



IBM Developer  
SKILLS NETWORK

# Winning Space Race with Data Science

Abraham Tartalos  
Enero 2025



# Tabla de Contenido

---

- Resumen Ejecutivo
- Introducción
- Metodología
- Resultados
- Conclusión
- Apéndice

# Resumen Ejecutivo

---

Este proyecto busca predecir el éxito del aterrizaje del cohete Falcon 9 utilizando análisis de datos y aprendizaje automático. Los resultados muestran que el modelo de Regresión Logística fue el más preciso. Esto puede ayudar a SpaceX a optimizar sus operaciones reduciendo el costo de los lanzamientos.

# Introducción

---

- SpaceX ha revolucionado la industria aeroespacial con cohetes reutilizables, reduciendo drásticamente los costos de lanzamiento. Un aspecto crítico para lograr esto es el éxito del aterrizaje de la primera etapa del Falcon 9.
- Este proyecto tiene como objetivo predecir con precisión si un aterrizaje será exitoso utilizando datos históricos.
- En este informe se exploran factores claves que influyen en los resultados, como la masa de carga útil, el tipo de órbita, entre otras, aplicando modelos de aprendizaje automático para maximizar la precisión.



Section 1

# Methodology

# Methodology

---

## Resumen

- Metodología de recogida de datos:
  - Descripción de cómo se recopilaron los datos.
- Preprocesamiento de datos
  - Descripción de cómo se procesaron los datos.
- Análisis exploratorio de datos (EDA) utilizando visualización y SQL
- Análisis visuales interactivos utilizando Folium y Plotly Dash
- Análisis predictivos utilizando modelos de clasificación
  - Cómo construir, entrenar y evaluar modelos de clasificación.

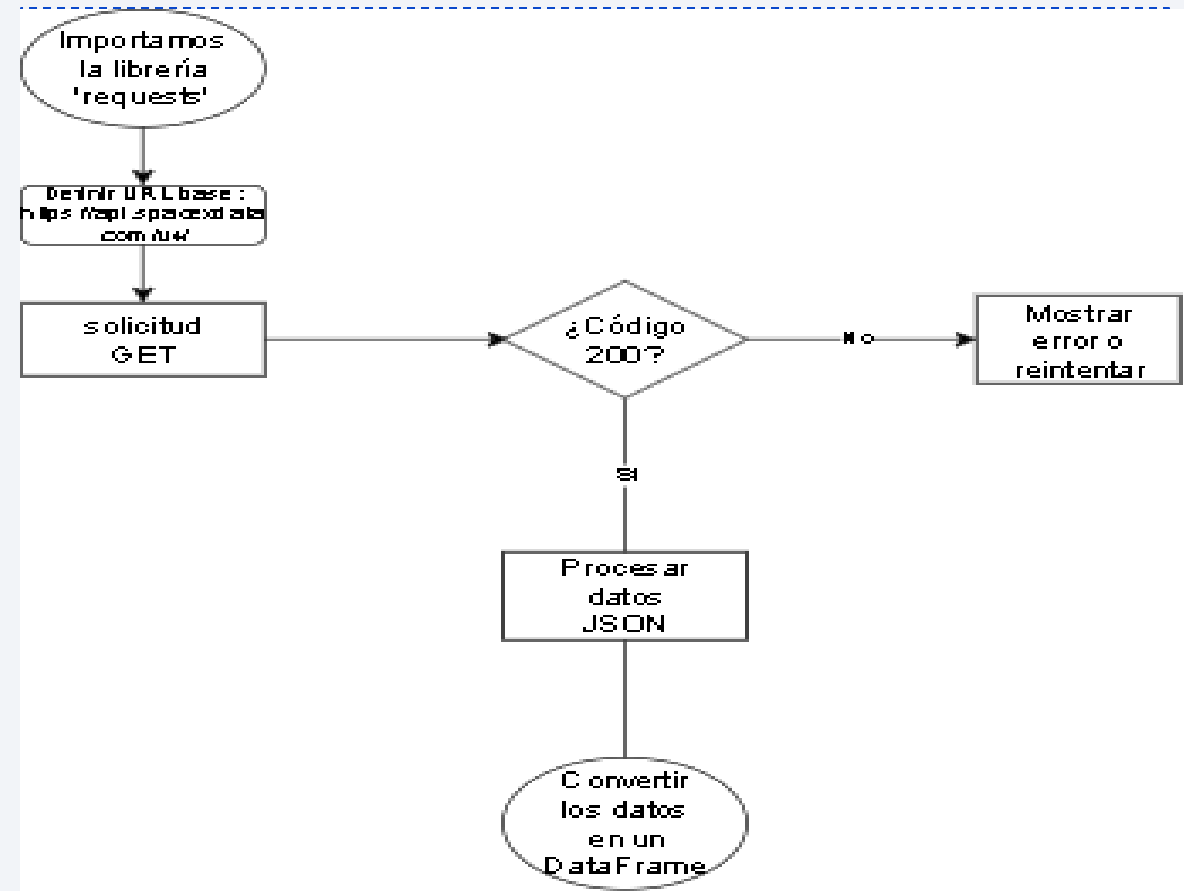
# Recopilación de datos

---

- Los conjuntos de datos utilizados en este proyecto fueron recolectados de dos fuentes, de la API publica de SpaceX y de la web.
- Luego extenderemos estas dos fuentes utilizadas y se mostrarán diagramas de flujo que permiten visualizar el proceso de obtención de datos.

# Recopilación de datos – API de SpaceX

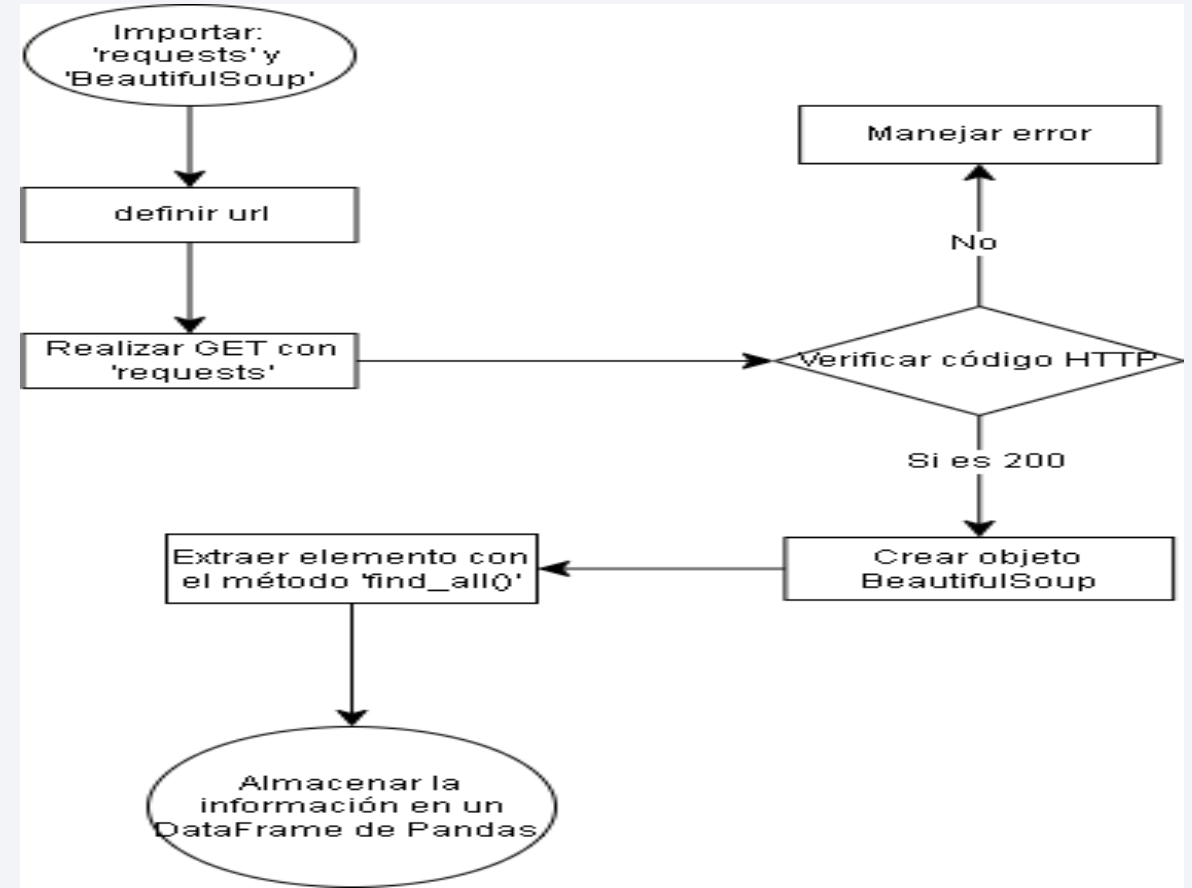
- Los datos se recopilaron de los registros históricos de lanzamiento de SpaceX, disponibles públicamente en la API de SpaceX.
- A continuación se muestra la URL del cuaderno de llamadas a la API de SpaceX: [click aquí](#)





