

Winning Space Race with Data Science

<Name>
<Date>



Outline

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

Executive Summary

Este proyecto tuvo como propósito explorar los lanzamientos de SpaceX mediante la integración de diferentes fuentes de datos, entre ellas API REST, procesos de web scraping y consultas SQL. A partir de estos datos, se desarrollaron procesos de limpieza y preparación que permitieron contar con un conjunto de información depurado y confiable. Posteriormente, se aplicaron técnicas de análisis exploratorio, visualización interactiva y construcción de dashboards dinámicos. Finalmente, se desarrolló un modelo predictivo basado en clasificación para anticipar la probabilidad de éxito en los aterrizajes. Todo este recorrido permitió comprender los factores más relevantes que inciden en los resultados de los lanzamientos, ofreciendo una base sólida para la toma de decisiones estratégicas.

Introduction

SpaceX, como empresa pionera en la reutilización de cohetes, ha transformado la industria espacial con el objetivo de optimizar los costos y mejorar la eficiencia de los lanzamientos. En este contexto surge la pregunta central: ¿qué factores determinan el éxito o fracaso de un aterrizaje? Para responderla se definieron tres interrogantes principales. En primer lugar, se buscó identificar cómo influyen el número de vuelos realizados y la carga útil en la probabilidad de éxito. En segundo lugar, se analizaron las diferencias en el desempeño según el sitio de lanzamiento y el tipo de órbita empleada. Finalmente, se exploró la posibilidad de construir un modelo predictivo capaz de anticipar los resultados de los lanzamientos con un grado de confianza adecuado.

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

- El proceso de recolección de datos se llevó a cabo en tres fases principales. En primer lugar, se utilizaron llamadas a la API REST de SpaceX para obtener información histórica de cada lanzamiento, incluyendo detalles como la fecha, el sitio, la carga útil, el tipo de órbita y el resultado del aterrizaje. En segundo lugar, se complementó esta información con técnicas de web scraping, particularmente en Wikipedia, para extraer registros asociados a las versiones de los cohetes y datos de apoyo. Finalmente, se aplicaron técnicas de data wrangling que permitieron limpiar, normalizar y consolidar toda la información en un único conjunto de datos apto para el análisis posterior.

From the `payload` we would like to learn the mass of the payload and the orbit that it is going to.

```
1 # Takes the dataset and uses the payloads column to call the API and append the data to the lists
2 def getPayloadData(data):
3     for load in data['payloads']:
4         if load:
5             response = requests.get("https://api.spacexdata.com/v4/payloads/"+load).json()
6             PayloadMass.append(response['mass_kg'])
7             Orbit.append(response['orbit'])
```

From the `rocket` column we would like to learn the booster name.

```
1 # Takes the dataset and uses the rocket column to call the API and append the data to the list
2 def getBoosterVersion(data):
3     for x in data['rocket']:
4         if x:
5             response = requests.get("https://api.spacexdata.com/v4/rockets/"+str(x)).json()
6             BoosterVersion.append(response['name'])
```

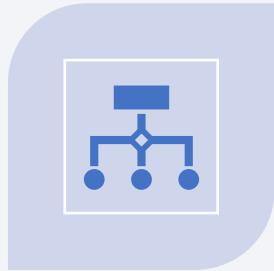
From the `launchpad` we would like to know the name of the launch site being used, the longitude, and the latitude.

```
1 # Takes the dataset and uses the launchpad column to call the API and append the data to the list
2 def getLaunchSite(data):
3     for x in data['launchpad']:
4         if x:
5             response = requests.get("https://api.spacexdata.com/v4/launchpads/"+str(x)).json()
6             Longitude.append(response['longitude'])
7             Latitude.append(response['latitude'])
8             LaunchSite.append(response['name'])
```

From cores we would like to learn the outcome of the landing, the type of the landing, number of flights with th 6 17 142 1 ^ v gridfins were used, wheter the core is reused, wheter legs were used, the landing pad used, the block of the core which is a number used to seperate version of cores, the number of times this specific core has been reused, and the serial of the core.

```
1 # Takes the dataset and uses the cores column to call the API and append the data to the lists
2 def getCoreData(data):
3     for core in data['cores']:
4         if core['core'] != None:
5             response = requests.get("https://api.spacexdata.com/v4/cores/"+core['core']).json()
6             Block.append(response['block'])
7             ReusedCount.append(response['reuse_count'])
8             Serial.append(response['serial'])
9         else:
10             Block.append(None)
11             ReusedCount.append(None)
12             Serial.append(None)
13             Outcome.append(str(core['landing_success'])+' '+str(core['landing_type']))
14             Flights.append(core['flight'])
15             GridFins.append(core['gridfins'])
16             Reused.append(core['reused'])
17             Legs.append(core['legs'])
18             LandingPad.append(core['Landpad'])
```

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

From cores we would like to learn the outcome of the landing, the type of the landing, number of flights with th 6 17 142 1 ^ v gridfins were used, wheter the core is reused, wheter legs were used, the landing pad used, the block of the core which is a number used to seperate version of cores, the number of times this specific core has been reused, and the serial of the core.

```
1 # Takes the dataset and uses the cores column to call the API and append the data to the lists
2 def getCoreData(data):
3     for core in data['cores']:
4         if core['core'] != None:
5             response = requests.get("https://api.spacexdata.com/v4/cores/"+core['core']).json()
6             Block.append(response['block'])
7             ReusedCount.append(response['reuse_count'])
8             Serial.append(response['serial'])
9         else:
10            Block.append(None)
11            ReusedCount.append(None)
12            Serial.append(None)
13            Outcome.append(str(core['landing_success'])+ ' '+str(core['landing_type']))
14            Flights.append(core['flight'])
15            GridFins.append(core['gridfins'])
16            Reused.append(core['reused'])
17            Legs.append(core['legs'])
18            LandingPad.append(core['landpad'])
```

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

```
Create a BeautifulSoup object from the HTML response

1 # Use BeautifulSoup() to create a BeautifulSoup object from a response text content
2 from bs4 import BeautifulSoup
3
4 # Crear un objeto BeautifulSoup a partir del contenido HTML
5 soup = BeautifulSoup(html_content, "html.parser")
6
7 # Mostrar una vista previa del HTML analizado
8 print(soup.prettify()[:1000]) # Muestra los primeros 1000 caracteres para verificar que se cargó correctamente

<!DOCTYPE html>
<html class="client-nojs vector-feature-language-in-header-enabled
vector-feature-language-in-main-page-header-disabled vector-feature-page-tools-pinned-disabled
vector-feature-toc-pinned-clientpref-1 vector-feature-main-menu-pinned-disabled
vector-feature-limited-width-clientpref-1 vector-feature-limited-width-content-enabled
vector-feature-custom-font-size-clientpref-1 vector-feature-appearance-pinned-clientpref-1
vector-feature-night-mode-enabled skin-theme-clientpref-day vector-sticky-header-enabled vector-toc-available"
dir="ltr" lang="en">
<head>
<meta charset="utf-8"/>
<title>
List of Falcon 9 and Falcon Heavy launches - Wikipedia
Print the page title to verify if the BeautifulSoup object was created properly
```

```
6
7     except requests.HTTPError as e:
8         print(f"Error HTTP: {e.response.status_code}")
9
10    except requests.RequestException as e:
11        print(f"Error de red: {e}")
12
13    print(response.content)

/03/spacex-successfully-completes-first-mission-geostationary-transfer-orbit">the original</a>
2019<span class="reference-accessdate">. Retrieved <span class="nowrap">25 November</span> 20
.</cite><span title="ctx_ver=Z39.88-2004&rft_val_fmt=info%3Aofi%2Ffmt%3Akev%3Abook&
.genre=unknown&rft.btitle=SpaceX+Successfully+Completes+First+Mission+to+Geostationary+Tr
rft.pub=SpaceX& rft.date=2013-12-03& rft_id=https%3A%2Fwww.spacex
.com%2Fpress%2F2013%2F12%2F03%2Fspacex-successfully-completes-first-mission-geostationary-tr
rfr_id=info%3Asid%2Fen.wikipedia.org%3Alist+of+Falcon+9+and+Falcon+Heavy+launches"
class="Z3988"></span></span><n><li id="cite_note-spx-pr-42"><span class="mw-cite-backli
href="#cite_ref-spx-pr_42-0"><sup><i><b>a</b></i></sup></a> <a
href="#cite_ref-spx-pr_42-1"><sup><i><b>b</b></i></sup></a></span> <span class="reference-text
rel="mw-deduplicated-inline-style" href="mw-data:TemplateStyles:r1238218222" /><cite
id="CITEREFBrostFeltes2011" class="citation_bibliography cs1">Brost, Kirstin; Feltes, Yves (1
```

