IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

Cristian Sandoval
20/05/2025

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:
  - Describe how data was collected
- Perform data wrangling
  - Describe how data was processed
- 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

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 you PI calls here

jupyter-labs-spacex-data-collection-api.ipynb

Data Collection - Scraping

- Present your web scraping process using key phrases and flowcharts

- Add the GitHub URL of the completed web scraping notebook, as an external reference and peer-review purpose

jupyter-labs-webscraping.ipynb

Data Wrangling

- Describe how data were processed
- You need to present your data wrangling process using key phrases and flowcharts
- Add the GitHub URL of your completed data wrangling related notebooks, as an external reference and peer-review purpose

labs-jupyter-spacex-Data wrangling.ipynb

EDA with Data Visualization

- Summarize what charts were plotted and why you used those charts
- Add the GitHub URL of your completed EDA with data visualization notebook, as an external reference and peer-review purpose

El primer grafico de puntos: el número de vuelo frente a la masa de la carga útil y superponer el resultado del lanzamiento. A medida que aumenta el número de vuelo, la primera etapa tiene más probabilidades de aterrizar con éxito. La masa de la carga útil también parece ser un factor; incluso con cargas más pesadas, la primera etapa suele regresar con éxito.
El gráfico de puntos de dispersión de masa de carga útil versus lugar de lanzamiento muestra que para el sitio de lanzamiento VAFB-SLC no hay cohetes lanzados para una masa de carga útil pesada (superior a 10 000).
El siguiente gráfico muestra que, con cargas útiles pesadas, la tasa de aterrizajes exitosos o positivos es mayor en la Estación Polar, la LEO y la ISS. Sin embargo, para la GTO, es difícil distinguir entre aterrizajes exitosos y fallidos, ya que ambos resultados están presentes.
El siguiente gráfico lineal muestra que la tasa de éxito desde 2013 siguió aumentando hasta 2020

labs-jupyter-spacex-Data wrangling.ipynb

edadataviz.ipynb

EDA with SQL

- Using bullet point format, summarize the SQL queries you performed
- Add the GitHub URL of your completed EDA with SQL notebook, as an external reference and peer-review purpose

- Primero se carga la extensión SQL y establezcamos una conexión con la base de datos
- El primer objetivo es mostrar los nombres de los sitios de lanzamiento únicos de la misión espacial.
- Mostrar 5 registros donde los sitios de lanzamiento comienzan con la cadena 'CCA'
- Mostrar la masa total de carga útil transportada por los propulsores lanzados por la NASA (CRS)
- Mostrar la masa promedio de carga útil transportada por la versión F9 v1.1 del propulsor
- Enumerar la fecha en la que se logró el primer aterrizaje exitoso en la plataforma de tierra.
- Enumerar los nombres de los propulsores que tienen éxito en los barcos no tripulados y tienen una masa de carga útil mayor a 4000 pero menor a 6000
- Enumerar el número total de resultados de misiones exitosas y fallidas
- Enumerar todas las booster_versions que han transportado la carga útil máxima. Use una subconsulta.
- Enumerar los registros que mostrarán los nombres de los meses, los resultados de los aterrizajes fallidos en el barco no tripulado, las versiones de refuerzo y el sitio de lanzamiento para los meses del año 2015.
- Clasificar el recuento de resultados de aterrizaje (como Fracaso (nave no tripulada) o Éxito (plataforma de tierra)) entre la fecha del 4 de junio de 2010 y el 20 de marzo de 2017, en orden descendente.

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
- Add the GitHub URL of your completed interactive map with Folium map, as an external reference and peer-review purpose

Se importan las librerías incluyendo Folium
Se utilizan para crear mapas interactivos, definiendo lugares específicos a través de coordenadas

Primero: Marcar todos los sitios de lanzamiento en un mapa
Segundo: Marcar los lanzamientos exitosos/fallidos de cada sitio en el mapa
Tercero: Calcular las distancias entre un sitio de lanzamiento y sus proximidades

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

- Agregar un Componente de Entrada de Menú Desplegable para Sitios de Lanzamiento.
Tenemos cuatro sitios de lanzamiento diferentes y nos gustaría ver primero cuál tiene la mayor cantidad de éxitos. Luego,
nos gustaría seleccionar un sitio específico y verificar su tasa de éxito detallada (clase=0 vs. clase=1).
- Agrega una función de callback para renderizar success-pie-chart basado en el sitio seleccionado del menú desplegable
La idea general de esta función de callback es obtener el sitio de lanzamiento seleccionado del site-dropdown y renderizar un gráfico de pastel que visualice los conteos de éxito de lanzamientos.
- Agregar un Control deslizante de rango para seleccionar la carga útil
Averiguar si la variable carga útil está correlacionada con el resultado de la misión. Desde el punto de vista de un panel de control, queremos poder seleccionar fácilmente diferentes rangos de carga útil y ver si podemos identificar algunos patrones visuales.
- Agrega una función de callback para renderizar el gráfico de dispersión success-payload-scatter-chart
Trazar un gráfico de dispersión con el eje x como la carga útil y el eje y como el resultado del lanzamiento (es decir, la columna class). De esta manera, podemos observar visualmente cómo la carga útil puede estar correlacionada con los resultados de la misión para el(los) sitio(s) seleccionado(s).