# Recopilación de datos - Scraping

- A continuación se muestra el flujo que se llevó a cabo, utilizando como técnica el Web Scraping, para extraer datos alojados en unas tablas en Wiki sobre registros de lanzamiento de Falcon 9.
- A continuación se muestra la URL del cuaderno de Web Scraping: [click aquí](#)



# Preprocesamiento de Datos (Data Wrangling)

---

- En esta etapa se identificaron los valores nulos, de lo cual se obtuvo que solo la columna 'LandingPad' presentaba valores nulos (28.888889% de VN).
- Luego se calcularon estadísticas como el número de aterrizajes para cada sitio de lanzamiento o el número de ocurrencias de lanzamiento para cada órbita.
- Por último se creó una variable numérica llamada 'landing\_outcome' que representa los resultados de los aterrizaje (0 si el aterrizaje no fue exitoso y 1 si lo fue).
- A continuación se muestra la URL del cuaderno de preprocesamiento de datos: [\*click aquí\*](#)

# EDA con Visualización de Datos

---

- En este análisis exploratorio de datos se trazaron gráficos de dispersión entre distintas variables para entender la correlación entre ellas, algunas de las variables que se enfrentaron son: 'flight\_number' vs. 'launch\_site', 'payload\_mass' vs. 'launch\_site' o 'flight\_number' vs. 'orbit\_type', entre otros. A demás también se puede ver un histograma que muestra la tasa de éxito por tipo de orbita.
- A continuación se muestra la URL del cuaderno de Análisis Exploratorio de Datos y visualización: [click aquí](#)

# EDA con SQL

---

- Entre las consultas SQL que se realizaron en esta cuaderno están:
- Consulta para mostrar los nombres de cada uno de los sitios de lanzamiento.
- Consulta para obtener la masa total de carga útil transportada por los propulsores de la NASA (CRS).
- Consulta para obtener la media de la carga útil transportada en la versión de booster F9 v1.1.
- Consulta para enumerar los nombres de los propulsores que tienen éxito en naves no tripuladas y con una carga útil de entre 4000 y 6000 kg, entre otras.
- A continuación se muestra la URL del cuaderno de Análisis Exploratorio de Datos con SQL: [click aquí](#)

# Mapa Interactivo con Folium

---

- Entre los objetos creados y agregados a los mapas generados en este apartado se encuentran: Marcadores (Marker), Conjunto de marcadores (MarkerCluster), objetos línea (Polyline), objetos MousePosition, objetos circulo (Circle).
- Los objetos Marker y Circle fueron utilizados para marcar y delimitar cada sitio de lanzamiento en el mapa, por su parte los MarkerCluster se utilizaron para agrupar marcadores que tienen la misma coordenada y así poder subdividirlos en aterrizajes exitosos y fallidos.
- Por otro lado el objeto MousePosition nos sirve para obtener la coordenada de un mouse sobre un punto en el mapa. Por último, Polyline lo utilizamos para marcar líneas en el mapa, que muestran directamente la distancia entre dos puntos en el mapa, por ejemplo; la distancia/el camino desde un sitio de lanzamiento hasta la ciudad más cercana.
- A continuación se muestra la URL del cuaderno de Análisis Visual Interactivo con Folium: [click aquí](#)



# Panel de Control con Plotly Dash

---

- En este dashboard o panel de control podemos observar, como componentes de entrada, una lista desplegable y un control deslizante de rango para interactuar con un gráfico circular y un gráfico de dispersión.
- Estos gráficos e interacciones fueron agregados para poder obtener un análisis a partir de cada sitio de lanzamiento y un rango de carga útil.
- A continuación se muestra la URL del cuaderno de Plotly Dash: [click aquí](#)

# Análisis Predictivo (Clasificación)

---

- Para encontrar el modelo de clasificación con mejor rendimiento, creamos modelos de regresión logística, SVM (Maquina de Soporte de Vectores), arboles de decisión y K-NN (K vecinos más cercanos).
- Luego para cada uno de estos modelos se creó y entrenó un objeto GridSearchCV, con el objetivo de encontrar los mejores hiper parámetros para su construcción.
- Luego se calcula la precisión de cada uno de los modelos en los datos de prueba utilizando el método 'score', por último se traza una matriz de confusión para entender visualmente el rendimiento de los modelos.
- A continuación se muestra la URL del cuaderno de Análisis Predictivo: [click aquí](#)

# Resultados

---

- El análisis exploratorio de datos da como resultado la obtención de la correlación de las distintas variables entre sí, además se presenta un gráfico con la tendencia anual de éxito de lanzamiento y luego se realizan trabajos de Feature Engineering.
- Más adelante se mostrarán capturas de pantalla como demostración del análisis interactivo.
- El análisis predictivo dio como resultado que el modelo más optimizado es el modelo de Regresión Logística con una precisión en los datos de prueba del 83,33%.



The background of the slide is an abstract composition. It features a dark blue base color. Overlaid on this are numerous diagonal streaks in shades of blue and red, creating a sense of motion or data flow. A faint, light blue grid pattern is also visible, particularly in the lower-left quadrant. The overall effect is high-tech and digital.