Create a BeautifulSoup object from the HTML response

```
1 # Use BeautifulSoup() to create a BeautifulSoup object from a response text content
2 from bs4 import BeautifulSoup
3
4 # Crear un objeto BeautifulSoup a partir del contenido HTML
5 soup = BeautifulSoup(html_content, "html.parser")
6
7 # Mostrar una vista previa del HTML analizado
8 print(soup.prettify()[:1000]) # Muestra los primeros 1000 caracteres para verificar que se cargó correctamente

<!DOCTYPE html>
<html class="client-nojs vector-feature-language-in-header-enabled
vector-feature-language-in-main-page-header-disabled vector-feature-page-tools-pinned-disabled
vector-feature-toc-pinned-clientpref-1 vector-feature-main-menu-pinned-disabled
vector-feature-limited-width-clientpref-1 vector-feature-limited-width-content-enabled
vector-feature-custom-font-size-clientpref-1 vector-feature-appearance-pinned-clientpref-1
vector-feature-night-mode-enabled skin-theme-clientpref-day vector-sticky-header-enabled vector-toc-available"
dir="ltr" lang="en">
<head>
<meta charset="utf-8"/>
<title>
List of Falcon 9 and Falcon Heavy Launches - Wikipedia
```

Print the page title to verify if the BeautifulSoup object was created properly

Data Wrangling

- Durante la fase de preparación de datos se identificaron valores nulos y atípicos, los cuales fueron tratados mediante imputaciones apropiadas o eliminaciones cuando resultaban irrelevantes para el análisis. Se convirtieron variables categóricas en representaciones numéricas mediante variables dummy, facilitando su procesamiento en modelos estadísticos y predictivos. Asimismo, se crearon columnas derivadas como la variable de clase, que indica el éxito o fracaso del aterrizaje, y una categorización de la masa de carga útil en distintos rangos. Como resultado de este proceso, se obtuvo un conjunto de datos final compuesto por más de 90 registros válidos de lanzamientos, lo que permitió realizar un análisis consistente y confiable.

```
> 1 # landing_outcomes = values on Outcome column
 2 # landing_outcomes = values on Outcome column
 3 landing_outcomes = df['Outcome'].dropna().value_counts()
 4 landing_outcomes
 5

  Outcome
True ASDS      41
None None      19
True RTLS       14
False ASDS      6
True Ocean      5
False Ocean     2
None ASDS       2
False RTLS       1
Name: count, dtype: int64
```

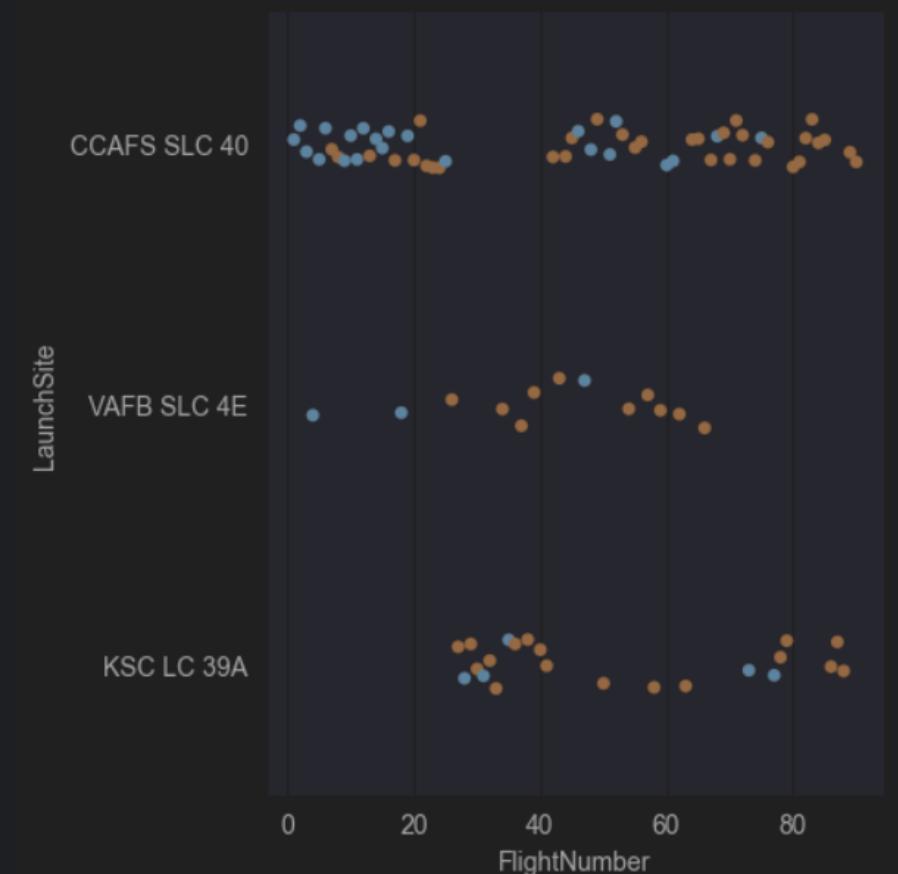
```
1 df.isnull().sum()/len(df)*100
```

Date	0.000000
BoosterVersion	0.000000
PayloadMass	0.000000
Orbit	0.000000
LaunchSite	0.000000
Outcome	0.000000
Flights	0.000000
GridFins	0.000000
Reused	0.000000
Legs	0.000000
LandingPad	28.888889
Block	0.000000

EDA with Data Visualization

- El análisis exploratorio mediante visualizaciones permitió descubrir patrones significativos en los datos. Al relacionar el número de vuelos con el sitio de lanzamiento, se observó que la experiencia acumulada en cada lugar incrementa la probabilidad de éxito. Al analizar la relación entre la carga útil y el sitio de lanzamiento, se identificó que las cargas intermedias presentan mejores resultados en comparación con cargas extremadamente bajas o altas. En el estudio de las tasas de éxito por tipo de órbita, se evidenció que las órbitas LEO y GTO mostraron mayores probabilidades de éxito. Finalmente, al revisar la tendencia de éxito anual, se constató una mejora sostenida a partir del año 2013, alcanzando porcentajes superiores al 80% en los últimos periodos analizados.

```
# Plot a scatter point chart with x axis to be Flight Number and y axis  
# class value  
# Crear el gráfico de dispersión con catplot  
sns.catplot(x="FlightNumber", y="LaunchSite", hue="Class", data=df)  
  
plt.show()
```



Tendencia de Éxito en los Lanzamientos por Año





EDA with SQL

- El análisis con SQL permitió responder a preguntas clave relacionadas con la operación de SpaceX. Se identificaron todos los sitios de lanzamiento únicos y se filtraron aquellos que comenzaban con la sigla CCA, lo que facilitó un enfoque más preciso sobre ubicaciones específicas. Además, se calcularon los volúmenes totales de carga útil transportada por los boosters en misiones de la NASA y se determinó la fecha del primer aterrizaje exitoso en tierra firme. Asimismo, se generó un ranking de resultados de aterrizaje entre los años 2010 y 2017, lo que permitió visualizar la evolución en el desempeño. Este enfoque demostró cómo SQL no solo facilita consultas directas, sino que también aporta respuestas estratégicas que complementan los análisis visuales.

```
1 # Function to assign color to launch outcome
2 def assign_marker_color(launch_outcome):
3     if launch_outcome == 1:
4         return 'green'
5     else:
6         return 'red'
7
8 spacex_df['marker_color'] = spacex_df['class'].apply(assign_marker_color)
9 spacex_df.tail(10)
```

10 rows ▾ 10 rows × 5 cols

	Launch Site	Lat	Long	class	marker_color
46	KSC LC-39A	28.573255	-80.646895	1	green
47	KSC LC-39A	28.573255	-80.646895	1	green
48	KSC LC-39A	28.573255	-80.646895	1	green
49	CCAFS SLC-40	28.563197	-80.576820	1	green
50	CCAFS SLC-40	28.563197	-80.576820	1	green
51	CCAFS SLC-40	28.563197	-80.576820	0	red
52	CCAFS SLC-40	28.563197	-80.576820	0	red
53	CCAFS SLC-40	28.563197	-80.576820	0	red
54	CCAFS SLC-40	28.563197	-80.576820	1	green
55	CCAFS SLC-40	28.563197	-80.576820	0	red

```
1 %sql SELECT DISTINCT "Launch_Site" FROM SPACEXTBL;  
2  
3
```