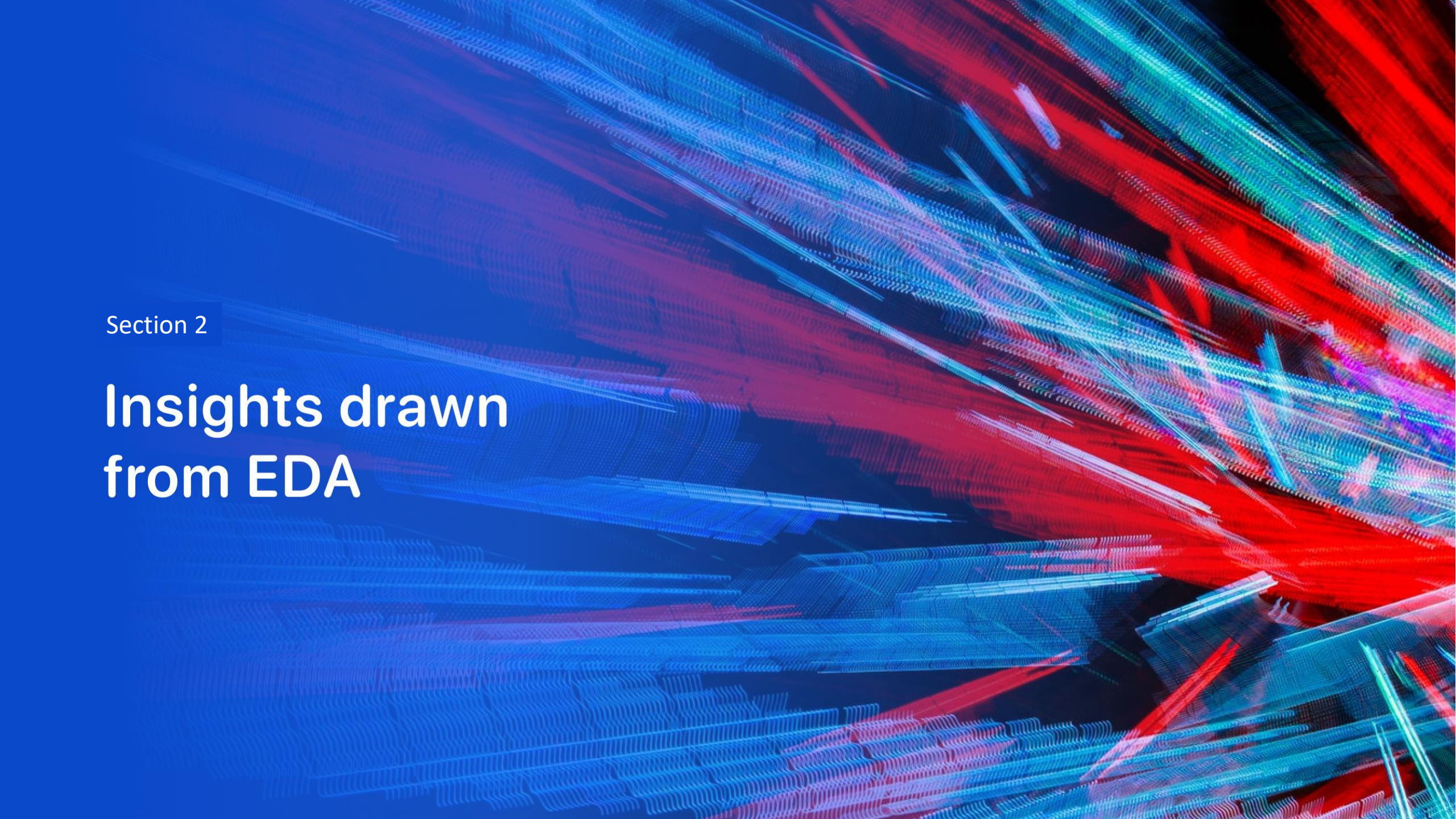
Predictive Analysis (Classification)

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

- Crear una matriz NumPy a partir de la columna Class en datos, aplicando el método to_numpy().
- Estandarizar los datos en X y luego reasignelos a la variable X
- Usar la función train_test_split para dividir los datos X e Y en datos de entrenamiento y de prueba. Establezca el parámetro test_size en 0.2 y random_state en 2.
- Crear un objeto de regresión logística y luego cree un objeto GridSearchCV logreg_cv con cv = 10.
- Calcular la precisión de los datos de prueba
- Crear un objeto de máquina de vectores de soporte y luego cree un objeto GridSearchCV svm_cv con cv = 10.
- Crear un objeto clasificador de árbol de decisión y luego cree un objeto GridSearchCV tree_cv con cv = 10.
- Calcule la precisión de tree_cv en los datos de prueba
- Crear un objeto de k vecinos más cercanos y luego cree un objeto GridSearchCV knn_cv con cv = 10.
- Encuentra el método que funciona mejor