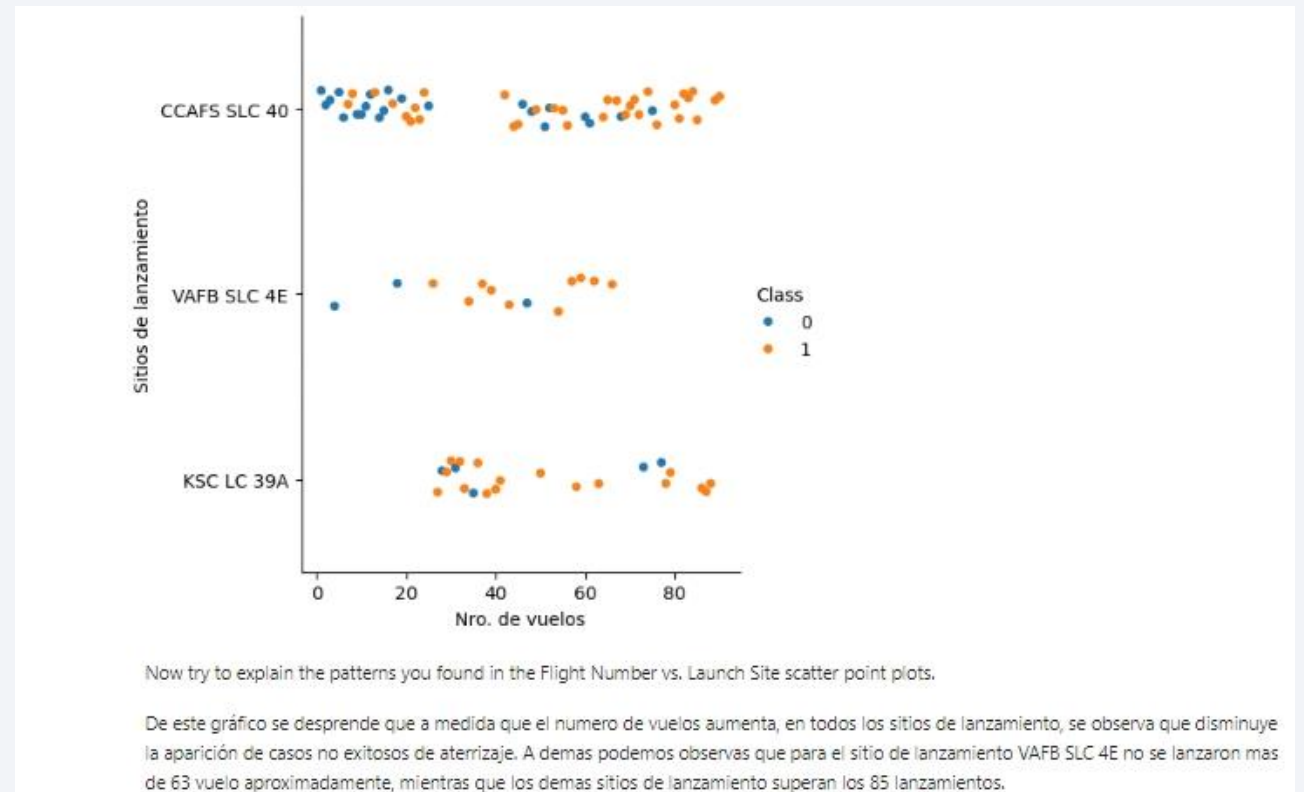
Section 2

# Insights drawn from EDA



# Número de vuelo vs. Sitio de lanzamiento

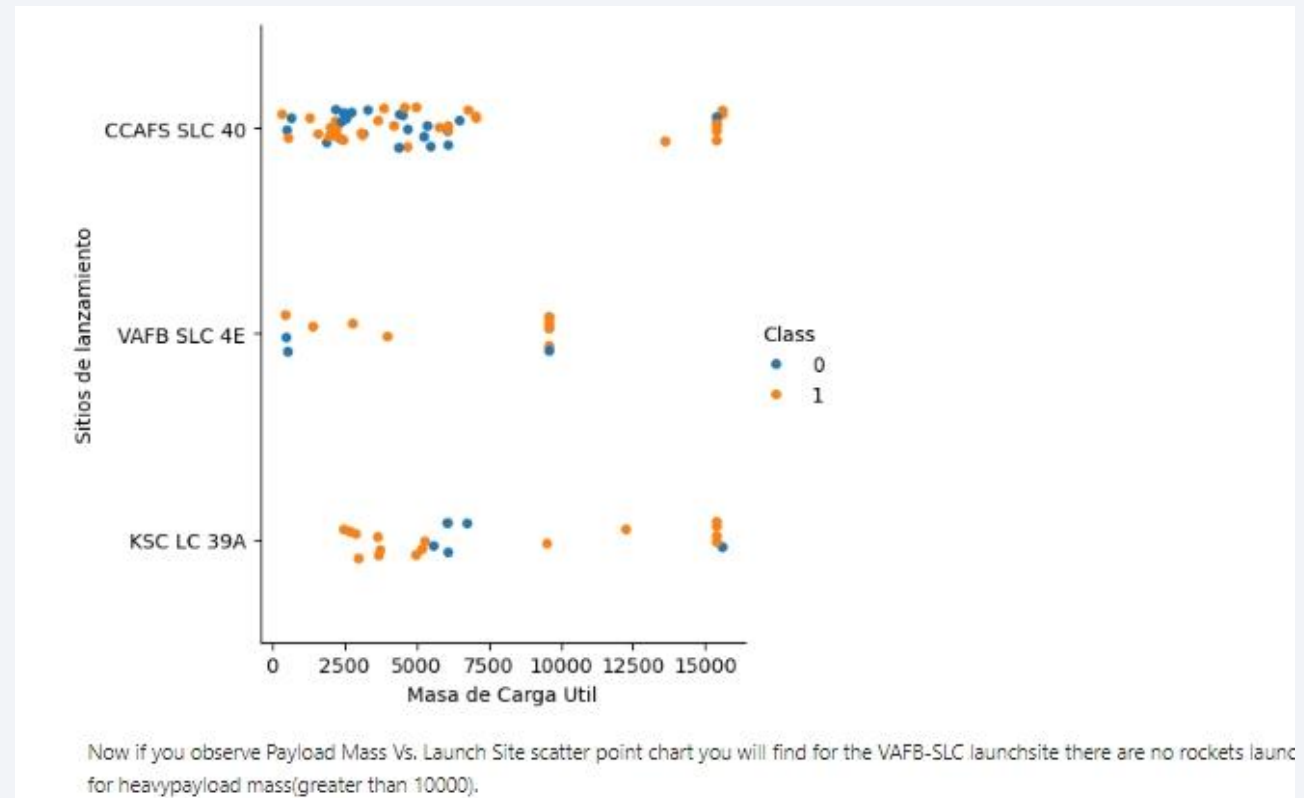
- A continuación se muestra un diagrama de dispersión que enfrenta las variables Flight Number vs. Launch Site.
- A medida que el nro. de vuelos aumenta, en todos los sitios de lanzamiento, se observa que disminuyen los casos no exitosos de aterrizaje.





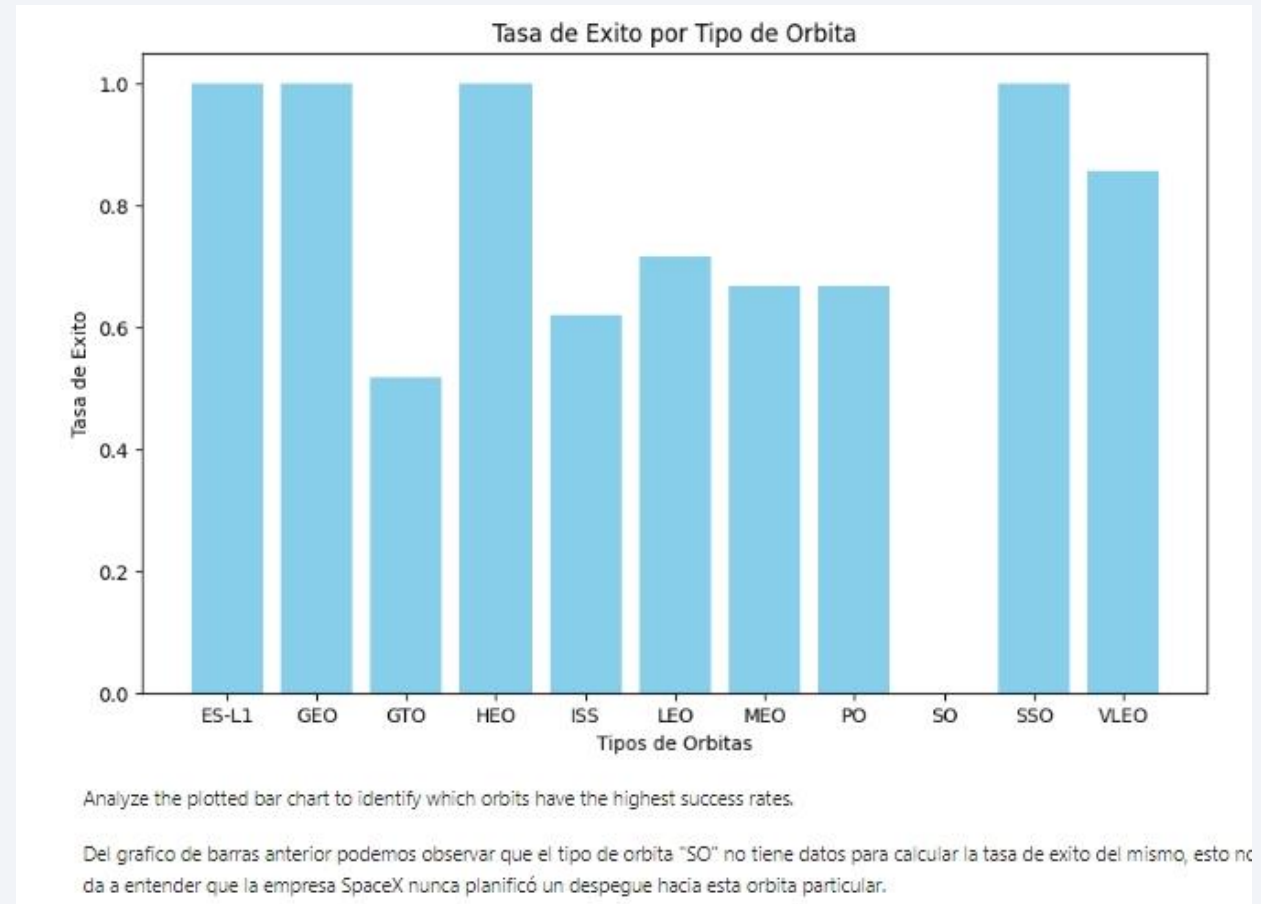
# Carga útil vs. Sitio de lanzamiento

- A continuación se muestra un diagrama de dispersión que enfrenta las variables Payload vs. Launch Site
- De este gráfico se desprende que en el sitio de lanzamiento VAFB SLC 4E no tiene lanzamientos registrados de cohetes que transporten una carga útil pesada (mayor a 10000 toneladas).



