IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

Eder Zamudio
15 de mayo 2025

Estructura

- Resumen ejecutivo
- Introducción
- Metodología
- Resutados
- Conclusión
- Apéndice

Resumen ejecutivo

- Resumen de metodologías
  - Recopilación de datos mediante API
  - Recopilación de datos con web scraping
  - Procesamiento de datos
  - Análisis exploratorio de datos con SQL
  - Análisis exploratorio de datos con visualización de datos
  - Análisis visual interactivo con Folium
  - Predicción de aprendizaje automático
- Resumen de todos los resultados
  - Resultado del análisis exploratorio de datos
  - Análisis interactivo en capturas de pantalla
  - Resultado del análisis predictivo

Introducción

- Contexto y antecedentes del proyecto

SpaceX anuncia los lanzamientos del cohete Falcon 9 en su sitio web con un costo de 62 millones de dólares por lanzamiento. En comparación, otros proveedores cobran más de 165 millones de dólares por lanzamiento. Gran parte de este ahorro se debe a que SpaceX puede reutilizar la primera etapa del cohete. Por lo tanto, si podemos predecir si la primera etapa aterrizará exitosamente, podremos estimar el costo real de un lanzamiento.

Esta información es valiosa para empresas alternativas que deseen competir con SpaceX en licitaciones para lanzamientos espaciales. El objetivo de este proyecto es desarrollar una pipeline de machine learning que prediga si la primera etapa del cohete Falcon 9 aterrizará exitosamente.

- Problemas que queremos resolver

¿Qué factores influyen en el éxito del aterrizaje de la primera etapa del cohete?

¿Cómo interactúan las distintas variables para determinar la probabilidad de un aterrizaje exitoso?

¿Cuáles son las condiciones operativas necesarias para asegurar un programa de aterrizaje exitoso?

Section 1

Methodology

Metodología

Resumen ejecutivo

- Metodología de recolección de datos:
  - Se obtuvieron datos utilizando la API de SpaceX y mediante web scraping de Wikipedia.
- Limpieza y preparación de datos:
  - Se aplicó one-hot encoding a las variables categóricas para convertirlas en formato numérico.
- Se reallizó un análisis exploratorio de datos (EDA) mediante visualización y SQL.
- Se realizaron análisis visuales interactivos con Folium y Plotly Dash.
- Se realizó un análisis predictivo utilizando modelos de clasificación.
  - Cómo construir, ajustar y evaluar modelos de clasificación.

Recolección de Datos

- La recolección de datos se realizó mediante una solicitud GET a la API de SpaceX.
- Luego, decodificamos el contenido de la respuesta como JSON usando la función .json() y lo convertimos en un DataFrame de pandas utilizando json_normalize().
- Posteriormente, limpiamos los datos, verificamos si había valores faltantes y completamos los valores ausentes cuando fue necesario.
- Además, realizamos web scraping desde Wikipedia para obtener los registros de lanzamientos del Falcon 9, utilizando BeautifulSoup.
- El objetivo fue extraer los registros de lanzamiento desde una tabla HTML, analizar su contenido y convertirlo en un DataFrame de pandas para su análisis posterior.

Recolección de Datos – SpaceX API

- Usamos la solicitud GET a la API de SpaceX para recopilar datos, limpiarlos y realizar algunas operaciones básicas de manipulación y formato.
- El enlace del notebok: https://github.com/EderZC12/TestRepo/tree/c08c47a0b1e89e840156cc25ba8cc1c15d9da4fb

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

[9]: response = requests.get(spacex_url)
```

```
[14]: # Use json_normalize meethod to convert the json result into a dataframe
      from pandas import json_normalize
      # Decodificamos la respuesta como JSON
      data_json = response.json()
      data = json_normalize(data_json)
```

```
[16]: # 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']]

      # Eliminar filas con múltiples cores o payloads
      data = data[data['cores'].map(len) == 1]
      data = data[data['payloads'].map(len) == 1]

      # Extraer elementos únicos de las listas
      data['cores'] = data['cores'].map(lambda x: x[0])
      data['payloads'] = data['payloads'].map(lambda x: x[0])

      # Convertir fecha y filtrar hasta el 13 de noviembre de 2020
      data['date'] = pd.to_datetime(data['date_utc']).dt.date
      data = data[data['date'] <= datetime.date(2020, 11, 13)]
```

Recolección de Datos – Web Scraping

- Aplicamos web scraping para extraer los registros de lanzamientos del Falcon 9 utilizando BeautifulSoup. Además, analizamos la tabla HTML y la convertimos en un DataFrame de pandas.
- El enlace del notebook: https://github.com/EderZC12/TestRepo/tree/c08c47a0b1e89e840156cc25ba8cc1c15d9da4fb

[4]:
```
static_url = "https://en.wikipedia.org/w/index.php?title=List_of_Falcon_9_and_Falcon_Heavy_launches&oldid=1027686922"
```

[5]:
```
# use requests.get() method with the provided static_url
# assign the response to a object
# Perform an HTTP GET request
response = requests.get(static_url)

# Optional: Check the response status code to verify the request succeeded
print("Status Code:", response.status_code)
```

```
Status Code: 200
```

[8]:
```
# Use soup.title attribute
soup = BeautifulSoup(response.text, 'html.parser')
# Print the page title
print("Page Title:", soup.title.string)
```

```
Page Title: List of Falcon 9 and Falcon Heavy launches - Wikipedia
```

[9]:
```
# Use the find_all function in the BeautifulSoup object, with element type `table`
# Assign the result to a list called `html_tables`
# Find all table elements on the page
html_tables = soup.find_all("table")

