



Applied Data Science Capstone

By Anderson Quiroz Gonzales

August 10, 2024

A large, dark blue rectangular image on the left side of the slide features the word "SPACEX" in white, sans-serif capital letters at the bottom. Above it, a white Boeing 747 aircraft is shown from a low angle, flying towards the right. The background is a dark, textured space.

Indice

- Resumen ejecutivo
- Introducción
- Metodología
- Resultados
- Conclusión

Resumen Ejecutivo

- Resumen de metodologías
 - Recopilación de datos con API de SpaceX
 - Recopilación de datos con Web Scraping
 - Data Wrangling
 - Análisis exploratorio de datos (EDA) con SQL
 - EDA con visualización de datos
 - Análisis visual interactivo con Folium y Dashboard
 - Análisis predictivo con aprendizaje automático
- Resumen de todos los resultados
 - Resultados del análisis exploratorio de datos
 - Demostración de análisis interactivo en capturas
 - Resultados del análisis predictivo





Introducción

Contexto: "SpaceX ha revolucionado la industria aeroespacial al reutilizar las primeras etapas de sus cohetes, reduciendo significativamente los costos de lanzamiento".

Objetivo: "Este proyecto tiene como objetivo predecir el éxito de los aterrizajes de la primera etapa del Falcon 9 utilizando un enfoque basado en la ciencia de datos".

Metodología general: "Se utilizaron técnicas de análisis exploratorio de datos (EDA), modelado predictivo y visualización interactiva para desarrollar y evaluar varios modelos".

Metodologia



Metodología

Resumen

Metodología de recolección de datos:

- Los datos se obtuvieron utilizando la API de SpaceX y la técnica de web scraping desde Wikipedia.

Realización de preparación de datos:

- Se realizó la codificación one-hot en las características categóricas.

Análisis exploratorio de datos (EDA):

- Se utilizó visualización y SQL para llevar a cabo el análisis.

Análisis visual interactivo:

- Se emplearon las herramientas Folium y Plotly Dash para crear visualizaciones interactivas.

Análisis predictivo usando modelos de clasificación:

- Se utilizaron modelos como Regresión Logística, SVM, Árbol de Decisión y KNN.
- Los hiperparámetros se ajustaron mediante validación cruzada con GridSearch.
- La precisión en los datos de prueba se calculó utilizando el método de puntuación, y se generó una matriz de confusión.



Recolección de Datos:

Se realizaron los siguientes pasos para la recolección de datos:

API de SpaceX:

- Los datos se obtuvieron enviando una solicitud GET a la API de SpaceX.
- Los datos se decodificaron como JSON utilizando `.json()` y se convirtieron en un DataFrame usando `.json.normalize()`.
- Luego, se limpiaron los datos, se identificaron los valores faltantes y se completaron adecuadamente cuando fue necesario.

Web-Scraping:

- Se recopilaron registros de lanzamientos del Falcon 9 mediante web scraping desde Wikipedia.
- Se envió una solicitud HTTP GET a la página HTML de lanzamiento del Falcon 9.
- Los datos se analizaron y almacenaron como un DataFrame utilizando BeautifulSoup.



Recolección de Datos en la API de SpaceX

- Se envió una solicitud GET a la API de SpaceX.
- Los datos se decodificaron utilizando `.json()` y se convirtieron en un DataFrame con `.json_normalize()`.
- Se trajeron los valores faltantes sustituyéndolos por la media.
- Primero, se realizó una solicitud GET a la API de SpaceX. Luego, los datos fueron decodificados utilizando `.json()` y se transformaron en un DataFrame con la función `.json_normalize()`. Por último, se abordó el manejo de los valores faltantes.

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

response = requests.get(spacex_url)

static_json_url='https://cf-courses-data.s3.us.cloud-object-storage.appdomain.cloud/IBM-DS0321EN-SkillsNetwork/datasets/Spacex_Falcon_9_Launches.json'

We should see that the request was successful with the 200 status response code

response.status_code

200

Now we decode the response content as a Json using .json() and turn it into a Pandas dataframe using .json_normalize()

# Use json_normalize method to convert the json result into a dataframe
data = pd.json_normalize(response.json())

mean_payload_mass = data_falcon9['PayloadMass'].mean()
# Replace the np.nan values with its mean value
data_falcon9['PayloadMass'].replace(np.nan, mean_payload_mass, inplace=True)
```



Recolección de Datos a través de la web scraping

- Se envió una solicitud HTTP GET a la página HTML del lanzamiento del Falcon 9. Posteriormente, los datos fueron analizados y almacenados en un DataFrame utilizando BeautifulSoup.
- Primero, se envió una solicitud HTTP GET a la página HTML del lanzamiento del Falcon 9 de SpaceX. Luego, se utilizó BeautifulSoup para extraer todos los nombres de columnas o variables desde el encabezado de la tabla HTML. Los registros del lanzamiento del Falcon 9 fueron analizados y se creó un DataFrame a partir de ellos, el cual finalmente fue exportado a un archivo CSV.

```
static_url = "https://en.wikipedia.org/w/index.php?title=List_of_Falcon_9_an  
  
response = requests.get(static_url)  
  
Beautifulsoup = BeautifulSoup(response.text, 'html.parser')  
  
page_title = Beautifulsoup.title  
print(f"Page Title: {page_title}")  
  
Page Title: <title>List of Falcon 9 and Falcon Heavy launches - Wikipedia</title>  
  