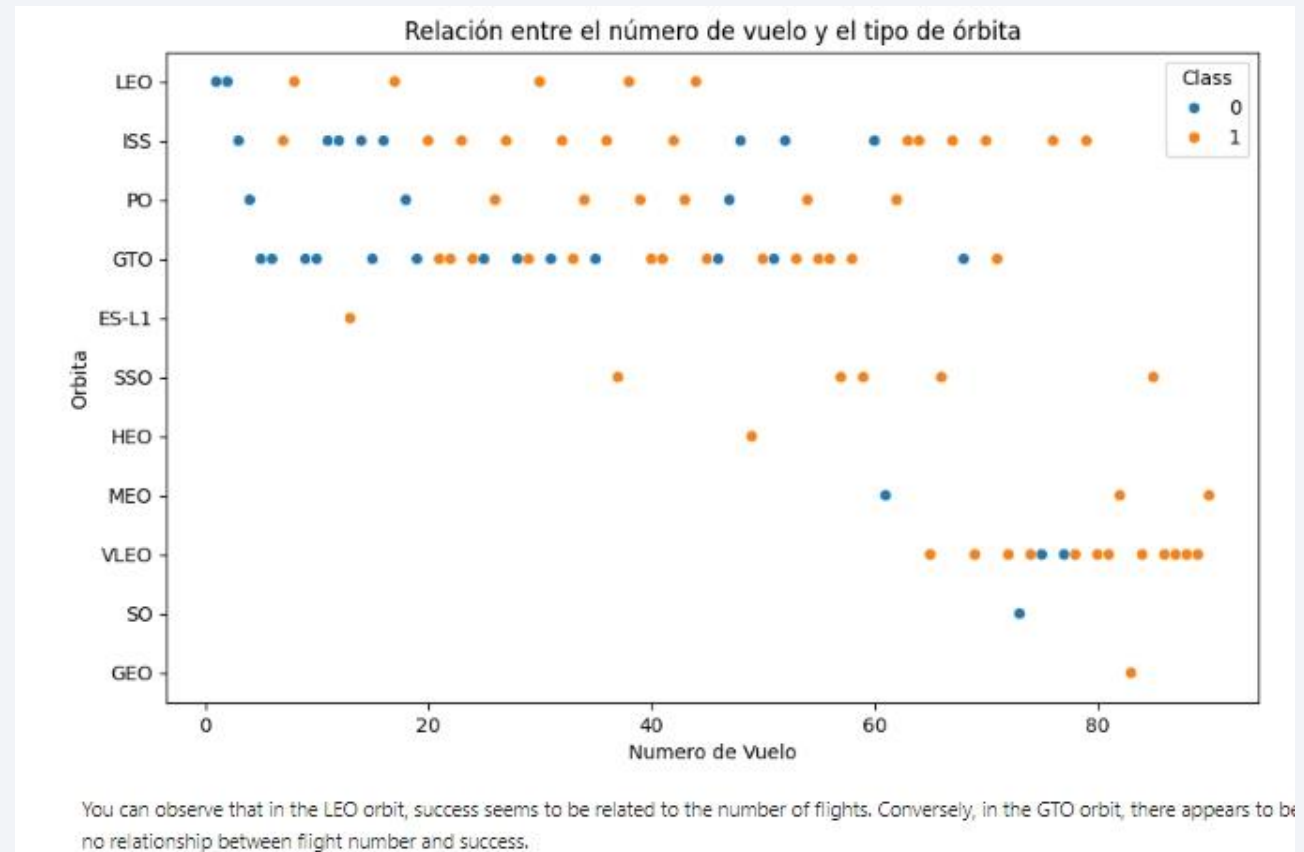
# Tasa de éxito vs. Tipo de órbita

- A continuación se muestra un gráfico de barras que muestra la tasa de éxito por tipo de órbita.
- Se observa que para la órbita SO no se registran lanzamientos.
- También podemos ver que la órbita con menos tasa de éxito es GTO.



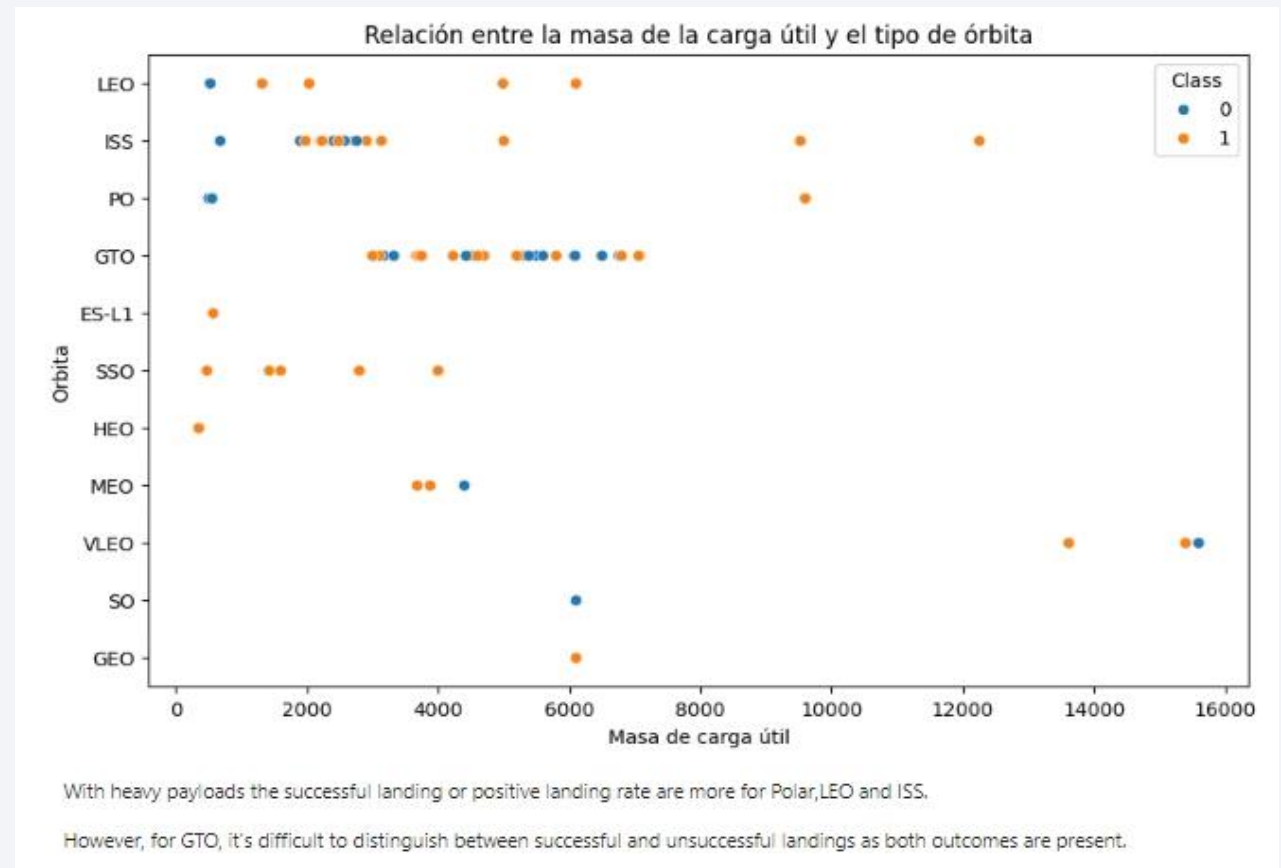
# Número de vuelo vs. Tipo de órbita

- A continuación se muestra un diagrama de dispersión que enfrenta las variables Flight number vs. Orbit type
- Se puede ver que en la órbita LEO el éxito está correlacionado con el número de vuelo.
- Por el contrario, en la órbita GTO no parece haber relación con el número de vuelo.