# Print number of tables found
print("Total tables found:", len(html_tables))
```

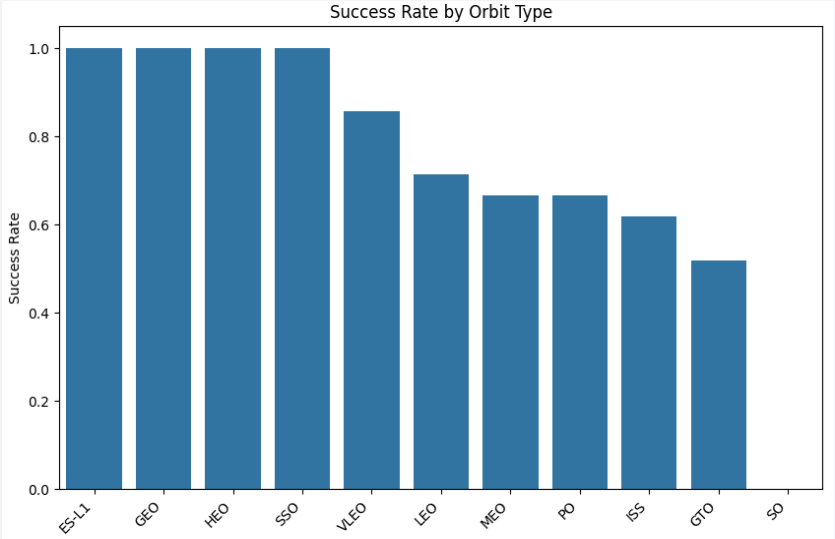
```
Total tables found: 25
```

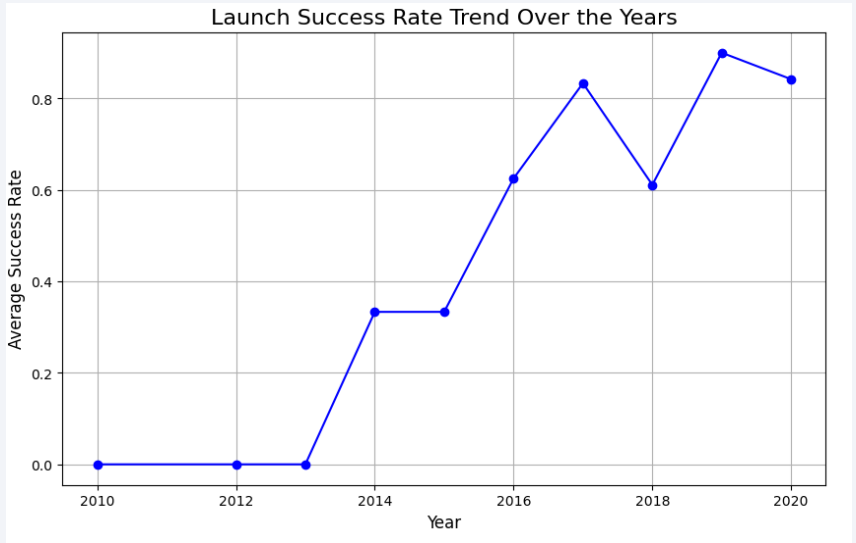
Limpieza de datos

- Se realizó un análisis exploratorio de los datos (EDA) para determinar las etiquetas de entrenamiento.
- Se calculó el número de lanzamientos por sitio y la frecuencia de cada tipo de órbita.
- Se creó una etiqueta de resultado de aterrizaje a partir de la columna de resultados y se exportaron los datos procesados a un archivo CSV.
- El enlace del notebook: https://github.com/EderZC12/TestRepo/tree/c08c47a0b1e89e840156cc25ba8cc1c15d9da4fb

Análisis Exploratorio de Datos con Visualización (EDA)

- Exploramos los datos visualizando la relación entre el número de vuelo y el lugar de lanzamiento, la carga útil y el lugar de lanzamiento, la tasa de éxito de cada tipo de órbita, el número de vuelo y el tipo de órbita, y la tendencia anual de éxito de los lanzamientos.
- El enlace del notebook: https://github.com/EderZC12/TestRepo/tree/c08c47a0b1e89e840156cc25ba8cc1c15d9da4fb





Análisis Exploratorio de Datos con SQL

- Cargamos el conjunto de datos de SpaceX en una base de datos PostgreSQL directamente desde el cuaderno Jupyter.
- Aplicamos EDA utilizando consultas SQL para obtener información relevante. Algunas de las consultas realizadas fueron:
- Obtener los nombres de los sitios de lanzamiento únicos en las misiones espaciales.
- Calcular la masa total de carga útil transportada por boosters lanzados por NASA (CRS).
- Determinar la masa promedio de carga útil transportada por la versión de booster F9 v1.1.
- Contar el número total de resultados exitosos y fallidos de las misiones.
- Identificar los aterrizajes fallidos en barcos drones, junto con su versión de booster y nombre del sitio de lanzamiento.
- El enlace del notebook: https://github.com/EderZC12/TestRepo/tree/c08c47a0b1e89e840156cc25ba8cc1c15d9da4fb

Crear un Mapa Interactivo con Folium

- Marcamos todos los sitios de lanzamiento y añadimos objetos al mapa como marcadores, círculos y líneas para señalar el éxito o fallo de los lanzamientos en cada sitio usando Folium.
- Asignamos los resultados de lanzamiento como clases: 0 para fallos y 1 para éxitos.
- Mediante clusters de marcadores codificados por colores, identificamos cuáles sitios de lanzamiento tienen una tasa de éxito relativamente alta.
- Calculamos las distancias desde cada sitio de lanzamiento hacia sus proximidades. Respondimos preguntas como:
- ¿Están los sitios de lanzamiento cerca de ferrocarriles, autopistas o costas?
- ¿Mantienen los sitios de lanzamiento una distancia prudente de las ciudades?
- El enlace del notebook: https://github.com/EderZC12/TestRepo/tree/c08c47a0b1e89e840156cc25ba8cc1c15d9da4fb

Construcción de un Dashboard con Plotly Dash

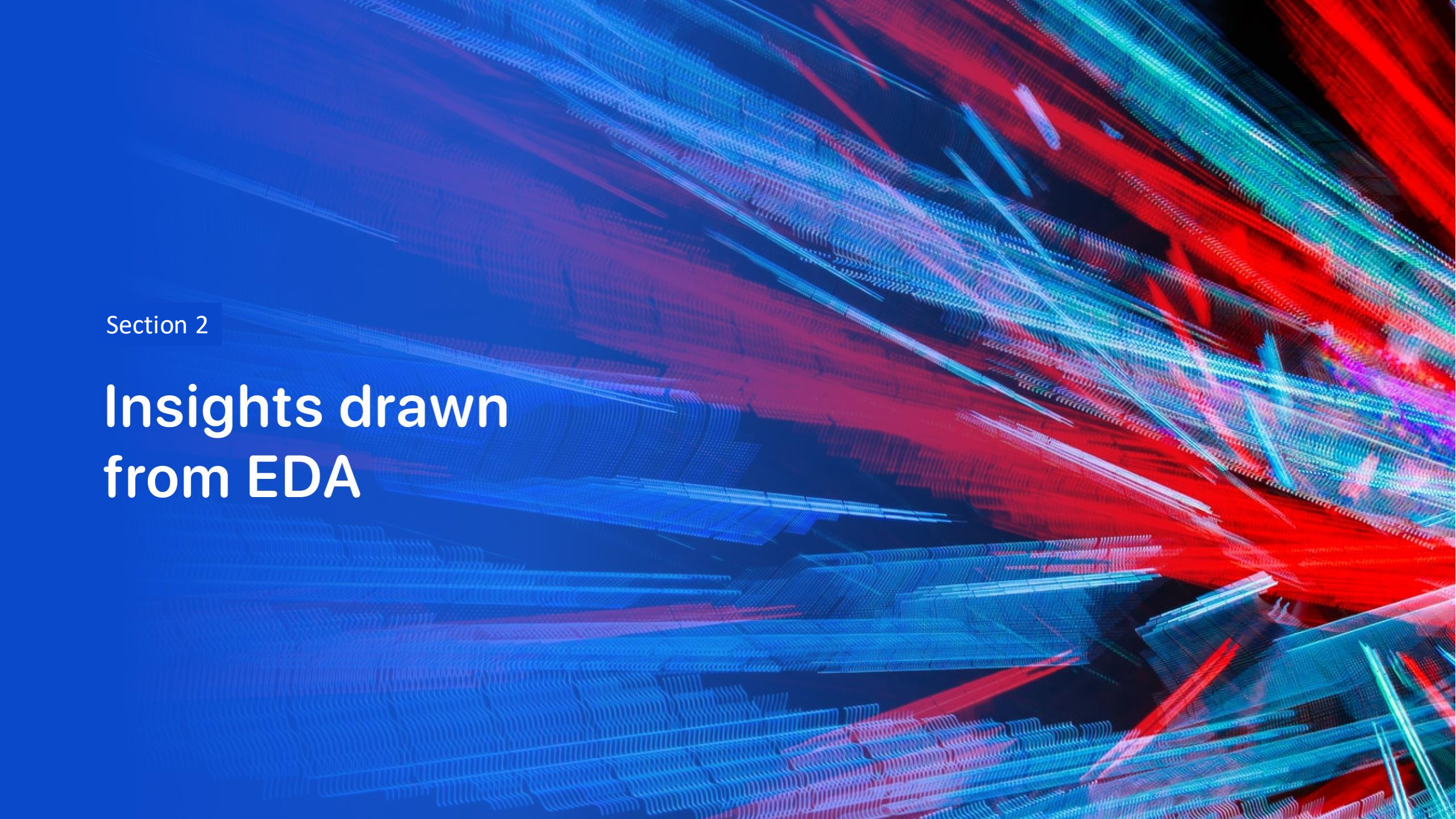
- Creamos un panel interactivo utilizando Plotly Dash.
- Trazamos gráficos de pastel que muestran el total de lanzamientos por sitio específico.