for th in th_elements:  
    name = extract_column_from_header(th)  
    # Append the non-empty column name to the list  
    if name is not None and len(name) > 0:  
        column_names.append(name)
```





Data Wrangling

Se calcularon la cantidad de lanzamientos en cada sitio, el número y la frecuencia de cada órbita, así como los tipos y la cantidad respectiva de resultados de aterrizaje. Además, se creó una etiqueta de resultado de aterrizaje a partir de la columna "Outcome".

Primeramente se determinó el total de lanzamientos en cada ubicación y se contó el número de veces que se realizó cada órbita.

GTO	27
ISS	21
VLEO	14
PO	9
LEO	7
SSO	5
MEO	3
ES-L1	1
HEO	1
SO	1
GEO	1

CCAFS SLC 40 55
KSC LC 39A 22
VAFB SLC 4E 13
Name: LaunchSite, dtype: int64



Data Wrangling

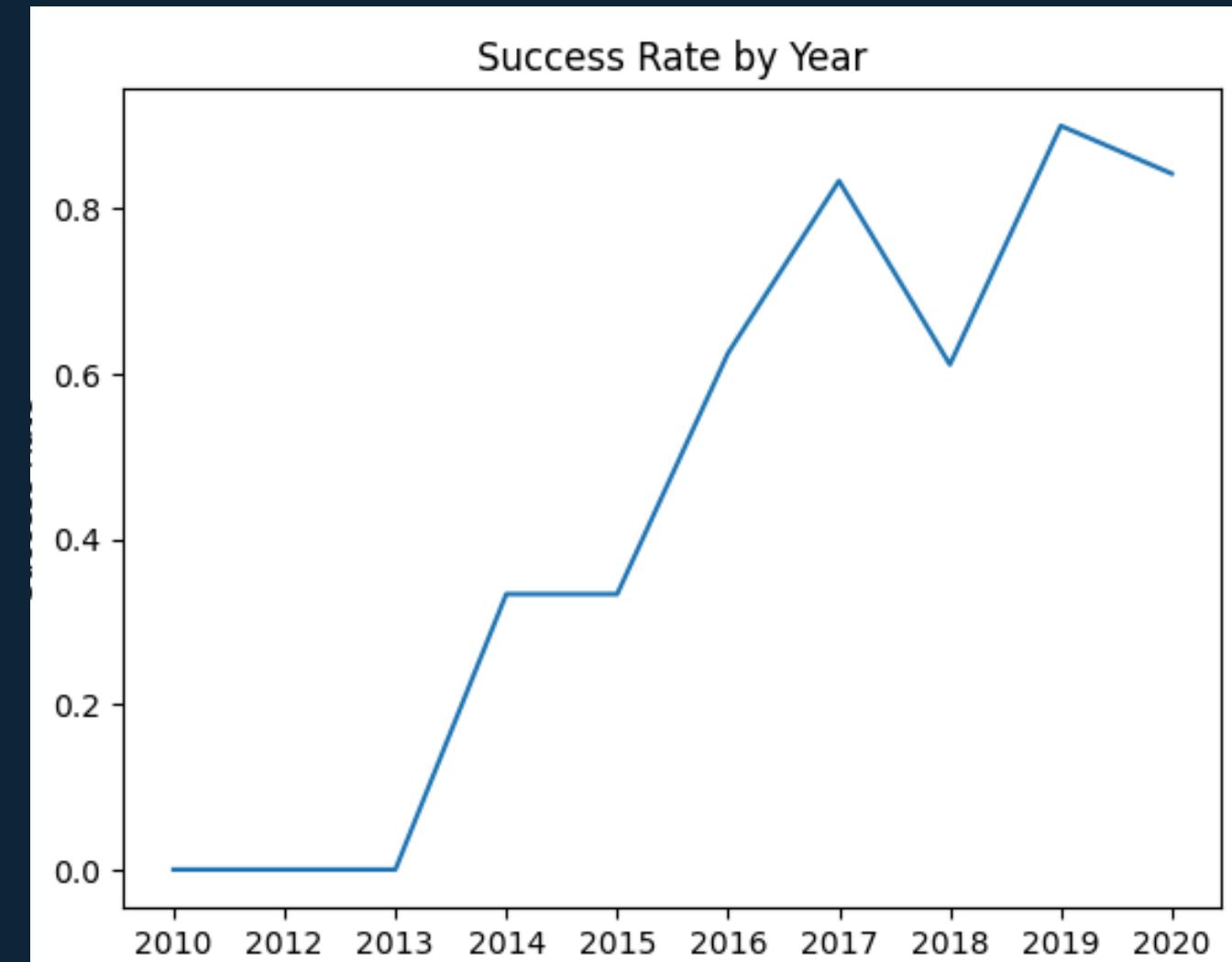
Se determinó el número de lanzamientos en cada sitio, así como la cantidad y frecuencia de cada órbita, junto con los tipos y números correspondientes de resultados de aterrizaje. Además, se generó una etiqueta para los resultados de aterrizaje a partir de la columna "Outcome".

En tercer lugar, se identificaron los tipos y la cantidad de resultados de aterrizaje. Luego, se generó una etiqueta para la columna de aterrizaje a partir de la columna "Outcome". Finalmente, el DataFrame resultante fue exportado a un archivo CSV.

```
landing_outcomes = df['Outcome'].value_counts()  
landing_outcomes  
  
True ASDS      41  
None None      19  
True RTLS      14  
False ASDS     6  
True Ocean     5  
False Ocean    2  
None ASDS      2  
False RTLS     1  
  
df['landing_class'] = landing_class  
print(landing_class)  
  
