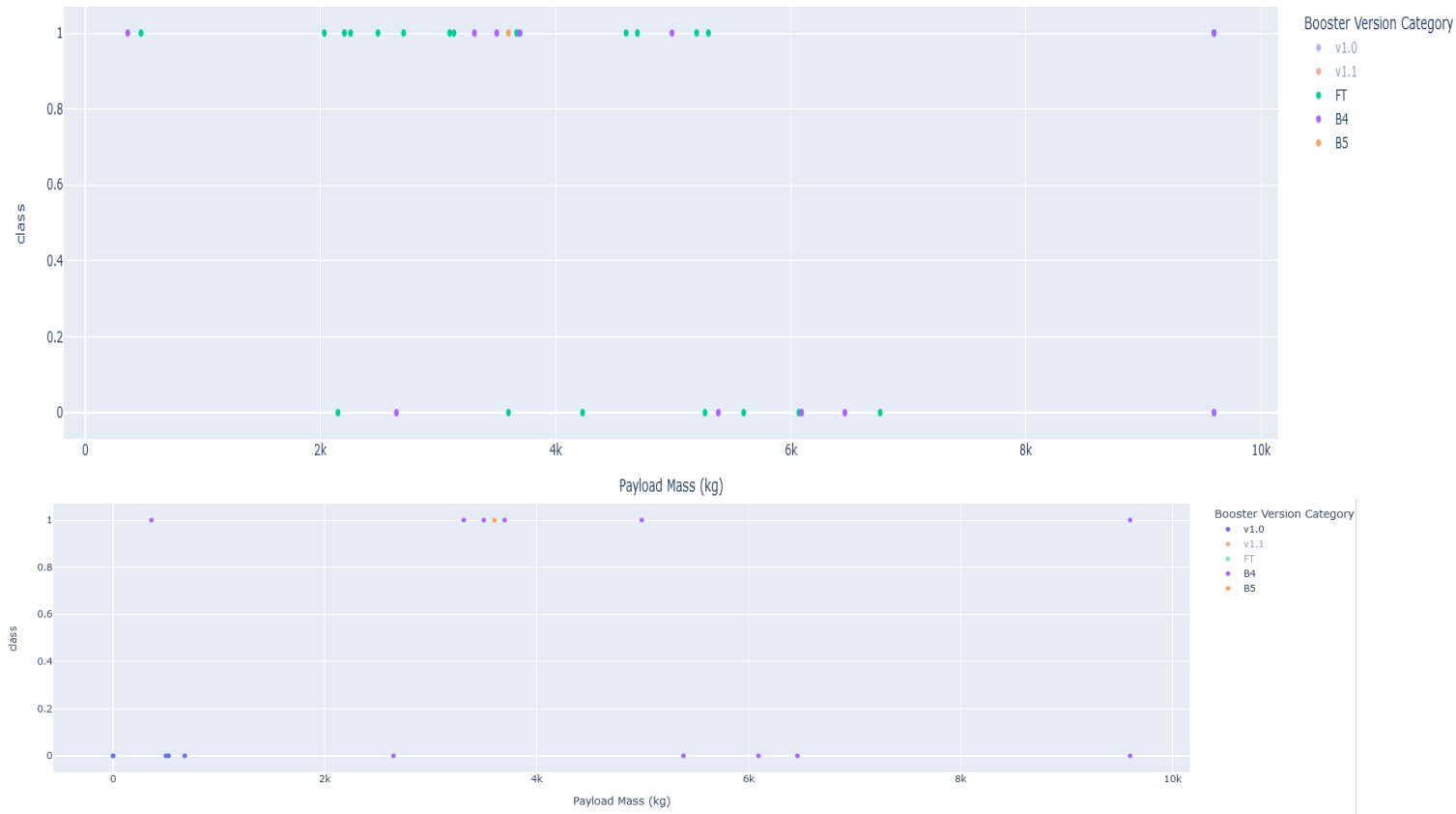
- Replace <Dashboard screenshot 1> title with an appropriate title
- Show the screenshot of launch success count for all sites, in a piechart
- KSC is the best site

correlation between payload and success for all sites



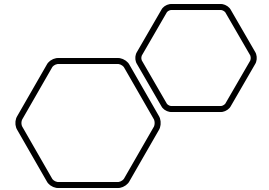
launch site with highest launch success ratio