- Trazamos un gráfico de dispersión que muestra la relación entre el resultado del lanzamiento (éxito o fallo) y la masa de carga útil (Kg) para las distintas versiones del propulsor.
- El enlace del notebook: https://github.com/EderZC12/TestRepo/tree/c08c47a0b1e89e840156cc25ba8cc1c15d9da4fb

Análisis Predictivo (Clasificación)

- Cargamos los datos utilizando NumPy y Pandas, transformamos los datos y los dividimos en conjuntos de entrenamiento y prueba.
- Construimos diferentes modelos de aprendizaje automático y ajustamos distintos hiperparámetros utilizando GridSearchCV.
- Utilizamos la precisión (accuracy) como métrica principal para evaluar el modelo, y mejoramos su rendimiento aplicando ingeniería de características y ajuste de algoritmos.
- Identificamos el modelo de clasificación con mejor rendimiento.
- El enlace del notebook: https://github.com/EderZC12/TestRepo/tree/c08c47a0b1e89e840156cc25ba8cc1c15d9da4fb

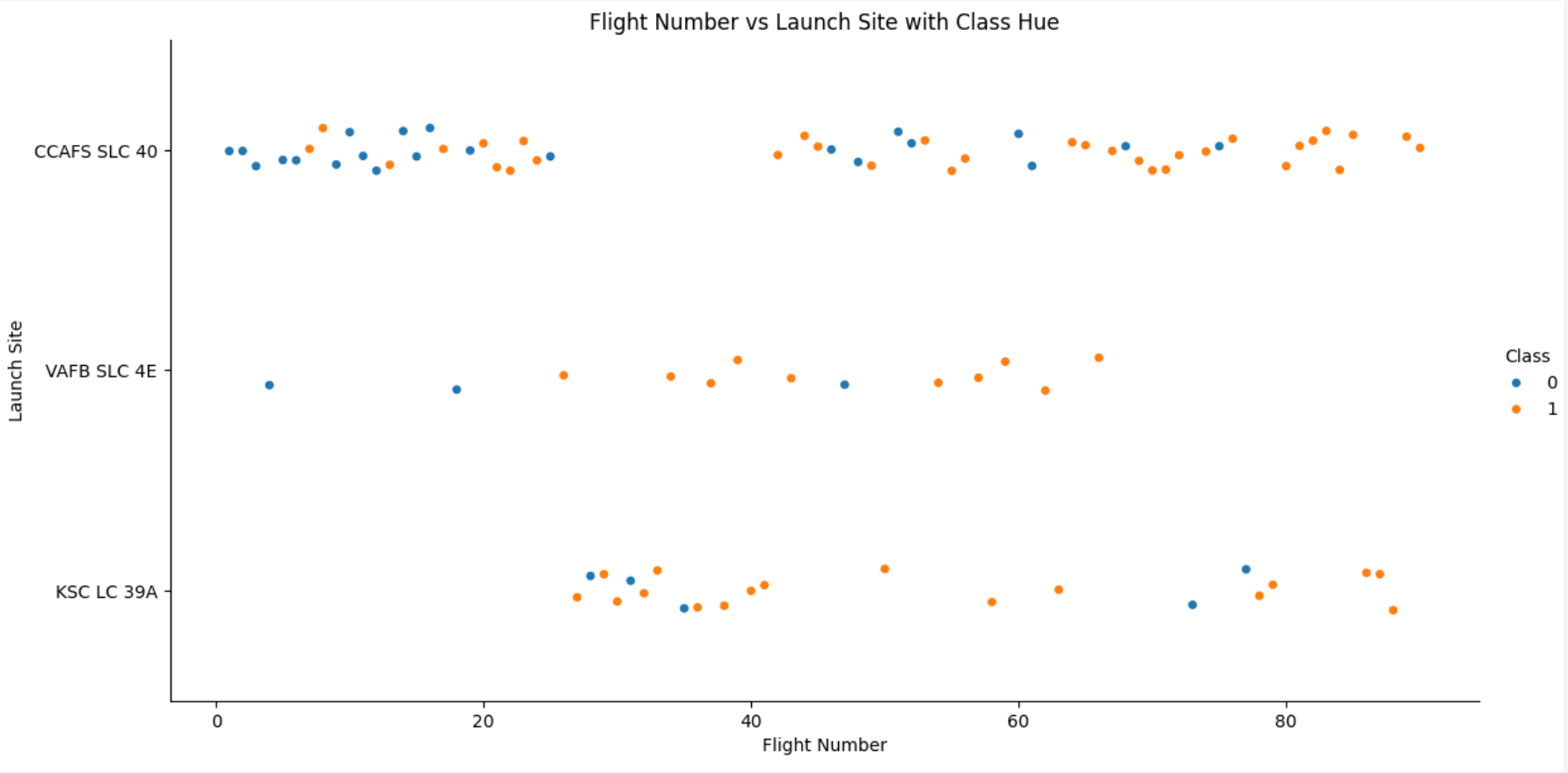
Resultados

- Resultados del análisis exploratorio de datos
- Demostración de análisis interactivo en capturas de pantalla
- Resultados del análisis predictivo


Section 2
Insights drawn from EDA

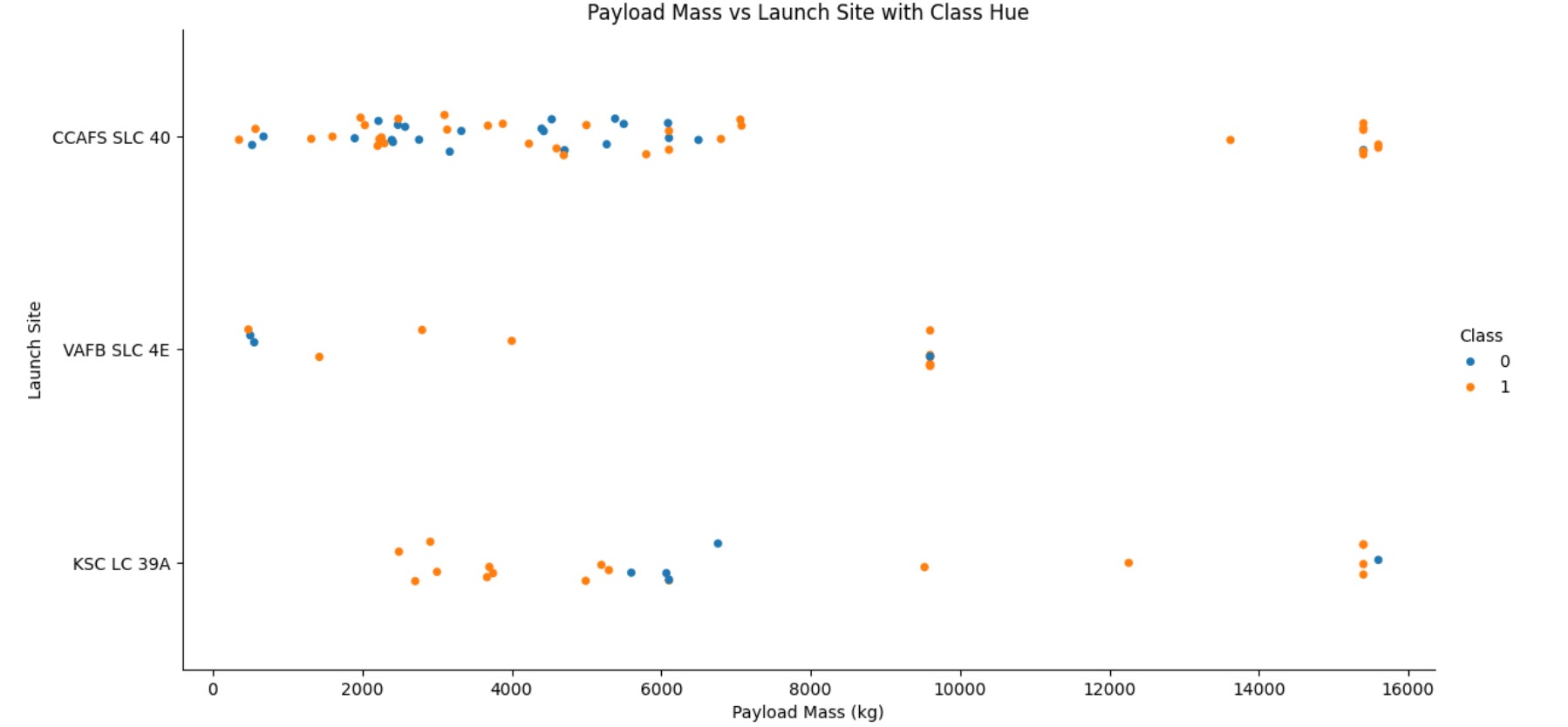
Flight Number vs. Launch Site

- La mayoría de los lanzamientos exitosos ocurrieron en CCAFS SLC 40 y VAFB SLC 4E, mientras que KSC LC 39A tuvo una distribución más equilibrada. Además, el número de vuelos varía entre sitios, lo que sugiere una diferencia en la frecuencia de uso de los mismos.



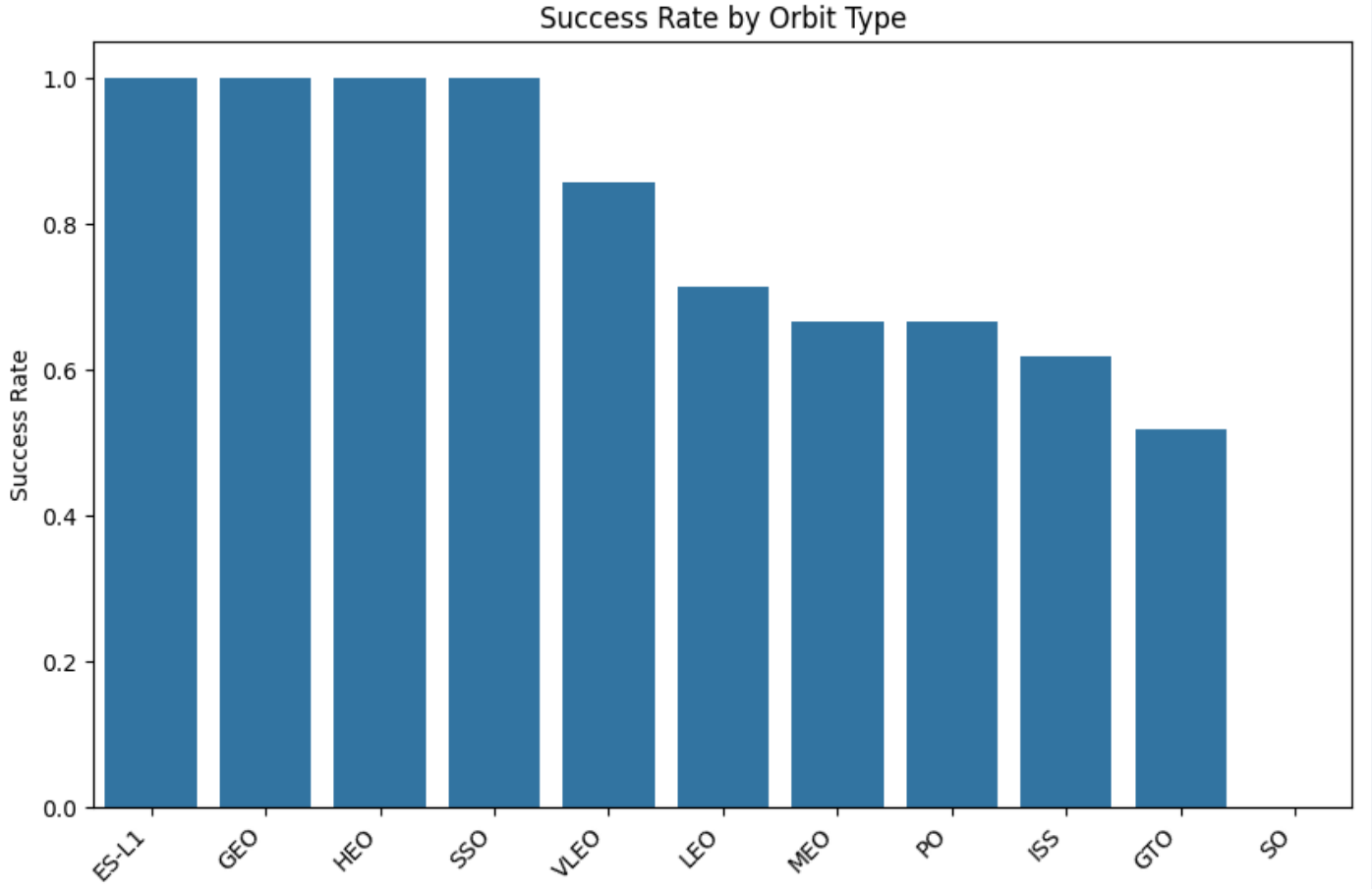
Payload vs. Launch Site

- CCAFS SLC 40 y KSC LC 39A son los sitios de lanzamiento más activo y muestra una relación positiva entre mayor masa de carga útil y lanzamientos exitosos. En cambio, el otro podria estar destinados a misiones diferentes.



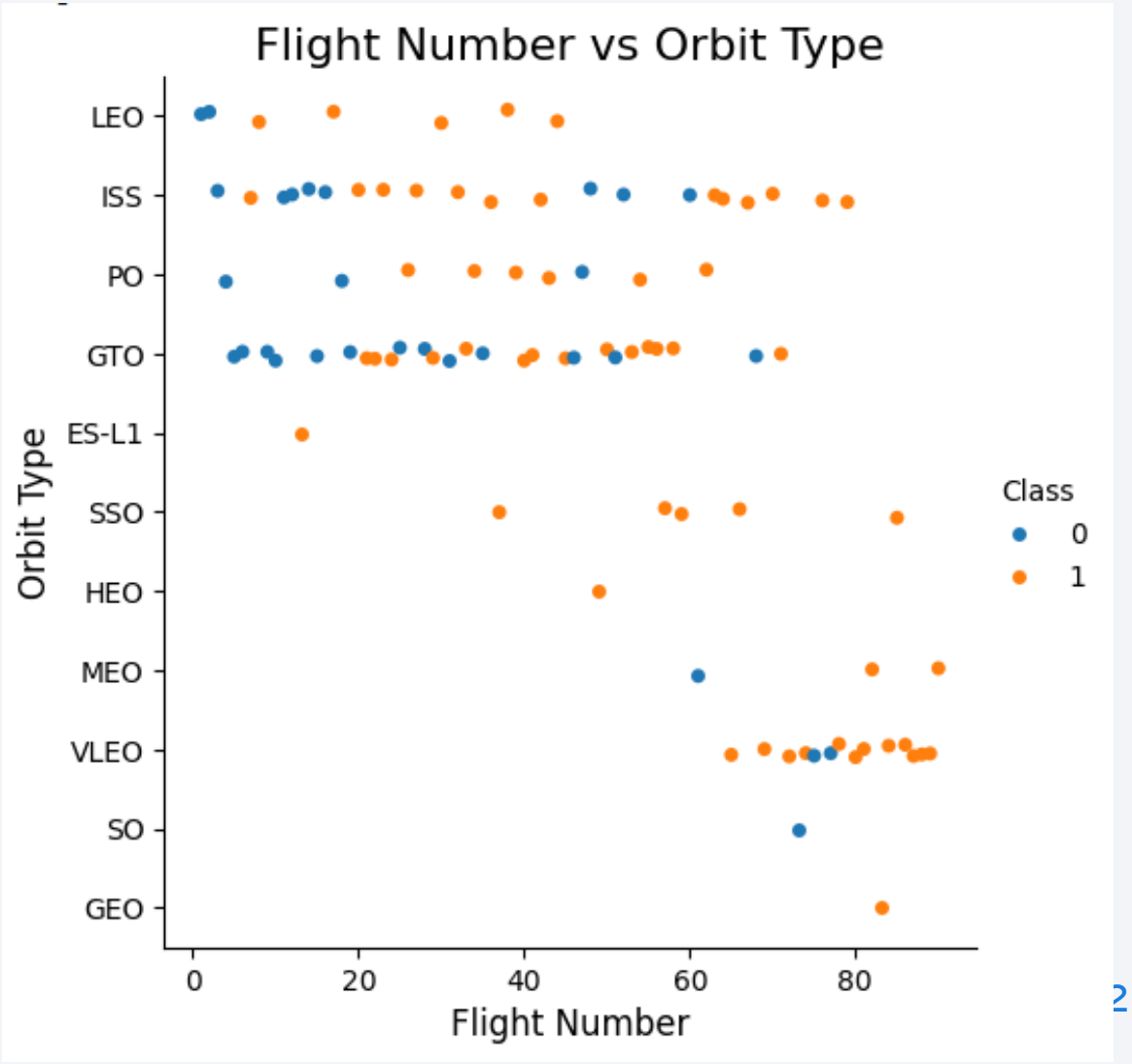
Success Rate vs. Orbit Type

- Del gráfico, podemos ver que ES-L1, GEO, HEO, SSO son los que tuvieron la mayor tasa de éxito con 100%. En cambio, el SO es el que tuvo menor tasa de éxito con 0%.



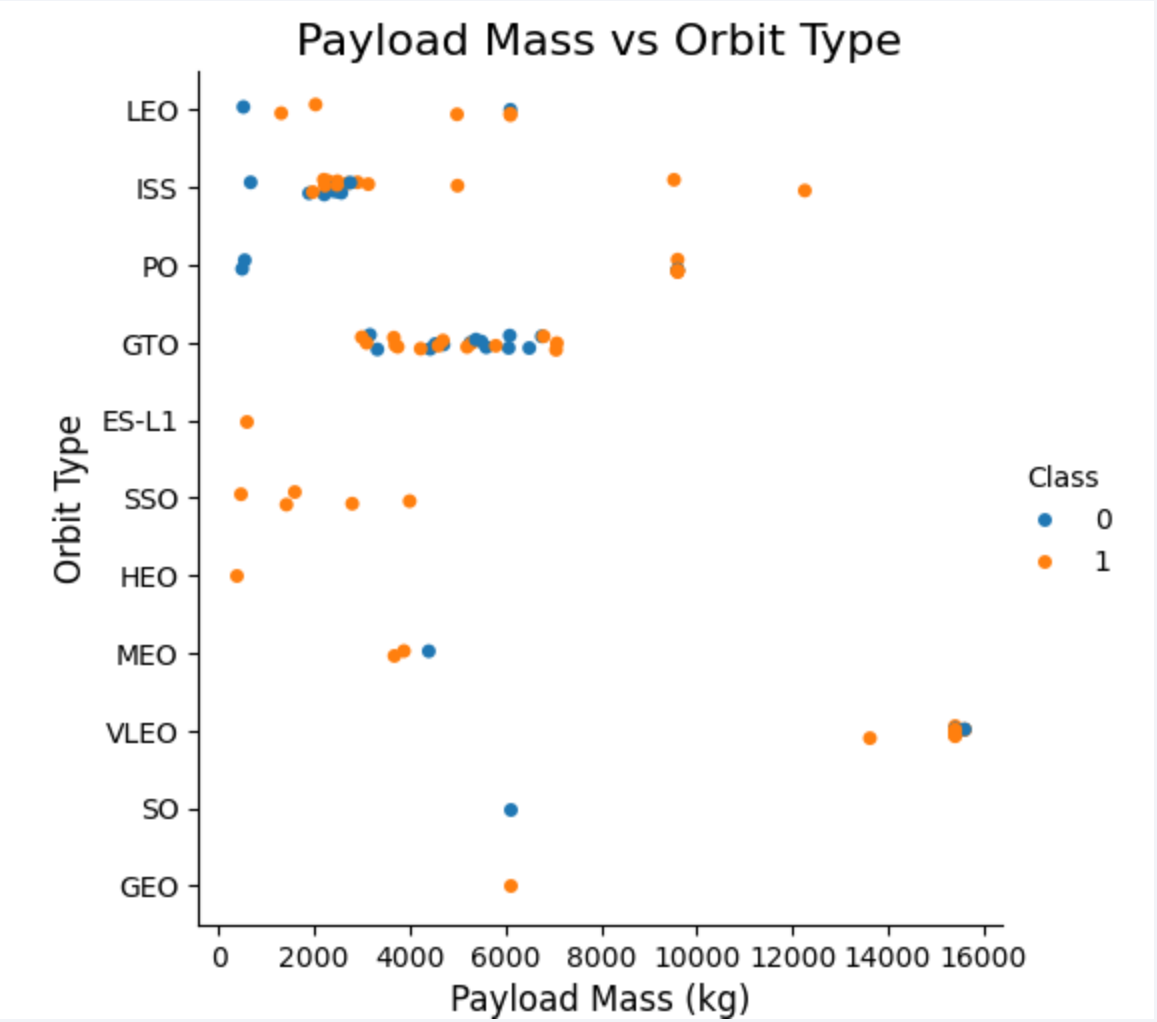
Flight Number vs. Orbit Type

- El gráfico a continuación muestra el Número de Vuelo frente al tipo de Órbita. Observamos que en las órbitas LEO, ISS, PO, MEO y VLEO el éxito aumenta con el número de vuelos. Esto se puede observar de forma más directa en la órbita LEO Además,que en la órbita GTO no existe relación entre el número de vuelos y la órbita.



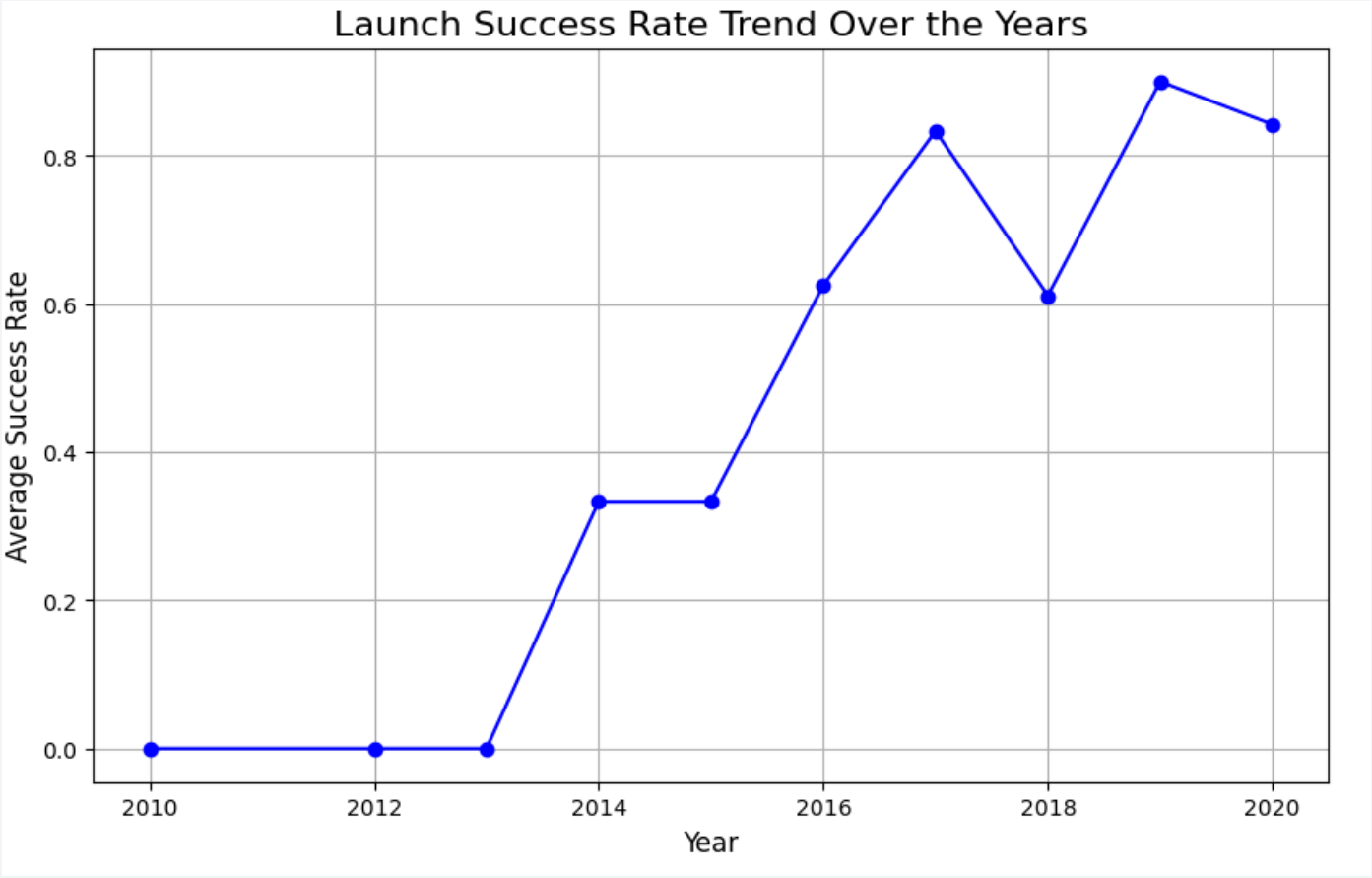
Payload vs. Orbit Type

- Podemos observar que con cargas útiles pesadas, los aterrizajes exitosos son más frecuentes en las órbitas PO, LEO e ISS y VLEO. Además observamos que las cargas mas ligeras se realizan en las órbitas ES-L1 y HEO.



Launch Success Yearly Trend

- Del gráfico se observa que la tasa de éxito de los lanzamientos ha mostrado una tendencia general de crecimiento desde 2013 hasta 2020, con excepciones en los años 2018 y 2020, donde se registra una leve disminución en el porcentaje de éxitos.



All Launch Site Names

- Usamos la palabra clave DISTINCT para mostrar solo los sitios de lanzamiento únicos en los datos de SpaceX.

