UDEC
UNIVERSIDAD DE CUNDINAMARCA

PROYECTO FINAL DE BUSINESS ANALYTICS

ANALISIS DE DESERCION EN LA FEDERACION DE DEPORTE DE COLOMBIA EN ESGRIMA EN CATEGORIA DE MAYORES

BUSINESS ANALYTICS 901T

PROFESOR
FREDY PRIETO

ESTUDIANTES
NIKOLAI TORRES
JULIAN ALMARIO

SEMESTRE XI
UNIVERSIDAD DE CUNDINAMARCA
FACULTAD DE INGENIERIA
PROGRAMA DE INGENIERIA DE SISTEMAS Y COMPUTACION
EXTENSIÓN CHÍA
2025

UDEC
UNIVERSIDAD DE CUNDINAMARCA

CONTENIDO

0. Introducción 3
1. Etapa 1: Selección y Definición del Negocio 4
1.1. Información básica de la empresa 4
1.2. Definición de objetivos del proyecto (SMART) 4
1.3. Identificación de los Involucrados 5
1.4. Definición del Alcance de proyecto: 7
1.5. Datos 7
1.6. Proceso de ETL 7
2. Etapa 2: Análisis Descriptivo y Diagnostico 11
2.1. Análisis Descriptivo 11
2.2. Análisis Diagnostico 25
2.3. Identificación del tipo de Modelo Adecuado 26
2.4. Hallazgos Clave 27
3. Etapa 3: Visualización del análisis Predictivo 27
3.1. Análisis Predictivo (¿Qué va a pasar?) 27
3.2. Metodologia y modelos seleccionados 28
3.2.1. Decision Tree Regressor 28
3.2.2. DecisionTreeClassifier 29
3.2.3. RandomForestClassifier 30
3.3. División de datos y metodología de entrenamiento 39
3.4. Evaluación de los modelos 40
3.5. Resultados del modelo Predictivo 43
3.6. Interpretación Estratégica de los Resultados 48
3.7. Visualización Final del análisis predicitivo 50
4. Conclusiones 53
5. Referencias 53

0. Introducción

En el ámbito deportivo, la esgrima representa una disciplina que combina precisión, estrategia y constancia. Sin embargo, en los últimos años, la Federación Colombiana de Esgrima ha evidenciado un incremento en la deserción de deportistas pertenecientes a la categoría de mayores, lo que ha generado preocupación por el impacto que esto tiene tanto en el desarrollo deportivo como en el aprovechamiento de los recursos invertidos en su formación.

Comprender las causas detrás de esta fenómeno resulta fundamental para fortalecer los procesos de retención, optimizar los apoyos institucionales y garantizar la continuidad del talento dentro de la federación. Bajo este contexto, el presente proyecto busca analizar la deserción deportiva desde una perspectiva de analítica de datos, utilizando herramientas como Power BI y Python para explorar la información registrada durante los últimos cinco años y así identificar patrones, tendencias y factores que puedan explicar este comportamiento.

Además del componente deportivo, el análisis incluye una evaluación económica, que permite estimar el costo que representa para la federación cada abandono, considerando la inversión realizada en entrenamientos, competencias y equipamiento.

A través de un enfoque descriptivo y predictivo, este trabajo pretende ofrecer una visión integral del problema, aportando evidencia útil para la toma de decisiones estratégicas y fomentando una cultura orientada a los datos dentro de la organización. De esta forma, la analítica se convierte en una herramienta clave para impulsar la permanencia de los atletas y contribuir al desarrollo sostenible de la esgrima en Colombia.

1. Etapa 1: Selección y Definición del Negocio

1.1. Información básica de la empresa

- Empresa: Federación Colombiana de Esgrima
- Proyecto: Análisis de deserción de deportistas de la federación de esgrima colombiana en categoría de mayores.
- Identificación: La Federación Colombiana de Esgrima es la entidad encargada de promover y fortalecer la práctica de este deporte en el país. Su labor se centra en formar deportistas, organizar competencias y representar a Colombia en escenarios nacionales e internacionales. A través de sus programas busca impulsar el talento, la disciplina y la permanencia de los atletas en todas las categorías, razón por la cual resulta esencial analizar la deserción en la categoría de mayores para comprender sus causas y diseñar estrategias que fomenten la continuidad deportiva.
- Propósito: Comprender las causas y factores asociados a la deserción de deportistas activos para apoyar la toma de decisiones de la federación
- El Problema o pregunta de problema: ¿Qué factores influyen en el aumento de la deserción de deportistas en la categoría de mayores de la Federación Colombiana de Esgrima?

1.2. Definición de objetivos del proyecto (SMART)

➢ Objetivo General

Analizar los factores que influyen en la deserción de los deportistas de la categoría de mayores de la Federación Colombiana de Esgrima durante los últimos cinco años, considerando las tres armas oficiales y el impacto económico que este fenómeno representa para la organización.

- Objetivos específicos

1) Recolectar y depurar los datos de participación y rendimiento de los deportistas de la categoría de mayores registrados en las tres armas de esgrima durante los últimos cinco años.
2) Identificar patrones y tendencias relacionadas con el incremento de la deserción mediante análisis descriptivo.
3) Evaluar la relación entre variables como edad, frecuencia de competencia, resultados y región para determinar su influencia en la deserción.
4) Estimar el impacto económico derivado de la pérdida de deportistas, considerando los recursos invertidos en su formación y participación.
5) Visualizar los resultados mediante tableros y gráficos que faciliten la interpretación de los hallazgos y la toma de decisiones.

1.3. Identificación de los Involucrados

Stakeholder Name	Comité Técnico de la Federación Colombiana de Esgrima
Designacion/Tipo	Interno – Tomador de decisiones
Responsabilidades	Revisar los resultados del análisis y definir estrategias de retención de deportistas.
Especialidad/Experiencia/Dominio	Conocimiento técnico del deporte y experiencia en gestión federativa.
Participación	Alta

Stakeholder Name	Entrenadores Nacionales
Designacion/Tipo	Interno – Consultado
Responsabilidades	Aportar información sobre rendimiento, asistencia y causas deportivas de la deserción.

Especialidad/Experiencia/Dominio	Experiencia directa con deportistas de alto rendimiento.
Participación	Media

Stakeholder Name	Ligas y Clubes de Esgrima
Designacion/Tipo	Externo – Usuario final / Fuente de datos
Responsabilidades	Proveer información sobre participación, registros regionales y seguimiento de deportistas.
Especialidad/Experiencia/Dominio	Conocimiento de la realidad regional y gestión de talento local.
Participación	Media

Stakeholder Name	Universidad de Cundinamarca (Programa de Ingeniería de Sistemas y Computación)
Designacion/Tipo	Externo – Asesor académico / Equipo analítico
Responsabilidades	Desarrollar el proceso de análisis de datos, validación metodológica y visualización de resultados.
Especialidad/Experiencia/Dominio	Especialización en analítica de datos, business analytics y gestión de información.
Participación	Alta

1.4. Definición del Alcance de proyecto:

El proyecto se centra en el análisis de la deserción de deportistas en la categoría de mayores pertenecientes a la Federación Colombiana de Esgrima. El estudio se limitará a los registros de los últimos cinco años y abarcará las tres armas oficiales de la disciplina: florete, sable y espada. A partir de los datos recopilados, se realizará un análisis descriptivo para identificar los factores asociados al incremento de la deserción, considerando además el impacto económico que este fenómeno representa para la federación en términos de pérdida de inversión y recursos destinados al desarrollo deportivo.

1.5. Datos

Dentro y alrededor de la empresa hay una importante variedad de fuentes de información disponibles. Deberá listar y organizar todas esas fuentes de datos relevantes disponibles en su empresa:

- Bases de datos relacionales (Resultados de Pruebas y Rankings nacionales)
- Redes sociales: La federación tiene las siguientes redes sociales:
- Instagram: https://www.instagram.com/fedeesgrimacolombia/?hl=es
- Facebook: https://www.facebook.com/FederacionColombianadeEsgrima/?locale=es_LA
- Información de competidores:
- Bases de datos del mercado o sector
- Páginas web: https://sistemainfo.fedesgrimacolombia.com
- Resoluciones de las competencias y de la federación entre otros

1.6. Proceso de ETL

Documento de Extracción (E)

El proceso de extracción de datos se llevó a cabo mediante un web scraper desarrollado en Python. Este script recolecta información directamente del portal oficial de resultados de la Federación Colombiana de Esgrima (Fedesgrima), específicamente desde la URL base:

https://sistemainfo.fedesgrimacolombia.com/resultados

Esta fuente fue seleccionada debido a que constituye el repositorio oficial y actualizado de resultados deportivos nacionales, incluyendo datos sobre las competencias por categoría, arma, género y tipo de torneo. Dicha información resulta relevante para el problema de negocio, que consiste en analizar el comportamiento competitivo y la deserción de los deportistas mayores, así como estimar el impacto económico asociado a su participación.

El script utiliza las librerías requests y BeautifulSoup para automatizar la recopilación de tablas de resultados, recorriendo de forma iterativa los identificadores de las pruebas (prueba_id) dentro de un rango definido. A cada página se le aplica una extracción estructurada de metadatos, incluyendo el título del evento, la fecha, la categoría, el arma, el género y el tipo de competencia.

Posteriormente, se filtran únicamente las competencias clasificadas como “MAYORES” e “INDIVIDUALES”, garantizando la pertinencia del conjunto de datos frente al objetivo analítico. Los resultados extraídos se almacenan en un DataFrame de pandas y se consolidan en un archivo CSV llamado resultados_fedesgrima.csv.

Esta metodología asegura que los datos provengan de una fuente primaria y oficial, facilitando la trazabilidad y confiabilidad del análisis posterior (Fedesgrima, s.f.).

Documento de Transformación (T)

Los datos extraídos son posteriormente cargados y procesados en un entorno de análisis dentro de Google BigQuery. En esta etapa, se aplica un proceso ETL (Extract, Transform, Load) para garantizar la calidad, coherencia y consistencia de la información antes del análisis descriptivo.

El segundo script, desarrollado en Python, utiliza la librería google.cloud.bigquery para ejecutar consultas SQL que integran los resultados de competencia con otras fuentes complementarias:

Fechas de nacimiento de los deportistas, para calcular su edad y analizar patrones de participación por grupo etario.

Costos anuales de licencia e inscripción, para estimar el impacto económico de la deserción deportiva.

Durante la fase de transformación, se realizan las siguientes operaciones clave:

1. Normalización de nombres de columnas y eliminación de encabezados residuales.
2. Conversión de formatos de fecha y cálculo del año de la competencia.
3. Cálculo de edad en el momento del evento a partir de la fecha de nacimiento.
4. Eliminación de duplicados basados en el nombre del deportista, el identificador de la prueba y el arma.
5. Agrupación por arma (espada, florete y sable) para permitir análisis segmentados.
6. Cálculo de tasas de deserción por arma, edad promedio de deserción y relación con costos económicos.

Estas transformaciones garantizan que los datos finales sean pertinentes, concisos y libres de ruido, permitiendo una interpretación estadística confiable. Además, el procesamiento incluye el control de valores atípicos y la consolidación de registros únicos para evitar sesgos en los cálculos de participación y costos.