- 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



• payload selected in the range slider

- 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)

```
gscv=GridSearchCV(svm,parameters,scoring='accuracy',cv=10)
logreg_cv=gscv.fit(X_train,Y_train)
print("tuned hpyerparameters : (best parameters) ",logreg_cv.best_params_)
print("accuracy :",logreg_cv.best_score_)
logreg_cv.score(X_train, Y_train)

tuned hpyerparameters : (best parameters) {'C': 1.0, 'gamma': 0.03162277660168379, 'kernel': 'sigmoid'}
accuracy : 0.8482142857142856
0.8888888888888888

[ ] parameters = {'criterion': ['gini', 'entropy'],
                  'splitter': ['best', 'random'],
                  'max_depth': [2*n for n in range(1,10)],
                  'max_features': ['auto', 'sqrt'],
                  'min_samples_leaf': [1, 2, 4],
                  'min_samples_split': [2, 5, 10]}

tree = DecisionTreeClassifier()
gscv=GridSearchCV(tree,parameters,scoring='accuracy',cv=10)
logreg_cv=gscv.fit(X_train,Y_train)
print("tuned hpyerparameters : (best parameters) ",logreg_cv.best_params_)
print("accuracy :",logreg_cv.best_score_)
logreg_cv.score(X_train, Y_train)

tuned hpyerparameters : (best parameters) {'criterion': 'entropy', 'max_depth': 4, 'max_features': 'sqrt', 'min_samples_leaf': 2, 'min_samples_split': 2, 'splitter': 'random'}
accuracy : 0.8857142857142858
0.8611111111111112

[ ] parameters = {'n_neighbors': [1, 2, 3, 4, 5, 6, 7, 8, 9, 10],
                  'algorithm': ['auto', 'ball_tree', 'kd_tree', 'brute'],
                  'p': [1,2]}

KNN = KNeighborsClassifier()
gscv=GridSearchCV(KNN,parameters,scoring='accuracy',cv=10)
logreg_cv=gscv.fit(X_train,Y_train)
print("tuned hpyerparameters : (best parameters) ",logreg_cv.best_params_)
print("accuracy :",logreg_cv.best_score_)
logreg_cv.score(X_train, Y_train)

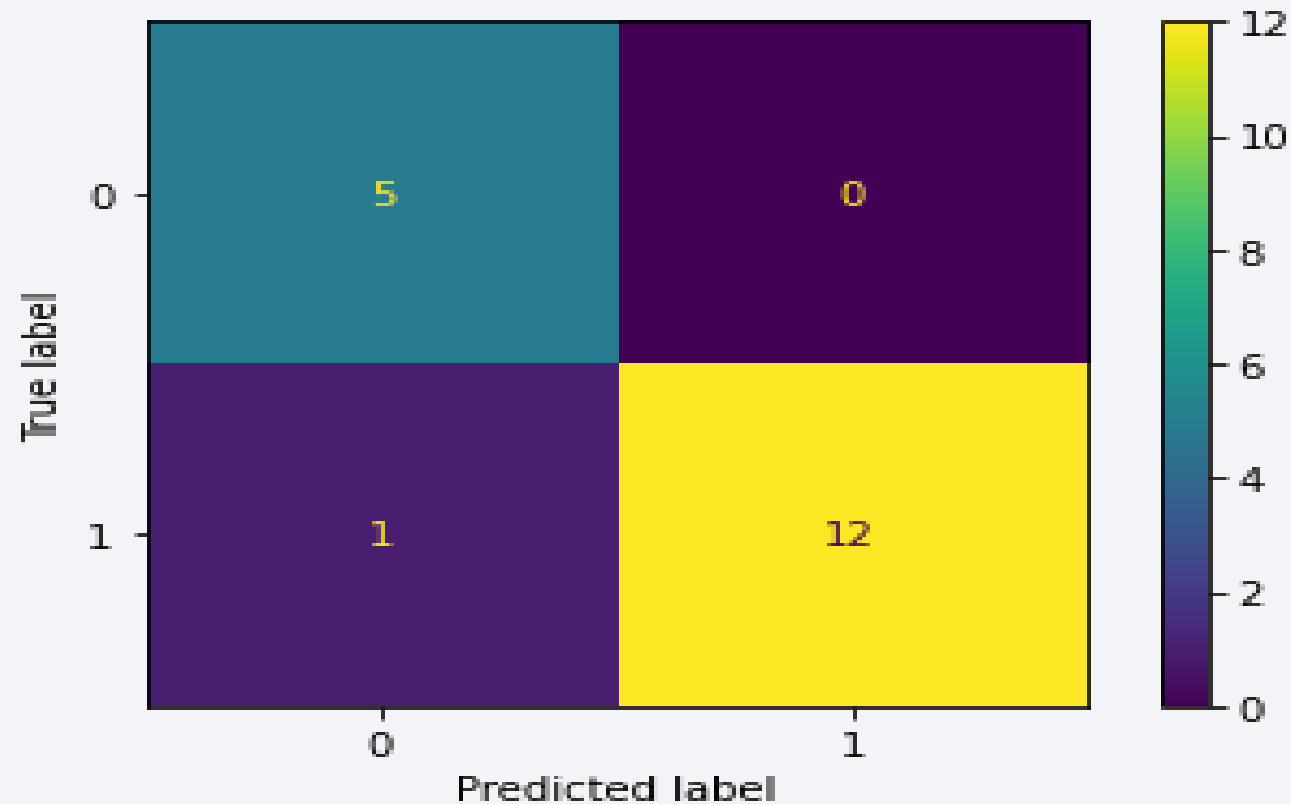
tuned hpyerparameters : (best parameters) {'algorithm': 'auto', 'n_neighbors': 10, 'p': 1}
accuracy : 0.8482142857142858
0.8611111111111112
```

Classification Accuracy

- Visualize the built model accuracy for all built classification models, in a bar chart
- The best accuracy is DecisionTreeClassifier with an accuracy of 0.8857142857142858

Confusion Matrix

- Show the confusion matrix of the best performing model with an explanation



Conclusions

- space x has been increasing its successful rocket launch since 2013
- KSC LC-39A is the best site for successful launches
- Es-L1,GEO,HEO,SSO are the best orbits
- Decision Tree classifier is the best algorithm for this situation

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!