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

Launch_Site

CCAFS LC-40

VAFB SLC-4E

KSC LC-39A

CCAFS SLC-40

```
%sql SELECT * FROM SPACEXTBL WHERE "Launch_Site" LIKE 'CCA%' LIMIT 5;
```

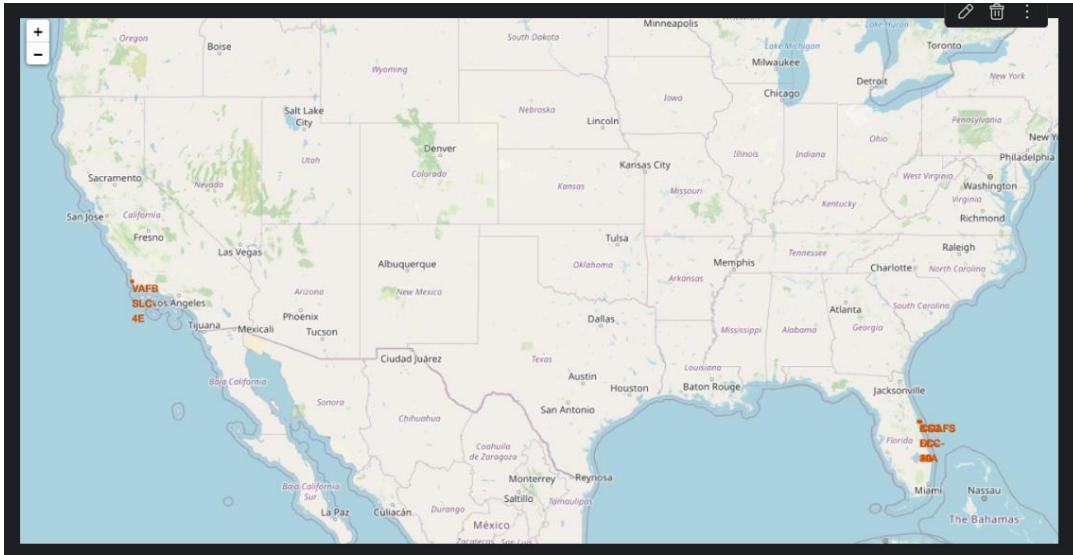
```
* sqlite:///my_data1.db  
Done.
```

Date	Time (UTC)	Booster_Version	Launch_Site	Payload	PAYLOAD_MASS_KG_	Orbit	Customer	Mission_Outcome	Landing_Outcome
2010-06-04	18:45:00	F9 v1.0 B0003	CCAFS LC-40	Dragon Spacecraft Qualification Unit	0	LEO	SpaceX	Success	Failure (parachute)
2010-12-08	15:43:00	F9 v1.0 B0004	CCAFS LC-40	Dragon demo flight C1, two CubeSats, barrel of Brouere cheese	0	LEO (ISS)	NASA (COTS) NRO	Success	Failure (parachute)
2012-05-22	7:44:00	F9 v1.0 B0005	CCAFS LC-40	Dragon demo flight C2	525	LEO (ISS)	NASA (COTS)	Success	No attempt
2012-10-08	0:35:00	F9 v1.0 B0006	CCAFS LC-40	SpaceX CRS-1	500	LEO (ISS)	NASA (CRS)	Success	No attempt
2013-03-01	15:10:00	F9 v1.0 B0007	CCAFS LC-40	SpaceX CRS-2	677	LEO (ISS)	NASA (CRS)	Success	No attempt



Build an Interactive Map with Folium

- Los mapas interactivos desarrollados con Folium aportaron una dimensión geográfica al análisis de los lanzamientos. En una primera visualización se representaron todos los sitios de lanzamiento de SpaceX a nivel global, lo que permitió comprender la distribución espacial de sus operaciones. En un segundo mapa, los resultados fueron codificados por colores según éxito o fracaso, lo que facilitó la identificación visual de patrones de desempeño asociados a la ubicación. Finalmente, un tercer mapa se centró en la proximidad de uno de los sitios respecto a infraestructuras críticas como carreteras, costas y ferrocarriles, evidenciando la importancia de la logística y la localización estratégica en el éxito de los lanzamientos.
- https://github.com/eddymorfeo/Lab_Modulo_10



Build a Dashboard with Plotly Dash

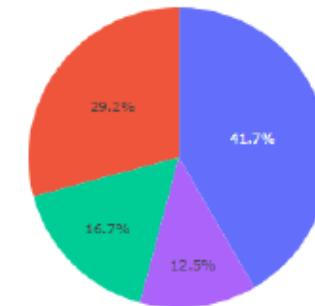


- La construcción de un dashboard interactivo con Plotly Dash permitió condensar información clave de manera dinámica. Una de las visualizaciones principales fue un gráfico de pastel que mostró la proporción de lanzamientos exitosos en cada sitio, ofreciendo un panorama inmediato del desempeño. Además, se incluyó una representación que relacionó la masa de la carga útil con el resultado del aterrizaje, evidenciando cómo los rangos intermedios de peso tendían a obtener mejores resultados. El uso de controles como rangos deslizantes y menús desplegables posibilitó a los usuarios interactuar con los datos y filtrar según sus necesidades, transformando el dashboard en una herramienta de exploración y análisis flexible.
- https://github.com/eddymorfeo/Lab_Modulo_10

SpaceX Launch Records Dashboard

All Sites

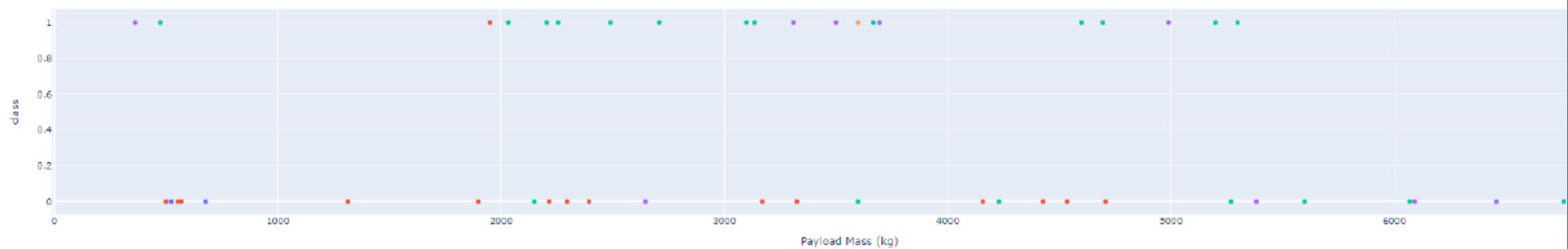
Total Success Launches By Site



Payload range (Kg):



Correlation between Payload and Success for all sites



Predictive Analysis (Classification)