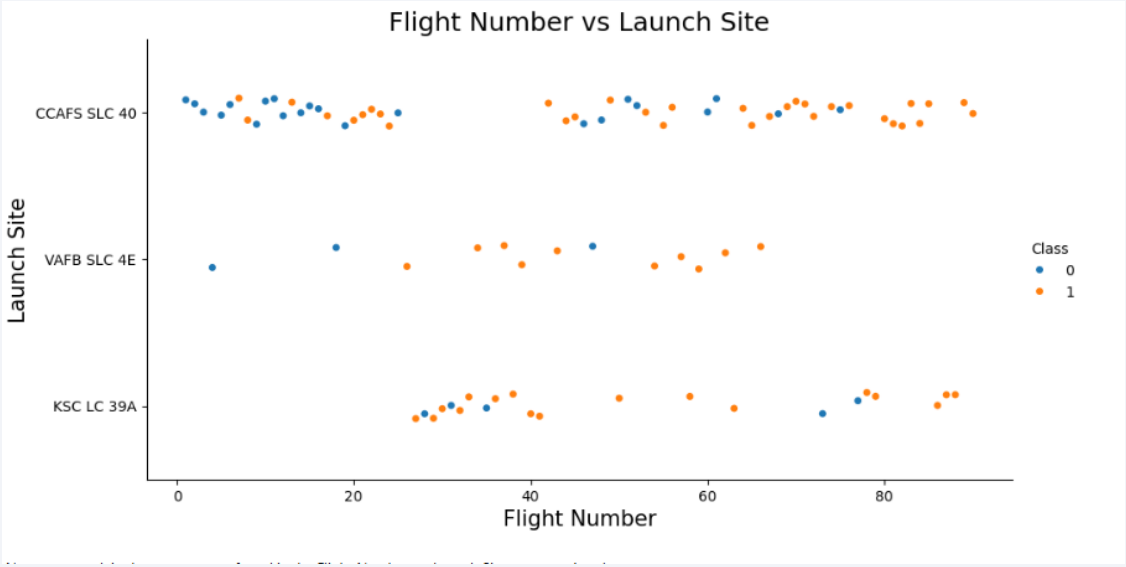
Results

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


Section 2
Insights drawn from EDA

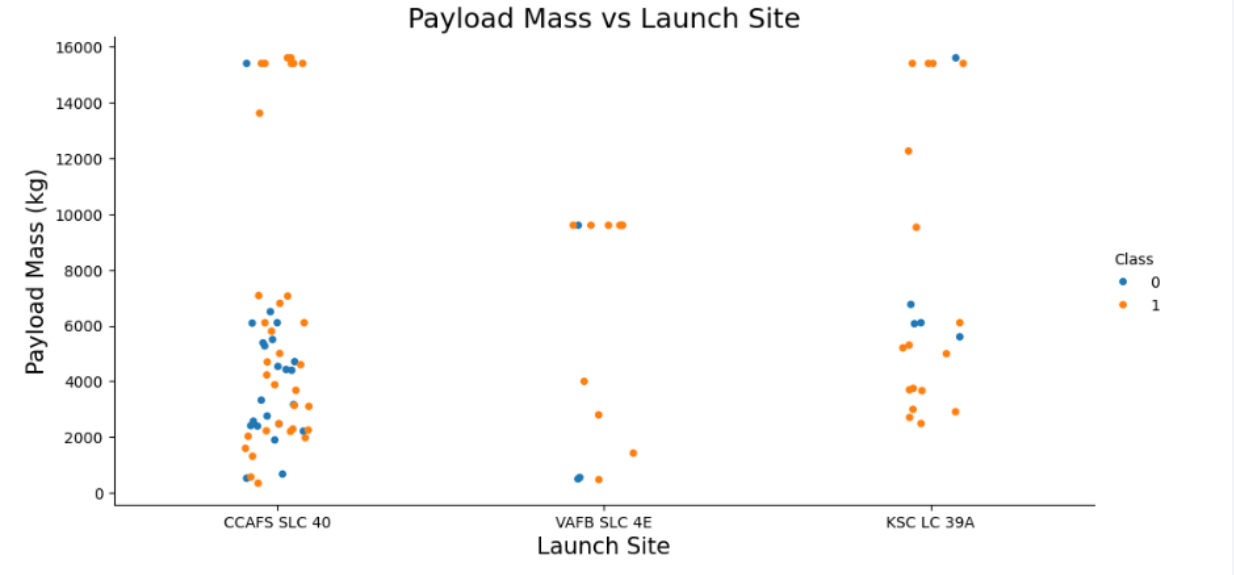
Flight Number vs. Launch Site

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



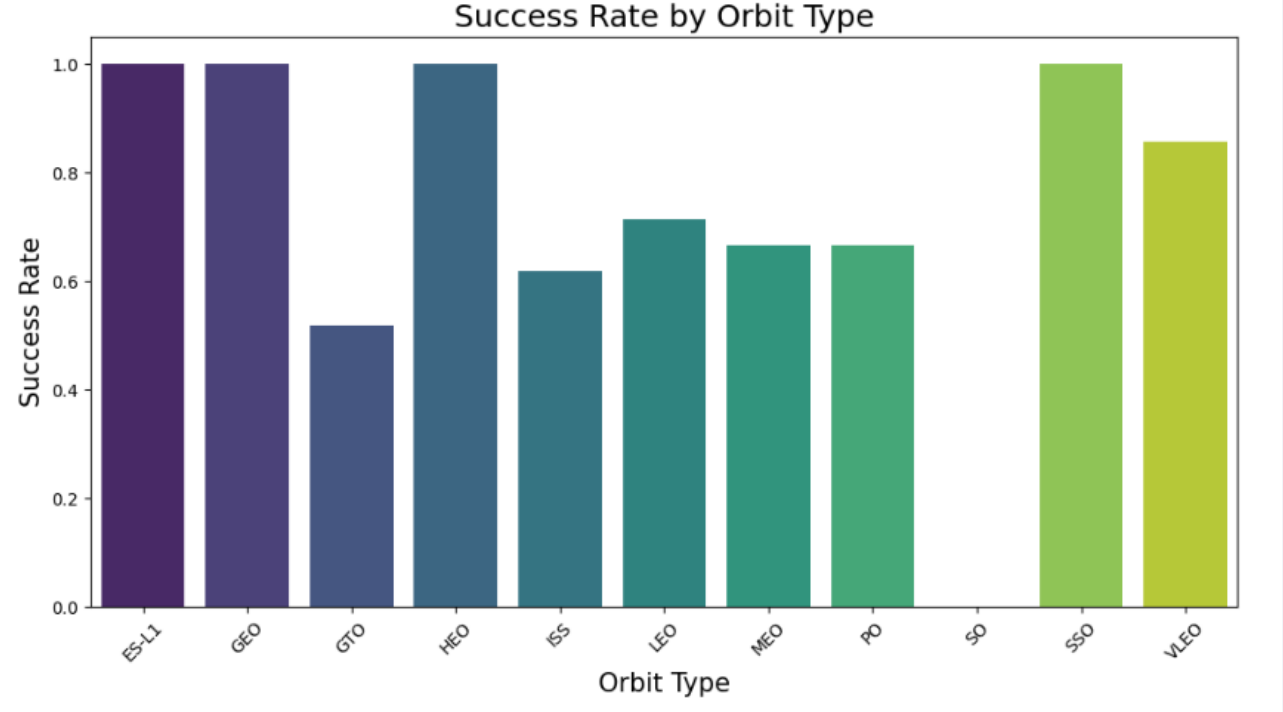