```
[23]: query1 = 'SELECT DISTINCT "Launch_Site" FROM SPACEXTBL;'
      df1 = pd.read_sql_query(query1, con)
      print(df1)
```

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

Launch Site Names Begin with 'CCA'

- Usamos la consulta anterior para mostrar 5 registros donde los sitios de lanzamiento comienzan con CCA.

[23]:
```
query2 = '''
SELECT * FROM SPACEXTBL
WHERE "Launch_Site" LIKE 'CCA%'
LIMIT 5;
'''
df2 = pd.read_sql_query(query2, con)
df2
```

[23]:

	Date	Time (UTC)	Booster_Version	Launch_Site	Payload	PAYLOAD_MASS__KG_	Orbit	Customer	Mission_Outcome	Landing_Outcome
0	2010-06-04	18:45:00	F9 v1.0 B0003	CCAFS LC-40	Dragon Spacecraft Qualification Unit	0	LEO	SpaceX	Success	Failure (parachute)
1	2010-12-08	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	7:44:00	F9 v1.0 B0005	CCAFS LC-40	Dragon demo flight C2	525	LEO (ISS)	NASA (COTS)	Success	No attempt
3	2012-10-08	0:35:00	F9 v1.0 B0006	CCAFS LC-40	SpaceX CRS-1	500	LEO (ISS)	NASA (CRS)	Success	No attempt
4	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

- Calculamos que la masa total de la carga útil transportada por los propulsores de la NASA es 48,213 usando la siguiente consulta.

[28]:
```
query3 = '''
SELECT SUM("Payload_Mass__kg_") AS Total_Payload_Mass
FROM SPACEXTBL
WHERE "Customer" LIKE '%NASA (CRS)%';
'''