- El análisis predictivo se centró en la implementación de modelos de clasificación para estimar la probabilidad de éxito en los aterrizajes. Se probaron distintos algoritmos, incluyendo regresión logística, máquinas de soporte vectorial, árboles de decisión y K-Nearest Neighbors. Mediante la técnica de GridSearchCV se optimizaron los hiperparámetros de cada modelo para mejorar su rendimiento. Los resultados mostraron que el árbol de decisión fue el modelo con mayor precisión, alcanzando aproximadamente un 83%. El análisis de la matriz de confusión permitió identificar los aciertos y errores más comunes, evidenciando un buen equilibrio en la clasificación de éxitos y fracasos. En conclusión, el modelo predictivo demostró ser una herramienta robusta y confiable para anticipar resultados futuros.
- https://github.com/eddymorfeo/Lab_Modulo_10

```

9 # Evaluar cada modelo en X_test
10 accuracy_results = {}
11 for model_name, model in models.items():
12     accuracy = model.score(X_test, Y_test)
13     accuracy_results[model_name] = accuracy
14
15 # Mostrar resultados
16 for model, acc in accuracy_results.items():
17     print(f"{model}: Precisión en datos de prueba = {acc:.4f}")
18
19 # Identificar el modelo con mejor precisión
20 best_model = max(accuracy_results, key=accuracy_results.get)
21 print(f"\nEl mejor modelo es: {best_model} con una precisión de {accuracy_results[best_model]:.4f}")

    Regresión Logística: Precisión en datos de prueba = 0.8333
    SVM: Precisión en datos de prueba = 0.8333
    Árbol de Decisión: Precisión en datos de prueba = 0.8333
    k-NN: Precisión en datos de prueba = 0.8333

    El mejor modelo es: Regresión Logística con una precisión de 0.8333

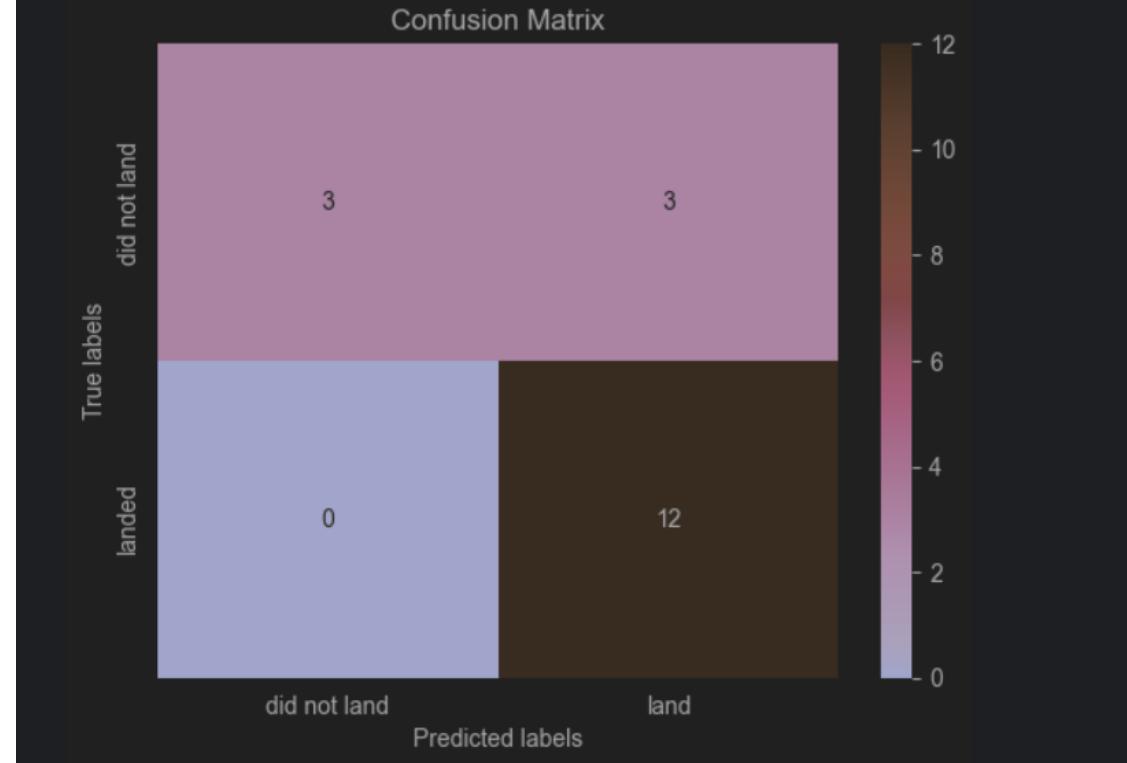
1 print("Número de registros en la muestra de prueba:", len(X_test))
    Número de registros en la muestra de prueba: 18

```

```

1 yhat=logreg_cv.predict(X_test)
2 plot_confusion_matrix(Y_test,yhat)

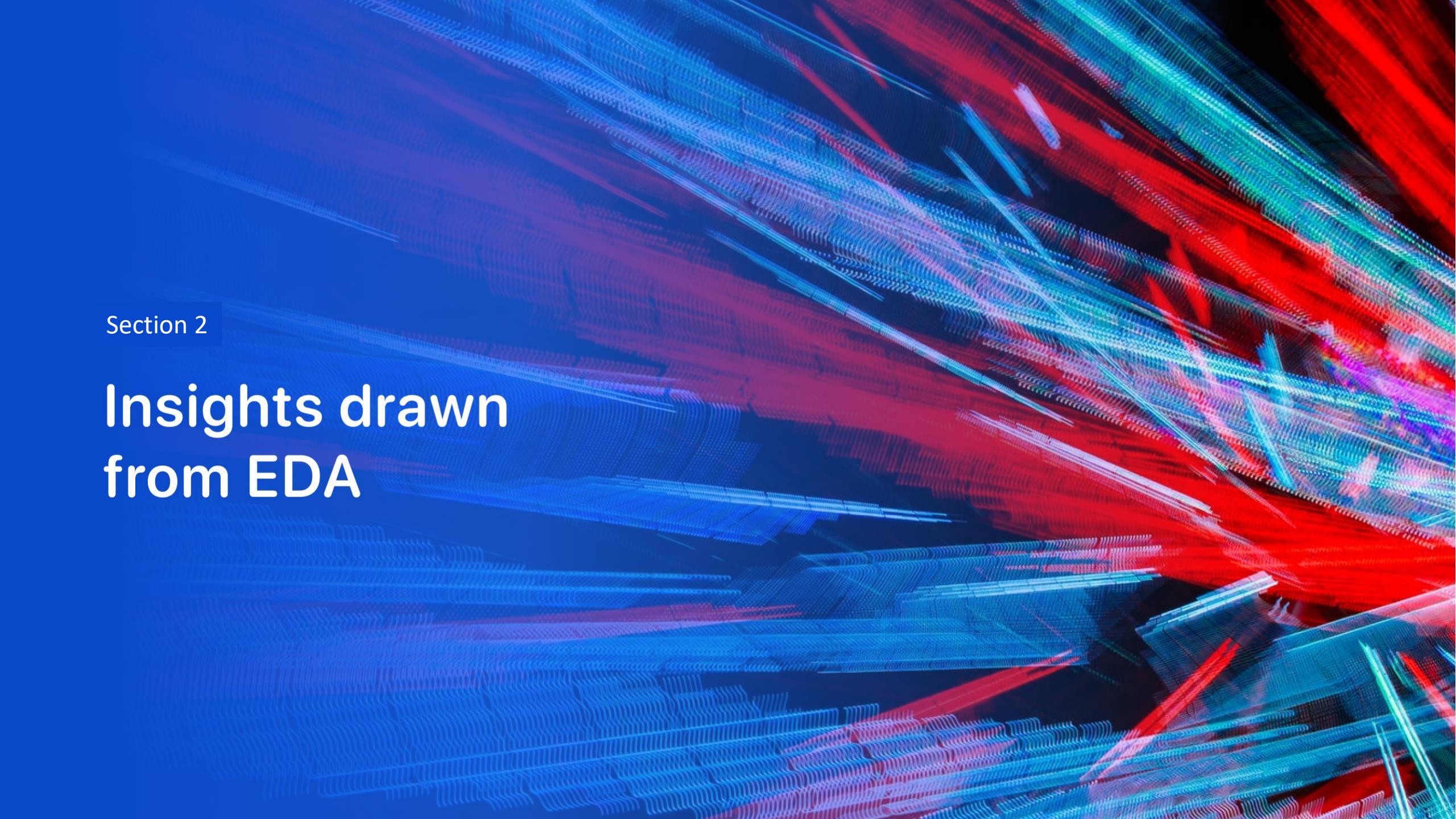
```





Results

- El análisis integral de los datos reveló tendencias claras y consistentes. Se comprobó que los lanzamientos acumulados en un sitio incrementan la experiencia y mejoran la tasa de éxito. Asimismo, los sitios de lanzamiento como KSC LC-39A y CCAFS SLC-40 registraron los mejores desempeños, consolidándose como ubicaciones clave para la empresa. En cuanto a la masa de las cargas útiles, aquellas comprendidas entre tres mil y seis mil kilogramos mostraron un balance óptimo entre riesgo y probabilidad de éxito. Finalmente, la tendencia histórica evidenció un progreso sostenido en los resultados, consolidando a SpaceX como un referente en la reutilización de cohetes y en la eficiencia de sus operaciones.

The background of the slide features a complex, abstract digital visualization. It consists of numerous thin, glowing lines that create a sense of depth and motion. The lines are primarily blue and red, with some green and purple highlights. They form a grid-like structure that curves and twists across the frame, resembling a three-dimensional space or a network of data points. The overall effect is futuristic and dynamic.