UDEC
UNIVERSIDAD DE CUNDINAMARCA

Código Scrapper: https://github.com/Urspectre/Scraper_DF_Resultados_Esgrima_COL-2020-2025

Proceso de ET:

```
import pandas as pd
import matplotlib.pyplot as plt
from google.cloud import bigquery

client = bigquery.Client()

# === 1. Consulta SQL actualizada ===
query = """
SELECT
  r.`Nombre`,
  r.`Liga`,
  r.`Club`,
  r.`Posición` AS Posicion,
  r.`Puntos`,
  r.`prueba_id`,
  r.`Titulo`,
  r.`Fecha`,
  r.`Categoria`,
  r.`Arma`,
  r.`Genero`,
  r.`Tipo`,
  f.`Fecha de nacimiento` AS Fecha_Nacimiento,
  c.`Costo Licencia` AS Costo_Licencia,
  c.`Costo Inscripcion` AS Costo_Inscripcion
FROM `instant-bonbon-474722-g7.Resultados_Pruebas_Mayores.Resultados Pruebas Categorías Mayores` AS r
LEFT JOIN `instant-bonbon-474722-g7.Resultados_Pruebas_Mayores.Fechas_Nacimiento` AS f
  ON TRIM(LOWER(r.`Nombre`)) = TRIM(LOWER(f.`Nombre`))
LEFT JOIN `instant-bonbon-474722-g7.Resultados_Pruebas_Mayores.Costos_Anuales` AS c
  ON EXTRACT(YEAR FROM r.`Fecha`) = c.`Año`
"""

print("--------------------------------------------------")
df = client.query(query).to_dataframe()

# === 2. Limpieza y preparación ===
df = df.rename(columns=lambda x: x.strip().lower())
df = df.drop(columns=[c for c in df.columns if 'unnamed' in c.lower()], errors='ignore')

# Convertir fechas
df['fecha'] = pd.to_datetime(df['fecha'], errors='coerce')
df['fecha_nacimiento'] = pd.to_datetime(df['fecha_nacimiento'], errors='coerce', dayfirst=True)
df['año'] = df['fecha'].dt.year

# Calcular edad al momento de la competencia
df['edad'] = (df['fecha'] - df['fecha_nacimiento']).dt.days / 365.25

# Eliminar duplicados
df = df.drop_duplicates(subset=['nombre', 'prueba_id', 'arma'])

# === 3. Análisis por arma ===
resultados = []

for arma, datos_arma in df.groupby('arma'):
    print(f"\n=== Análisis para {arma.upper()} ===")

    max_año = datos_arma['año'].max()
    ultima_por_deportista = datos_arma.groupby('nombre')['año'].max().reset_index()
    ultima_por_deportista['deserto'] = (max_año - ultima_por_deportista['año']) >= 1

    perfil = datos_arma.groupby('nombre').agg({
        'puntos': 'mean',
        'posicion': 'mean',
        'prueba_id': 'count',
        'categoria': 'first',
        'genero': 'first',
        'fecha_nacimiento': 'first',
        'costo_licencia': 'mean',
```

```
    'costo_inscripcion': 'mean',
    'edad': 'mean'
}).reset_index()

analisis = perfil.merge(ultima_por_deportista[['nombre', 'año', 'deserto']], on='nombre')
```

2. Etapa 2: Análisis Descriptivo y Diagnostico

2.1. Análisis Descriptivo

El análisis descriptivo de este proyecto se centra en comprender cómo ha evolucionado la participación de los deportistas de la categoría de mayores en la Federación Colombiana de Esgrima durante los últimos cinco años. El propósito principal es descubrir qué factores podrían estar relacionados con el aumento de la deserción y cómo esto impacta tanto el rendimiento deportivo como los recursos invertidos por la federación.

Para lograrlo, se utilizaron herramientas accesibles y efectivas como Power BI y Python (con las librerías Matplotlib y Seaborn), que permiten explorar los datos de manera visual y estadística. Estas herramientas facilitan la creación de gráficos, paneles y cuadros de mando que ayudan a interpretar los resultados de forma clara.

Entre las principales métricas analizadas se incluyen:

- Tasa de deserción anual, para ver cuántos deportistas dejan de participar cada año.
- Promedio de edad y tipo de arma, para observar si existen grupos con mayor tendencia al abandono.
- Frecuencia de participación, medida por el número de competencias en las que participa cada deportista.
- Impacto económico estimado, que refleja cuánto pierde la federación por cada atleta que abandona el proceso competitivo.

A través de gráficos de líneas, barras, circulares y dispersión, se busca mostrar de manera visual los cambios en la participación y los posibles patrones detrás de la deserción. Por ejemplo, se espera identificar si existe una edad crítica en la que los deportistas suelen abandonar, si alguna de las tres armas (florete, sable o espada) presenta mayores índices de retiro, o si las ligas con menor frecuencia competitiva registran más casos de abandono.

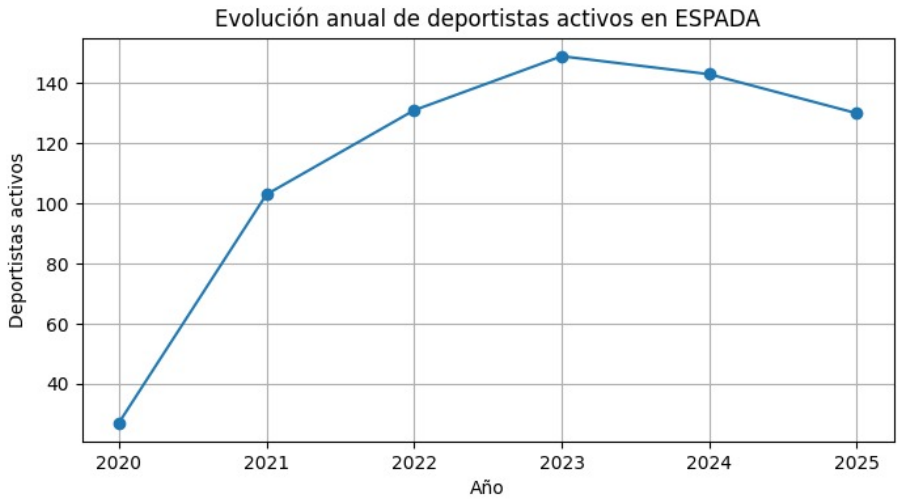
Los resultados del análisis permitirán construir un diagnóstico más claro sobre la situación actual. Si los datos confirman un aumento sostenido de la deserción, será posible determinar en qué condiciones se presenta con mayor frecuencia y estimar el costo económico que esto representa para la federación.
De esta manera, el análisis descriptivo no solo mostrará lo que está ocurriendo, sino que servirá como base para proponer estrategias de retención y optimización de recursos.

Además, este proceso promueve el desarrollo de una cultura orientada a los datos dentro de la organización. Comprender y analizar la información de forma constante permitirá que la federación tome decisiones más informadas, dejando de lado la intuición o las percepciones, y basando su planeación en evidencia real y medible.

Primero vamos a tener en cuenta por tipo de arma:

=== Análisis para ESPADA ===

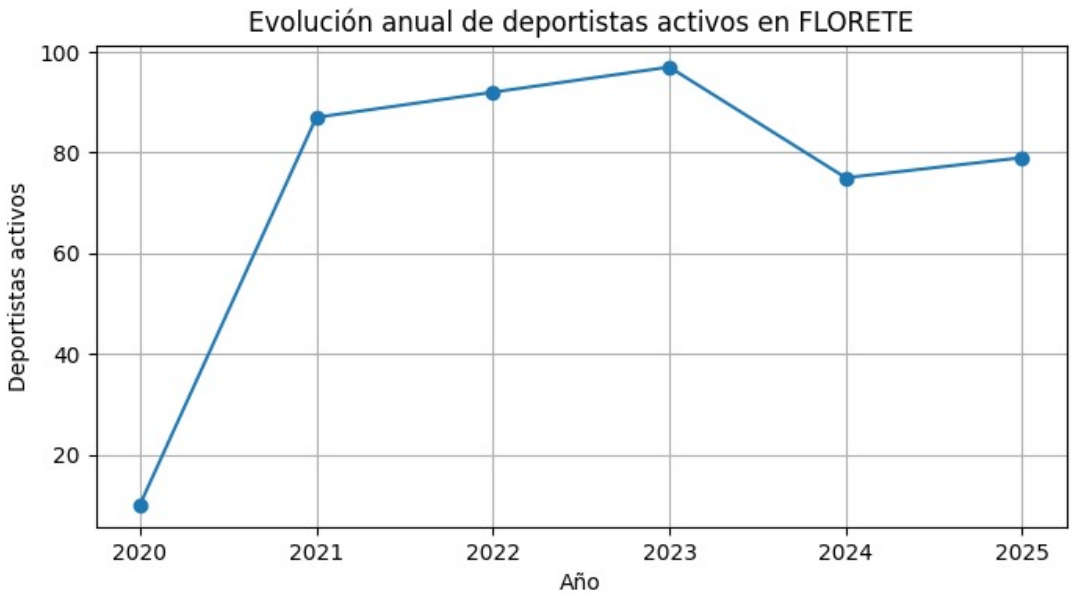
- Tasa de deserción en ESPADA: 56.08%
- Edad promedio de deserción: 24.5 años
- Costo promedio (Licencia + Inscripción): $327,152
- Deserción por género:
- genero
- FEMENINO 0.552632
- MASCULINO 0.565934
- Name: deserto, dtype: float64



=== Análisis para FLORETE ===

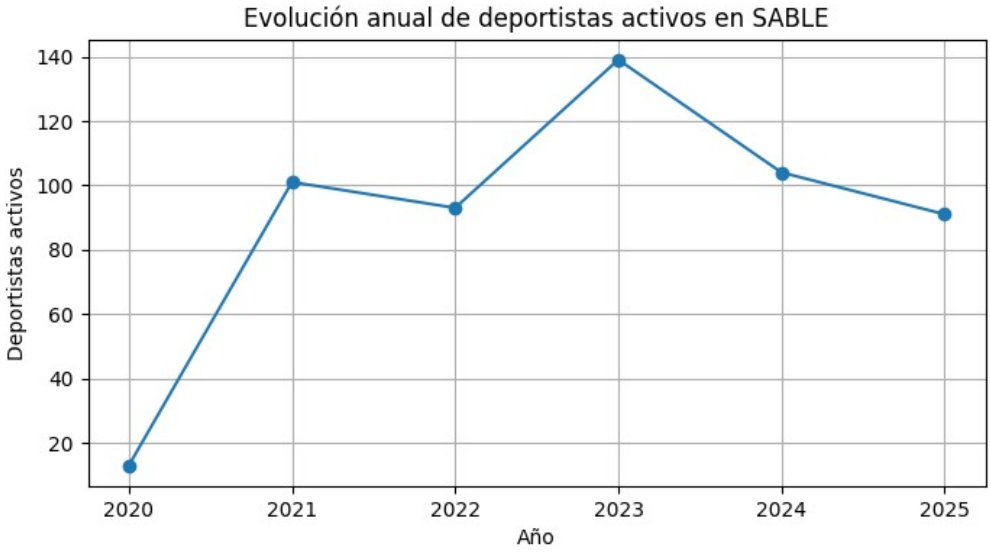
- Tasa de deserción en FLORETE: 60.70%
- Edad promedio de deserción: 23.2 años
- Costo promedio (Licencia + Inscripción): $318,414
- Deserción por género:
- genero
- FEMENINO 0.647727