df3 = pd.read_sql_query(query3, con)
print(df3)
```

```
   Total_Payload_Mass
0               48213
```

Average Payload Mass by F9 v1.1

- Calculamos que la masa promedio de la carga útil transportada por la versión del propulsor F9 v1.1 es 2928.4.

[29]:
```
query4 = '''
SELECT AVG("Payload_Mass__kg_") AS Average_Payload
FROM SPACEXTBL
WHERE "Booster_Version" = 'F9 v1.1';
'''
df4 = pd.read_sql_query(query4, con)
print(df4)
```

```
   Average_Payload
0           2928.4
```

First Successful Ground Landing Date

- Se observa que la fecha del primer aterrizaje exitoso en la plataforma terrestre fue el 22 de diciembre de 2015.

```
[30]: query5 = '''
      SELECT MIN(Date) AS First_Successful_Landing
      FROM SPACEXTBL
      WHERE "Landing_Outcome" = 'Success (ground pad)';
      '''
      df5 = pd.read_sql_query(query5, con)
      print(df5)
```

```
  First_Successful_Landing
0               2015-12-22
```

Successful Drone Ship Landing with Payload between 4000 and 6000

- Utilizamos la cláusula WHERE para filtrar los propulsores que han aterrizado exitosamente en el barco dron y aplicamos la condición AND para determinar los aterrizajes exitosos con una masa de carga útil mayor a 4000 pero menor a 6000.

[31]:
```
query6 = ''
SELECT DISTINCT "Booster_Version"
FROM SPACEXTBL
WHERE "Landing_Outcome" = 'Success (drone ship)'
  AND "Payload_Mass__kg_" > 4000
  AND "Payload_Mass__kg_" < 6000;
''
df6 = pd.read_sql_query(query6, con)
print(df6)
```

```
  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

- La mayoría de las misiones tuvieron un desenlace exitoso, destacando 38 aterrizajes exitosos convencionales, 14 en plataformas no tripuladas (drone ship) y 9 en plataformas terrestres, lo que suma un total de 61 éxitos. En contraste, se registraron 10 fallos distribuidos en distintas causas, además de 22 misiones sin intento de aterrizaje. Esto indica un alto nivel de eficacia en las misiones de SpaceX.

[32]:
```
query7 = '''
SELECT "Landing_Outcome", COUNT(*) AS Count
FROM SPACEXTBL
GROUP BY "Landing_Outcome";
'''
df7 = pd.read_sql_query(query7, con)
print(df7)
```

	Landing_Outcome	Count
0	Controlled (ocean)	5
1	Failure	3
2	Failure (drone ship)	5
3	Failure (parachute)	2
4	No attempt	21
5	No attempt	1
6	Precluded (drone ship)	1
7	Success	38
8	Success (drone ship)	14
9	Success (ground pad)	9
10	Uncontrolled (ocean)	2

Boosters Carried Maximum Payload

- Los booster versions que han transportado la máxima masa de carga útil pertenecen todos a la serie Falcon 9 Block 5 (F9 B5), destacando versiones como B1048.4, B1051.6 y B1049.7, entre otras. Este resultado indica que la versión Block 5 del Falcon 9 es la más potente y confiable para misiones con cargas pesadas.

[32]:
```
query8 = '''
SELECT "Booster_Version"
FROM SPACEXTBL
WHERE "Payload_Mass__kg_" = (
    SELECT MAX("Payload_Mass__kg_")
    FROM SPACEXTBL
);
'''
df8 = pd.read_sql_query(query8, con)
print(df8)
```

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

2015 Launch Records

- En el año 2015, se registraron dos fallos de aterrizaje en plataformas no tripuladas (drone ship), ambos ocurridos en el sitio de lanzamiento CCAFS LC-40 durante los meses de enero (01) y abril (04). Las versiones de los cohetes involucrados fueron F9 v1.1 B1012 y F9 v1.1 B1015, lo que refleja los desafíos iniciales de SpaceX en la recuperación de boosters sobre plataformas móviles.

[33]:
```
query9 = '''
SELECT strftime('%m', Date) AS Month,
       "Landing_Outcome",
       "Booster_Version",
       "Launch_Site"
FROM SPACEXTBL
WHERE substr(Date, 0, 5) = '2015'
  AND "Landing_Outcome" LIKE '%Failure (drone ship)%';
'''
df9 = pd.read_sql_query(query9, con)
print(df9)
```

	Month	Landing_Outcome	Booster_Version	Launch_Site
0	01	Failure (drone ship)	F9 v1.1 B1012	CCAFS LC-40
1	04	Failure (drone ship)	F9 v1.1 B1015	CCAFS LC-40

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

- Entre el 4 de junio de 2010 y el 20 de marzo de 2017, la mayoría de los lanzamientos no intentaron aterrizar el propulsor. Le siguen los aterrizajes exitosos y fallidos en plataformas no tripuladas (drone ship), con 5 casos cada uno, lo que muestra un periodo de experimentación y transición hacia recuperaciones más complejas.