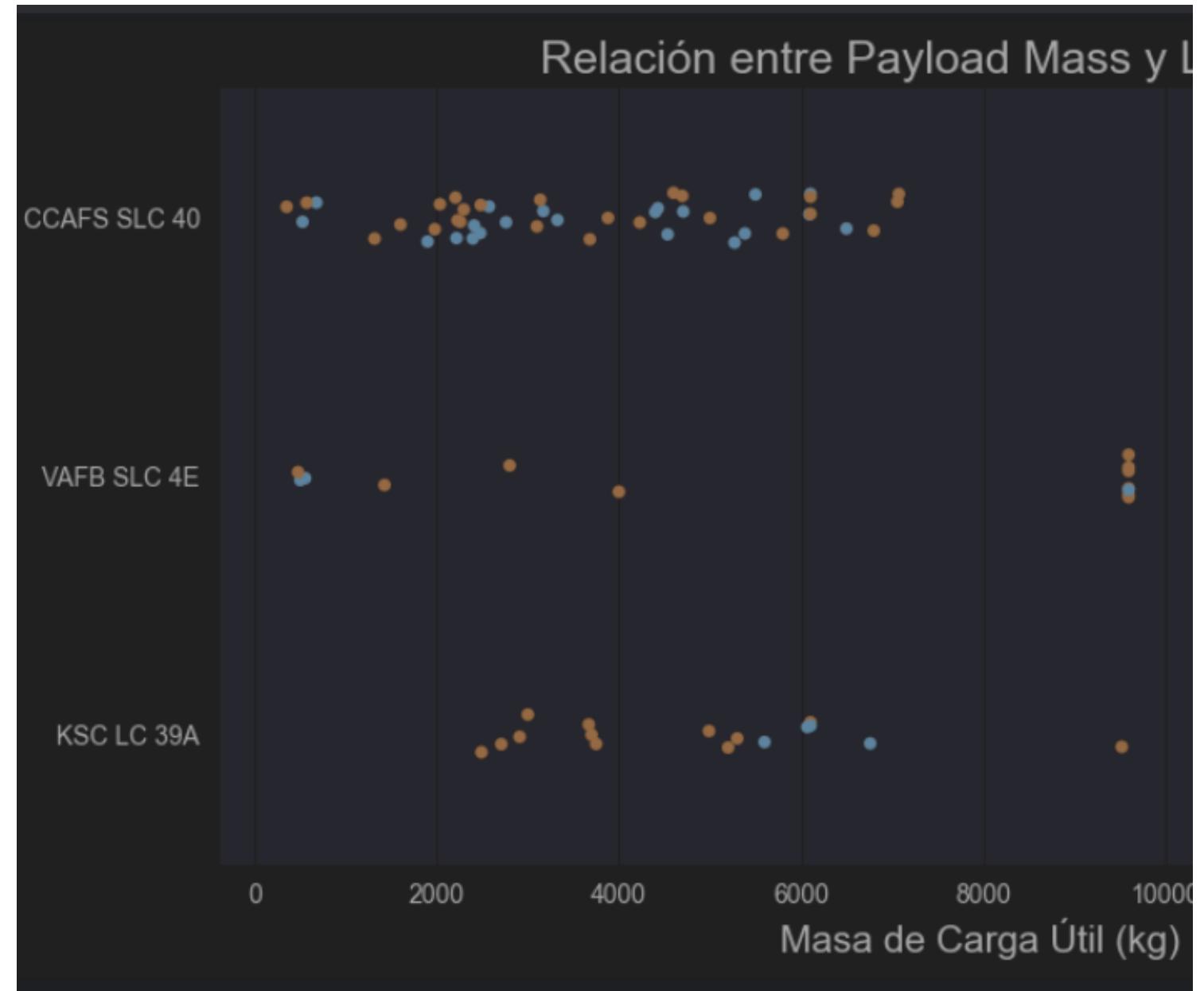
Section 2

Insights drawn from EDA

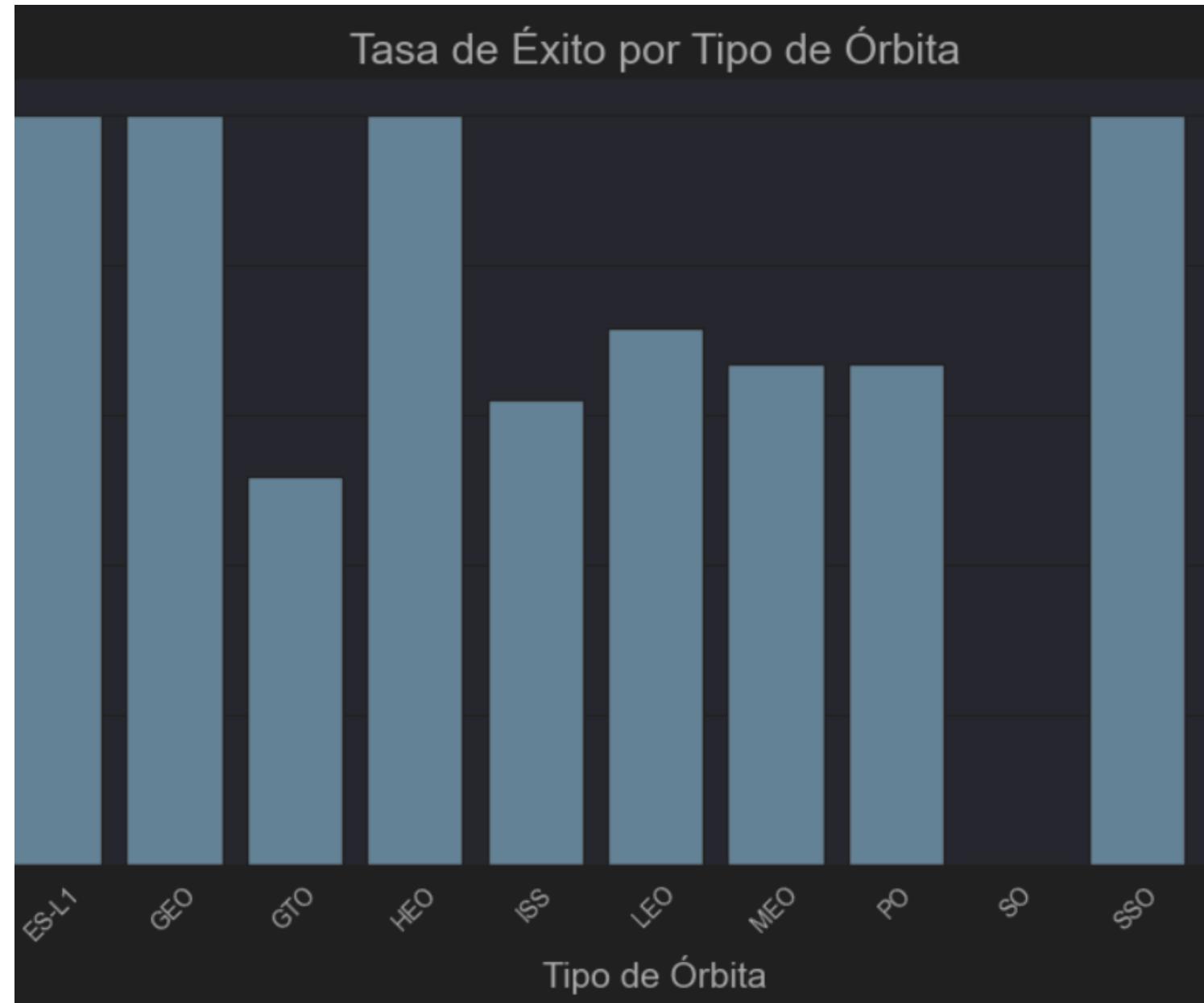
Flight Number vs. Launch Site



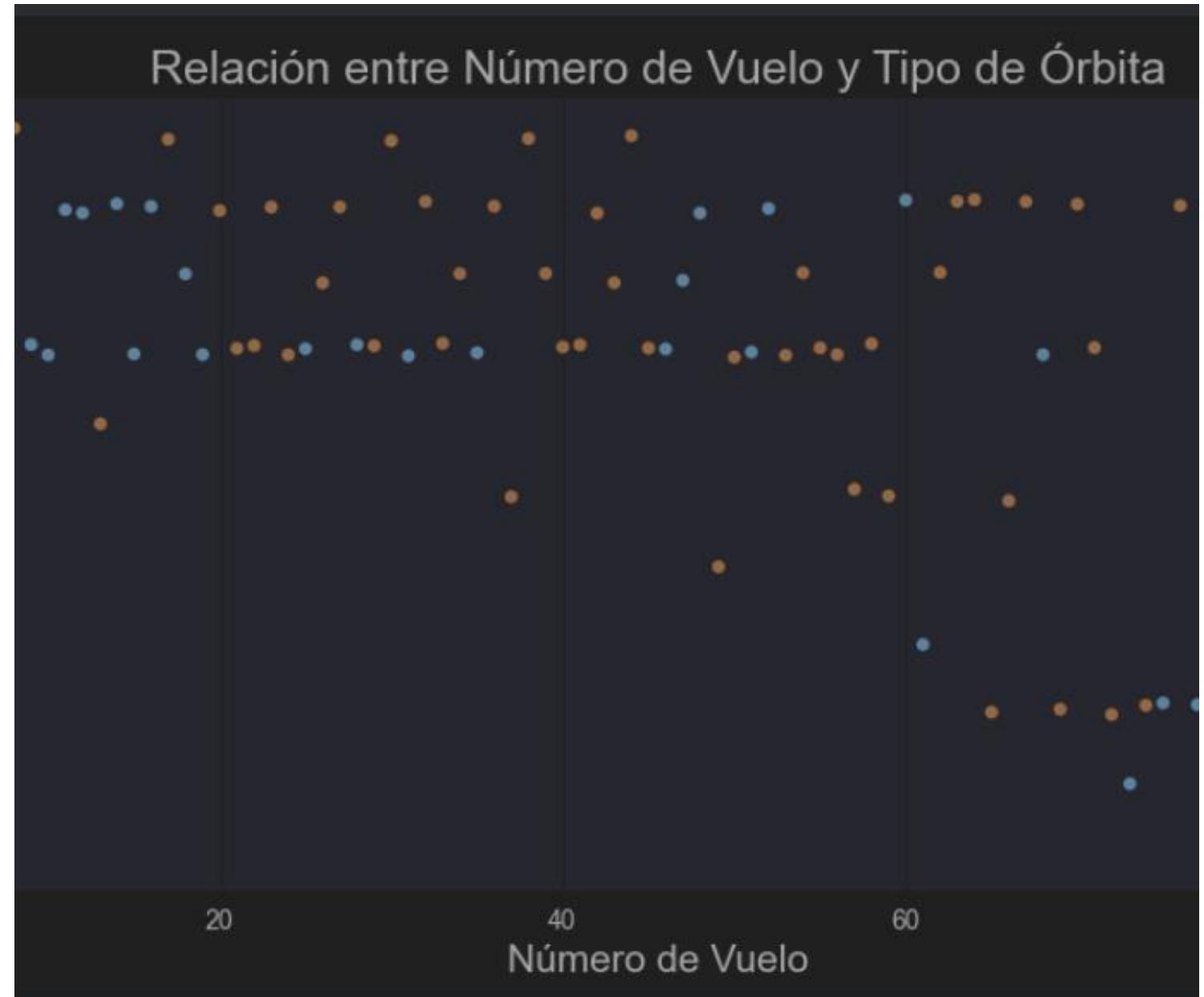
Payload vs. Launch Site



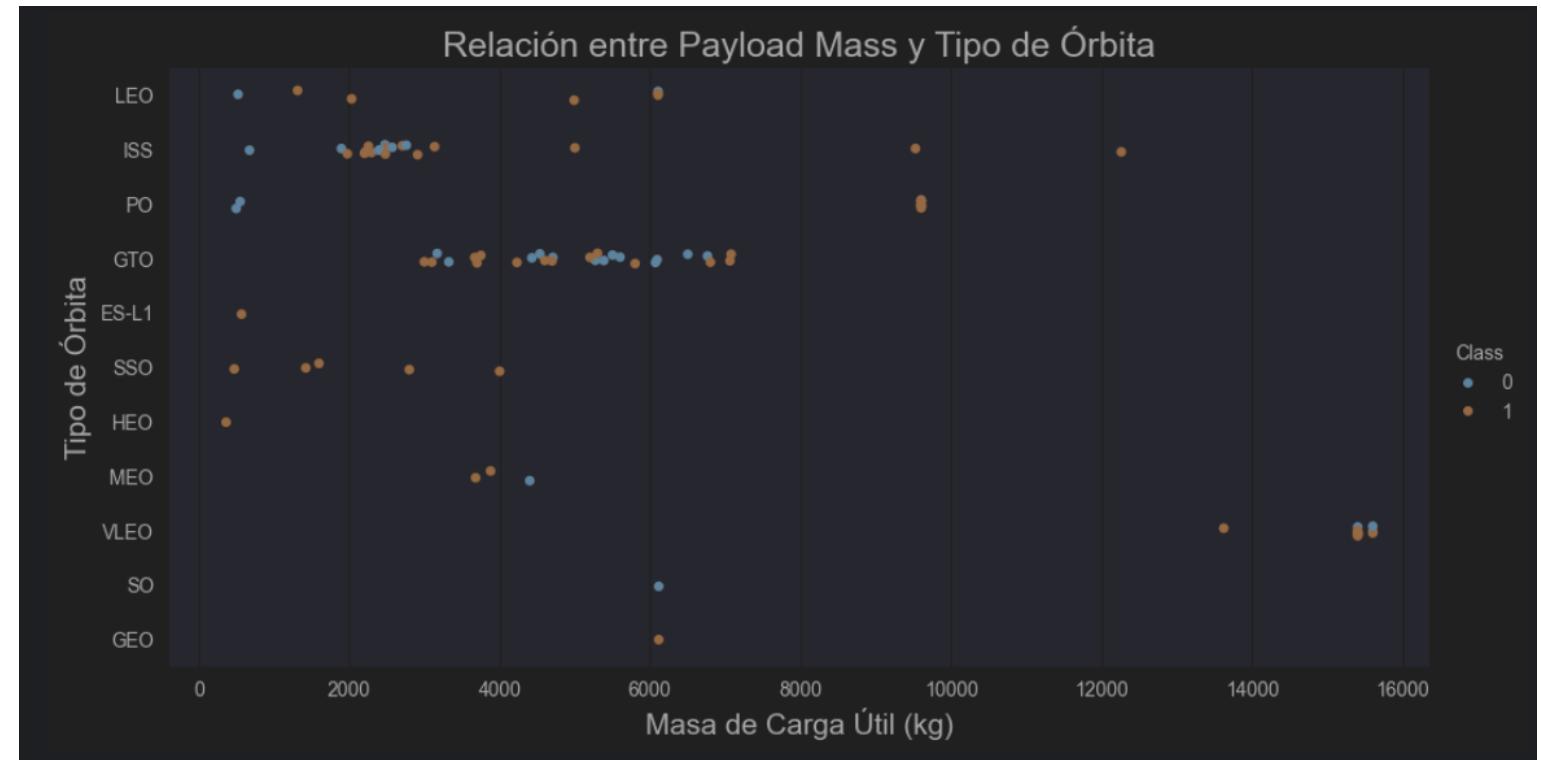
Success Rate vs. Orbit Type



Flight Number vs. Orbit Type



Payload vs. Orbit Type



Launch Success Yearly Trend



All Launch Site Names

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

```
1 %sql SELECT DISTINCT "Launch_Site" FROM SPACEXTBL;
```

```
2  
3
```

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

Launch_Site

CCAFS LC-40

VAFB SLC-4E

KSC LC-39A

CCAFS SLC-40

Launch Site Names Begin with 'CCA'

```
1 %sql SELECT * FROM SPACEXTBL WHERE "Launch_Site" LIKE 'CCA%' LIMIT 5;
2
3
```

* sqlite:///my_data1.db
Done.