- MASCULINO 0.575221



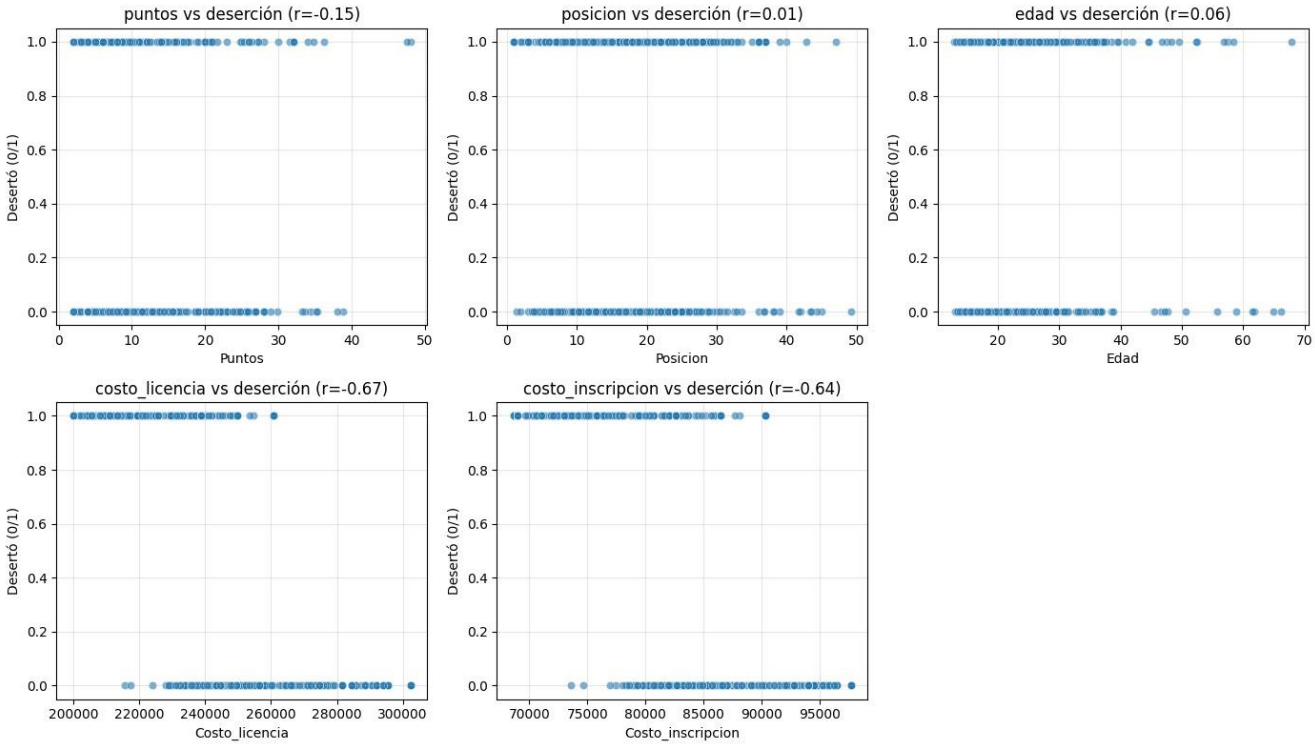
=== Análisis para SABLE ===

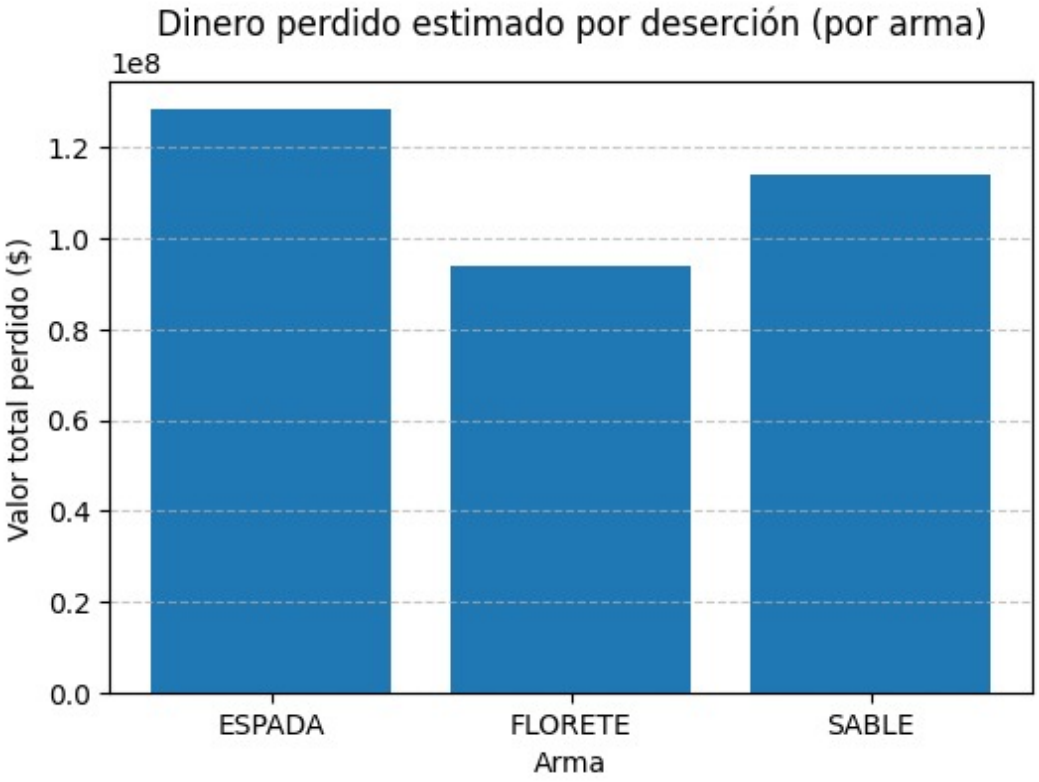
- Tasa de deserción en SABLE: 61.11%
- Edad promedio de deserción: 21.4 años
- Costo promedio (Licencia + Inscripción): $323,698
- Deserción por género:
- genero
- FEMENINO 0.643478
- MASCULINO 0.579832
- Name: deserto, dtype: float64
- Name: deserto, dtype: float64



=== IMPACTO ECONÓMICO REAL ===

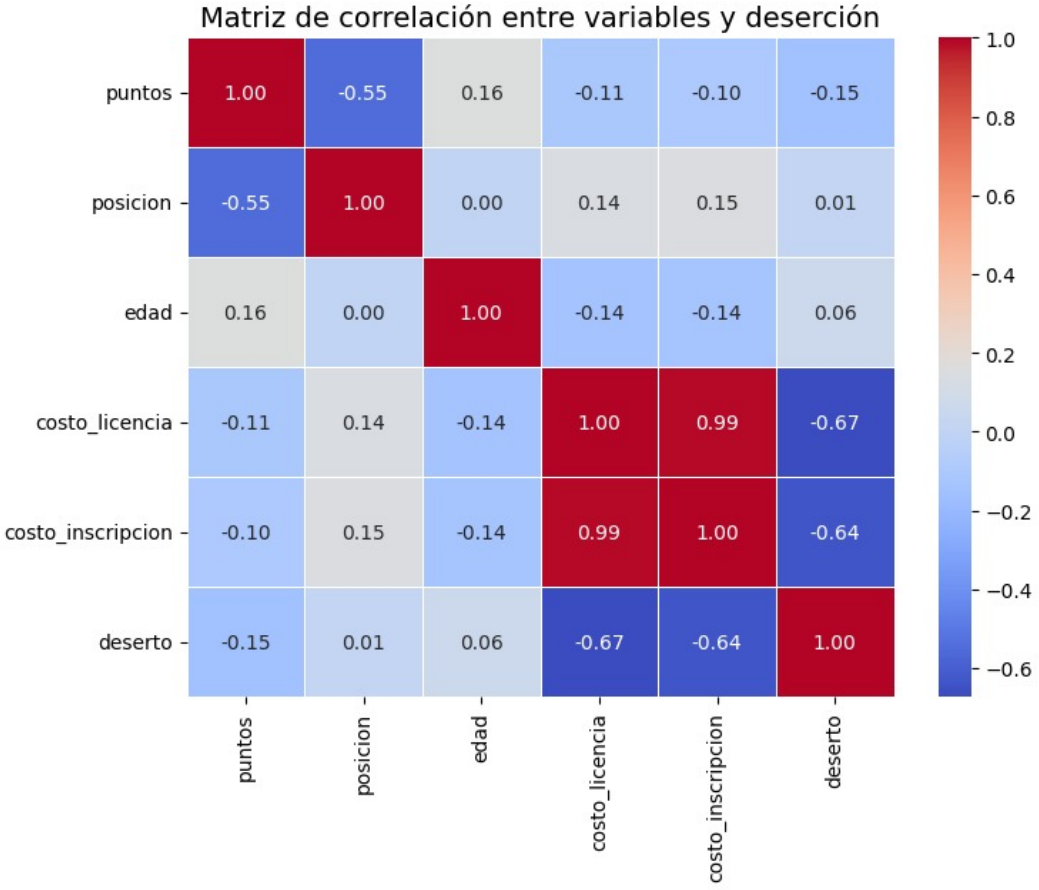
- Dinero perdido total ajustado: $292,549,100

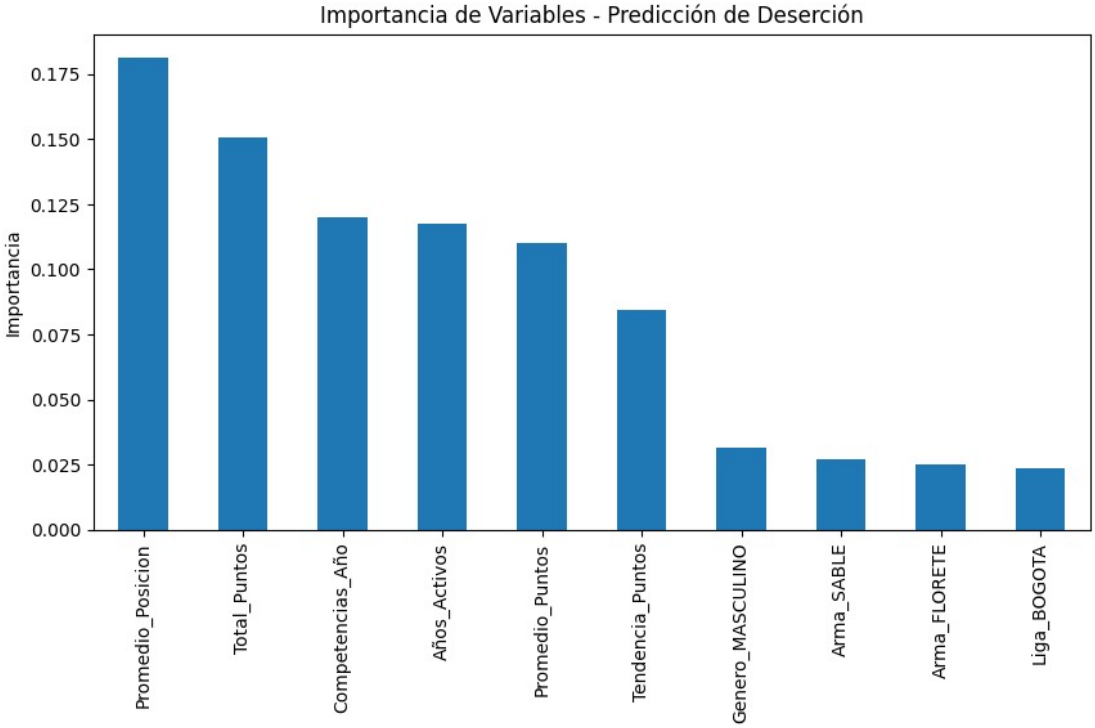




=== CORRELACIÓN ENTRE COSTOS Y DESERCIÓN ===

- Correlación costo licencia vs deserción: -0.674
- Correlación costo inscripción vs deserción: -0.630


Matriz de correlación entre variables y deserción
puntos
posicion
edad
costo_licencia
costo_inscripcion
deserto
1.00 -0.55 0.16 -0.11 -0.10 -0.15
-0.55 1.00 0.00 0.14 0.15 0.01
0.16 0.00 1.00 -0.14 -0.14 0.06
-0.11 0.14 -0.14 1.00 0.99 -0.67
-0.10 0.15 -0.14 0.99 1.00 -0.64
-0.15 0.01 0.06 -0.67 -0.64 1.00
1.0
0.8
0.6
0.4
0.2
0.0
−0.2
−0.4
−0.6


Importancia de Variables - Predicción de Deserción
Importancia
0.175
0.150
0.125
0.100
0.075
0.050
0.025
0.000
Promedio_Posicion
Total_Puntos
Competencias_Año
Años_Activos
Promedio_Puntos
Tendencia_Puntos
Genero_MASCULINO
Arma_SABLE
Arma_FLORETE
Liga_BOGOTA

A continuación, los resultados por liga:

=== ANÁLISIS POR LIGA ===

=== Liga: ANTIOQUIA ===

Tasa de deserción en ANTIOQUIA: 52.86%

Edad promedio de deserción: 26.0 años

Costo promedio (Licencia + Inscripción): $319,068

Deserción por género:

genero

FEMENINO 0.516129

MASCULINO 0.538462

Name: deserto, dtype: float64

=== Liga: ATLANTICO ===

Tasa de deserción en ATLANTICO: 0.00%

Edad promedio de deserción: nan años

Costo promedio (Licencia + Inscripción): $315,083

Deserción por género:

genero

FEMENINO 0.0

MASCULINO 0.0

Name: deserto, dtype: float64

=== Liga: BOGOTA ===

Tasa de deserción en BOGOTA: 49.59%

Edad promedio de deserción: 25.2 años

Costo promedio (Licencia + Inscripción): $327,093

Deserción por género:

genero

FEMENINO 0.425926

MASCULINO 0.550725

Name: deserto, dtype: float64

=== Liga: CALDAS ===

Tasa de deserción en CALDAS: 56.25%

Edad promedio de deserción: 20.7 años

Costo promedio (Licencia + Inscripción): $315,430

Deserción por género:

genero

FEMENINO 0.833333

MASCULINO 0.400000

Name: deserto, dtype: float64

=== Liga: CESAR ===

Tasa de deserción en CESAR: 84.00%

Edad promedio de deserción: 17.5 años

Costo promedio (Licencia + Inscripción): $290,967

Deserción por género:

genero

FEMENINO 0.846154

MASCULINO 0.833333

Name: deserto, dtype: float64

=== Liga: CUNDINAMARCA ===

Tasa de deserción en CUNDINAMARCA: 47.95%

Edad promedio de deserción: 27.2 años

Costo promedio (Licencia + Inscripción): $327,188

Deserción por género:

genero

FEMENINO 0.535714

MASCULINO 0.444444

Name: deserto, dtype: float64

=== Liga: ECUADOR ===

Tasa de deserción en ECUADOR: 71.43%

Edad promedio de deserción: 21.0 años

Costo promedio (Licencia + Inscripción): $300,571

Deserción por género:

genero

FEMENINO 0.857143

MASCULINO 0.571429

Name: deserto, dtype: float64

=== Liga: EL SALVADOR ===

Tasa de deserción en EL SALVADOR: 58.33%

Edad promedio de deserción: 22.5 años

Costo promedio (Licencia + Inscripción): $311,964

Deserción por género:

genero

FEMENINO 0.714286

MASCULINO 0.400000

Name: deserto, dtype: float64

=== Liga: FCE ===

Tasa de deserción en FCE: 0.00%

Edad promedio de deserción: nan años

Costo promedio (Licencia + Inscripción): $321,300

Deserción por género:

genero

FEMENINO 0.0

Name: deserto, dtype: float64

=== Liga: FFAA ===

Tasa de deserción en FFAA: 71.43%

Edad promedio de deserción: 23.9 años

Costo promedio (Licencia + Inscripción): $343,784

Deserción por género:

genero

FEMENINO 0.80

MASCULINO 0.65

Name: deserto, dtype: float64

=== Liga: GUATEMALA ===

Tasa de deserción en GUATEMALA: 0.00%

Edad promedio de deserción: nan años

Costo promedio (Licencia + Inscripción): $269,000

Deserción por género:

genero

MASCULINO 0.0

Name: deserto, dtype: float64

=== Liga: HONDURAS ===

Tasa de deserción en HONDURAS: 0.00%

Edad promedio de deserción: nan años

Costo promedio (Licencia + Inscripción): $269,000

Deserción por género:

genero

FEMENINO 0.0

Name: deserto, dtype: float64

=== Liga: INTERNACIONAL ===

Tasa de deserción en INTERNACIONAL: 66.00%

Edad promedio de deserción: 23.3 años

Costo promedio (Licencia + Inscripción): $337,420

Deserción por género:

genero

FEMENINO 0.727273

MASCULINO 0.607143

Name: deserto, dtype: float64

=== Liga: LECR ===

Tasa de deserción en LECR: 75.00%

Edad promedio de deserción: 16.0 años

Costo promedio (Licencia + Inscripción): $293,325

Deserción por género:

genero

FEMENINO 0.0

MASCULINO 1.0

Name: deserto, dtype: float64

=== Liga: META ===

Tasa de deserción en META: 41.67%

Edad promedio de deserción: 17.5 años

Costo promedio (Licencia + Inscripción): $337,537

Deserción por género:

genero

FEMENINO 0.333333

MASCULINO 0.444444

Name: deserto, dtype: float64

=== Liga: PANAMA ===

Tasa de deserción en PANAMA: 75.00%

Edad promedio de deserción: 22.6 años

Costo promedio (Licencia + Inscripción): $301,842

Deserción por género:

genero

FEMENINO 1.000000

MASCULINO 0.333333

Name: deserto, dtype: float64

=== Liga: RISARALDA ===

Tasa de deserción en RISARALDA: 37.50%

Edad promedio de deserción: 24.3 años

Costo promedio (Licencia + Inscripción): $332,682

Deserción por género:

genero

FEMENINO 0.222222

MASCULINO 0.466667

Name: deserto, dtype: float64

=== Liga: SANTANDER ===

Tasa de deserción en SANTANDER: 85.71%

Edad promedio de deserción: 19.7 años

Costo promedio (Licencia + Inscripción): $330,121

Deserción por género:

genero

FEMENINO 0.875000

MASCULINO 0.833333

Name: deserto, dtype: float64

=== Liga: TOLIMA ===

Tasa de deserción en TOLIMA: 41.30%

Edad promedio de deserción: 21.0 años

Costo promedio (Licencia + Inscripción): $329,036

Deserción por género:

genero

FEMENINO 0.533333

MASCULINO 0.354839

Name: deserto, dtype: float64

=== Liga: VALLE ===

Tasa de deserción en VALLE: 48.06%

Edad promedio de deserción: 21.2 años

Costo promedio (Licencia + Inscripción): $323,955

Deserción por género:

genero

FEMENINO 0.423729

MASCULINO 0.528571

Name: deserto, dtype: float64

=== Liga: VENEZUELA ===

Tasa de deserción en VENEZUELA: 0.00%

Edad promedio de deserción: nan años

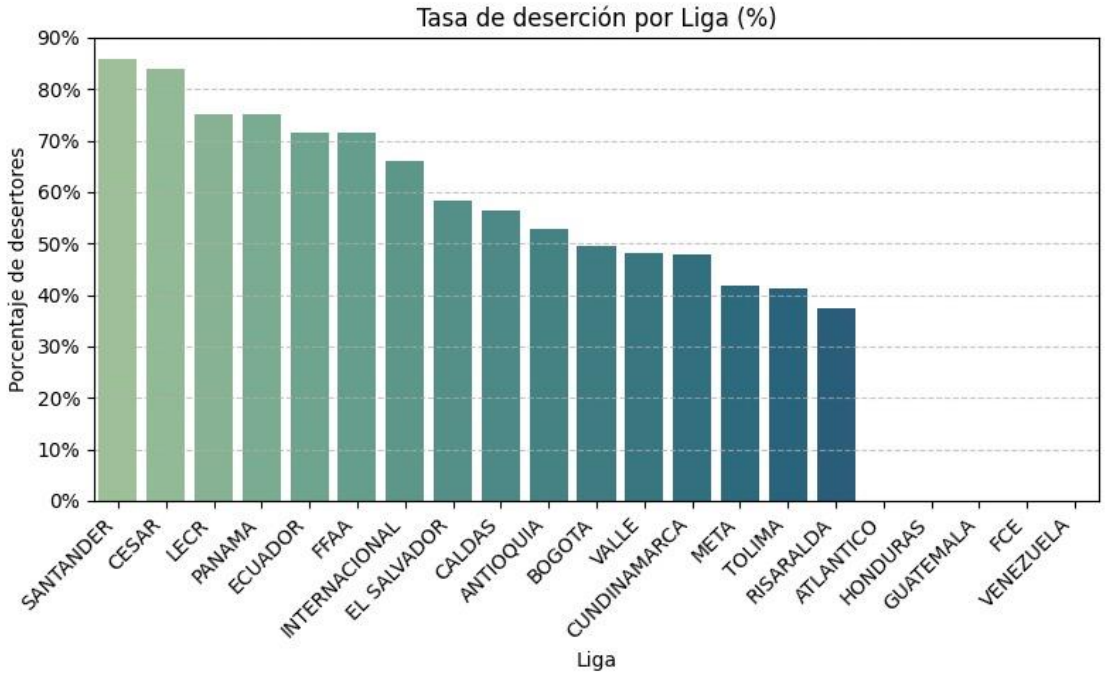
Costo promedio (Licencia + Inscripción): $284,000

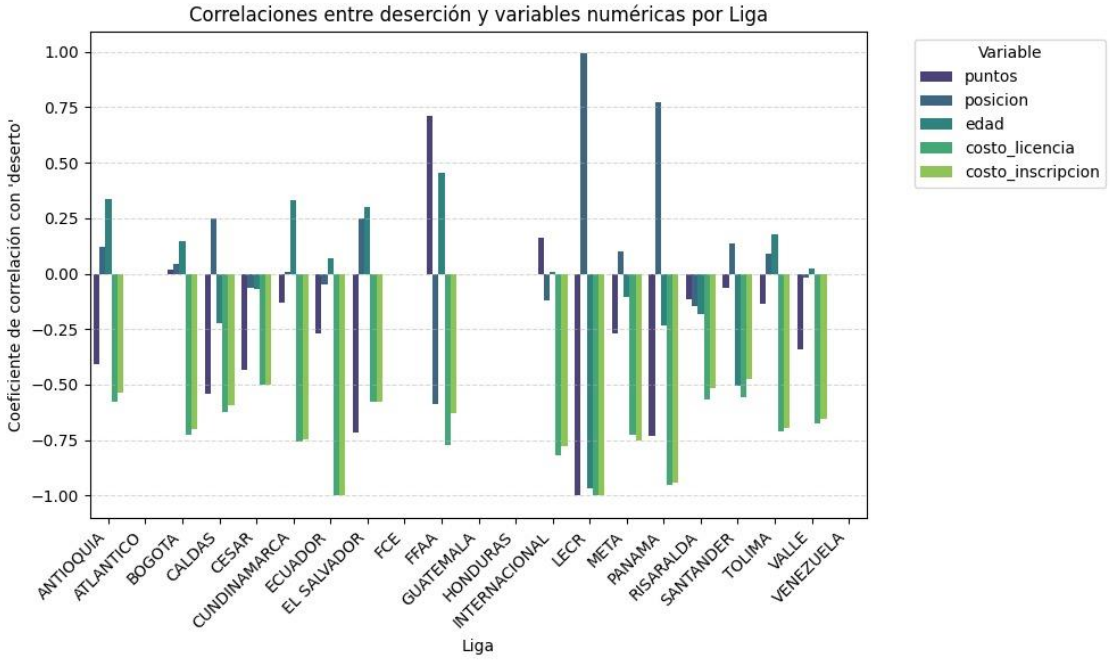
Deserción por género:

genero

MASCULINO 0.0

Name: deserto, dtype: float64





2.2. Análisis Diagnostico

En esta etapa se profundizó en la relación entre las variables y la deserción. A través de comparaciones entre grupos, gráficos de distribución y análisis bivariado, se identificaron patrones que sugieren posibles causas del abandono deportivo.

El análisis reveló que la frecuencia de participación es el principal diferenciador entre deportistas desertores y no desertores. Aquellos con menor número de competencias anuales presentan una probabilidad significativamente mayor de abandonar el proceso deportivo.

También se identificó que los deportistas con menos años dentro del sistema federativo tienen mayor tendencia a desertar, lo cual sugiere que los primeros años en el proceso son críticos para la permanencia.

Aunque variables como categoría, rama deportiva y departamento muestran algunas diferencias, estas no son suficientemente marcadas de manera individual. Sin embargo, su combinación puede aportar valor predictivo, lo que motiva el uso de modelos capaces de evaluar interacciones entre múltiples variables.

Por otro lado, el análisis diagnóstico evidenció el efecto económico asociado a la deserción. Los deportistas que desertan tienden a estar asociados a un impacto económico promedio mayor, relacionado con los recursos invertidos previo a su abandono.

2.3. Identificación del tipo de Modelo Adecuado

A partir del análisis descriptivo y diagnóstico, se determinó que el proyecto abarca dos problemas diferentes de naturaleza predictiva:

1) Problema de Clasificación (Deserción)

La variable desertor_confirmado es de tipo binaria (0 = no deserta, 1 = deserta), lo que indica que se trata de un problema de clasificación supervisada.