[34]:
```
query10 = ''
SELECT "Landing_Outcome", COUNT(*) AS Count
FROM SPACEXTBL
WHERE Date BETWEEN '2010-06-04' AND '2017-03-20'
GROUP BY "Landing_Outcome"
ORDER BY Count DESC;
''
df10 = pd.read_sql_query(query10, con)
print(df10)
```

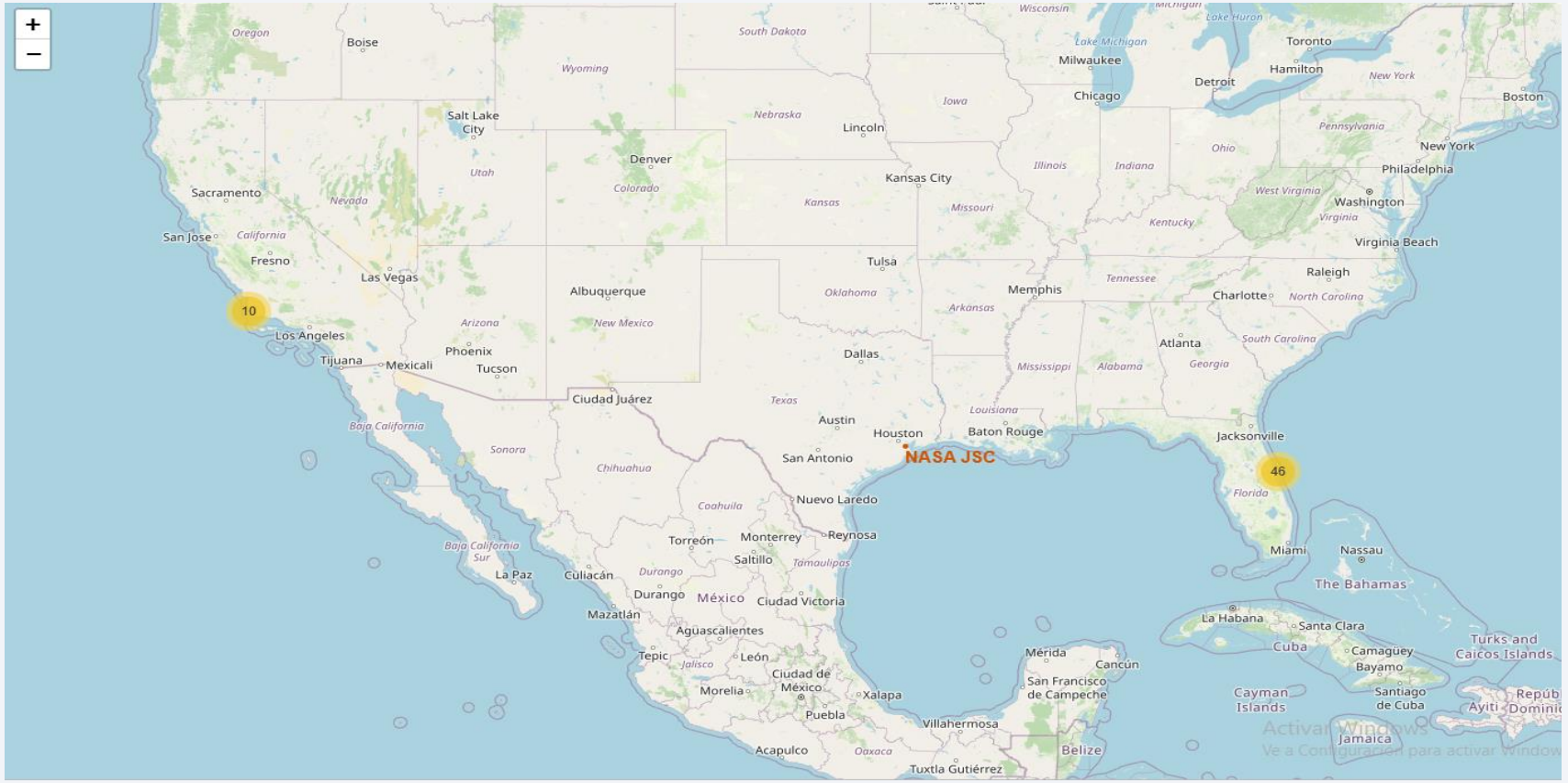
	Landing_Outcome	Count
0	No attempt	10
1	Success (drone ship)	5
2	Failure (drone ship)	5
3	Success (ground pad)	3
4	Controlled (ocean)	3
5	Uncontrolled (ocean)	2
6	Failure (parachute)	2
7	Precluded (drone ship)	1

Section 3

Launch Sites Proximities Analysis

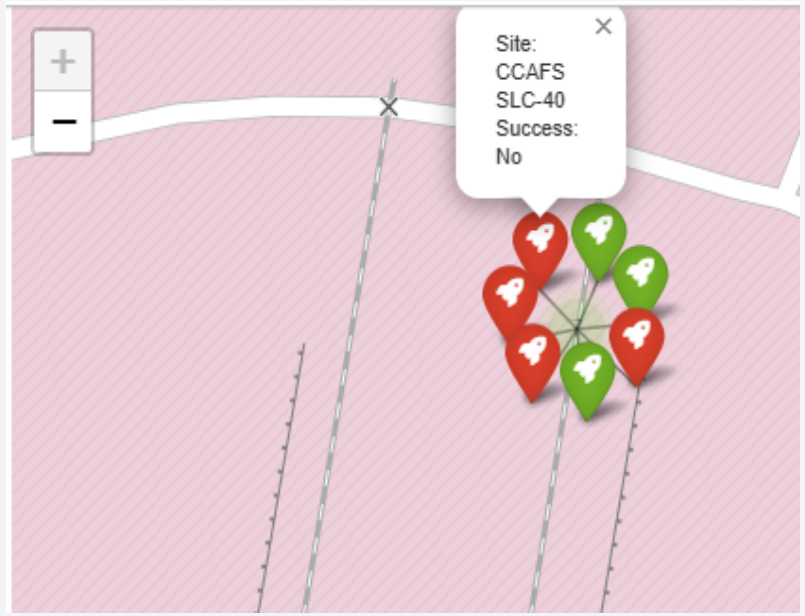
Todos los marcadores del mapa global de los sitios de lanzamiento

- Los sitios de lanzamiento de SpaceX están estratégicamente ubicados en las costas de Florida y California, lo que permite trayectorias de vuelo seguras sobre el océano y facilita el acceso eficiente a diferentes tipos de órbitas, minimizando riesgos para zonas habitadas.



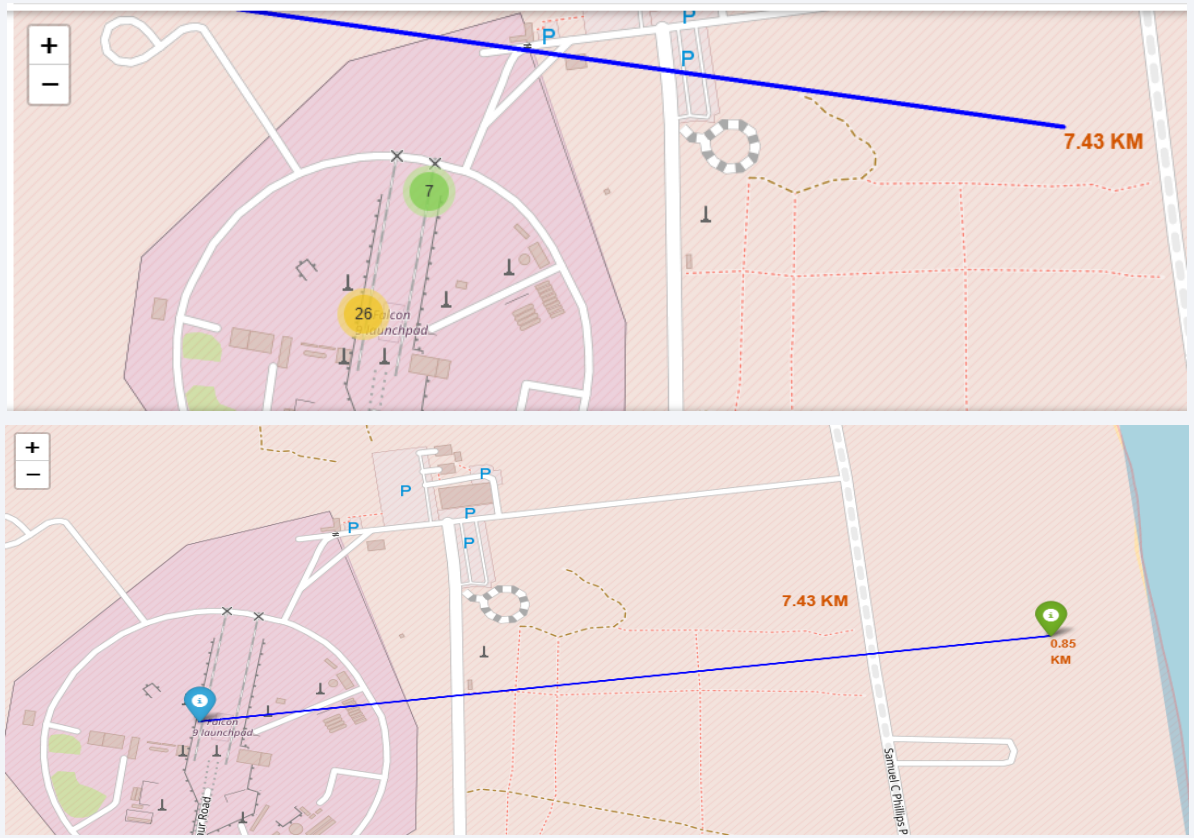
Marcadores que muestran sitios de lanzamiento con etiquetas de color