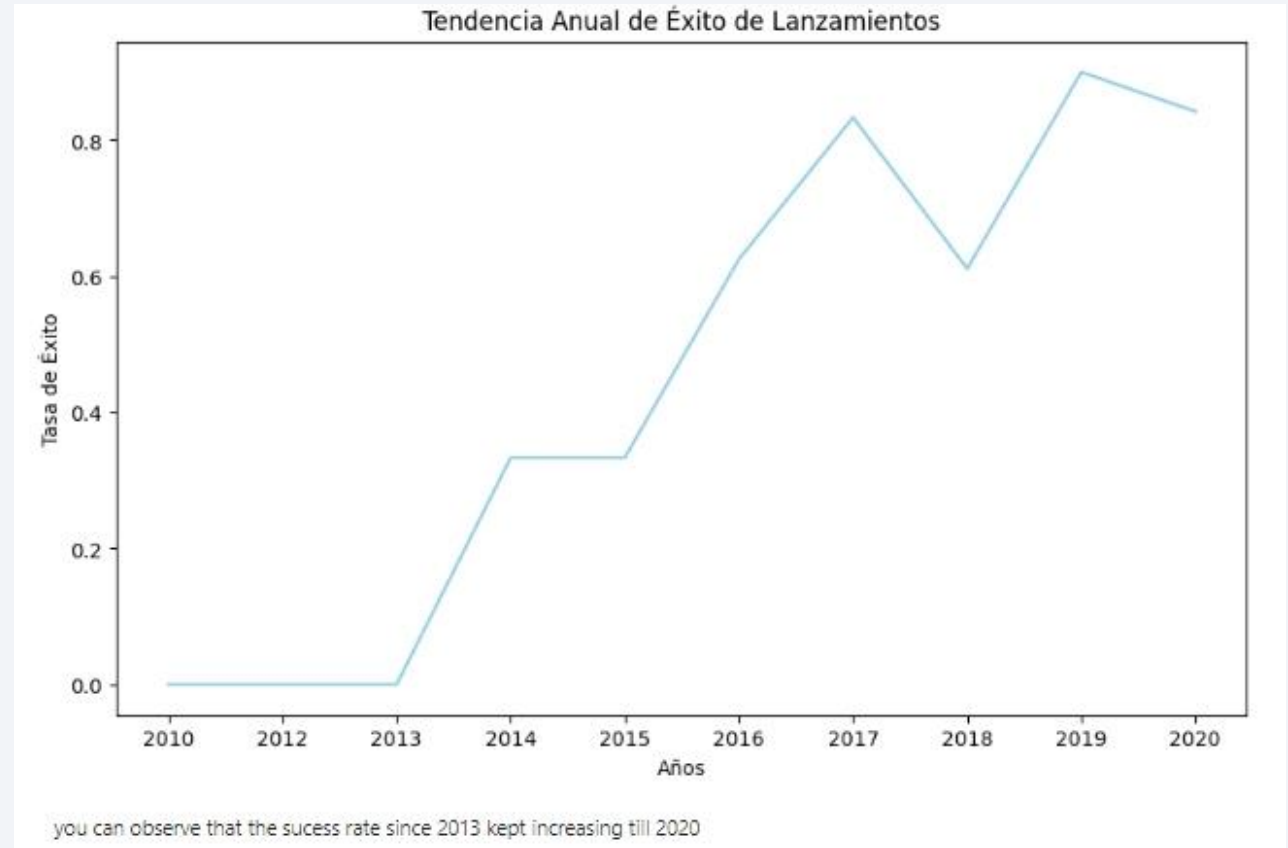
# Carga útil vs. Tipo de órbita

- A continuación se muestra un diagrama de dispersión que enfrenta las variables payload vs. orbit type.
- Con cargas útiles pesadas, la tasa de aterrizajes positivos es mayor para Polar, LEO e ISS.
- Sin embargo, para GTO es difícil distinguir entre aterrizajes exitosos o fallidos, ya que ambos están presentes.



# Tendencia Anual de Lanzamientos Exitosos

- A continuación se muestra un gráfico de líneas que muestra la tendencia de lanzamientos exitosos.
- Se puede observar que la tasa de éxito se mantiene en aumento desde 2013.





# Todos los Nombres de los Sitios de Lanzamiento

---

- A continuación, la consulta que muestra los 4 nombres únicos de los sitios de lanzamiento.

```
1: # Escribimos la consulta SQL
query = """
SELECT DISTINCT launch_site
FROM SPACEXTBL
"""

# Ejecutar la consulta
cur.execute(query)

# Obtener los resultados de la consulta
res = cur.fetchall()

# Mostrar los resultados
for fila in res:
    print(fila)
```

```
('CCAFS LC-40',)
('VAFB SLC-4E',)
('KSC LC-39A',)
('CCAFS SLC-40',)
```

# Nombres de los Sitios de Lanzamiento que comienzan con 'CCA'

- En la imagen se ve la consulta SQL para mostrar los 5 registros donde los sitios de lanzamiento comienzan con `CCA`.

```
27]: # Crear la consulta sql
      query = """
      SELECT *
      FROM SPACEXTBL
      WHERE launch_site
      LIKE 'CCA%'
      LIMIT 5
      """
      # Ejecutar la consulta
      cur.execute(query)
      # Obtener los resultados de la consulta
      res = cur.fetchall()
      # Mostrar los resultados
      for fila in res:
          print(fila)

('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')
```

# Masa Total de Carga Útil

---

- La carga útil total transportada por los cohetes de la NASA es de 45596KG.
- La siguiente captura de pantalla muestra la consulta SQL que deja ver este resultado.

```
: # Crear la consulta sql
query = """
SELECT SUM(PAYLOAD_MASS__KG_) as TOTAL_PAYLOAD_MASS__KG
FROM SPACEXTBL
WHERE Customer = 'NASA (CRS)'
"""

# Ejecutar consulta
cur.execute(query)
# Obtener los resultados de la consulta
res = cur.fetchall()
# Mostrar los resultados
print(res)

[(45596,)]
```

# Masa de Carga Útil Promedio para los F9 v1.1

---

- Para los cohetes de la versión F9 v1.1 se calculó una masa media de la carga útil de 2534,66KG aproximadamente.

```
]:
```

```
query = """
SELECT AVG(PAYLOAD_MASS__KG_)
FROM SPACEXTBL
WHERE Booster_Version
LIKE 'F9 v1.1%'
"""

cur.execute(query)

cur.fetchall()
```

```
]:
```

```
[(2534.6666666666665,)]
```

# Fecha del Primer Aterrizaje Exitoso en Tierra

---

- A continuación se muestra la captura de pantalla de la consulta que muestra la fecha del primer aterrizaje exitoso en plataforma de tierra.
- La fecha de este aterrizaje es '22-12-2015'

```
# crear la consulta sql
query = """
SELECT MIN(Date)
FROM SPACEXTBL
WHERE Landing_Outcome = 'Success (ground pad)'
"""

# Ejecutar la consulta
cur.execute(query)
#Obtener los resultados de la consulta anterior en una variable llamada res
res = cur.fetchone()

# Mostrar el resultado
print(f"El primer aterrizaje exitoso en una plataforma terrestre es: : {res}")
```

El primer aterrizaje exitoso en una plataforma de exito es: : ('2015-12-22',)



## Aterrizaje Exitoso de Nave No Tripulada con Carga Útil de entre 4000 kg a 6000 kg

---

- Ahora se puede ver una captura de pantalla que muestra los propulsores que cumplen con la condición del título.
- Los nombre de los propulsores son:
- F9 FT B1022, F9 FT B1026, F9 FT B1021.2, F9 FT B1031.2.

```
1: # crear la consulta sql
query = """
SELECT Booster_Version
FROM SPACEXTBL
WHERE Landing_Outcome = 'Success (drone ship)'
AND PAYLOAD_MASS_KG > 4000
AND PAYLOAD_MASS_KG < 6000;
"""

# Ejecutar la consulta
cur.execute(query)
# Obtener los resultados de la consulta anterior en una variable llamada res
res = cur.fetchall()

# Mostrar los resultados
print("Propulsores con éxito en naves no tripuladas y masa de carga útil entre 4000 y 6000:")
for fila in res:
    print(fila[0])
```

```
Propulsores con éxito en naves no tripuladas y masa de carga útil entre 4000 y 6000:
F9 FT B1022
F9 FT B1026
F9 FT B1021.2
F9 FT B1031.2
```

# Número Total de Resultados de Misiones Exitosas y Fallidas

---

- La captura de la consulta a continuación da como resultado:
- Misiones exitosas: 99
- Misiones no exitosas: 2

```
I: # Crear la consulta sql
query = """
SELECT Mission_Outcome, COUNT(*) as Total_Count
FROM SPACEXTBL
GROUP BY Mission_Outcome;
"""

# Ejecutar la consulta en el cursor de la coneccion que está abierta
cur.execute(query)
# Obtener Los resultados de la ejecucion anterior
res = cur.fetchall()
# Mostrar Los resultados
print("Numero total de resultados de misión exitosos y fallidos")
for fila in res:
    print(f"Resultado: {fila[0]}, Total: {fila[1]}")
```

```
Numero total de resultados de misión exitosos y fallidos
Resultado: Failure (in flight), Total: 1
Resultado: Success, Total: 98
Resultado: Success , Total: 1
Resultado: Success (payload status unclear), Total: 1
```