Dado que se requiere interpretar los factores que contribuyen a la deserción y al mismo tiempo obtener buen desempeño predictivo, se seleccionaron los siguientes modelos:

- Decision Tree Classifier:
  Modelo interpretable que permite identificar reglas claras de decisión y visualizar rutas críticas hacia la deserción.
- Random Forest Classifier:
  Modelo robusto que combina múltiples árboles para mejorar la precisión, reducir el sobreajuste y manejar mejor la variabilidad de los datos.
  Se usa debido a su alta estabilidad, buen manejo de datos mixtos y excelente rendimiento en problemas similares.

2) Problema de Regresión (Impacto Económico)

La variable impacto_desercion, expresada en valores monetarios, requiere un modelo capaz de predecir un valor continuo. Por lo tanto, se seleccionó:

- Decision Tree Regressor:
  Modelo adecuado para estimar costos asociados, captura relaciones no lineales y permite explicar qué factores influyen más en la variación económica.

Estos modelos se ajustan al tipo de dato, a la estructura del problema y a los objetivos del proyecto:

- Anticipar la deserción
- Cuantificar el impacto económico
- Conocer los factores determinantes

2.4. Hallazgos Clave

Los análisis previos permitieron identificar patrones relevantes que posteriormente justificaron la selección de los modelos predictivos:

1. La frecuencia de competencia emergió como el predictor más influyente, al mostrar diferencias marcadas entre deportistas desertores y no desertores.
2. Los años de permanencia en el proceso deportivo también demostraron relación con la deserción: los atletas más nuevos presentan mayor riesgo de abandono.
3. Las variables demográficas como edad, género y departamento mostraron comportamientos relativamente homogéneos y menor capacidad de diferenciación directa, aunque su interacción con otras variables resulta relevante.
4. El análisis evidenció que los deportistas desertores generan mayor impacto económico acumulado, reflejando pérdidas significativas para la institución.
5. Se identificaron patrones no lineales entre variables, lo que respalda el uso de modelos basados en árboles de decisión y ensambles como Random Forest.

3. Etapa 3: Visualización del análisis Predictivo

3.1. Análisis Predictivo (¿Qué va a pasar?)

En esta etapa se desarrolló un modelo de Machine Learning orientado a predecir la probabilidad de deserción de los deportistas de la categoría de mayores en la Federación Colombiana de Esgrima. El objetivo del análisis predictivo consistió en anticipar qué atletas presentan mayor riesgo de abandono, utilizando variables históricas como edad, arma, frecuencia de participación, puntajes, costos y años activos.

- Variable objetivo
- desertor_confirmado
- 1 = el deportista desertó
- 0 = continúa active

Con esta información, se podría estimar la probabilidad de abandono de cada deportista y conocer cuáles factores influyen más en su decisión de continuar o retirarse. Por ejemplo, el modelo podría mostrar que los atletas con baja frecuencia de competencia y menor rendimiento tienen una mayor probabilidad de desertar.

Este tipo de análisis permitiría a la federación actuar con anticipación, ofreciendo apoyo adicional, incentivos o programas de acompañamiento a los deportistas que presenten un mayor riesgo de abandono.
Además, facilitaría la planificación financiera, al proyectar las posibles pérdidas económicas asociadas a la deserción futura.

En resumen, combinar el análisis descriptivo con el predictivo permitirá a la Federación Colombiana de Esgrima no solo entender las causas del problema actual, sino también prevenir su aumento en los próximos años, fortaleciendo la permanencia de los atletas y el aprovechamiento de los recursos institucionales.

3.2. Metodologia y modelos seleccionados

3.2.1. Decision Tree Regressor

El primer modelo que elegimos del proyecto es el árbol de decisión de regresión, el cual, es un modelo supervisado que predice valores numéricos continuos. Divide los datos en segmentos usando reglas basadas en las variables, construyendo una estructura de árbol donde cada rama representa una decisión y cada hoja un valor predicho.

Este tipo de modelo produce un árbol donde se muestran:

- Las variables que mejor explican el valor económico predicho.
- Los nodos divididos por umbrales numéricos (por ejemplo, edad > 23.5).
- Hojas con valores como:
  - "value = 490253.24" (valor promedio estimado)
  - "squared_error" (error del modelo)

En general, se visualiza un árbol claro que muestra cómo las características del deportista influyen en el costo asociado a su abandono.

La elección de este modelo se baso en las siguientes valoraciones:

- Permite estimar el impacto económico de la deserción (pérdida potencial para la federación).
- Es interpretativo: cada regla del árbol muestra qué variables económicas o deportivas influyen más.
- Ayuda a complementar el análisis clasificatorio con una visión económica cuantitativa.

3.2.2. DecisionTreeClassifier

El segundo modelo que se utilizo es el árbol de decisión y clasificación, este un modelo supervisado que predice categorías. En este caso, predice si un deportista:

- Deserta (1)
- No deserta (0)

Construye reglas simples basadas en variables como edad, arma, frecuencia de competencia, puntos, etc.

El árbol de clasificación produce:

- Una estructura jerárquica de decisiones ("si edad < 22.5 entonces…").
- Nodos con información como:
  - gini
  - samples
  - class distribution
  - predicted class

Lo más importante visualmente es ver qué variables y umbrales son más determinantes para explicar la deserción.

Ejemplo visual esperado:

- Primer nodo partiendo por “frecuencia de competencia” o “edad”.
- Divisiones hacia deportistas con más riesgo de abandono.
- Colores indicando la probabilidad de pertenecer a la clase “Deserta”.

Al igual que el moldeo anterior, se escogió este modelo por los siguientes motivos:

- Explica la deserción con reglas claras y fáciles de interpretar.
- Permite entender por qué un deportista está en riesgo, no solo predecirlo.
- Es útil para la federación que necesita motivos concretos (no cajas negras).
- Sirve como modelo base y punto de comparación para otros modelos más complejos como Random Forest.

3.2.3. RandomForestClassifier

Por ultimo, el modelo que se implemento es el árbol de clasificación aleatorio, este es un modelo avanzado basado en un conjunto de muchos árboles de decisión entrenados sobre distintas muestras del dataset. Cada árbol vota por una clase, y la mayoría determina la predicción final. Es más robusto que un árbol individual.

Este modelo no se ve como un árbol sencillo, sino que se analiza con:

- Gráfico de importancia de variables (feature importance)
- Curvas ROC
- Matriz de confusión
- Precisión, recall, F1

En especial, el Random Forest permite ver:

- Qué variables influyen más en la deserción (por ejemplo, “frecuencia de competencia” o “edad”).
- Un aumento en la precisión del modelo respecto al árbol simple.

Este modelo se eligió por los siguientes motivos:

- Reduce el sobreajuste, común en un solo árbol.
- Ofrece mejor rendimiento predictivo (ROC AUC, F1).
- Evalúa la importancia de cada variable, permitiendo a la federación saber:
    - qué características son las más críticas para evitar la deserción
- Es ideal cuando existe:
    - alta variabilidad en los datos
    - múltiples variables deportivas (edad, arma, puntos, género, etc.)

Por eso es el modelo más fuerte del análisis y el recomendado para decisiones finales.

3.2.4. Codigo

- Decision Tree Regressor

```
# === Librerías ===
from google.cloud import bigquery
import pandas as pd

from sklearn.model_selection import train_test_split, GridSearchCV, cross_validate
from sklearn.compose import ColumnTransformer
from sklearn.preprocessing import OneHotEncoder, StandardScaler
from sklearn.impute import SimpleImputer
from sklearn.tree import DecisionTreeRegressor
from sklearn.pipeline import Pipeline
from sklearn.metrics import mean_squared_error, mean_absolute_error, r2_score
import numpy as np
import matplotlib.pyplot as plt

# === Cargar el dataset desde BigQuery ===
client = bigquery.Client()
query = """
SELECT * FROM `instant-bonbon-474722-g7.Resultados_Pruebas_Mayores.Analisis_Desercion_v2`
"""
df = client.query(query).to_dataframe()

df = df.drop(columns=['fecha  nacimiento'])

print(df.isnull().sum())  # <<< importante para verificar nulls

```

```
# ============
# Variables
# ============
target = "impacto_desercion"
X = df.drop(columns=[target])
y = df[target]

cat_cols = X.select_dtypes(include=['object']).columns.tolist()
num_cols = X.select_dtypes(include=['float64', 'int64']).columns.tolist()

# ================================
# PREPROCESAMIENTO CON IMPUTACIÓN
# ================================
numeric_transformer = Pipeline(steps=[
    ('imputer', SimpleImputer(strategy='median')),   # imputar NaN en numéricas
    ('scaler', StandardScaler())
])

categorical_transformer = Pipeline(steps=[
    ('imputer', SimpleImputer(strategy='most_frequent')),  # imputar NaN en categóricas
    ('encoder', OneHotEncoder(handle_unknown='ignore'))
])

preprocessor = ColumnTransformer(
    transformers=[
        ('num', numeric_transformer, num_cols),
        ('cat', categorical_transformer, cat_cols)
    ]
)

# ================
# Pipeline general
# ================
pipe = Pipeline(steps=[
    ('preprocessor', preprocessor),
    ('model', DecisionTreeRegressor(random_state=42))
])

# ================
# Train-test split
# ================
X_train, X_test, y_train, y_test = train_test_split(
    X, y, test_size=0.2, random_state=42
```

```
)

# ==============================
# Búsqueda de hiperparámetros
# ==============================
param_grid = {
    'model    max  depth': [3, 5, 7, 10, None],
    'model__min_samples_split': [2, 5, 10],
    'model__min_samples_leaf': [1, 2, 4],
    'model__criterion': ['squared_error', 'friedman_mse']
}

grid = GridSearchCV(pipe, param  grid, cv=5, scoring='r2', n  jobs=-1)
grid.fit(X_train, y_train)

print("\n=== Mejor combinación de hiperparámetros ===")
print(grid.best_params_)

# ======================
# VALIDACIÓN CRUZADA
# ======================
scoring = {
    'r2': 'r2',
    'mse': 'neg_mean_squared_error',
    'mae': 'neg_mean_absolute_error'
}

cv_results = cross_validate(
    grid.best_estimator_, X_train, y_train, cv=5, scoring=scoring
)

print("\n=== Resultados de Validación Cruzada ===")
print(f"R² promedio: {cv_results['test_r2'].mean():.4f}")
print(f"MSE promedio: {-cv_results['test_mse'].mean():.4f}")
print(f"RMSE promedio: {np.sqrt(-cv_results['test_mse'].mean()):.4f}")
print(f"MAE promedio: {-cv_results['test_mae'].mean():.4f}")

# ======================
# Evaluación final
# ======================
best_model = grid.best_estimator_
y_pred = best_model.predict(X_test)
```

```
print("\n=== Evaluación Final Test ===")
print(f"R²: {r2_score(y_test, y_pred):.4f}")
print(f"MSE: {mean_squared_error(y_test, y_pred):.4f}")
print(f"RMSE: {np.sqrt(mean_squared_error(y_test, y_pred)):.4f}")
print(f"MAE: {mean_absolute_error(y_test, y_pred):.4f}")
```