[0, 0, 0, 0, 0, 1, 1, 0, 0, 0, 1, 0, 0, 0, 1, 0, 0,  
1, 1, 1, 1, 1, 1, 0, 0, 0, 1, 1, 0, 0, 1, 1, 1, 1,  
0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1]
```

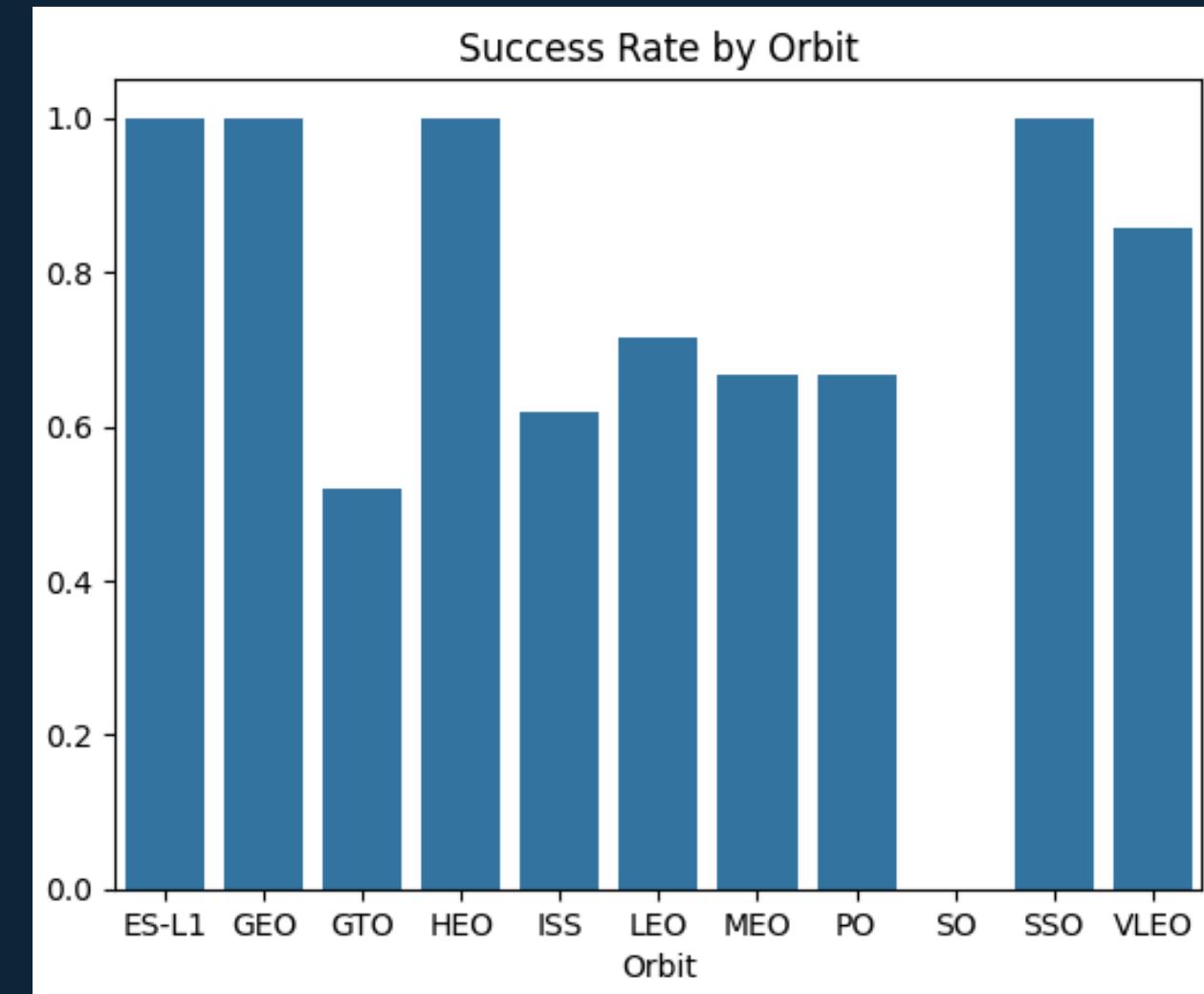
Análisis Exploratorio de Datos (EDA) con Visualización de Datos

Se visualizó la relación entre el número de vuelos y el sitio de lanzamiento, la carga útil y el sitio de lanzamiento.



Análisis Exploratorio de Datos (EDA) con Visualización de Datos

La tasa de éxito y la órbita, el número de vuelos y la órbita, la masa de la carga útil y la órbita, así como la tendencia anual del éxito en los lanzamientos.



Análisis Exploratorio de Datos (EDA) con SQL

Se cargó el conjunto de datos en una tabla DB2, y se realizaron diversas consultas SQL desde Jupyter Notebook. Primero, se identificaron los sitios de lanzamiento únicos en las misiones espaciales. A continuación, se calculó la masa total de la carga útil transportada por los propulsores lanzados por la NASA (CRS) y se estimó la masa promedio de carga útil para la versión del propulsor F9 v1.1. También se contabilizaron los resultados exitosos y fallidos de las misiones. Por último, se evaluaron los aterrizajes fallidos en la nave no tripulada, considerando la versión del propulsor y los nombres de los sitios de lanzamiento involucrados.

Crear un Panel Interactivo con Plotly Dash

Se desarrolló un panel interactivo utilizando Plotly Dash. En este panel, se trazaron gráficos de torta que muestran el total de lanzamientos por ciertos sitios, así como gráficos de dispersión que reflejan la relación entre los resultados de los lanzamientos y la masa de carga útil (en kilogramos) para las diferentes versiones de propulsores.

Crear un Mapa Interactivo con Folium