# Propulsores que Transportaron la Máxima Carga Útil

- La primer imagen (de izquierda a derecha) muestra la consulta SQL que devuelve el nombre de los propulsores que transportan la masa de carga útil mayor.
- La segunda imagen muestra el resultado de la consulta y por consiguiente el nombre de cada uno de los 12 propulsores que cumplen la condición.

```
I: # Crear la consulta sql
query = """
SELECT Booster_Version
FROM SPACEXTBL
WHERE PAYLOAD_MASS_KG_ = (
    SELECT MAX(PAYLOAD_MASS_KG_)
    FROM SPACEXTBL
);
"""
# Ejecutar la consulta en el cursor de la conexión que está abierta
cur.execute(query)
# Obtener los resultados de la ejecución anterior
res = cur.fetchall()
# Mostrar los resultados
print("Nombre de los Boosters que han transportado la masa maxima de carga util:\n\n", "-"*50)
for fila in res:
    print(f"Nombre de la versión del booster: {fila[0]}")
```

Nombre de los Boosters que han transportado la masa maxima de carga util:

```
-----
Nombre de la versión del booster: F9 B5 B1048.4
Nombre de la versión del booster: F9 B5 B1049.4
Nombre de la versión del booster: F9 B5 B1051.3
Nombre de la versión del booster: F9 B5 B1056.4
Nombre de la versión del booster: F9 B5 B1048.5
Nombre de la versión del booster: F9 B5 B1051.4
Nombre de la versión del booster: F9 B5 B1049.5
Nombre de la versión del booster: F9 B5 B1060.2
Nombre de la versión del booster: F9 B5 B1058.3
Nombre de la versión del booster: F9 B5 B1051.6
Nombre de la versión del booster: F9 B5 B1060.3
Nombre de la versión del booster: F9 B5 B1049.7
```

# Réconds de Lanzamiento de 2015

---

- A continuación se mostrará una captura de pantalla de una consulta SQL que permite obtener el récord de fallas en aterrizajes en naves no tripuladas.
- En 2015 se detectaron 2 fallas en naves no tripuladas.
- Los propulsores que utilizaron los cohetes de estos aterrizajes fallidos fueron: F9 v1.1 B1012 y F9 v1.1 B1015.
- Por último, los sitios de lanzamiento donde se efectuaron estos aterrizajes fallidos fueron: CCAFS LC-40 y CCAFS LC-40.

## Ranking de Resultado de Aterrizajes entre 2010-06-04 y 2017-03-20

---

- La consulta ejecutada en este apartado dio como resultado los siguientes resultados de aterrizaje:
- Aterrizajes Exitosos ➡ Plataformas de Tierra ➡ Total = 3.
- Aterrizajes Fallidos ➡ Naves No Tripuladas ➡ Total = 5.

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 blue, with numerous bright yellow and orange lights representing cities and urban areas. The horizon line of the Earth is visible, separating the dark surface from the blackness of space.

Section 3

# Launch Sites Proximities Analysis



# Ubicación de Todos los Sitios de Lanzamiento en un Mapa

---

- En la siguiente captura de pantalla se puede observar el mapa generado en el laboratorio de "Análisis de proximidades de los sitios de lanzamiento" en el que se puede ver un marcador naranja para cada sitio (4) de lanzamiento.

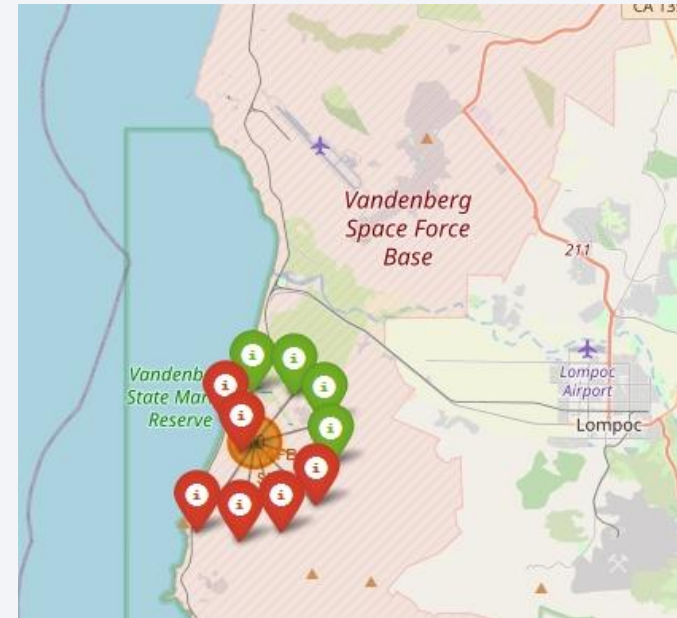
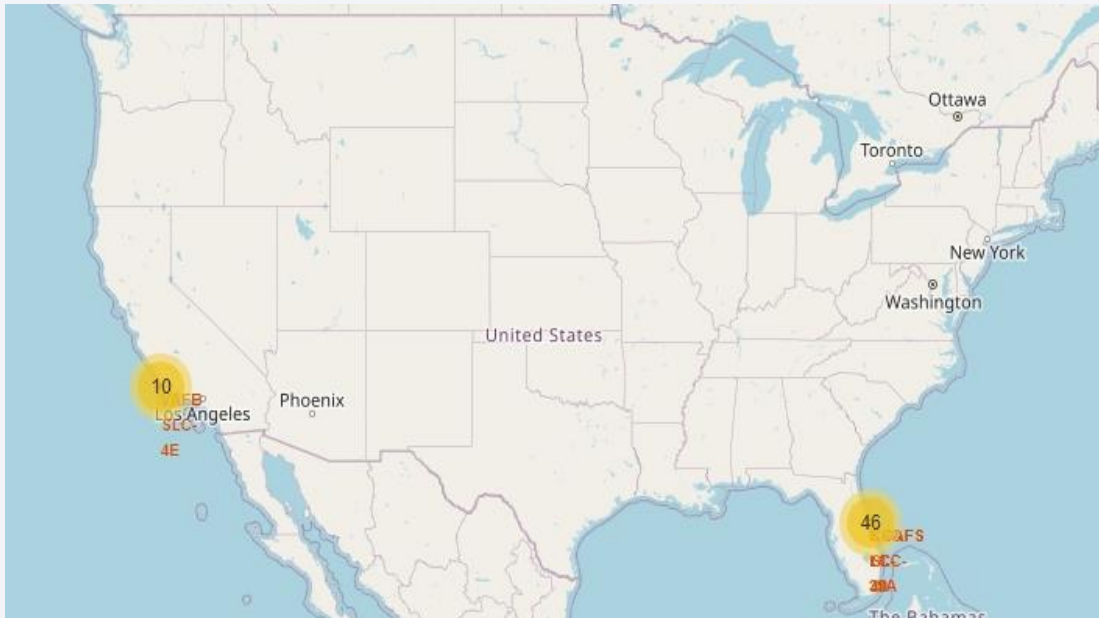




# Resultados del Lanzamiento Etiquetados con Colores en un Mapa

---

- El primer mapa (de izquierda a derecha) muestra el número total de aterrizajes en todos los sitios de lanzamiento. El segundo mapa muestra los aterrizajes exitosos y no exitosos para la Base Espacial Vandenberg.
- En un análisis visual de este mapa se puede obtener que 32 de los 56 aterrizajes fueron fallidos, y los 24 restantes fueron exitosos.



# Sitios de Lanzamiento y sus Proximidades

- Las capturas a continuación muestran sitios de lanzamiento unidos, por líneas azules, con sus proximidades.
- La primera imagen muestra el sitio KSC LC-39A junto a la autopista



más cercana a casi 1 kilómetro, y también muestra al sitio CCAFS SLC-40 unido a la costa más cercana a 1 km aproximadamente.

- La segunda imagen muestra el sitio VAFB SLC-4E unido a la ciudad más cercana a unos 13,9 KM.







Section 4

# Build a Dashboard with Plotly Dash

# Total de Lanzamientos Exitosos por Sitio de Lanzamiento

- La imagen a continuación muestra un gráfico circular con el porcentaje de lanzamientos exitosos para cada sitio de lanzamiento.
- Del gráfico se desprende que el sitio con la tasa de éxito más alta es KSC LC-39A con un 41.7%.
- El sitio con la tasa de éxito menos alta es CCAFS SLC-40 con un 12.5%.