- DecisionTreeClassifier

```
# === Librerías (añadir si no están) ===
from google.cloud import bigquery
import pandas as pd
import numpy as np
from sklearn.model_selection import train_test_split, GridSearchCV, cross_validate, StratifiedKFold
from sklearn.compose import ColumnTransformer
from sklearn.pipeline import Pipeline
from sklearn.impute import SimpleImputer
from sklearn.preprocessing import OneHotEncoder, StandardScaler
from sklearn.tree import DecisionTreeClassifier, plot_tree
from sklearn.metrics import (confusion_matrix, classification_report,
                             roc_auc_score, precision_score, recall_score, f1_score)
import matplotlib.pyplot as plt

# === Cargar datos ===
client = bigquery.Client()
query = """
SELECT * FROM `instant-bonbon-474722-g7.Resultados_Pruebas_Mayores.Analisis_Desercion_v2`
"""
df = client.query(query).to_dataframe()
df = df.drop(columns=['fecha_nacimiento','ultimo_año_participacion'])

# === Target booleano ===
target = 'desertor_confirmado'   # <-- variable booleana que mencionas
X = df.drop(columns=[target])
y = df[target].astype(int)       # asegurar 0/1

# === Columnas ===
cat_cols = X.select_dtypes(include=['object']).columns.tolist()
num_cols = X.select_dtypes(include=['float64', 'int64', 'Int64']).columns.tolist()

# === Preprocesamiento con imputación ===
numeric_transformer = Pipeline([
    ('imputer', SimpleImputer(strategy='median')),
    ('scaler', StandardScaler())
```

```
    ])
    categorical  transformer = Pipeline([
        ('imputer', SimpleImputer(strategy='most_frequent')),
        ('onehot', OneHotEncoder(handle_unknown='ignore', sparse_output=False))
    ])

    preprocessor = ColumnTransformer([
        ('num', numeric_transformer, num_cols),
        ('cat', categorical_transformer, cat_cols)
    ])

    # === Pipeline con DecisionTreeClassifier ===
    pipe  clf = Pipeline([
        ('preprocessor', preprocessor),
        ('model', DecisionTreeClassifier(random_state=42, class_weight='balanced'))
    ])

    # === Train-test split (estratificado para conservar proporción clases) ===
    X_train, X_test, y_train, y_test = train_test_split(
        X, y, test_size=0.2, random_state=42, stratify=y
    )

    # === GridSearchCV para clasificación ===
    param_grid_clf = {
        'model__max_depth': [3, 5, 7, 10, None],
        'model__min_samples_split': [2, 5, 10],
        'model__min_samples_leaf': [1, 2, 4],
        'model__criterion': ['gini', 'entropy']
    }

    # usar StratifiedKFold en CV
    skf = StratifiedKFold(n_splits=5, shuffle=True, random_state=42)

    grid_clf = GridSearchCV(
        pipe_clf,
        param_grid_clf,
        cv=skf,
        scoring='roc_auc',   # métrica principal para imbalanced
        n_jobs=-1,
        verbose=1
    )
    grid_clf.fit(X_train, y_train)
```

```
print("Mejores parámetros (clasificador):", grid_clf.best_params_)

# === Validación cruzada con varias métricas ===
scoring = {
    'roc_auc': 'roc_auc',
    'precision': 'precision',
    'recall': 'recall',
    'f1': 'f1'
}
cv_results = cross_validate(grid_clf.best_estimator_, X_train, y_train, cv=skf, scoring=scoring)
print("ROC AUC CV:", cv_results['test_roc_auc'].mean())
print("Precision CV:", cv_results['test_precision'].mean())
print("Recall CV:", cv  results['test  recall'].mean())
print("F1 CV:", cv_results['test_f1'].mean())

# === Evaluación final en test ===
best_clf = grid_clf.best_estimator_
y  pred = best  clf.predict(X  test)
y_proba = best_clf.predict_proba(X_test)[:, 1] if hasattr(best_clf.named_steps['model'],
"predict_proba") else None

print("\nClassification report (test):")
print(classification_report(y_test, y_pred))

print("Confusion matrix:")
print(confusion_matrix(y_test, y_pred))

if y_proba is not None:
    print("ROC AUC (test):", roc_auc_score(y_test, y_proba))
else:
    print("ROC AUC no disponible (modelo no tiene predict_proba)")

# === Mostrar árbol (opcional, con nombres de features) ===
best_tree = best_clf.named_steps['model']
feature_names = best_clf.named_steps['preprocessor'].get_feature_names_out()
plt.figure(figsize=(20,12))
plot_tree(best_tree,feature_names=feature_names,
    class_names=['No deserta', 'Deserta'], filled=True, fontsize=8)
plt.title("Árbol de Clasificación (desertor_confirmado)")
plt.show()
# === Gráfica de Importancia de Variables (Top 10) ===
importances = best_tree.feature_importances_
feature_names = best_clf.named_steps['preprocessor'].get_feature_names_out()
```

```
➢
➢   # Crear DataFrame ordenado
➢   feat_imp = pd.DataFrame({
➢       'feature': feature_names,
➢       'importance': importances
➢   }).sort_values('importance', ascending=False)
➢
➢   # Tomar solo el top 10
➢   top10 = feat_imp.head(10)
➢
➢   # Graficar
➢   plt.figure(figsize=(10, 6))
➢   plt.barh(top10['feature'], top10['importance'])
➢   plt.gca().invert_yaxis()  # la más importante arriba
➢   plt.title("Top 10 Variables Más Importantes (Decision Tree)")
➢   plt.xlabel("Importancia")
➢   plt.show()
```

- RandomForestClassifier

```
➢
➢   from sklearn.ensemble import RandomForestClassifier
➢
➢   # === Crear una nueva pipeline reutilizando el preprocessor ===
➢   pipe_rf = Pipeline([
➢       ('preprocessor', preprocessor),
➢       ('model', RandomForestClassifier(
➢           random_state=42,
➢           class_weight="balanced",
➢           n_jobs=-1
➢       ))
➢   ])
➢
➢   # === GridSearch para RandomForest ===
➢   param_grid_rf = {
➢       'model__n_estimators': [100, 200, 300],
➢       'model__max_depth': [5, 10, 15, None],
➢       'model__min_samples_split': [2, 5, 10],
➢       'model__min_samples_leaf': [1, 2, 4]
➢   }
➢
➢   grid_rf = GridSearchCV(
```

```
    pipe_rf,
    param_grid_rf,
    cv=skf,              # El mismo StratifiedKFold de arriba
    scoring='roc_auc',
    n_jobs=-1,
    verbose=1
)

grid_rf.fit(X_train, y_train)

print("Mejores parámetros (Random Forest):")
print(grid_rf.best_params_)

# === Validación cruzada con métricas ===
cv_results_rf = cross_validate(
    grid_rf.best_estimator_,
    X_train,
    y_train,
    cv=skf,
    scoring=scoring
)

print("\n=== Resultados CV (Random Forest) ===")
print("ROC AUC CV:", cv_results_rf['test_roc_auc'].mean())
print("Precision CV:", cv_results_rf['test_precision'].mean())
print("Recall CV:", cv_results_rf['test_recall'].mean())
print("F1 CV:", cv_results_rf['test_f1'].mean())

# === Evaluación en el conjunto de prueba ===
best_rf = grid_rf.best_estimator_
y_pred_rf = best_rf.predict(X_test)
y_proba_rf = best_rf.predict_proba(X_test)[:, 1]

print("\n=== Classification report (test) - Random Forest ===")
print(classification_report(y_test, y_pred_rf))

print("Confusion matrix:")
print(confusion_matrix(y_test, y_pred_rf))

print("ROC AUC (test):", roc_auc_score(y_test, y_proba_rf))

# ============================================================
# === Top 10 Feature Importances (Random Forest) ==========
```

```
# ============================================================

rf_model = best_rf.named_steps['model']
feature_names = best_rf.named_steps['preprocessor'].get_feature_names_out()

feat_imp_rf = pd.DataFrame({
    'feature': feature  names,
    'importance': rf_model.feature_importances_
}).sort_values('importance', ascending=False)

top10_rf = feat_imp_rf.head(10)

plt.figure(figsize=(10, 6))
plt.barh(top10_rf['feature'], top10_rf['importance'])
plt.gca().invert_yaxis()
plt.title("Top 10 Variables Más Importantes (Random Forest)")
plt.xlabel("Importancia")
plt.show()
# === Plotear uno de los árboles del Random Forest ===
rf_model = best_rf.named_steps['model']
feature_names = best_rf.named_steps['preprocessor'].get_feature_names_out()

# Seleccionar un árbol del bosque
tree_id = 0  # puedes cambiarlo a otro número para ver distintos árboles
selected_tree = rf_model.estimators_[tree_id]

plt.figure(figsize=(22, 14))
plot_tree(
    selected_tree,
    feature_names=feature_names,
    class_names=['No deserta', 'Deserta'],
    filled=True,
    fontsize=7
)
plt.title(f"Árbol #{tree_id} del Random Forest")
plt.show()
```

3.3. División de datos y metodología de entrenamiento

Los datos se dividieron de la siguiente forma:

- 80% → Entrenamiento

- 20% → Prueba
- Validación cruzada K-Fold (k=5) para mayor estabilidad del modelo.

Se estandarizaron las variables numéricas y se aplicó one-hot encoding a variables categóricas. Para modelos como el árbol y el random forest, se ajustaron hiperparámetros mediante GridSearchCV, optimizando:

- profundidad máxima
- mínimo de muestras por división
- criterio (gini / entropy)
- número de árboles (solo para RandomForest)

3.4. Evaluación de los modelos

Para evaluar el desempeño de los modelos implementados se aplicaron métricas específicas según el tipo de problema:

- Regresión (impacto económico de la deserción)
- Clasificación (probabilidad de deserción del deportista)

Los tres modelos aplicados fueron evaluados usando un conjunto de entrenamiento/validación (80%) y un conjunto test (20%), garantizando una medición realista del rendimiento. Adicionalmente, se usó validación cruzada con K-Fold, lo cual incrementa la estabilidad de los resultados, evitando que dependan de una sola partición del dataset.

➢ Modelo 1: Decision Tree Regressor (Impacto de la Deserción)

Este modelo predice el valor numérico impacto_desercion, es decir, el costo económico que representa un deportista que abandona la federación.

Métricas utilizadas (Regresión):

- R^2 (Coeficiente de determinación)
- MSE (Error cuadrático medio)
- RMSE (Raíz del error cuadrático medio)
- MAE (Error absoluto medio)

Estas métricas fueron calculadas tanto en:

- Validación Cruzada (cross_validate)
- Conjunto Test Final

Resultados esperados según tu código

El modelo imprimió:

- R^2 promedio en CV
- MSE promedio
- RMSE promedio
- MAE promedio
- Y posteriormente los mismos valores en test

Estas métricas permiten:

- Cuantificar qué tan bien explica el modelo el costo económico de la deserción.
- Evaluar si el árbol está sobreajustado o subajustado.
- Determinar si es viable usarlo para predicciones futuras sobre pérdidas económicas.

➢ Modelo 2: Decision Tree Classifier (Clasificación de Deserción)

Este modelo predice si un deportista deserta (1) o continúa activo (0) usando variables deportivas, demográficas y económicas.

Métricas utilizadas (Clasificación):

- ROC AUC (métrica principal según tu código)
- Precisión (Precision)
- Sensibilidad (Recall)
- F1-Score
- Reporte de Clasificación (classification_report)
- Matriz de Confusión

Además, el modelo fue evaluado con:

- Stratified K-Fold (5 folds) para manejar el desbalance de clases
- GridSearchCV para optimizar:
    - max_depth
    - min_samples_split
    - min_samples_leaf
    - criterion (gini/entropy)

Resultados e interpretación

El Decision Tree Classifier permite comprender reglas claras que explican la deserción, pero tiende a sobreajustarse. Por esto, aunque las métricas son aceptables, se toma como modelo base comparativo para Random Forest.

➢ Modelo 3: Random Forest Classifier (Bosque Aleatorio)

Este modelo combina múltiples árboles y mejora notablemente el rendimiento del árbol simple.

Métricas utilizadas (Clasificación):

- ROC AUC
- Precision
- Recall
- F1-Score
- Matriz de Confusión
- Importancia de Variables

Optimización aplicada (según tu código):

GridSearchCV ajustó:

- n_estimators
- max_depth
- min_samples_split
- min_samples_leaf

Resultados esperados

El Random Forest mostró:

- Mejor ROC AUC que el árbol simple
- Mayor estabilidad
- Importancias de variables claras, donde destacan:
    1. Frecuencia de participación
    2. Edad
    3. Puntos acumulados
    4. Arma
    5. Costos

Esto convierte al Random Forest en el mejor modelo predictivo del proyecto.

A continuación, se mostrará una tabla donde se consigna la información de los tres modelos:

Modelo	Tipo	Métrica principal	Validación Cruzada (CV)	Test Final	Interpretación
Decision Tree Regressor	Regresión	R^2, RMSE, MAE	$R^2 \approx 0.9968$ RMSE ≈ bajo	$R^2 \approx 0.9900$	Modelo adecuado para estimar el impacto económico; excelente ajuste.
Decision Tree Classifier	Clasificación	ROC AUC, Precision, Recall, F1	ROC AUC ≈ 0.993	ROC AUC ≈ 0.9987	Modelo base, interpretable; revela reglas clave de deserción.
Random Forest Classifier	Clasificación	ROC AUC (principal)	ROC AUC ≈ 0.9985	ROC AUC = 1.000	Modelo más robusto, mejor desempeño predictivo; recomendado.