Se marcaron todos los sitios de lanzamiento y se añadieron objetos al mapa, como marcadores, círculos y líneas, para indicar el éxito o fracaso de los lanzamientos en cada sitio en el mapa de Folium. Se asignaron los resultados de los lanzamientos (éxito o fracaso) a las clases 0 y 1, es decir, 0 para fracaso y 1 para éxito. Utilizando clusters de marcadores codificados por color, se identificaron los sitios de lanzamiento con tasas de éxito relativamente altas. Además, se calcularon las distancias entre el sitio de lanzamiento y sus proximidades, como ferrocarriles, autopistas y ciudades.

Análisis Predictivo

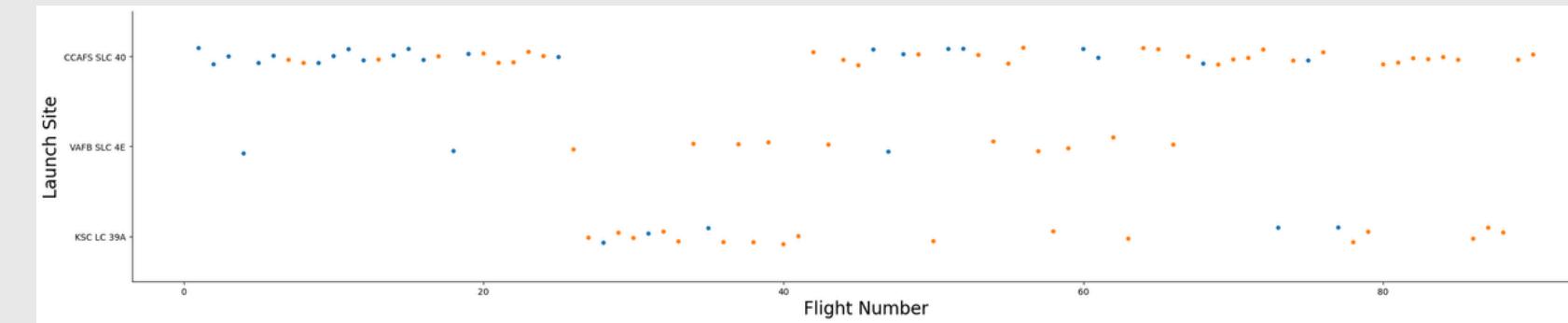
El conjunto de datos se dividió en conjuntos de entrenamiento y prueba. Se construyeron varios modelos de machine learning y se ajustaron diferentes hiperparámetros utilizando GridSearchCV. La precisión se utilizó como métrica para los modelos, y se mejoraron utilizando ingeniería de características y ajuste de hiperparámetros. Finalmente, se identificó el modelo de clasificación con el mejor rendimiento.

Conclusiones extraídas del Análisis Exploratorio de Datos (EDA)



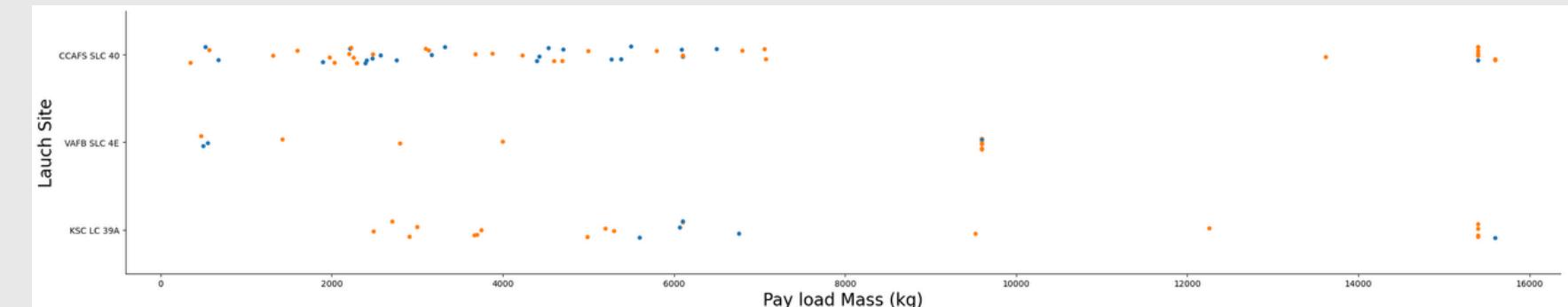
Número de Vuelo vs. Sitio de Lanzamiento

A partir del gráfico de dispersión entre el número de vuelo y el sitio de lanzamiento, se observa que a mayor número de vuelos en un sitio de lanzamiento, mayor es la tasa de éxito en dicho sitio.



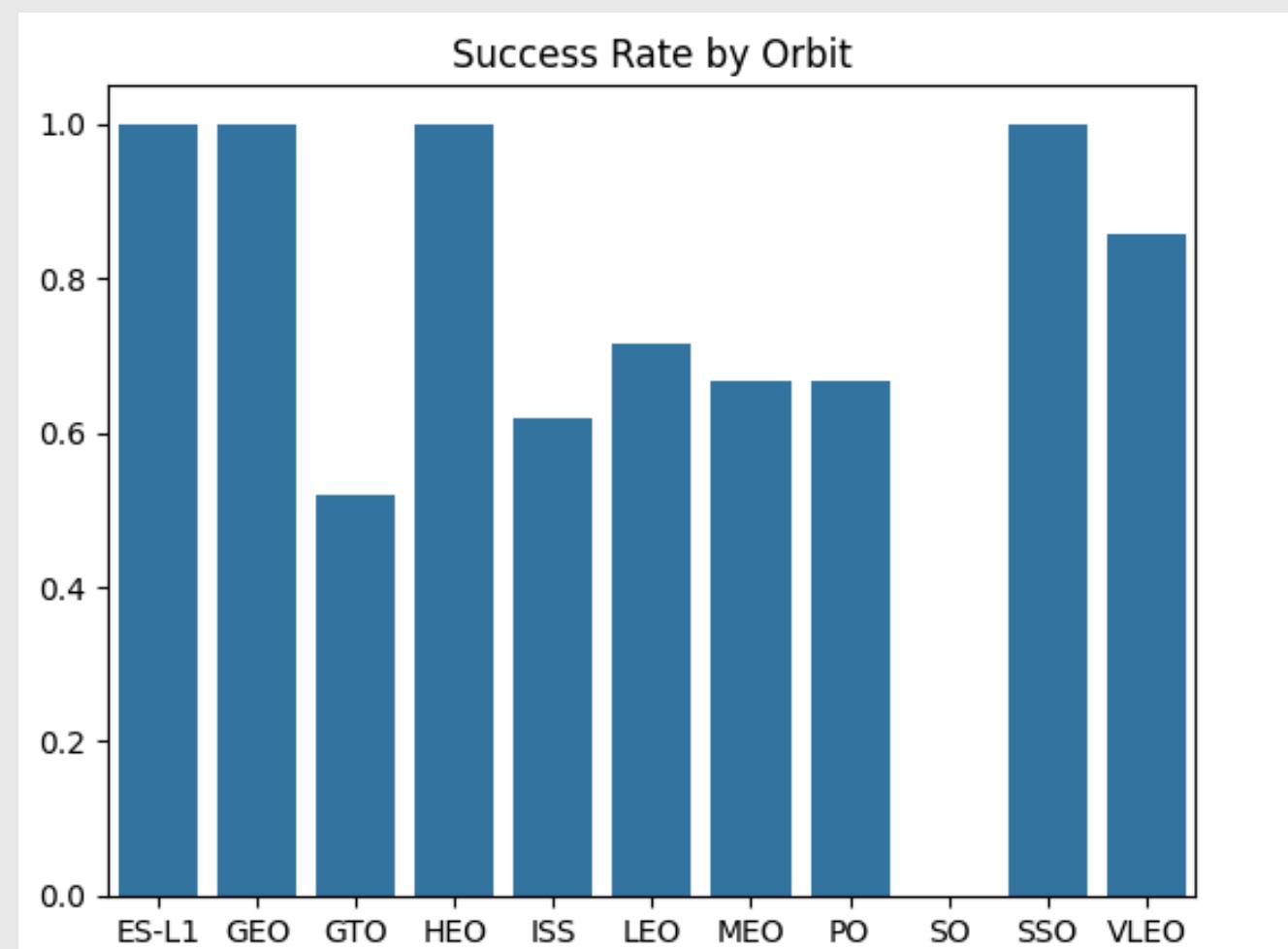
Carga Útil vs. Sitio de Lanzamiento

Para el sitio de lanzamiento CCSFS SLC 40, se observa que a medida que aumenta la masa de la carga útil, es más probable que la primera etapa aterrice con éxito. En el sitio de lanzamiento VAFB-SLC, no se lanzaron cohetes con una carga útil superior a 10,000 kg. Por otro lado, en el sitio de lanzamiento KDC LC 39A, no se realizaron lanzamientos con una carga útil inferior a 2,000 kg.



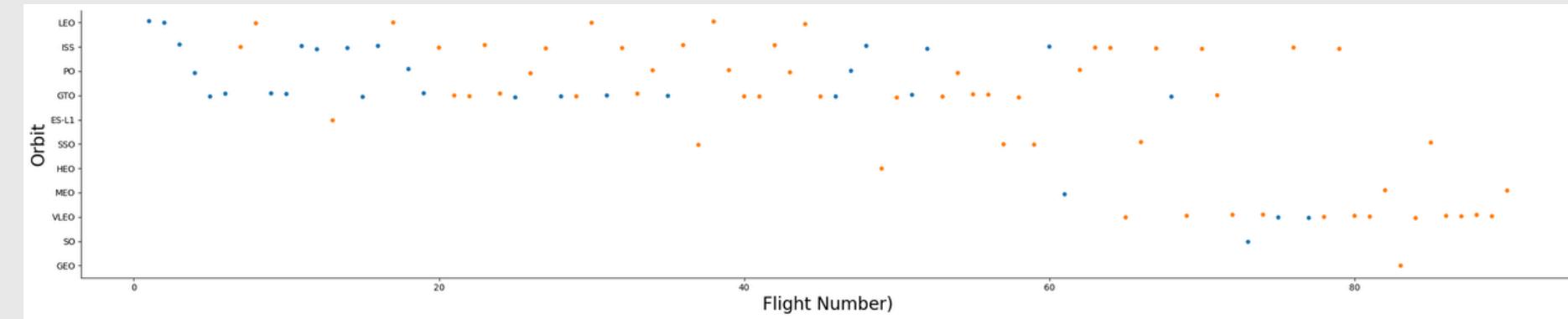