Date	Time (UTC)	Booster_Version	Launch_Site	Payload	PAYLOAD_MASS_KG_	Orbit	Customer	Mission_Outcome	Landing_Outcome
2010-06-04	18:45:00	F9 v1.0 B0003	CCAFS LC-40	Dragon Spacecraft Qualification Unit	0	LEO	SpaceX	Success	Failure (parachute)
2010-12-08	15:43:00	F9 v1.0 B0004	CCAFS LC-40	Dragon demo flight C1, two CubeSats, barrel of Brouere cheese	0	LEO (ISS)	NASA (COTS) NRO	Success	Failure (parachute)
2012-05-22	7:44:00	F9 v1.0 B0005	CCAFS LC-40	Dragon demo flight C2	525	LEO (ISS)	NASA (COTS)	Success	No attempt
2012-10-08	0:35:00	F9 v1.0 B0006	CCAFS LC-40	SpaceX CRS-1	500	LEO (ISS)	NASA (CRS)	Success	No attempt
2013-03-01	15:10:00	F9 v1.0 B0007	CCAFS LC-40	SpaceX CRS-2	677	LEO (ISS)	NASA (CRS)	Success	No attempt

Total Payload Mass

```
1 %sql SELECT SUM("Payload_Mass__kg_") FROM SPACEXTBL WHERE "Customer" = 'NASA (CRS)';  
2  
* sqlite:///my_data1.db  
Done.  
  
SUM(Payload_Mass_kg_)  
45596
```

Average Payload Mass by F9 v1.1

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

```
1 %sql SELECT AVG("Payload_Mass__kg_") FROM SPACEXTBL WHERE "Booster_Version" = 'F9 v1.1';
2
* sqlite:///my_data1.db
Done.

AVG(Payload_Mass_kg_)
2928.4
```

First Successful Ground Landing Date

```
1 %sql SELECT MIN(Date) FROM SPACEXTBL WHERE "Landing_Outcome" = 'Success (ground pad)';  
2  
* sqlite:///my_data1.db  
Done.  
  
MIN(Date)  
2015-12-22
```

Successful Drone Ship Landing with Payload between 4000 and 6000

```
List the names of the boosters which have success in drone ship and have payload mass greater than 4000 but less than 6000

1 %sql SELECT "Booster_Version" FROM SPACEXTBL WHERE "Landing_Outcome" = 'Success (drone ship)' AND
  "Payload_Mass__kg_" > 4000 AND "Payload_Mass__kg_" < 6000;
2
  * sqlite:///my_data1.db
Done.

Booster_Version
F9 FT B1022
F9 FT B1026
F9 FT B1021.2
F9 FT B1031.2
```

Total Number of Successful and Failure Misso n Outcomes

List the total number of successful and failure mission outcomes

```
1 %sql SELECT "Landing_Outcome", COUNT(*) FROM SPACEXTBL GROUP BY "Landing_Outcome";  
2
```

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

Landing_Outcome	COUNT(*)
Controlled (ocean)	5
Failure	3
Failure (drone ship)	5
Failure (parachute)	2
No attempt	21
No attempt	1
Precluded (drone ship)	1
Success	38
Success (drone ship)	14
Success (ground pad)	9
Uncontrolled (ocean)	2

Boosters Carried Maximum Pa yload

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

1 %sql SELECT "Booster_Version" FROM SPACEXTBL WHERE "Payload_Mass__kg_" = (SELECT MAX("Payload_Mass__kg_") FROM SPACEXTBL);

2
* sqlite:///my_data1.db
Done.

Booster_Version
F9 B5 B1048.4
F9 B5 B1049.4
F9 B5 B1051.3
F9 B5 B1056.4
F9 B5 B1048.5
F9 B5 B1051.4
F9 B5 B1049.5
F9 B5 B1060.2
F9 B5 B1058.3
F9 B5 B1051.6
F9 B5 B1060.3
F9 B5 B1049.7
```

2015 Launch Records

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.

```
1 %sql SELECT SUBSTR(Date, 6, 2) AS Month,"Landing_Outcome","Booster_Version","Launch_Site" FROM SPACEXTBL WHERE  
SUBSTR(Date, 1, 4) = '2015' AND "Landing_Outcome" = 'Failure (drone ship)';  
2  
* sqlite:///my_data1.db  
Done.
```

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

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

Rank the count of landing outcomes (such as Failure (drone ship) or Success (ground pad)) between the date 2010-06-04 and 2017-03-20, in descending order.

```
1 %sql SELECT "Landing_Outcome", COUNT(*) FROM SPACEXTBL WHERE Date BETWEEN '2010-06-04' AND '2017-03-20' GROUP BY "Landing_Outcome" ORDER BY COUNT(*) DESC;
```

```
2
```

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

```
Done.
```

Landing_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

The background of the slide is a photograph taken from space at night. It shows the curvature of the Earth's horizon against a dark blue sky. Numerous glowing yellow and white points represent city lights, concentrated in coastal and urban areas. In the upper right quadrant, there are bright green and yellow bands of light, likely the Aurora Borealis or Australis. The overall atmosphere is dark and mysterious.

Section 3

Launch Sites Proximities Analysis

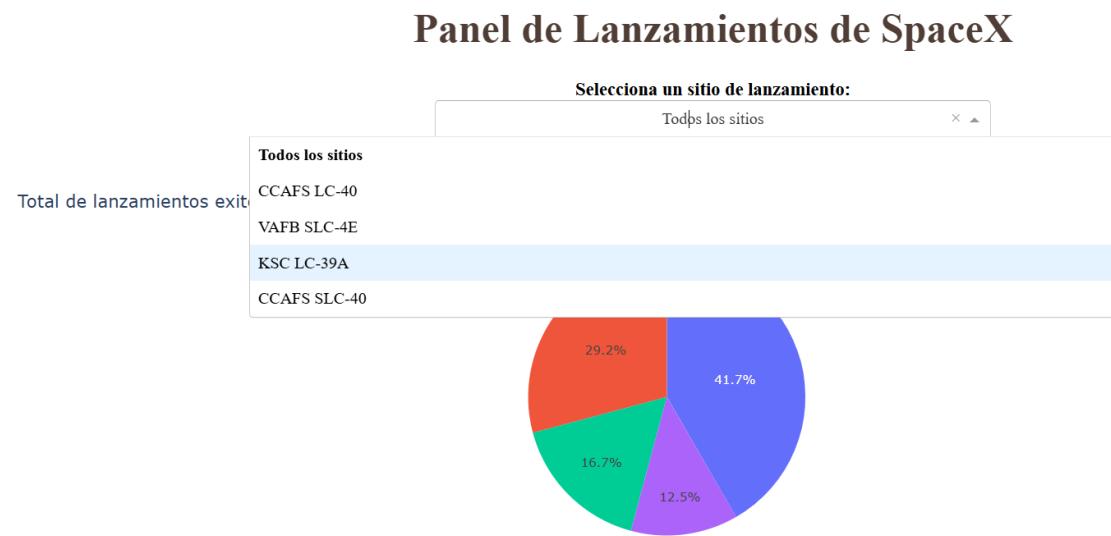
<Folium Map Screenshot 1>

```
# Definir la estructura de la aplicación con todas las tareas
app.layout = html.Div(children=[