Payload vs. Launch Site

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



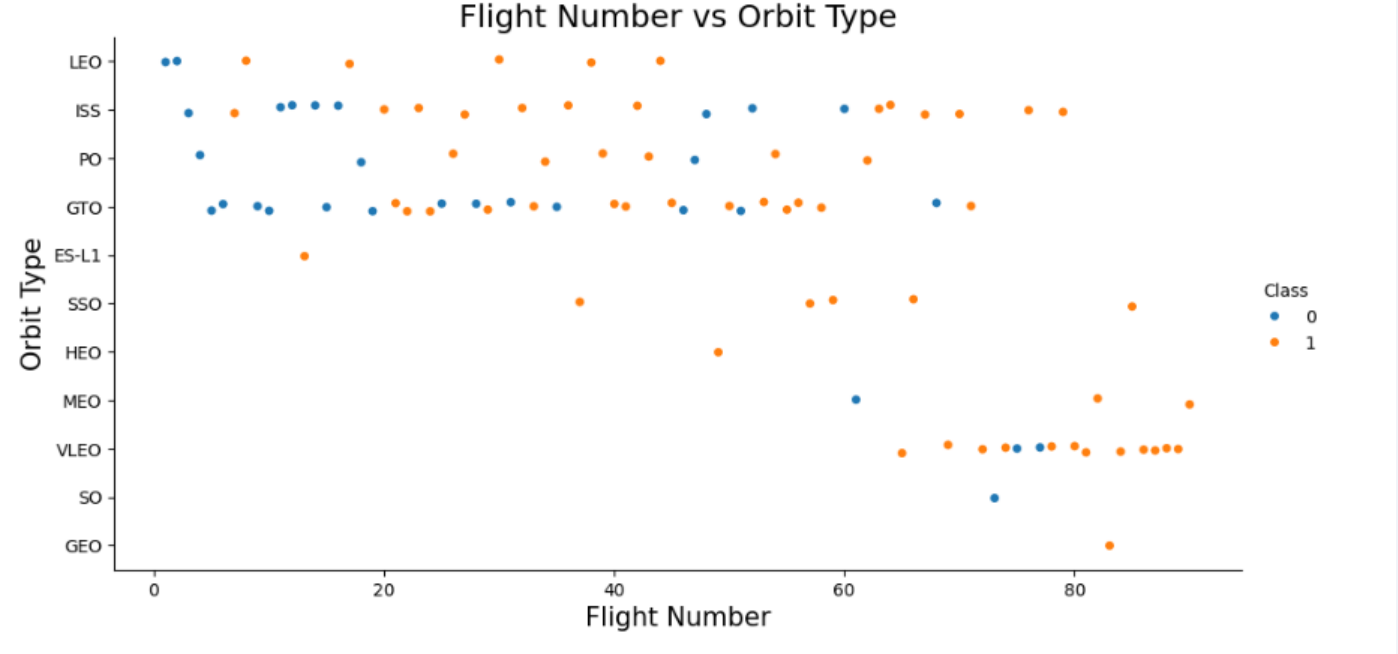
Success Rate vs. Orbit Type

- Show a bar chart for the success rate of each orbit type
- Show the screenshot of the scatter plot with explanations