Tasa de Éxito vs. Tipo de Órbita

Según el gráfico de barras, las órbitas ES-L1, GEO, HEO y SSO son las que presentan las tasas de éxito más altas.



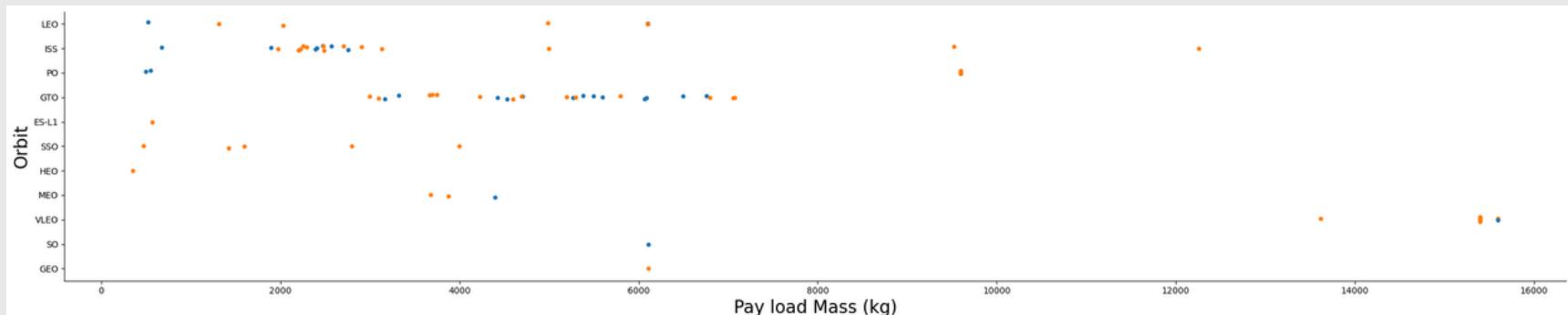
Número de Vuelo vs. Tipo de Órbita

Del gráfico de dispersión entre el número de vuelo y el tipo de órbita, se deduce que la tasa de éxito de las órbitas LEO está relacionada con el número de vuelos, mientras que en las órbitas GEO, la tasa de éxito parece ser independiente del número de vuelos.



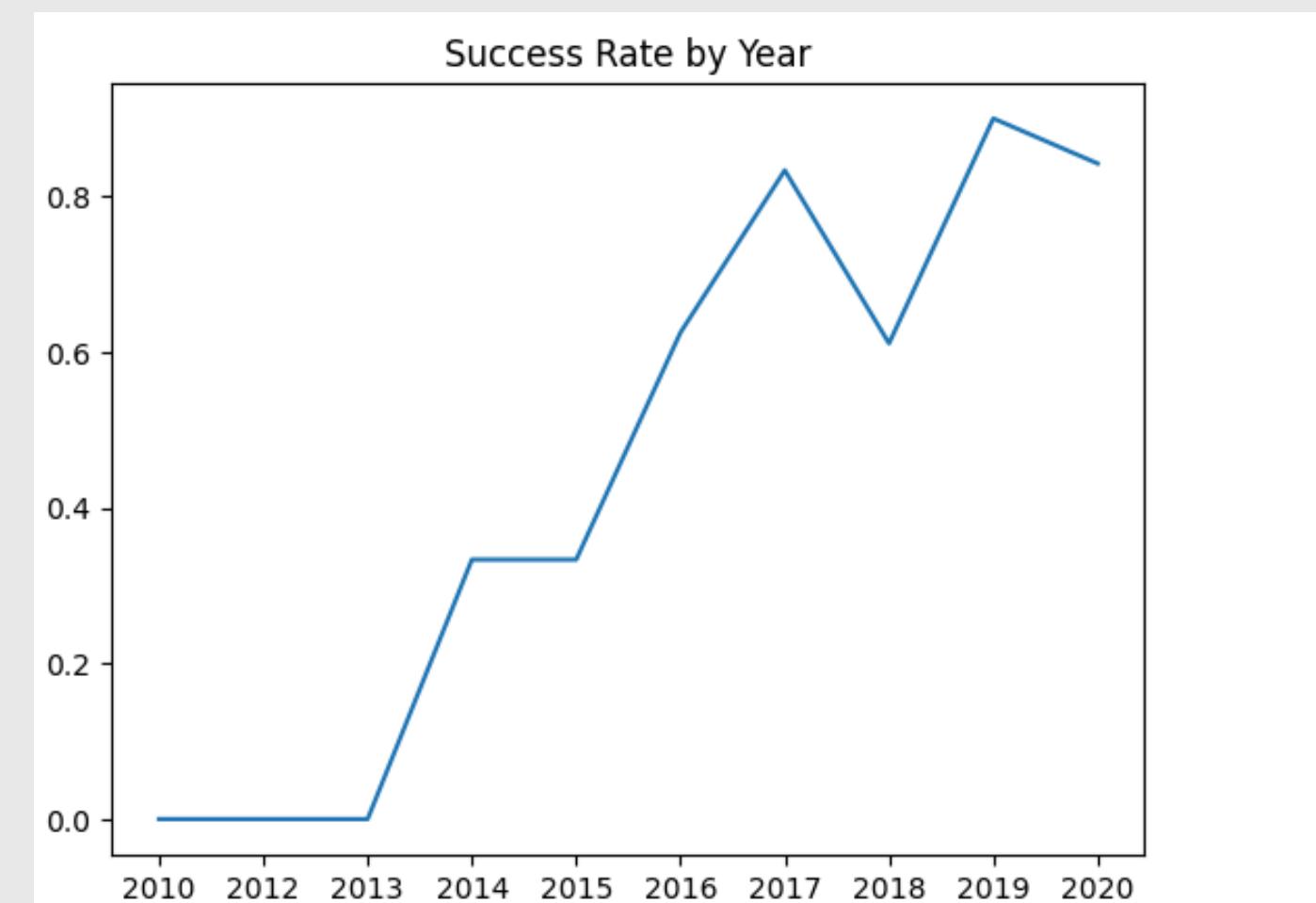
Carga Útil vs. Tipo de Órbita

Se observa que para cargas útiles más pesadas, la tasa de éxito aumenta en las órbitas LEO, ISS y Polares. No parece haber una relación observable entre la órbita GTO y la masa de la carga útil.



Tendencia Anual de Éxito en los Lanzamientos

El gráfico de líneas muestra que la tasa de éxito aumentó de 2013 a 2020, con pequeñas fluctuaciones a lo largo de este período. Sin embargo, se observa una caída significativa en la tasa de éxito en el año 2018.



Nombres de Todos los Sitios de Lanzamiento

Se utilizó la palabra clave DISTINCT para seleccionar los nombres únicos de los sitios de lanzamiento en la tabla.

Nombres de Sitios de Lanzamiento que Comienzan con 'CCA'

Se empleó la palabra clave LIKE para especificar que los nombres de los sitios de lanzamiento comienzan con 'CCA'. Además, se utilizó la palabra clave LIMIT para consultar solo 5 registros.

Done.	Launch_Site
	CCAFS LC-40
	VAFB SLC-4E
	KSC LC-39A
	CCAFS SLC-40

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

Masa Total de la Carga Útil

La masa total de la carga útil para el cliente NASA (CSR) es de 45,596 kg. Se utilizó la función SUM para obtener la suma de la masa de la carga útil y la cláusula WHERE para especificar que el cliente debe ser NASA (CSR).