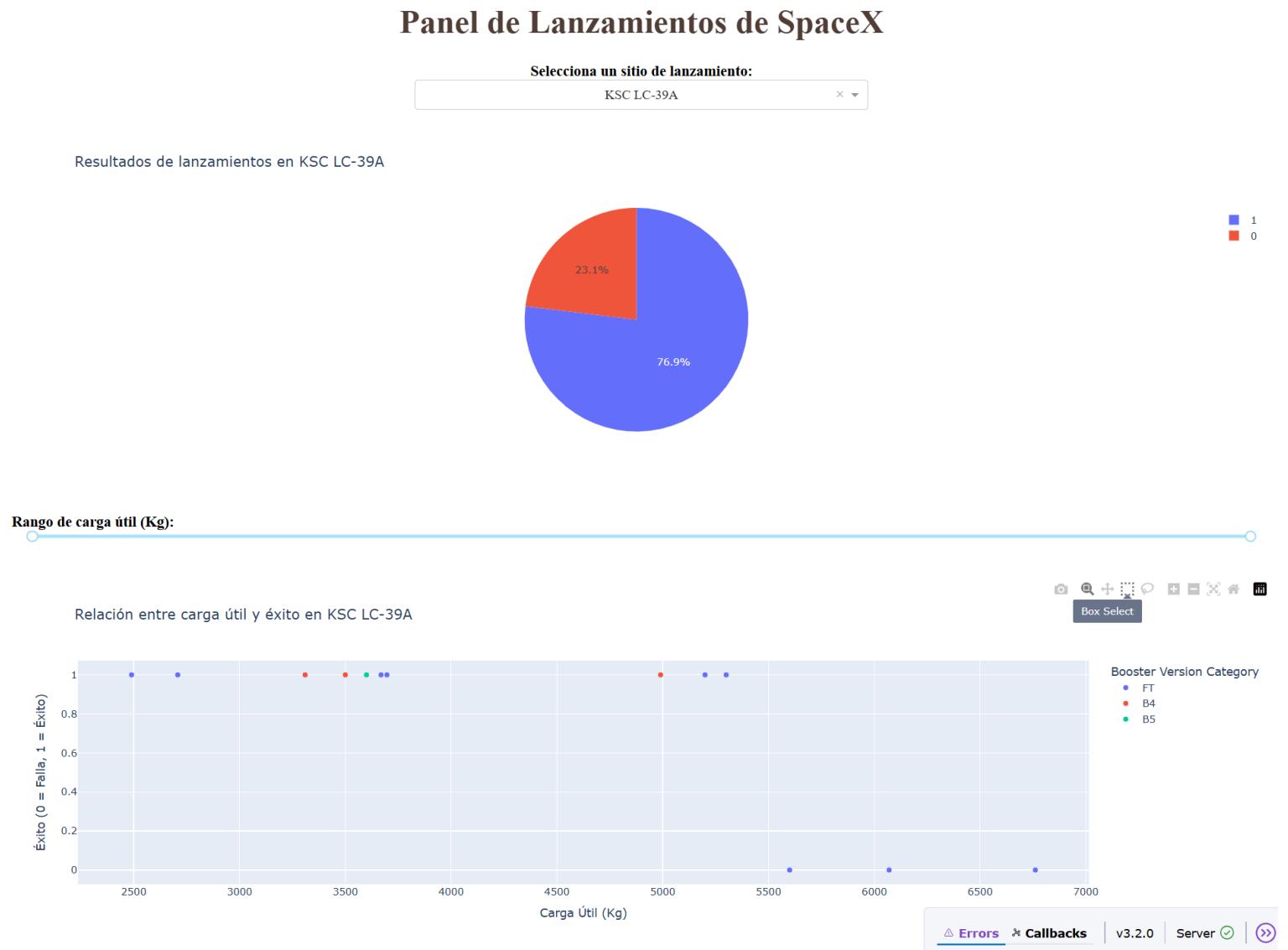
    html.H1( children: 'Panel de Lanzamientos de SpaceX',
             style={'textAlign': 'center', 'color': '#503D36', 'font-size': 40}),

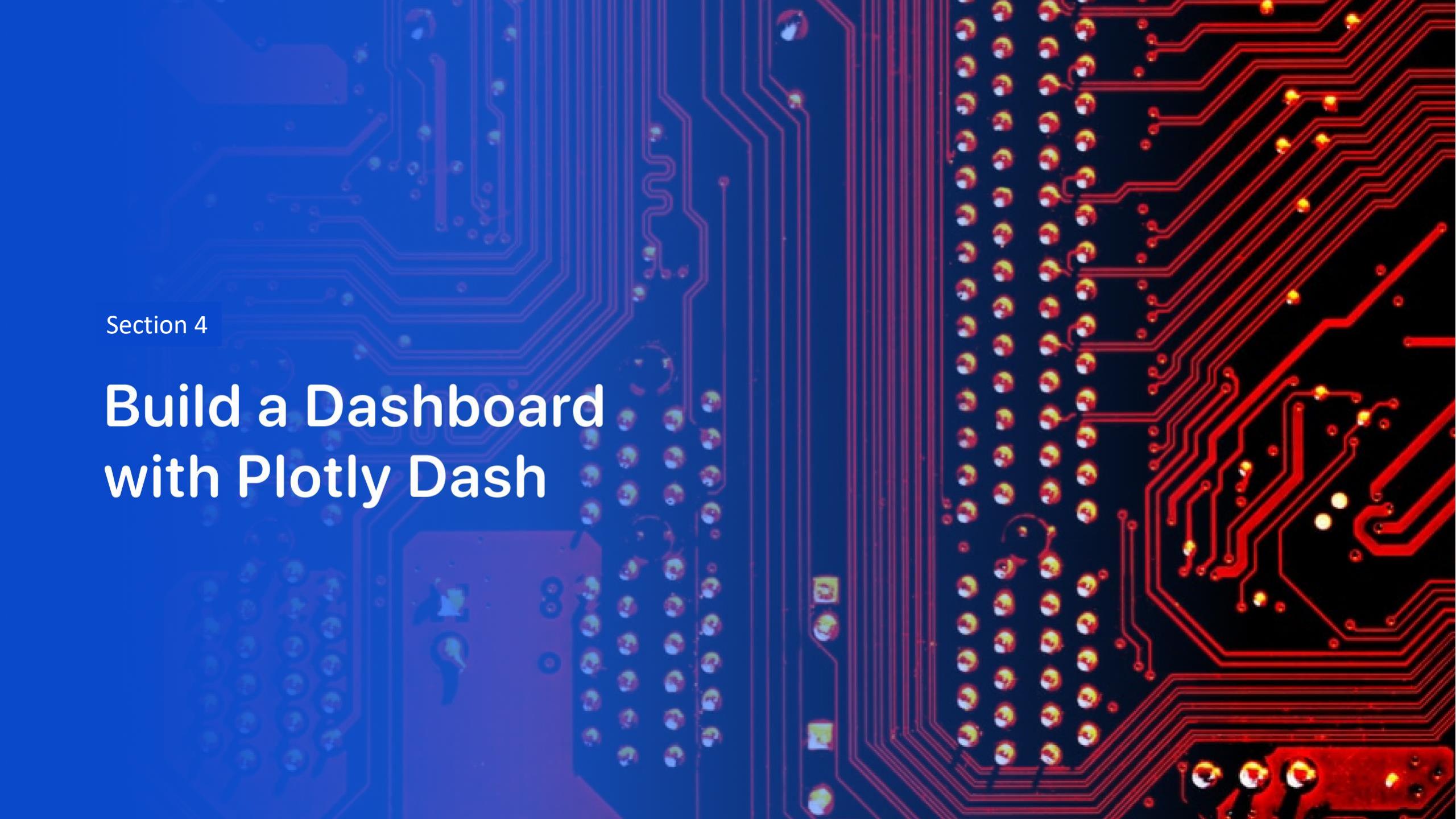
    # TAREA 1: Menú desplegable para seleccionar el sitio de lanzamiento
    html.Div( children: [
        html.Label( children: "Selecciona un sitio de lanzamiento:", style={'font-size': '18px', 'font-weight': 'bold'}),
        dcc.Dropdown(
            id='site-dropdown',
            options=[{'label': 'Todos los sitios', 'value': 'ALL'} + 
                     [{'label': site, 'value': site} for site in spacex_df['Launch_Site'].unique()],
            value='ALL',
            placeholder="Selecciona un sitio de lanzamiento",
            searchable=True,
            style={'width': '60%', 'margin': 'auto'}
        )
    ], style={'textAlign': 'center'})
```

<Folium Map Screenshot 2>



<Folium Map Screenshot 3>





Section 4

Build a Dashboard with Plotly Dash

The background of the slide features a dynamic, abstract design. It consists of several thick, curved lines that transition from a bright yellow at the top right to a deep blue at the bottom left. These lines create a sense of motion and depth, resembling a tunnel or a stylized landscape. The overall effect is modern and professional.

Section 5

Predictive Analysis (Classification)



Conclusions

- Los hallazgos permiten concluir que SpaceX ha mejorado de manera notable la tasa de éxito de sus lanzamientos en la última década, confirmando la efectividad de su estrategia de reutilización de cohetes. Se comprobó que factores como el sitio de lanzamiento, la carga útil y el tipo de órbita tienen una influencia directa en los resultados. Además, la construcción de un modelo predictivo validó la posibilidad de anticipar con un alto grado de confianza la probabilidad de éxito de futuros aterrizajes. En términos metodológicos, la combinación de API, web scraping, SQL, análisis exploratorio y aprendizaje automático demostró ser un enfoque integral y sólido para el estudio de datos espaciales.

Appendix

- El apéndice reúne los enlaces a los cuadernos de Jupyter desarrollados a lo largo del proyecto, incluyendo la recolección de datos mediante API, el web scraping, la preparación y limpieza de datos, el análisis exploratorio tanto visual como mediante SQL, la generación de mapas interactivos con Folium, el desarrollo del dashboard en Plotly Dash y la implementación del modelo predictivo. Asimismo, se listan los principales recursos tecnológicos utilizados, entre los que destacan Python, Pandas, Folium, Plotly, SQL y Scikit-learn, los cuales hicieron posible la construcción de un análisis completo y detallado del desempeño de SpaceX.



Thank you!