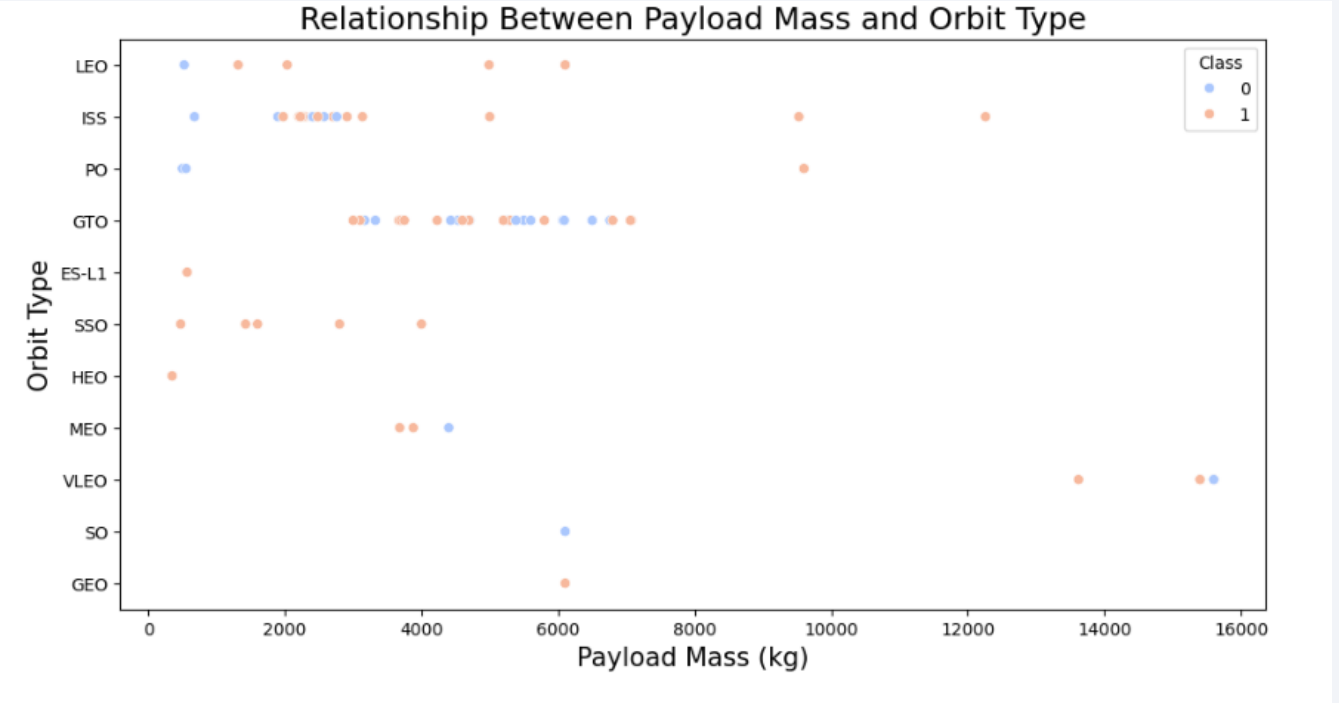
Flight Number vs. Orbit Type

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



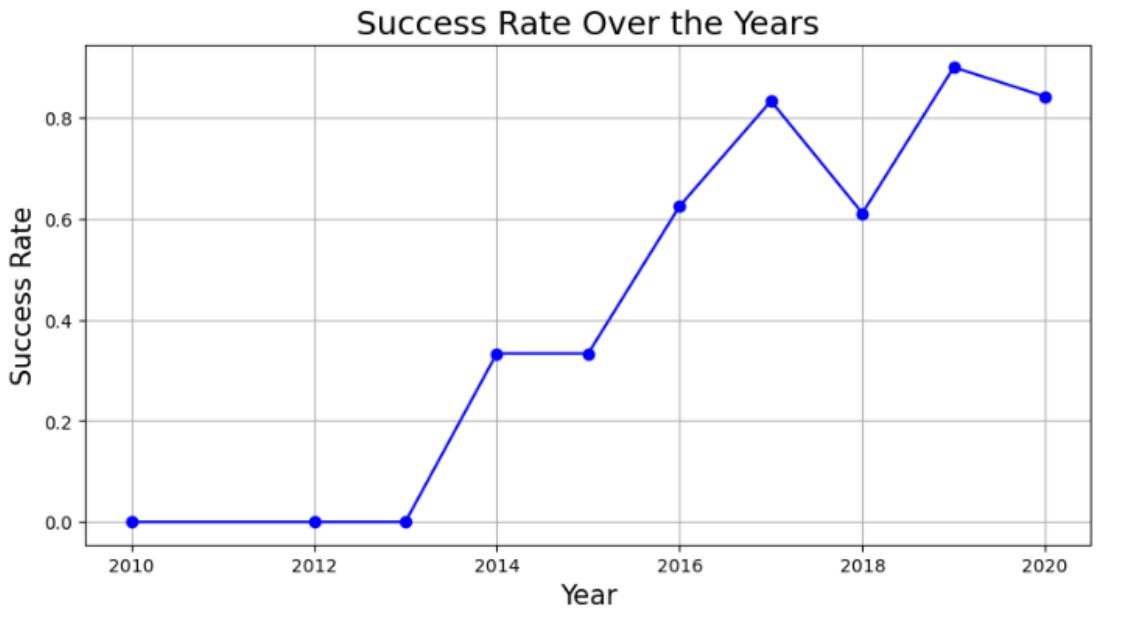
Payload vs. Orbit Type

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



Launch Success Yearly Trend

- Show a line chart of yearly average success rate
- 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

Task 1

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

```
[10]: %sql SELECT DISTINCT "LaunchSite" FROM SPACEXTABLE;
```

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

[10]:

"LaunchSite"
LaunchSite

Launch Site Names Begin with 'CCA'

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

Task 2

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

```
[11]: %sql SELECT * FROM SPACEXTABLE WHERE "LaunchSite" LIKE 'CCA%' LIMIT 5;
```

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

[11]:

Date	Time (UTC)	Booster_Version	Launch_Site	Payload	PAYLOAD_MASS__KG_	Orbit	Customer	Mission_Outcome	Landing_Outcome

Total Payload Mass

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

Task 3

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

```
[12]: %sql SELECT SUM("PayloadMass") AS Total_Payload_Mass FROM SPACEXTABLE WHERE "Mission" LIKE '%NASA (CRS)%';
 * sqlite:///my_data1.db
Done.
```

[12]:

Total_Payload_Mass
None

Average Payload Mass by F9 v1.1

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

Task 4

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

```
[13]: %sql SELECT AVG("PayloadMass") AS Average_Payload_Mass FROM SPACEXTABLE WHERE "BoosterVersion" = 'F9 v1.1';
 * sqlite:///my_data1.db
Done.
```

[13]:

Average_Payload_Mass
None

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

Task 5 ¶

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

Hint:Use min function

[14]:
```
%sql SELECT MIN("Date") AS First_Successful_Landing FROM SPACEXTABLE WHERE "Landing_Outcome" = 'True Ground Pad';
```

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

[14]:

First_Successful_Landing
None

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

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

```
[15]: %sql SELECT DISTINCT "BoosterVersion" FROM SPACEXTABLE WHERE "Landing_Outcome" = 'True ASDS' AND "PayloadMass" BETWEEN 4000 AND 6000;
 * sqlite:///my_data1.db
Done.
[15]: "BoosterVersion"
```

- Present your query result with a short explanation here

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

Task 7

List the total number of successful and failure mission outcomes

```
[16]: %sql SELECT "Mission_Outcome", COUNT(*) AS Total_Missions FROM SPACEXTABLE GROUP BY "Mission_Outcome";
```

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

[16]:

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

Task 8

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

```
[17]: %sql SELECT DISTINCT "BoosterVersion" FROM SPACEXTABLE WHERE "PayloadMass" = (SELECT MAX("PayloadMass") FROM SPACEXTABLE);
 * sqlite:///my_data1.db
Done.
```

[17]:

"BoosterVersion"
BoosterVersion

2015 Launch Records

- List 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.

```
[20]: %sql SELECT SUBSTR("Date", 6, 2) AS Month, "Landing_Outcome", "BoosterVersion", "LaunchSite" FROM SPACEXTABLE WHERE "Landing_Outcome" LIKE 'False ASDS'  AND SUBSTR("Date", 0, 5) = '2015';
 * sqlite:///my_data1.db
Done.
[20]: Month  Landing_Outcome  "BoosterVersion"  "LaunchSite"
```

- Present your query result with a short explanation here

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.

```
[22]: %sql SELECT "Landing_Outcome", COUNT(*) AS Outcome_Count  FROM SPACEXTABLE  WHERE "Date" BETWEEN '2010-06-04' AND '2017-03-20'  GROUP BY "Landing_Outcome"  ORDER BY Outcome_Count DESC;
 * sqlite:///my_data1.db
Done.
```

[22]:

Landing_Outcome	Outcome_Count
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

```
[23]: %sql SELECT DISTINCT "LaunchSite" FROM SPACEXTBL;
 * sqlite:///my_data1.db
Done.
```

[23]:

"LaunchSite"
LaunchSite

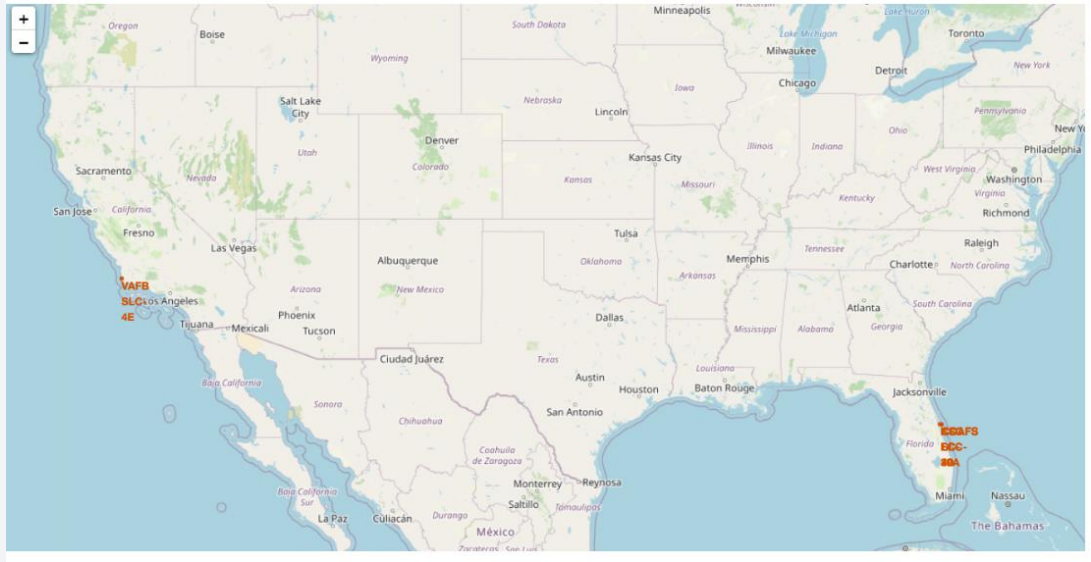
- Present your query result with a short explanation here

Section 3

Launch Sites Proximities Analysis

Launch sites location markers on a global map

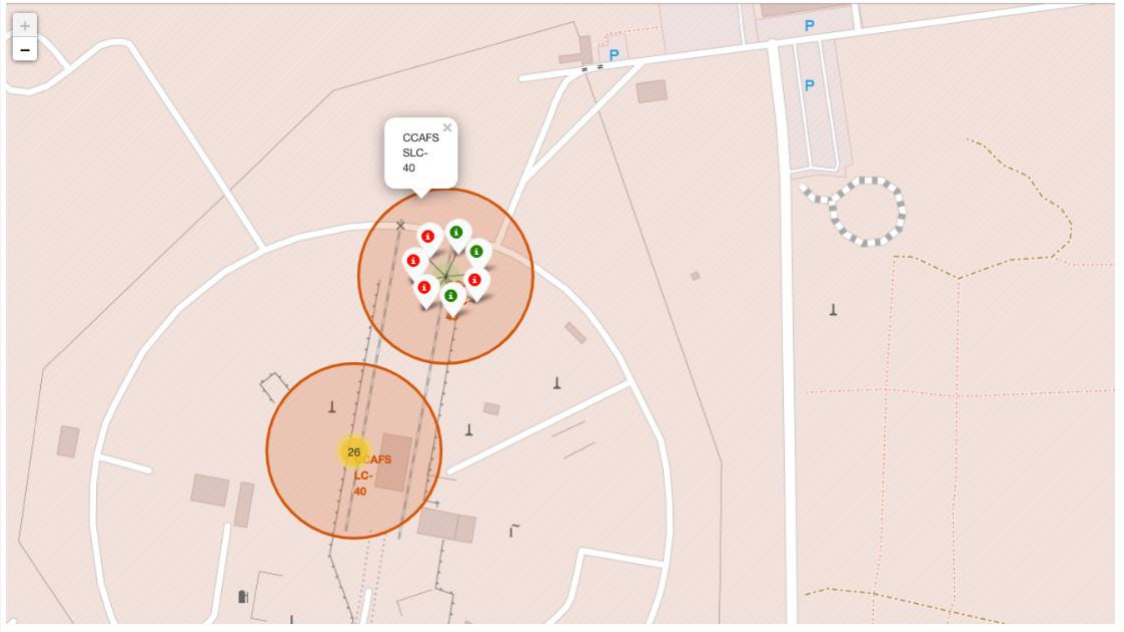
- Replace <Folium map screenshot 1> title with an appropriate title
- Explore the generated folium map and make a proper screenshot to include all launch sites' location markers on a global map
- Explain the important elements and findings on the screenshot