```
%sql select sum("PAYLOAD_MASS__KG_") from SPACEXTABLE where Customer = 'NASA (CRS)'  
* sqlite:///my_data1.db  
Done.  
  
sum("PAYLOAD_MASS__KG_")  
  
45596
```

Masa Promedio de la Carga Útil para el F9 v1.1

La masa promedio de la carga útil para los cohetes con la versión de propelador F9 v1.1 es de 2,928.4 kg. Se utilizó la función AVG para calcular esta media y la cláusula WHERE para especificar que la versión del propelador debe ser F9 v1.1.

```
%sql select avg("PAYLOAD_MASS__KG_") from SPACEXTABLE where Booster_Version = 'F9 v1.1'  
* sqlite:///my_data1.db  
Done.  
  
avg("PAYLOAD_MASS__KG_")  
  
2928.4
```

Fecha del Primer Aterrizaje en Tierra Exitoso

El primer aterrizaje exitoso en tierra ocurrió el 22 de diciembre de 2015. Se utilizó la función MIN para encontrar la fecha más temprana y la cláusula WHERE para especificar que el resultado del aterrizaje fuera un aterrizaje exitoso en una plataforma terrestre.

```
%sql select min(Date) from SPACEXTABLE where Landing_outcome = 'Success (ground pad)'  
* sqlite:///my_data1.db  
Done.  
min(Date)  
2015-12-22
```

Aterrizaje Exitoso en Nave No Tripulada con Carga Útil entre 4000 y 6000 kg

Los nombres de los propulsores que lograron un aterrizaje exitoso en una nave no tripulada y que tenían una masa de carga útil superior a 4000 kg pero inferior a 6000 kg incluyen F9 FT B1022, F9 FT B1026, F9 FT B1021.2, y F9 FT B1031.2. La cláusula WHERE se utilizó para especificar que el aterrizaje fue exitoso en la nave no tripulada, mientras que la cláusula AND añadió la condición de que la masa de la carga útil estuviera entre 4000 y 6000 kg.

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

Número Total de Resultados de Misiones Exitosas y Fallidas

El número total de misiones exitosas es de 61, mientras que el de misiones fallidas es de 10. La función COUNT se utilizó para obtener el número de aterrizajes, y la cláusula LIKE se utilizó para especificar si el resultado fue exitoso o fallido.

```
%sql select count(*) as 'Success Outcomes' from SPACEXTABLE where Landing_Outcome like 'Success%'  
* sqlite:///my_data1.db  
Done.  
  
Success Outcomes  
61  
  
%sql select count(*) as 'Failure Outcomes' from SPACEXTABLE where Landing_Outcome like 'Failure%'  
* sqlite:///my_data1.db  
Done.  
  
Failure Outcomes  
10
```

Propulsores con la Carga Útil Máxima

La cláusula WHERE se utilizó para especificar que se seleccionara la versión del propulsor con la masa de carga útil máxima, mientras que la función MAX proporcionó la masa de carga útil máxima.

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

Registros de Lanzamientos en 2015

La función SUBSTR se utilizó para seleccionar el año a partir de la fecha, y la cláusula WHERE especificó que la fecha debía ser 2015 y que el resultado del aterrizaje debía ser un fallo en una nave no tripulada.

```
%sql select substr(Date,6,2) as 'Month', Landing_outcome, booster_version, launch_site
* 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
```

Clasificación de Resultados de Aterrizajes Entre el 04-06-2010 y el 20-03-2017

Se realizó un conteo de los resultados de aterrizajes (como fallos en nave no tripulada o éxitos en plataforma terrestre) ocurridos entre el 4 de junio de 2010 y el 20 de marzo de 2017, ordenándolos en orden descendente. Se utilizó la cláusula GROUP BY para agrupar los resultados de los aterrizajes y la cláusula ORDER BY para ordenar los resultados agrupados en orden descendente.

```
%sql
select Landing_outcome, count(*) as 'NoofSuccessfulLandings'
from SPACEXTABLE
where (Landing_outcome like 'Success%') and Date between '2010-06-04' AND '2017-03-20'
group by Landing_outcome
order by 'NoofSuccessfulLandings' desc
* sqlite:///my_data1.db
Done.