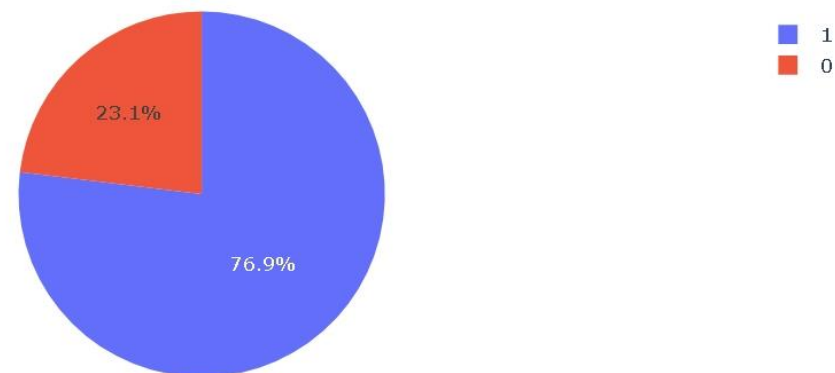


# Lanzamientos para el sitio KSC LC-39A

---

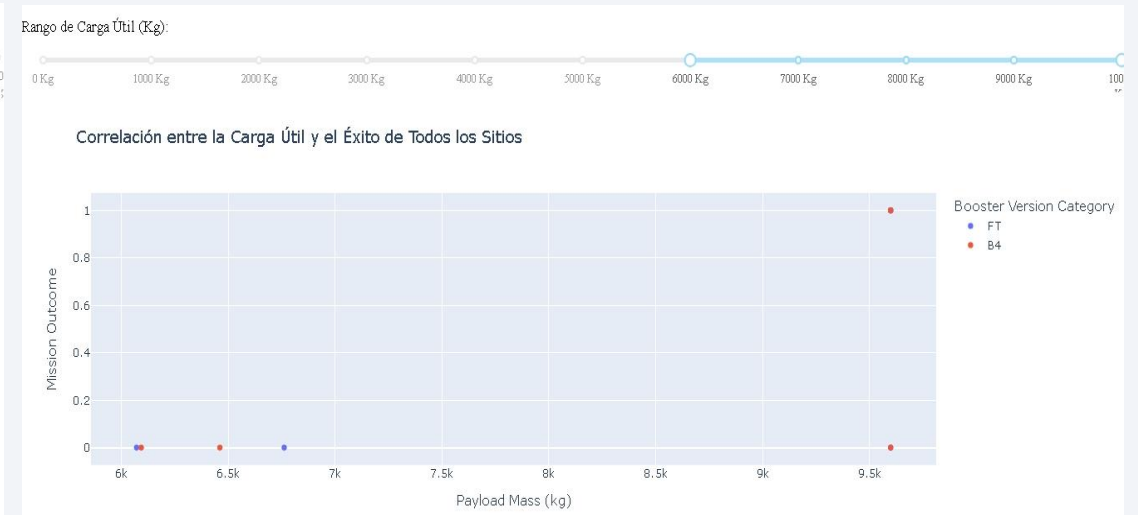
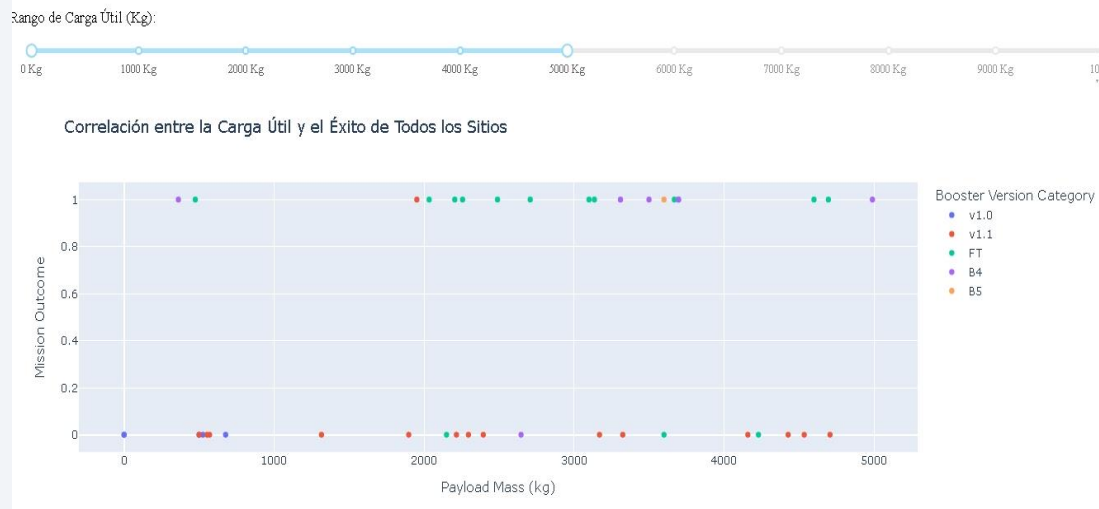
- Esta imagen muestra la comparación de aterrizajes exitosos y fallidos para el sitio con la tasa más alta de aterrizajes exitosos.
- El 76,9% (10 aterr.) de los aterrizajes fueron exitosos para este sitio.
- El 21,1% (3 aterr.) de los aterrizajes fueron fallidos para este sitio.

Lanzamientos Exitosos vs. Fallidos para KSC LC-39A



# Carga Útil vs. Resultado de Lanzamiento para Todos los Sitios

- Aquí se muestran diagramas de dispersión de carga útil vs. resultado para todos los sitios, con diferentes cargas útiles seleccionadas.
- Observamos que a medida que el rango de carga útil disminuye (por ej. De 0 a 5000kg) existen más muestras de lanzamiento.
- Y a medida que el rango de carga útil aumenta (opr ej. De 6000 a 10000Kg) existen menos muestras de lanzamientos.
- La versión de booster con mayor tasa de éxito es la versión 'FT'.







Section 5

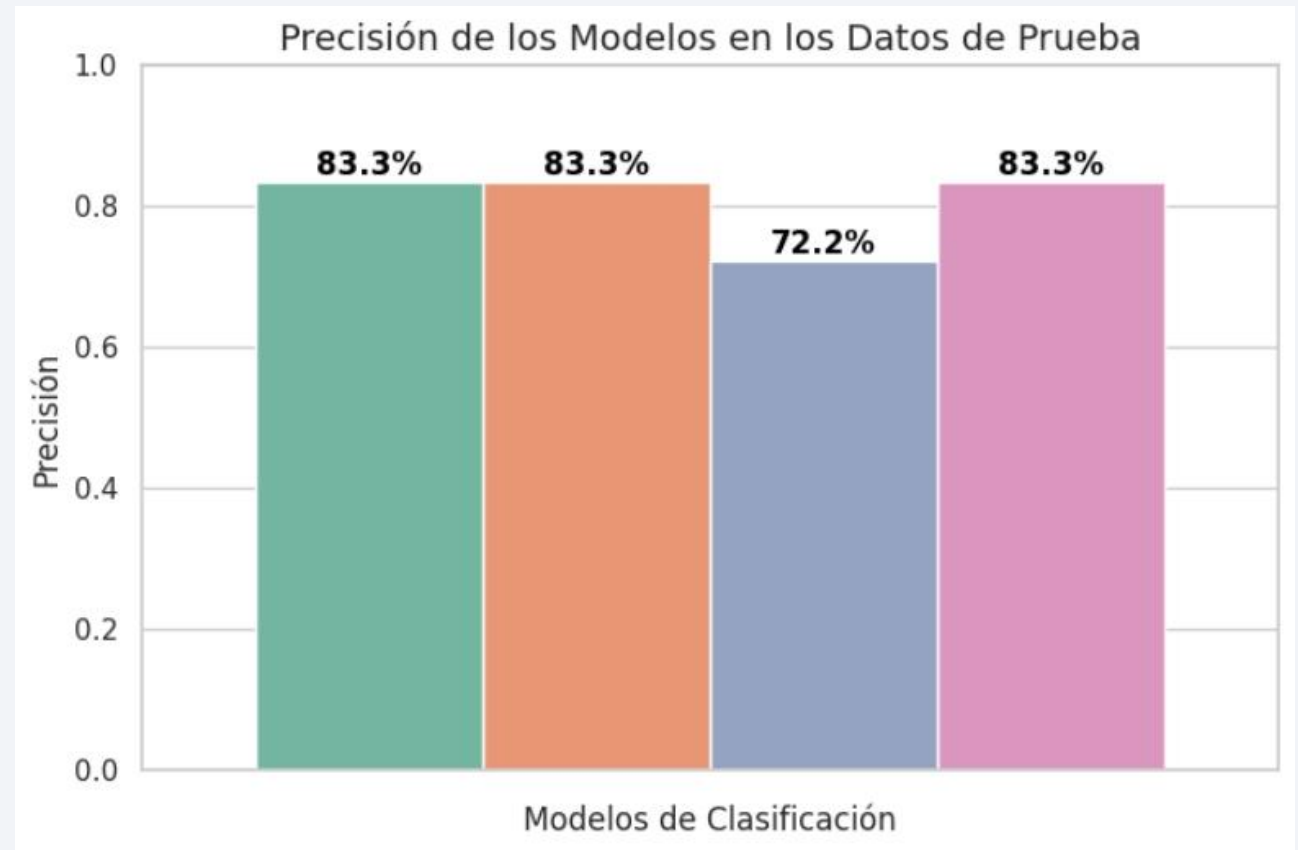
# Predictive Analysis (Classification)

# Precisión en la Clasificación

- Como podemos ver en este gráfico de barras, los siguientes modelos tienen la mayor precisión de clasificación en los datos de prueba:
- Regresión Logística,
- SVM,
- K-NN.