Lugares de lanzamiento de SpaceX

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

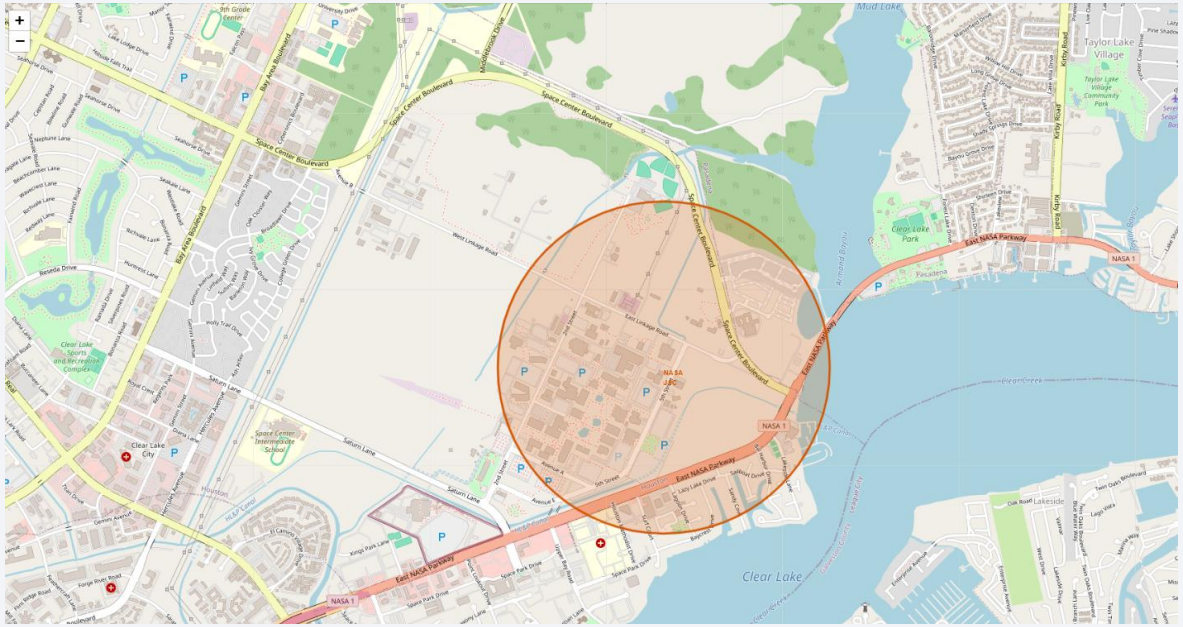


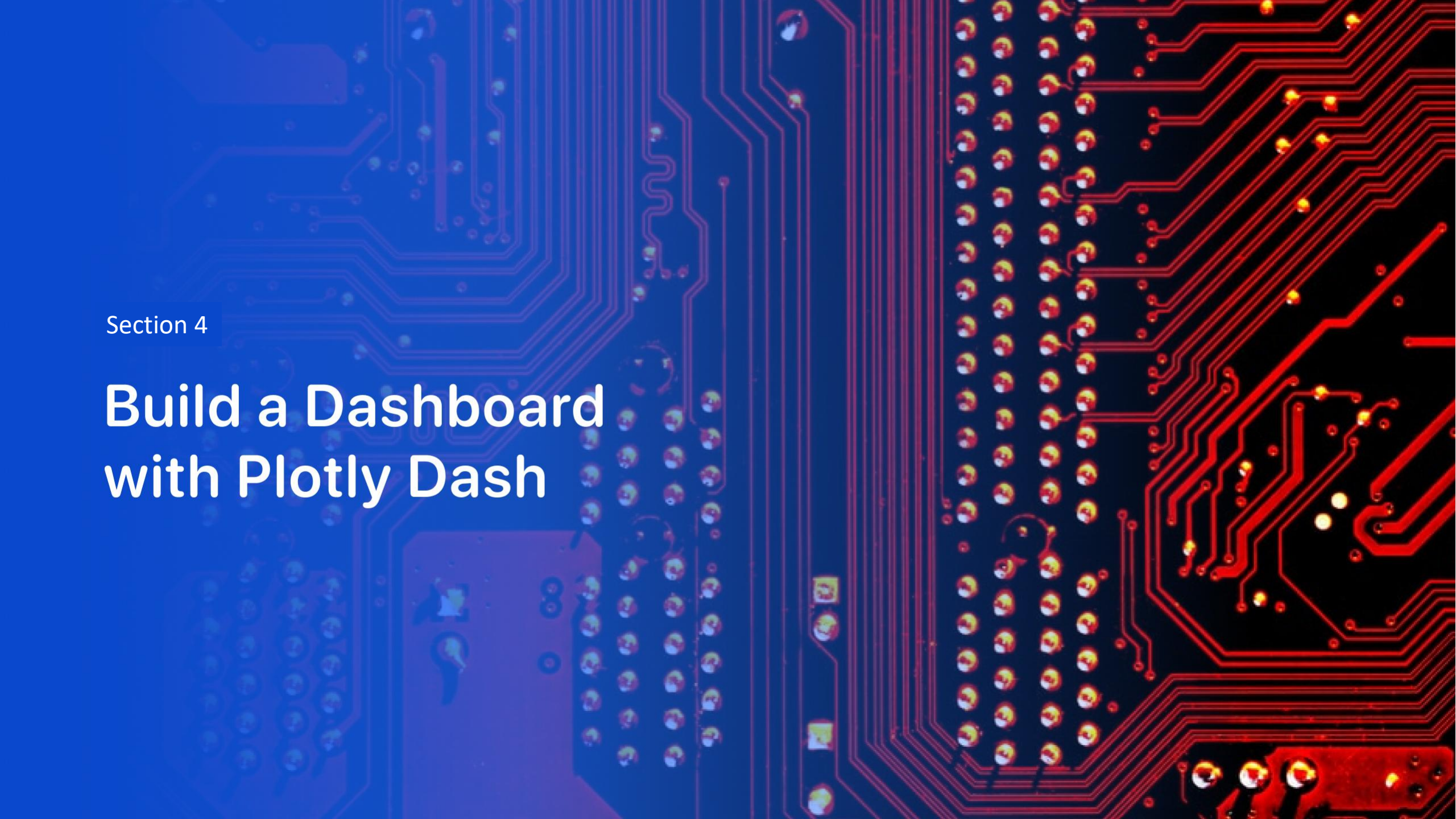
Launch site to its proximities

- 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 count for all sites

- 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

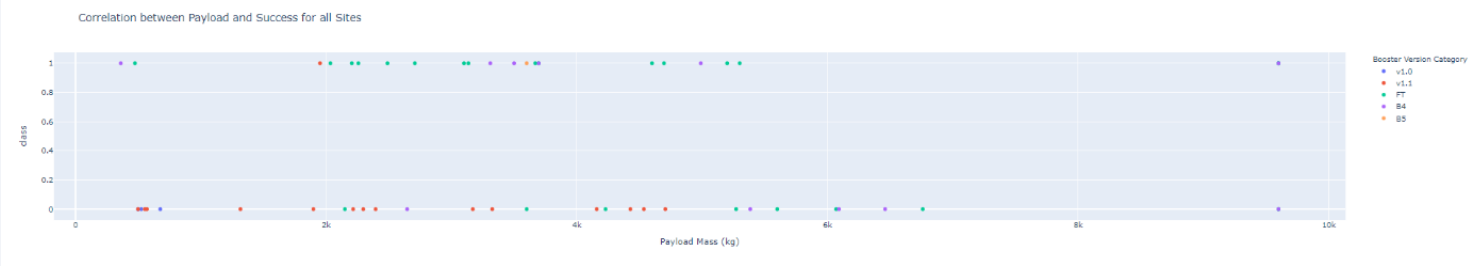
Launch site with highest launch success ratio

- 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 vs. Launch Outcome scatter plot for all sites

- 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



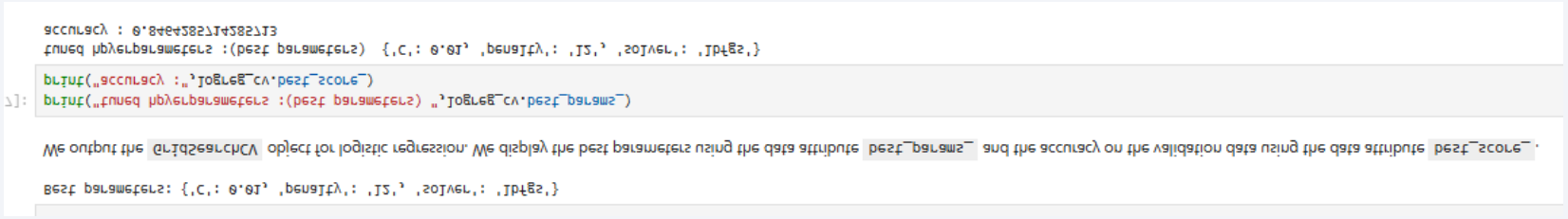
- Explain the important elements and findings on the screenshot, such as which payload range or booster version have the largest success rate, etc.

Section 5

Predictive Analysis (Classification)

Classification Accuracy

- Visualize the built model accuracy for all built classification models, in a bar chart

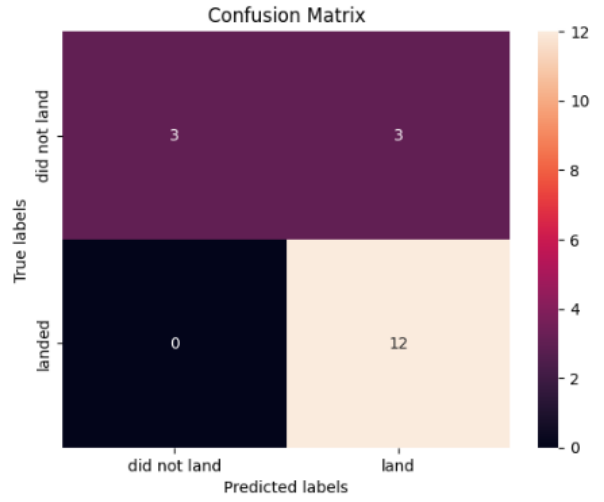


- Find which model has the highest classification accuracy

Confusion Matrix

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

```
[19]: yhat=logreg_cv.predict(X_test)
      plot_confusion_matrix(Y_test,yhat)
```



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

Overview:

True Postive - 12 (True label is landed, Predicted label is also landed)

False Postive - 3 (True label is not landed, Predicted label is landed)

Conclusions

- El Skill Network Labs es una herramienta muy interesante para aprender, sin embargo, en algunos momentos presenta caidas que impiden lograr los resultados esperados.
- Es trabajo ha permitido aprender sobre Data collection, scraping, wrangling, EDA, Folium, dashboard y predictive analysis.
- Ha sido possible obtener, distintos tipos de gráficos, mapas y ubicaciones, anális predictivos y búsqueda de información

Appendix

- Proyecto trabajado a través de laboratorio Skills Network, con apoyo de Copilot IA

Thank you!