Landing_Outcome  NoofSuccessfulLandings
Success (ground pad)  3
Success (drone ship)  5
```

Análisis de Proximidades de los Sitios de Lanzamiento



Los sitios de lanzamiento de SpaceX se encuentran en las costas de Florida y California en Estados Unidos. Solo el sitio de lanzamiento VAFB SLC-4E está en California; el resto se ubica en Florida.

Creacion de Dashboard con Plotly Dash



Gráfico de Torta para el Total de Lanzamientos Exitosos por Todos los Sitios

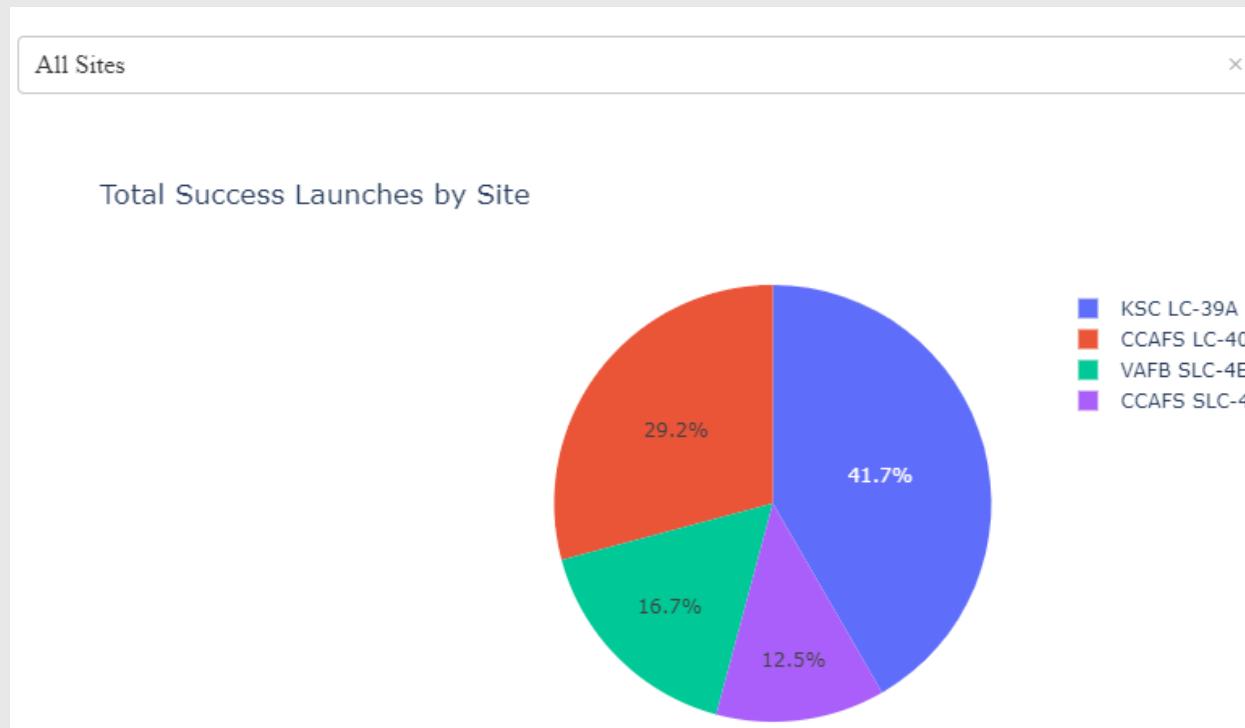
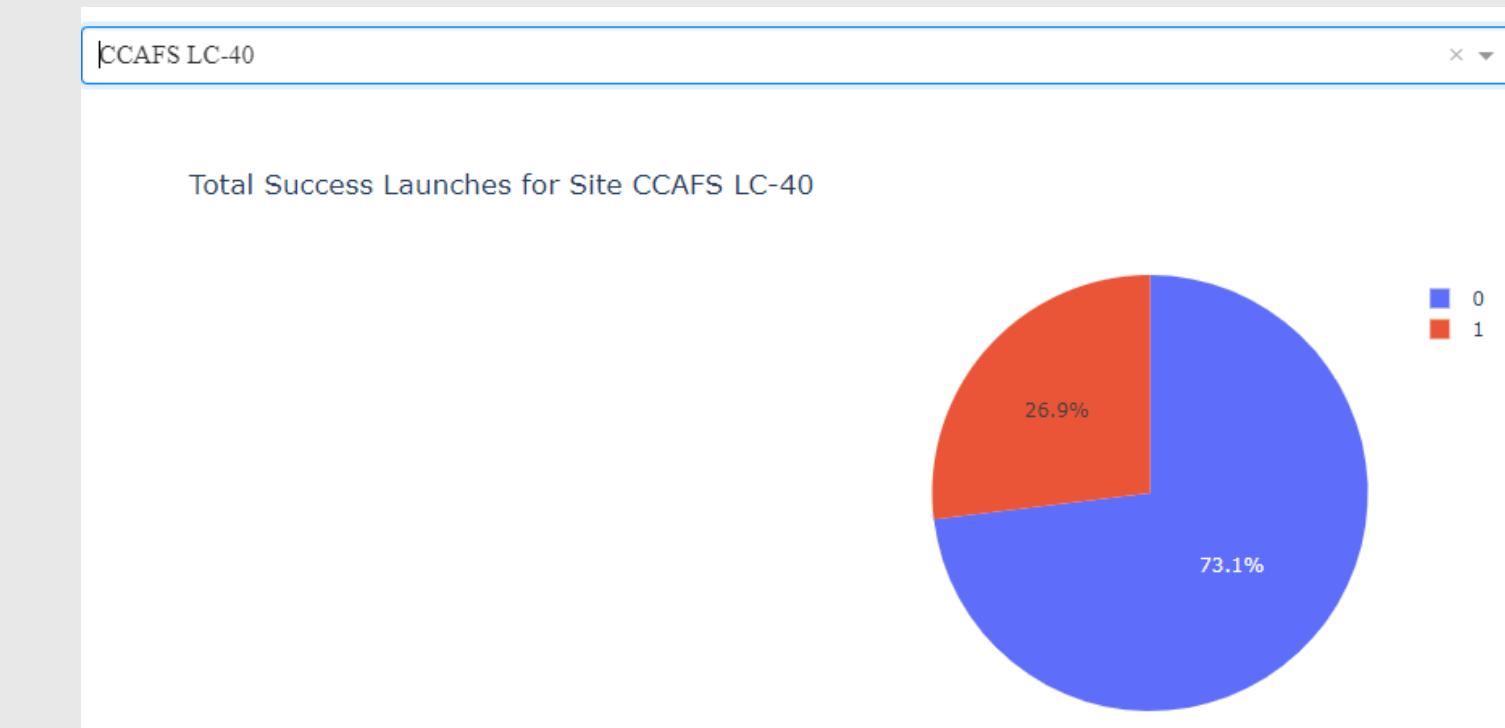
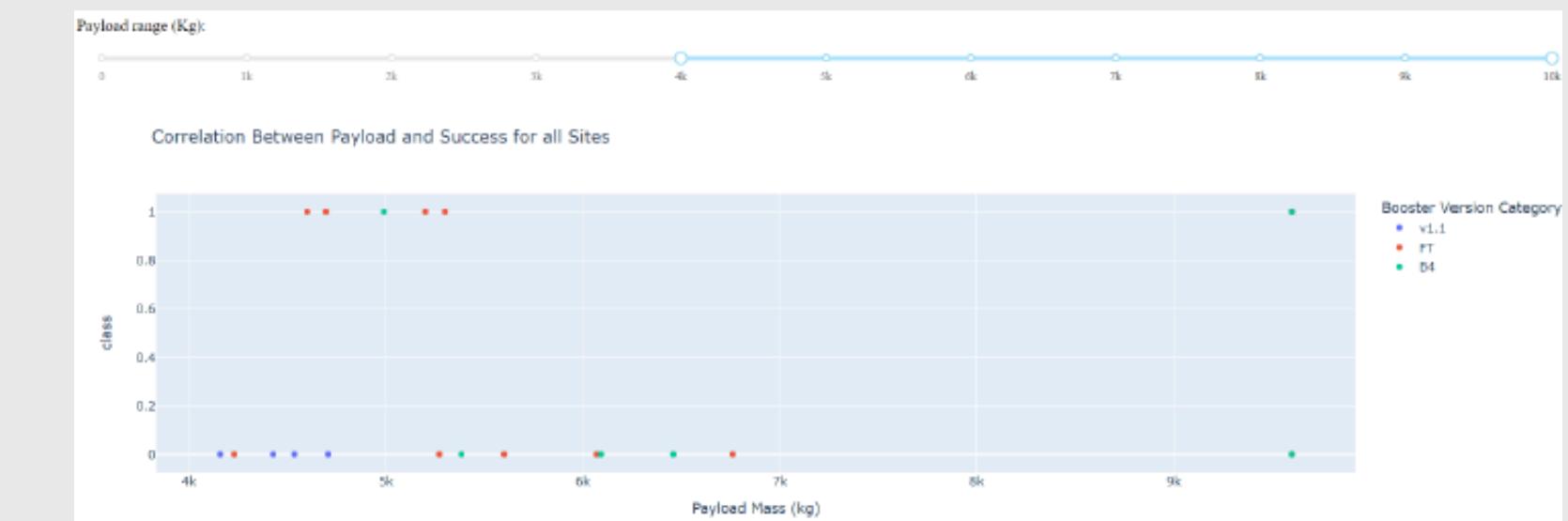


Gráfico de Torta del Índice de Éxito/Fracaso de Lanzamientos en CCAFS LC-40



Control Deslizante de Rango para Carga Útil vs Resultado del Lanzamiento



Análisis Predictivo



Clasificacion en la Presicion

El Clasificador de Árbol de Decisión tiene la mayor precisión de clasificación con un valor de 0.89.

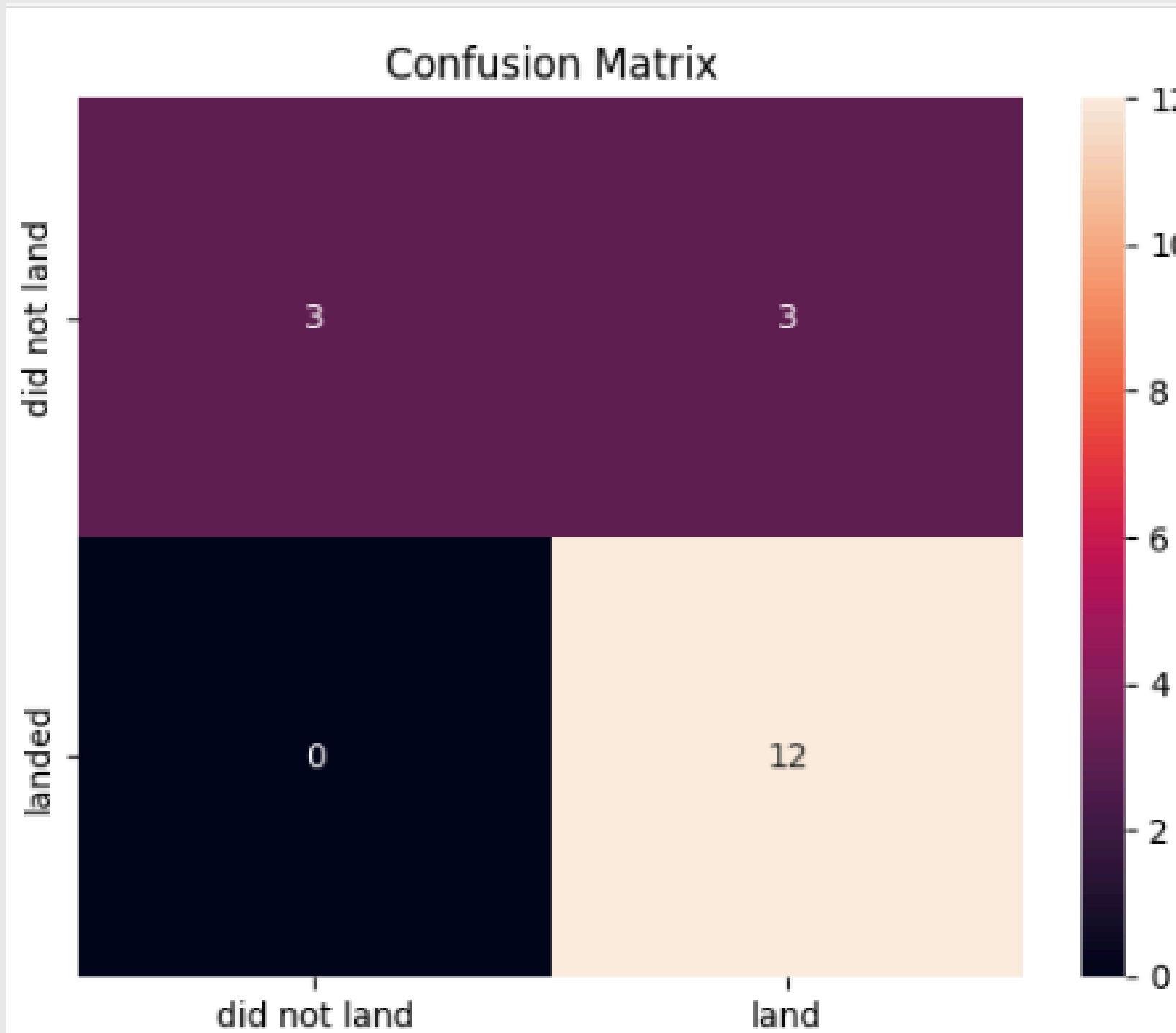
```
: models = [logreg_cv, svm_cv, tree_cv, knn_cv]
results = []
for model in models:
    result_dict = {'Model':str(model.estimator), 'Accuracy':str(model.best_score_), 'Score':str(model.score(X_test, Y_test))}
    results.append(result_dict)
performance_df = pd.DataFrame(results)
performance_df
```

	Model	Accuracy	Score
0	LogisticRegression()	0.8222222222222222	0.9444444444444444
1	SVC()	0.8482142857142856	0.8333333333333334
2	DecisionTreeClassifier()		0.8875 0.8333333333333334
3	KNeighborsClassifier()	0.8482142857142858	0.8333333333333334