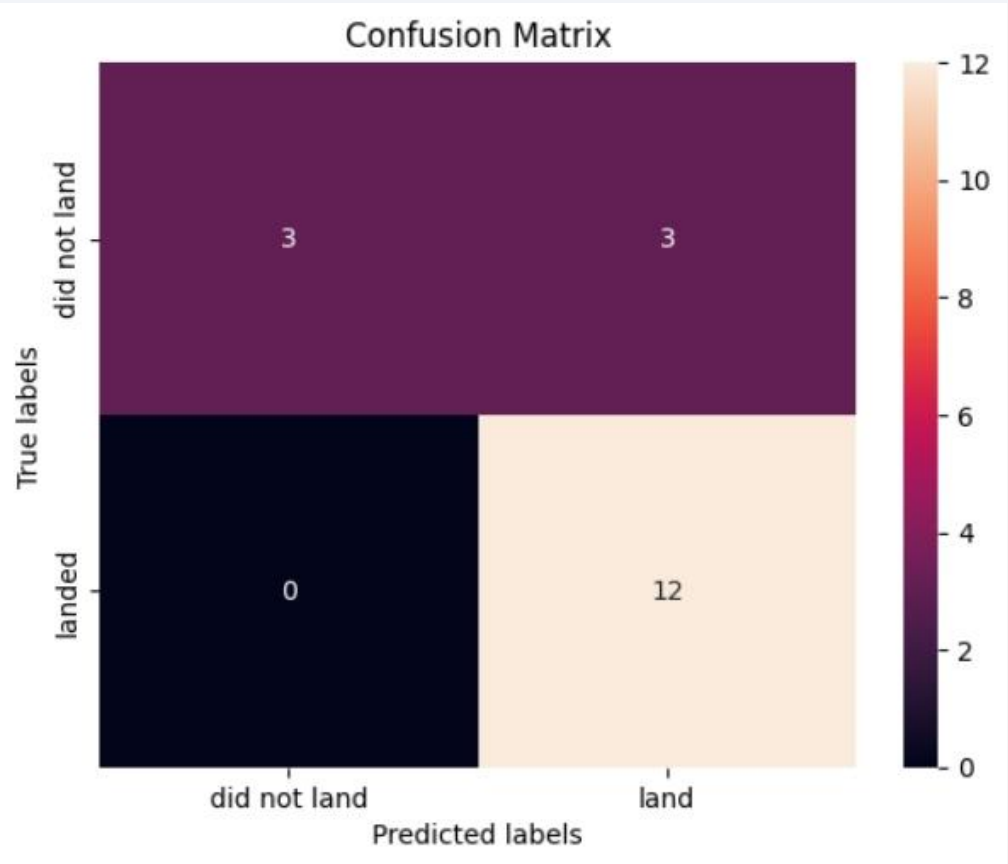
Todos con una precisión del 83.3% en los datos de prueba.

- El modelo que menos rindió (72.2%) fue el de árboles de decisión.







# Matriz de Confusión

- Podemos decir, por ejemplo, que el modelo de Regresión Logística es el mejor modelo en cuanto a desempeño. A continuación se muestra su matriz de confusión.
- Como podemos ver, el modelo puede distinguir bien entre clases.
- Vemos que el problema son los falsos positivos.



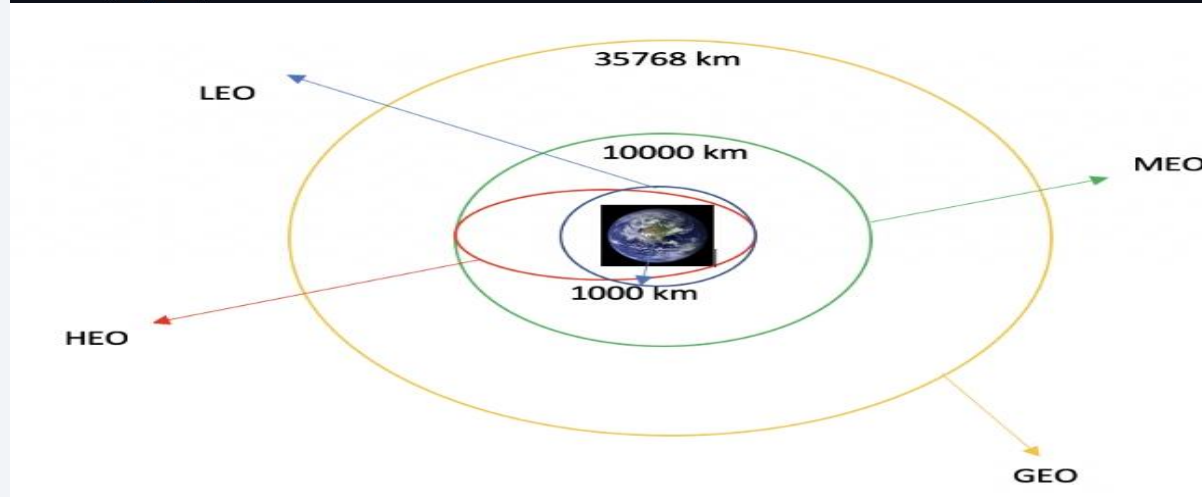
# Conclusiones

---

-  **El modelo de Regresión Logística** demostró la mayor precisión, alcanzando un **83.3%** en los datos de prueba.
-  **Factores críticos como la carga útil y el sitio de lanzamiento** impactan significativamente en los resultados del aterrizaje.
-  **La optimización de hiperparámetros** mejoró el rendimiento de los modelos de clasificación utilizados.
-  **El análisis de datos y la visualización** facilitaron la comprensión de los resultados para la toma de decisiones basada en datos.

# Apéndice

```
98 output(component_id='success-payload-scatter-chart', component_property='figure'),
99 [Input(component_id='site-dropdown', component_property='value'),
100  Input(component_id='payload-slider', component_property='value')]
101 )
102 def update_scatter_chart(selected_site, payload_range):
103     # Filtrar el dataframe según el rango de carga útil
104     filtered_df = spacex_df[(spacex_df['Payload Mass (kg)'] >= payload_range[0]) & (spacex_df['Payload Mass (kg)'] <= payload_range[1])]
105
106     # Si se seleccionan todos los sitios
107     if selected_site == 'ALL':
108         fig = px.scatter(
109             filtered_df,
110             x='Payload Mass (kg)',
111             y='class',
112             color='Booster Version Category',
113             title='Correlación entre la Carga Útil y el Éxito de Todos los Sitios',
114             labels={'class': 'Mission Outcome'})
115     else:
116         # Filtrar por el sitio seleccionado
117         site_filtered_df = filtered_df[filtered_df['Launch Site'] == selected_site]
118         # Crear el gráfico de dispersión con express de Plotly
119         fig = px.scatter(
120             site_filtered_df,
121             x='Payload Mass (kg)',
122             y='class',
123             color='Booster Version Category',
124             title=f'Correlación entre la Carga Útil y el Éxito para {selected_site}',
125             labels={'class': 'Mission Outcome'})
126     return fig
127
128
129 # Run the app
130 if __name__ == '__main__':
131     app.run_server()
```



- La primera imagen muestra el código de la aplicación Dash mencionada en este informe.
- La segunda imagen nos muestra las órbitas de las que se habla en la sección 'Insights drawn from EDA'.
- Este enlace lo redirigirá al repositorio donde se encuentran los datos utilizados para el análisis de este informe. [Click aquí.](#)



Thank you!