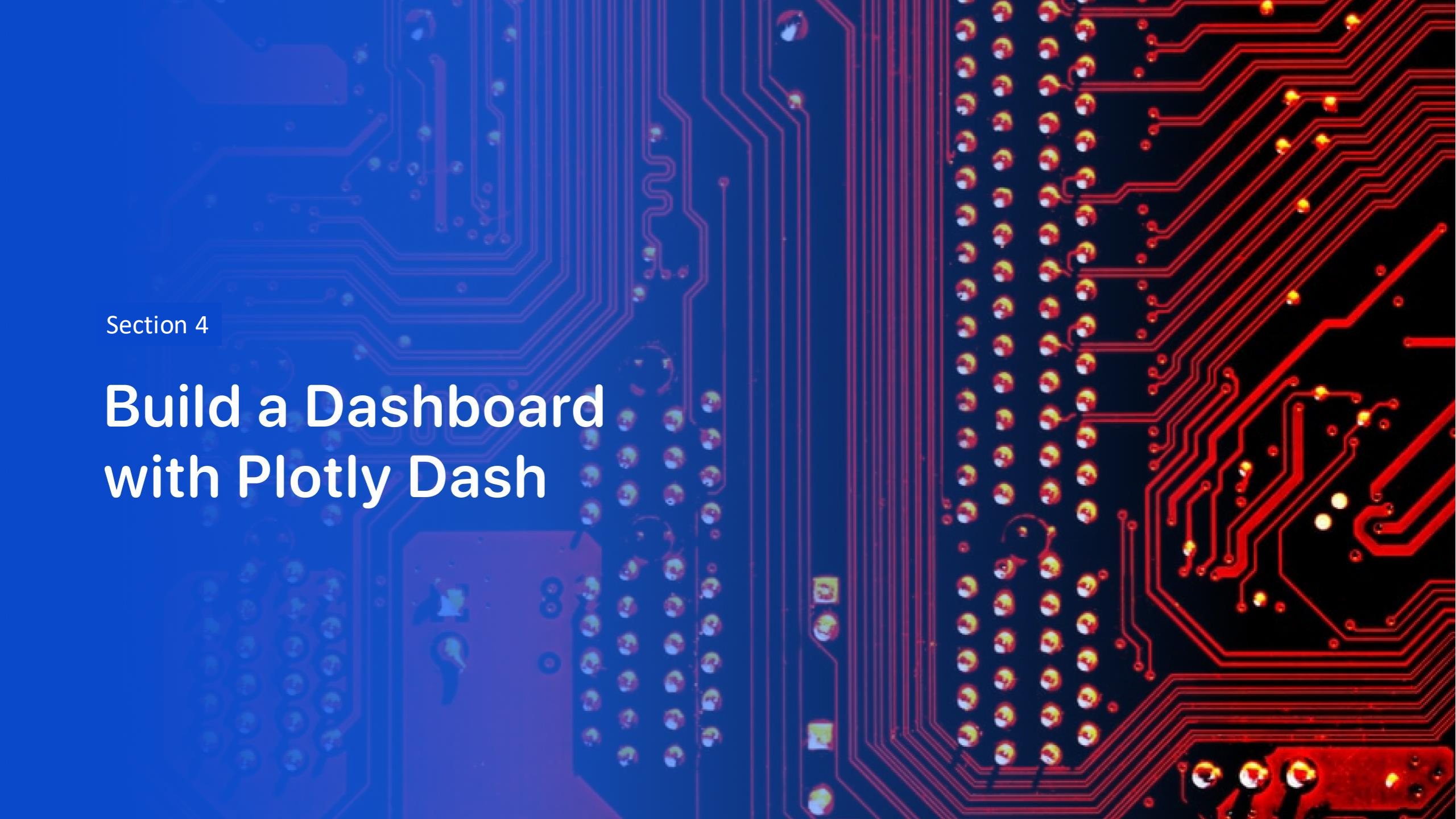
- El mapa evidencia el desempeño de los sitios de lanzamiento mediante marcadores de color, destacando que CCAFS LC-40 y KSC LC-39A en Florida concentran la mayoría de aterrizajes exitosos (verde), lo que resalta su alta confiabilidad. En cambio, otros sitios presentan más fallos (rojo).



Distancia del sitio de lanzamiento a puntos de referencia

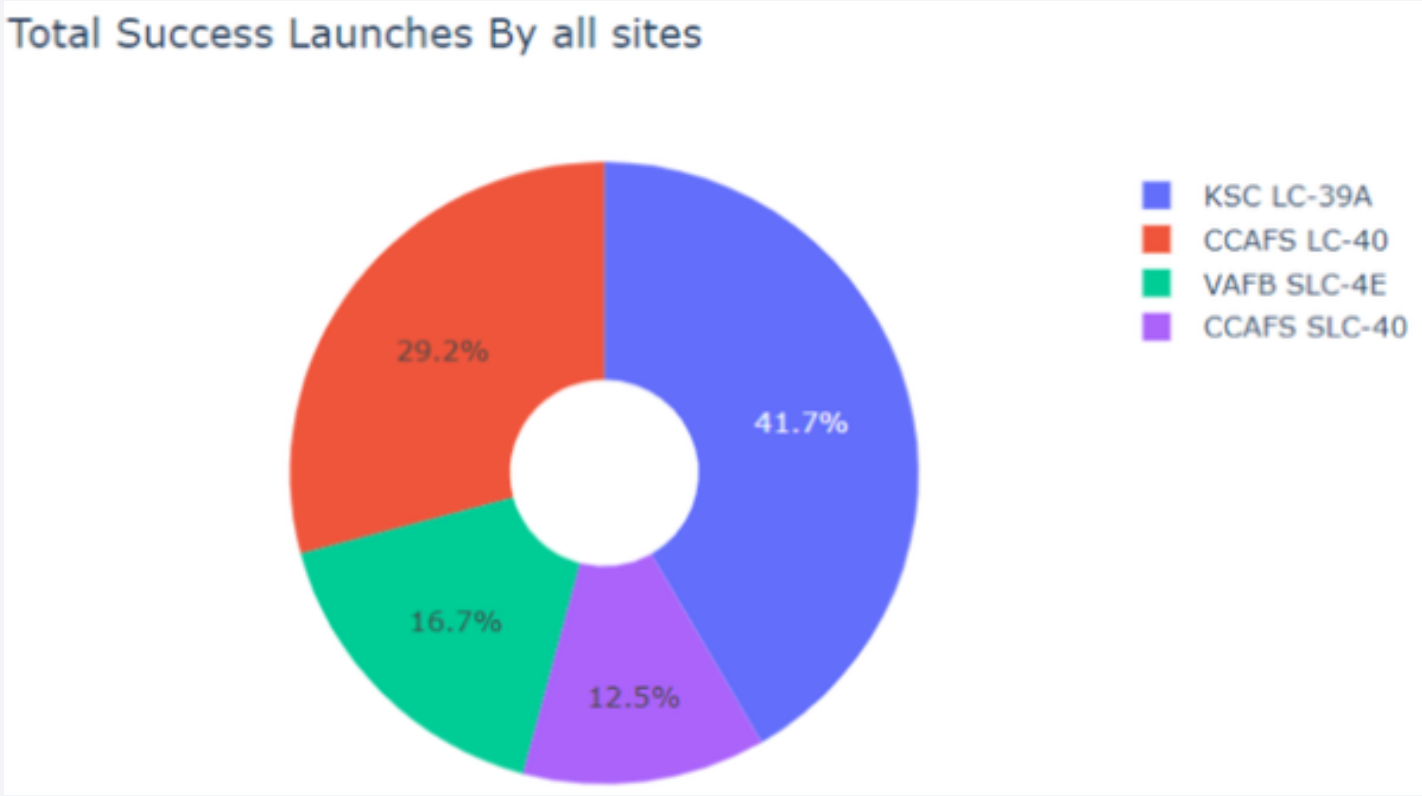
- Los sitios de lanzamiento están estratégicamente ubicados lejos de infraestructuras críticas y áreas pobladas, pero cerca del mar, lo que cumple con criterios de seguridad y eficiencia logística.