```
print("tuned hpyerparameters :(best parameters) ",tree_cv.best_params_)
print("accuracy :",tree_cv.best_score_)

tuned hpyerparameters :(best parameters)  {'criterion': 'gini', 'max_depth': 12, 'max_features': 'sqrt', 'min_samples_leaf': 4, 'min_samples_split': 5, 'splitter': 'random'}
accuracy : 0.8875
```

Matriz de Confusión



Mejor Modelo: Clasificador de Árbol de Decisión

La Matriz de Confusión para el Clasificador de Árbol de Decisión muestra que este clasificador presenta un alto número de falsos positivos, lo que significa que el cohete no aterrizó exitosamente, pero el clasificador predice que sí lo hizo.



Conclusion

- Se observa que a mayor número de vuelos realizados en un sitio de lanzamiento, la tasa de éxito en dicho sitio tiende a incrementarse significativamente, lo que sugiere una correlación positiva entre la frecuencia de lanzamientos y el perfeccionamiento de las operaciones en ese lugar.
- Las órbitas ES-L1, GEO, HEO y SSO destacan por tener las tasas de éxito más elevadas, lo que podría estar relacionado con las características específicas y las condiciones operativas de estas órbitas.
- En el caso de cargas útiles más pesadas, se evidencia un aumento en la tasa de éxito para las órbitas LEO, ISS y Polares, lo que sugiere que estas órbitas son más adecuadas para manejar misiones con mayores exigencias de carga.
- La tendencia general de la tasa de éxito en los lanzamientos ha mostrado un crecimiento constante desde 2013 hasta 2020, a pesar de presentar algunas fluctuaciones menores a lo largo de este periodo, lo que refleja mejoras continuas en los procedimientos y tecnologías de lanzamiento.
- El sitio de lanzamiento KDC LC-39A se destaca como el lugar con la mayor tasa de éxito en los lanzamientos, lo que lo posiciona como un sitio clave en las operaciones espaciales de SpaceX.
- Finalmente, el análisis sugiere que el Clasificador de Árbol de Decisión es el algoritmo de aprendizaje automático más eficaz para predecir con alta precisión el éxito del aterrizaje de la primera etapa del cohete Falcon 9 de SpaceX, destacando su potencial para aplicaciones en misiones futuras.