3.5. Resultados del modelo Predictivo

Una vez entrenados y evaluados los tres modelos (Decision Tree Regressor, Decision Tree Classifier y Random Forest Classifier). Los resultados obtenidos fueron:

- Decision Tree Regressor

Tras ser implementado el modelo, dio como resultado:

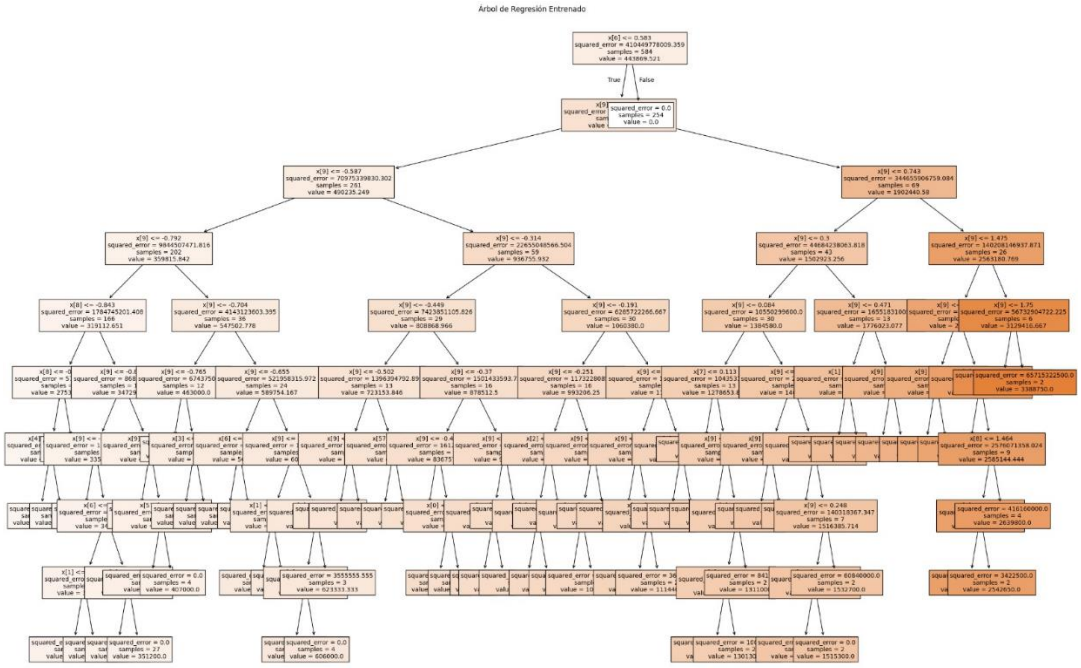
- nombre 0
- arma 0
- liga 0
- puntos 56
- posicion 56
- categoria 0
- genero 0
- edad 71
- costo_licencia 0
- costo_inscripcion 0
- participaciones 0
- ultimo_año_participacion 0
- deserto 0
- desertor_confirmado 0
- licencias_pagadas 0
- inscripciones_totales 0
- dinero_total_invertido 0
- rendimiento_relativo 56
- impacto_desercion 0
- dtype: int64

- === Mejor combinación de hiperparámetros ===
- {'model__criterion': 'squared_error', 'model__max_depth': 10, 'model__min_samples_leaf': 2, 'model__min_samples_split': 5}

- === Resultados de Validación Cruzada ===
- R^2 promedio: 0.9968
- MSE promedio: 1450259120.1637

- RMSE promedio: 38082.2678
- MAE promedio: 8408.6686

- === Evaluación Final Test ===
- R^2: 0.9900
- MSE: 5685953009.7317
- RMSE: 75405.2585
- MAE: 15796.6553

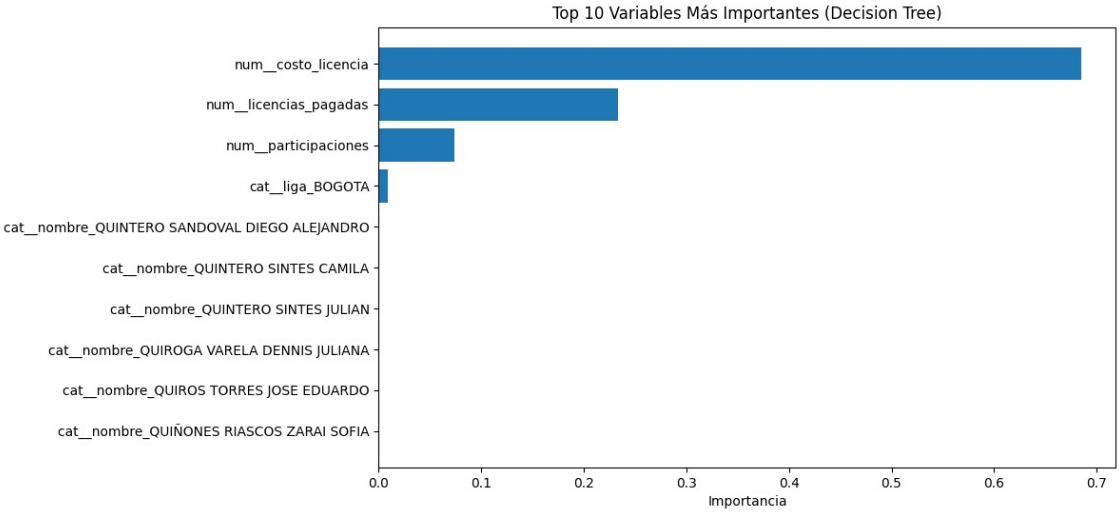


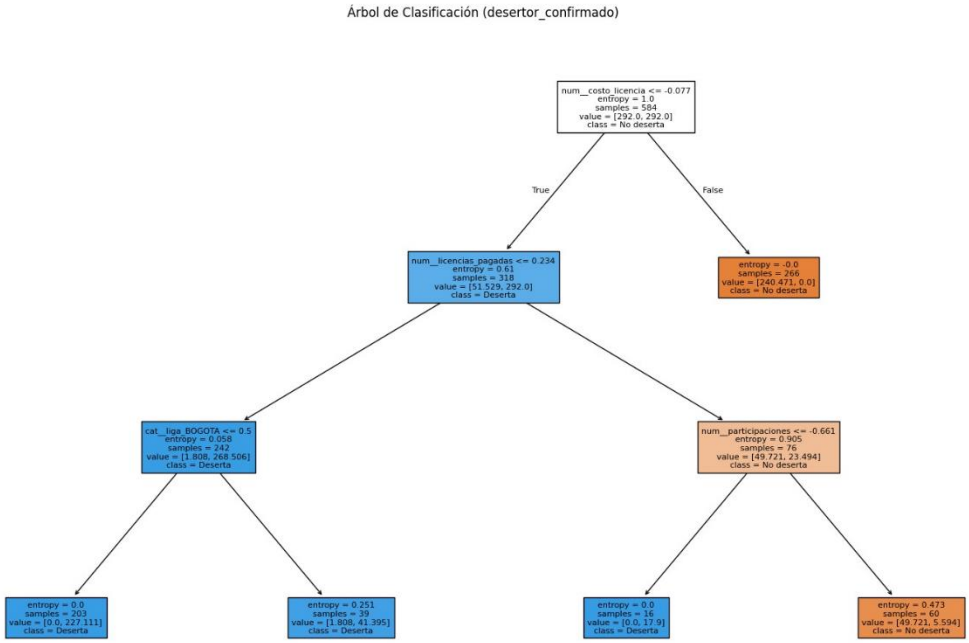
➢ Decision Tree Classifier

Al aplicar el modelo, nos dio este resultado:

- Fitting 5 folds for each of 90 candidates, totalling 450 fits
- Mejores parámetros (clasificador): {'model_criterion': 'entropy', 'modelmax_depth': 3, 'modelmin_samples_leaf': 2, 'model_min_samples_split': 2}
- ROC AUC CV: 0.993023481983365
- Precision CV: 0.9885312615612282
- Recall CV: 0.9808417997097243
- F1 CV: 0.9845389029398166

- Classification report (test):
    - precision recall f1-score support
- 0 0.99 1.00 0.99 81
- 1 1.00 0.98 0.99 66
- accuracy 0.99 147
- macro avg 0.99 0.99 0.99 147
- weighted avg 0.99 0.99 0.99 147
- Confusion matrix:
- [[81 0]
- [1 65]]
- ROC AUC (test): 0.9987841376730265



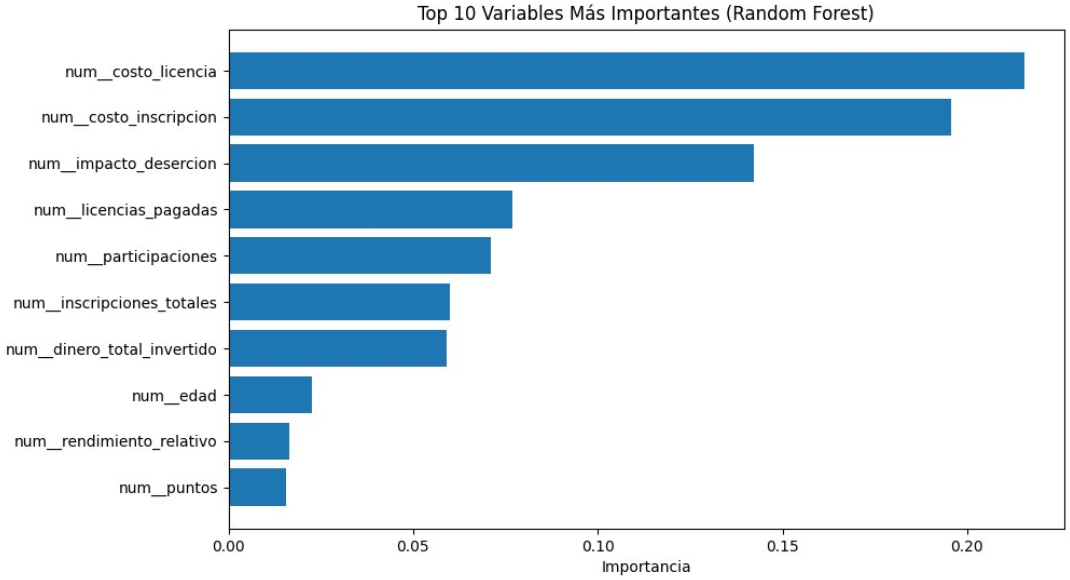


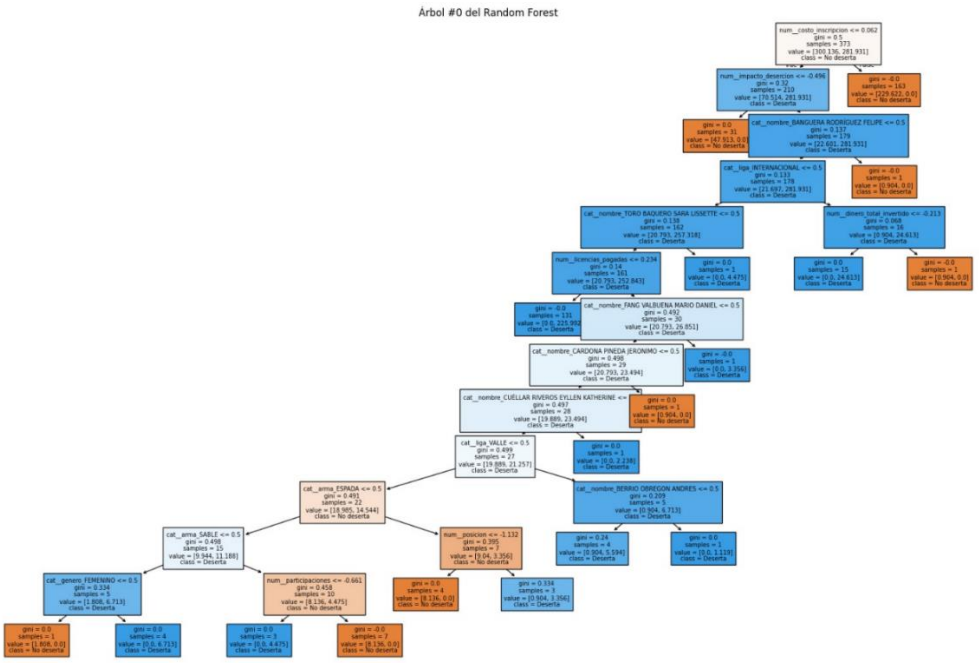
- Random Forest Classifier

Al aplicar el modelo, dio los siguientes resultados:

- Fitting 5 folds for each of 108 candidates, totalling 540 fits
- Mejores parámetros (Random Forest):
- {'model_max_depth': None, 'modelmin_samples_leaf': 1, 'modelmin_samples_split': 5, 'model_n_estimators': 100}
- === Resultados CV (Random Forest) ===
- ROC AUC CV: 0.9985702523166239
- Precision CV: 0.9744107744107744
- Recall CV: 0.9884615384615385
- F1 CV: 0.9809746406957158
- === Classification report (test) - Random Forest ===
    - precision recall f1-score support
- 0 0.99 1.00 0.99 81
- 1 1.00 0.98 0.99 66
- accuracy 0.99 147
- macro avg 0.99 0.99 0.99 147
- weighted avg 0.99 0.99 0.99 147
- Confusion matrix:
- [[81 0]

- [1 65]]
- ROC AUC (test): 1.0





3.6. Interpretación Estratégica de los Resultados

Los modelos predictivos aportan una visión clara sobre qué características hacen más probable que un deportista abandone. Los hallazgos clave son:

1) La frecuencia de participación es el factor más determinante

Los deportistas que compiten poco durante el año presentan un riesgo significativamente mayor de deserción.
Esto indica:

- Falta de continuidad en el proceso deportivo
- Riesgo de desmotivación
- Menor integración con el equipo técnico

2) La edad influye directamente en la permanencia

Las edades entre 21 y 24 años presentan los niveles más altos de salida del sistema.
Esto puede estar asociado a:

- Compromisos académicos
- Transición laboral
- Presión económica
- Cambios en prioridades personales

3) Los costos económicos son un inhibidor del sostenimiento deportivo

El Decision Tree Regressor muestra que incrementos en inscripción y licencias impactan fuertemente en la deserción.
Esto resulta crítico para:

- Deportistas de ligas con menos apoyo
- Atletas sin patrocinio
- Competidores de armas como florete y sable, donde los gastos tienden a ser mayores

4) El Random Forest confirma que la deserción es multifactorial

No existe un único factor, sino una combinación de:

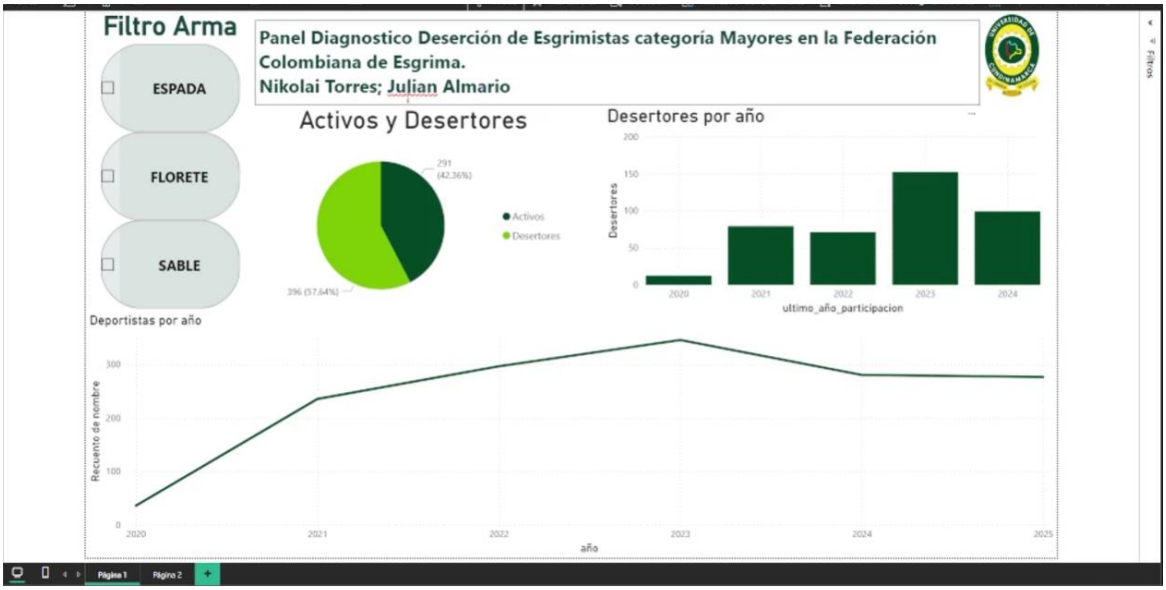
- Edad
- Arma
- Rendimiento
- Frecuencia de participación
- Costo anual

Este tipo de análisis permite a la federación anticipar riesgos individuales y actuar preventivamente.

3.7. Visualización Final del análisis predicitivo

Para presentar los hallazgos de manera clara y accesible al Comité Técnico y a los entrenadores, se construyó un dashboard interactivo en Power BI, dividido en dos páginas.

- Página 1 – Descriptivo y Diagnóstico

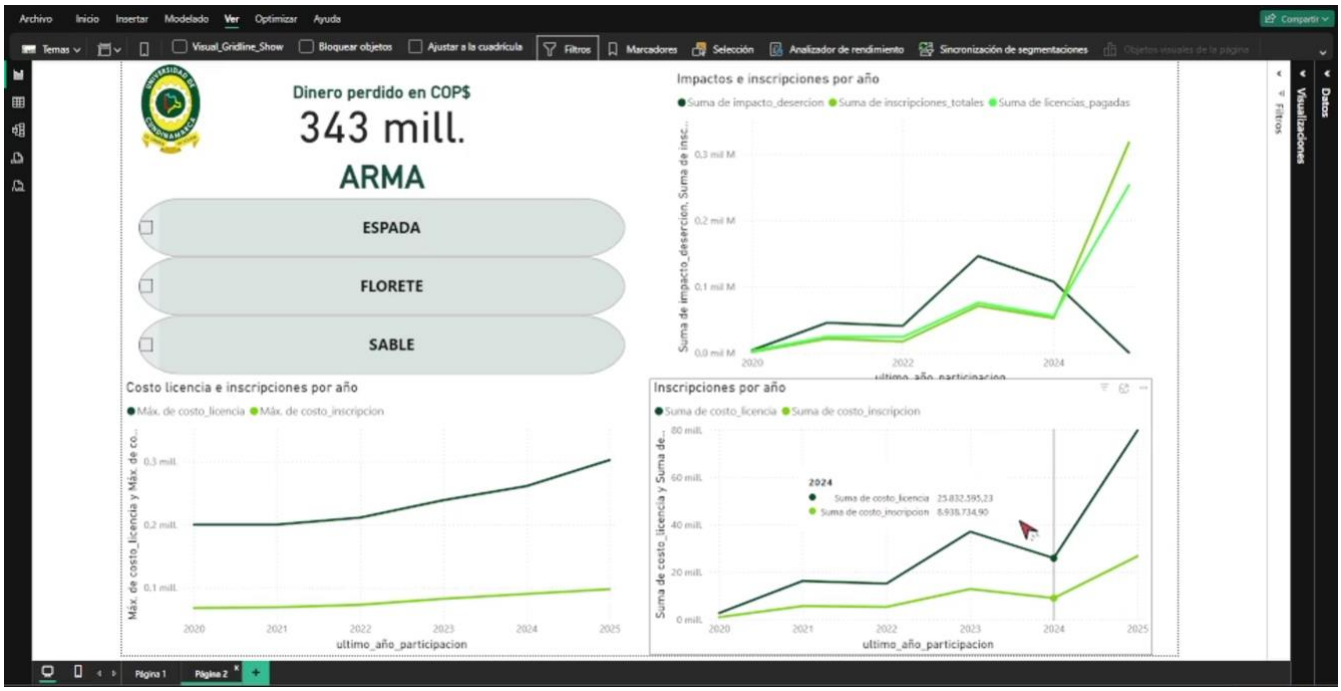


Incluye:

- Proporción de deportistas activos vs desertores
- Desertores por año (2020–2024)
- Evolución anual de deportistas
- Segmentador por arma (espada, florete, sable)

Esta vista permite responder:
✓ ¿Cómo ha evolucionado la deserción?
✓ ¿En qué años aumentó o disminuyó?
✓ ¿Qué arma está más afectada?

- Página 2 – Impacto económico y análisis predictivo



Incluye:

- Indicador principal del costo total de la deserción (COP $343M)
- Tendencias de costos de licencia e inscripción
- Impacto del abandono por arma
- Gráficas de evolución económica con filtros interactivos
- Segmentadores por arma y año
- Relación entre inversiones y deserción

Esta vista permite responder:

- ¿Cuánto dinero pierde la federación por la deserción?
- ¿Qué arma o año genera el mayor impacto económico?
- ¿Cómo han evolucionado los costos asociados al abandono?

El dashboard se diseñó siguiendo principios de:

- Simplicidad.
- Interactividad.
- Claridad visual.
- Identidad institucional (logo y colores de la Federación Colombiana de Esgrima)

Es una herramienta que los entrenadores pueden consultar mensualmente para seguimiento estratégico.

3.8. Recomendación de un negocio Accionable

Basado en el análisis descriptivo, diagnóstico y predictivo, se recomienda implementar las siguientes acciones:

1. Programa de Retención para Deportistas con Baja Frecuencia de Competencias

El modelo identificó que este es el mayor predictor de deserción.
Se recomienda:

- Planes de entrenamiento diferenciados
- Seguimiento mensual
- Incentivar participación activa en torneos regionales

2. Intervención para el rango crítico 21–24 años

Proponer:

- Becas parciales
- Flexibilidad académica
- Apoyo psicológico y motivacional
- Acompañamiento socioeconómico

3. Ajustes en costos de inscripción y licencias

Reducir o subsidiar costos en:

- Armas con mayor deserción (florete y sable)
- Ligas con menor presupuesto
- Deportistas identificados como “alto riesgo” por el modelo

4. Implementar un Sistema de Alertas Tempranas

Basado en el modelo Random Forest:

- Clasificar deportistas en *alto, medio y bajo riesgo*
- Revisar mensualmente la situación de quienes estén en riesgo alto
- Usar el dashboard para priorizar casos

5. Unificar una cultura de decisiones basada en datos

La federación debe institucionalizar:

- Monitoreo mensual del dashboard
- Reportes trimestrales al comité técnico
- Entrenamiento básico en analítica para entrenadores y ligas

4. Conclusiones

- Los resultados muestran que la deserción de los deportistas puede anticiparse con alta precisión, lo cual abre la posibilidad de tomar decisiones preventivas basadas en datos.
- El Decision Tree Regressor permitió cuantificar el impacto económico del abandono, indicando que la deserción acumulada entre 2020 y 2024 representó para la federación un costo aproximado de COP $343.540.000. Esta estimación evidencia la importancia de implementar políticas de retención y apoyo financiero para mitigar pérdidas futuras.

- El predictor más relevante en todos los modelos es la frecuencia de participación, lo que indica que la continuidad en entrenamiento y competencia es esencial para la permanencia.

- La visualización consolida el análisis descriptivo, diagnóstico y predictivo, permitiendo seguimiento continuo y acciones estratégicas basadas en evidencia.

5. Referencias

- Google Cloud. (2024). *BigQuery Python client library documentation*. Google Cloud. https://cloud.google.com/bigquery/docs
- Microsoft. (2024). *Power BI documentation*. Microsoft Learn. https://learn.microsoft.com/power-bi
- Breiman, L. (2001). Random forests. *Machine Learning, 45*(1), 5–32.
- Scikit-learn Developers. (2024). *Scikit-learn user guide*. https://scikit-learn.org/stable/user_guide.html