Section 4
Build a Dashboard with Plotly Dash

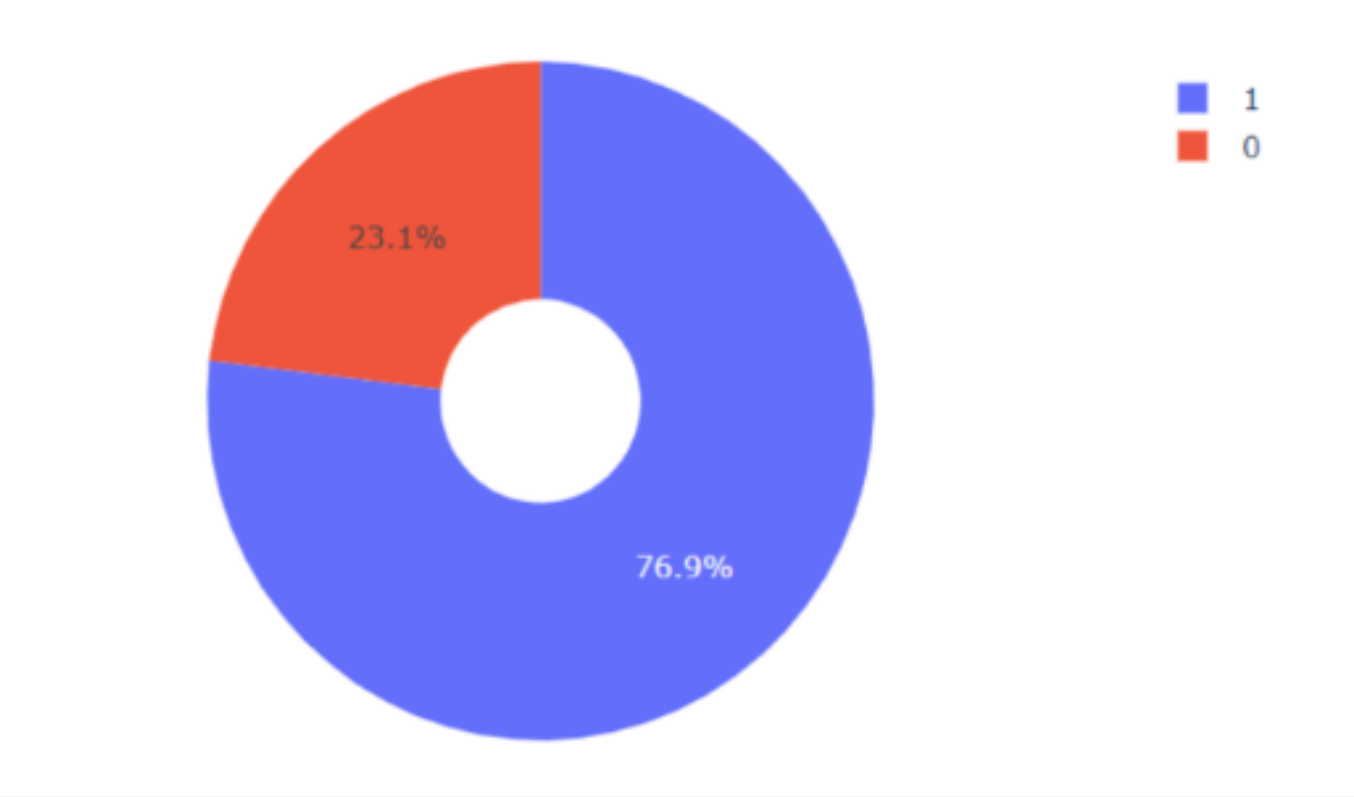
Porcentaje total de lanzamientos exitosos por sitio

- El alto porcentaje de 41.7% en KSC LC-39A evidencia su papel central en misiones exitosas, posiblemente gracias a su infraestructura, experiencia operativa y ubicación.



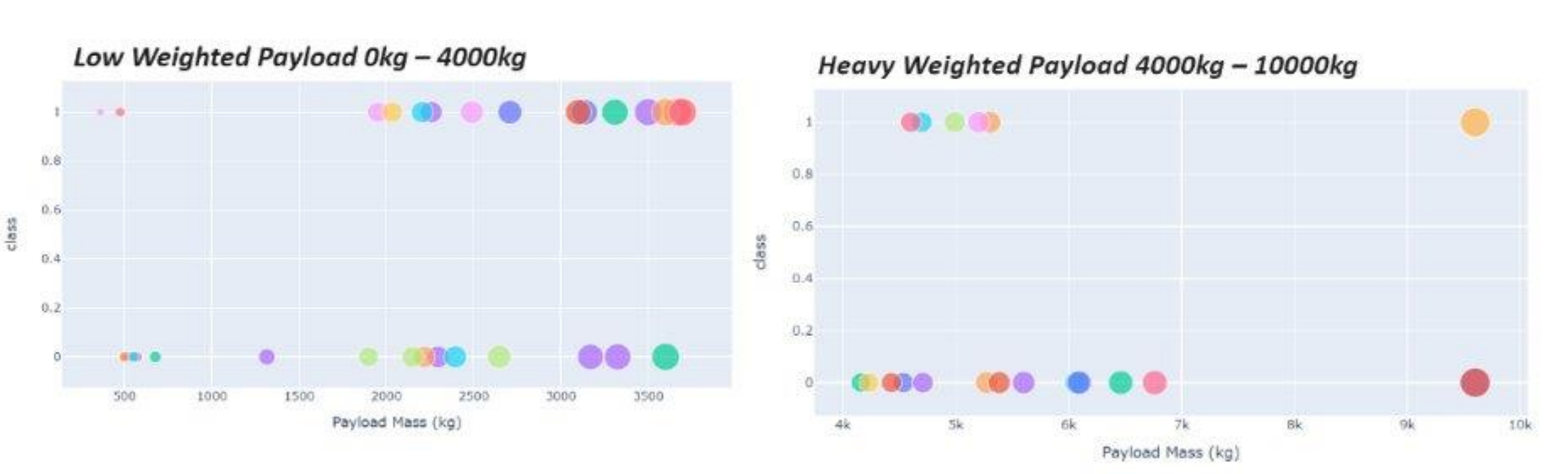
Tasa de éxito del sitio con mejor rendimiento

- La combinación de alto volumen y alta tasa de éxito refuerza a KSC LC-39A como el sitio más confiable con 76.9% para lanzamientos espaciales. En comparación con los fallos de 23.1%.



Payload vs Launch Outcome por sitio de lanzamiento

- Los lanzamientos con payloads ligeros tienen mayores tasas de éxito, mientras que los payloads pesados presentan más fallos, lo que indica que el peso influye directamente en la probabilidad de éxito de una misión.



Section 5

Predictive Analysis (Classification)

Precisión de la clasificación

- El clasificador de árbol de decisión es el modelo con mayor precisión de clasificación

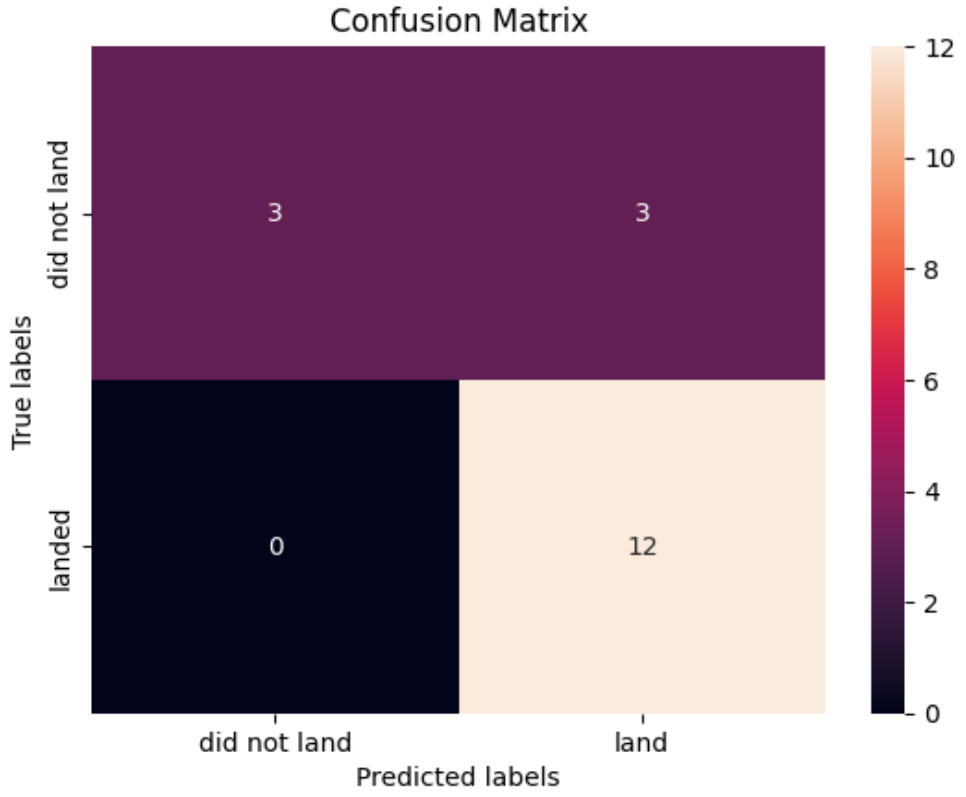
```
models = {'KNeighbors':knn_cv.best_score_,
              'DecisionTree':tree_cv.best_score_,
              'LogisticRegression':logreg_cv.best_score_,
              'SupportVector': svm_cv.best_score_}

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': 'auto', 'min_samples_leaf': 2, 'min_samples_split': 5, 'splitter': 'random'}
```

Matrix de confusión

- La matriz de confusión del clasificador de árbol de decisión indica que puede distinguir entre las diferentes clases. Sin embargo, un problema importante es la alta cantidad de falsos positivos, es decir, aterrizajes no exitosos que son clasificados incorrectamente como exitosos.

```
[31]: yhat = knn_cv.predict(X_test)
      plot_confusion_matrix(Y_test,yhat)
```



Conclusiones

- A mayor cantidad de lanzamientos en un sitio, mayor es la tasa de éxito registrada en dicho lugar.
- La tasa de éxito de los lanzamientos comenzó a incrementarse a partir del año 2013 y continuó en aumento hasta 2020.
- Las órbitas ES-L1, GEO, HEO, SSO y VLEO presentaron las tasas de éxito más altas.
- El sitio KSC LC-39A registró la mayor cantidad de lanzamientos exitosos entre todos los centros de lanzamiento analizados.
- El clasificador basado en árboles de decisión demostró ser el algoritmo de machine learning más adecuado para esta tarea, al ofrecer el mejor rendimiento en la predicción de lanzamientos exitosos.

Apéndice

- El enlace del notebook: https://github.com/EderZC12/TestRepo/tree/c08c47a0b1e89e840156cc25ba8cc1c15d9da4fb

Thank you!