Predicción de podios en carreras de Fórmula 1

Trabajo Final - Bases de datos masivas - UNLu

Rodriguez, Jose Emanuel
joseemaro@hotmail.com

30 de enero de 2025

1. Resumen

El rendimiento esperado de un piloto en una carrera de Fórmula 1 es complejo de predecir, influenciado por múltiples factores. Para abordar este desafío, se analizaron datos desde el año 2000 en adelante, recolectados de APIs como EargastF1[1], VisualCrossing [2]y sitios web como Wikipedia[3], que fueron integrados y enriquecidos mediante técnicas de preprocesamiento y limpieza de datos. A través de ingeniería de *features*, se generaron variables predictoras clave como el promedio de podios por piloto y escudería, condiciones climáticas durante la carrera y características del circuito (longitud y cantidad de curvas).

En el modelado, se utilizaron algoritmos clásicos de clasificación, incluidos *Logistic Regression, Random Forest* y *Support Vector Machines*, optimizados con *GridSearchCV* y validados mediante *StratifiedKFold*. Técnicas como normalización y análisis de correlación se emplearon para preparar las variables, mientras que SMOTE abordó el desbalance en la clase objetivo. Los modelos fueron evaluados con métricas como *accuracy*, F1-score y AUC-ROC, logrando mejoras significativas mediante la inclusión de variables derivadas. El modelo más preciso, con un *accuracy* cercano al 97%, destacó en ser apto para predecir la variable objetivo *(Top 3 finish)*. Finalmente, se evaluó su robustez temporal y se simuló un campeonato completo con las predicciones, validando su robustez para replicar rankings históricos de pilotos y escuderías.

2. Introducción

La Fórmula 1 representa el pináculo del automovilismo y un desafío significativo tanto para los pilotos como para los equipos. La competencia no solo depende de la habilidad del conductor, sino también de múltiples factores que influyen en el desempeño en cada carrera. Al analizar las estadísticas y resultados históricos, se observa que factores como la calidad del auto, el diseño del circuito y las condiciones climáticas tienen un impacto directo en los resultados. Sin embargo, la complejidad del problema aumenta significativamente cuando se consideran variables menos evidentes pero igualmente cruciales, como las estrategias implementadas por los equipos, el rendimiento de los rivales bajo diferentes circunstancias y el historial acumulado de desempeño de los pilotos a lo largo de la temporada. Estas múltiples variables interactúan de manera dinámica, haciendo que la predicción de resultados en Fórmula 1 sea un problema desafiante.

En este contexto, predecir si un piloto logrará posicionarse en el podio en una carrera es un problema altamente complejo. Factores como la posición en la parrilla, el historial de rendimiento en circuitos similares, o incluso variables externas como la probabilidad de lluvia, añaden un nivel de incertidumbre que hace que este análisis sea multifacético [1]. Además, el rendimiento puede variar significativamente entre temporadas debido a cambios en las regulaciones técnicas y las condiciones de los equipos.

[1] *https://ergast.com/mrd/*
[2] *https://www.visualcrossing.com*
[3] *https://es.wikipedia.org/wiki/Wikipedia:Portada*

En definitiva, la predicción de podios en Fórmula 1 no es un desafío trivial. Como en cualquier deporte, el éxito depende de la combinación de talento, tecnología y estrategia. En el caso de la Fórmula 1, las fluctuaciones inherentes a cada temporada y los avances tecnológicos constantes complican aún más el análisis. Esto plantea la necesidad de modelos predictivos que no solo sean precisos, sino también adaptables a las dinámicas de un deporte en constante evolución.

Respecto a la importancia de una predicción precisa, se encuentra la perspectiva del equipo que busca maximizar sus posibilidades de asegurar la pole position, y la del piloto que necesita demostrar su mejor desempeño. Además, una predicción acertada permite identificar si un piloto tiene un rendimiento adecuado en condiciones específicas, si está siendo superado por rivales con estrategias más eficaces, o si existe una oportunidad para optimizar la configuración del auto y asegurar una ventaja competitiva clave en la clasificación.

Por lo tanto, habiendo expuesto algunas dimensiones de la dificultad del problema, es necesario mencionar que existen ciertas concepciones, tal vez acertadas, sobre los factores determinantes para lograr la *pole position* en la Fórmula 1. En especial, se asocia la habilidad del piloto y las ventajas técnicas del constructor como elementos clave para este logro. De hecho, estudios recientes han analizado la contribución relativa de la destreza del piloto y las ventajas del constructor en los resultados de las carreras, destacando la importancia de ambos factores en el éxito en la clasificación. [2] En este trabajo se podrá verificar la validez de estas afirmaciones.

No existe un *dataset* armado previamente que pueda ser utilizado como punto de partida para la predicción, sin embargo esta limitación puede ser sorteada recolectando e integrando datos de múltiples fuentes, obteniendo la información necesaria.

2.1. Trabajos Relacionados

Por lo tanto, habiendo expuesto algunas dimensiones de la dificultad del problema, es necesario mencionar que existen ciertas aproximaciones, tal vez acertadas, sobre los factores determinantes en el desempeño en Fórmula 1. En especial, se ha investigado la relación entre la habilidad del piloto y la ventaja tecnológica de los constructores como elementos clave en los resultados de las carreras. De hecho, trabajos recientes han explorado esta problemática desde diversas perspectivas.

Van Kesteren y Bergkamp (2022)[3] utilizan un modelo bayesiano multinivel para analizar resultados históricos de la Fórmula 1, logrando separar el impacto relativo de la habilidad del piloto frente a la ventaja del constructor. Este enfoque permite identificar tendencias consistentes a lo largo de las temporadas.

Por otro lado, Fry et al. (2023)[4] propone un método innovador para identificar rápidamente a los pilotos más rápidos en una temporada, basándose en una dualidad tiempo-rango. Este método proporciona *insights* inmediatos para analizar el rendimiento de los pilotos carrera a carrera.

Otro trabajo interesante es el de Weiss y Behl (2020)[5] explora la aplicación de trayectorias parametrizadas en simulaciones de carreras. Aunque orientado a vehículos autónomos, este trabajo sienta bases importantes para comprender cómo las características del circuito influyen en el rendimiento.

Por otro lado, Sicoie (2020)[6] presenta un marco predictivo basado en algoritmos supervisados para estimar los ganadores de carreras y las posiciones finales en el campeonato. Su estudio enfatiza la importancia de seleccionar características representativas y combina diversas metodologías para optimizar la precisión de los modelos. Ambos trabajos destacan enfoques complementarios para analizar y predecir resultados en la Fórmula 1, uno desde una perspectiva probabilística y el otro mediante aprendizaje supervisado, contribuyendo significativamente al entendimiento de los factores clave que determinan el rendimiento en este deporte.

En este trabajo se busca ampliar el análisis hacia nuevas perspectivas que integren tanto datos históricos como predicciones basadas en aprendizaje automático.

2.2. Objetivos

El objetivo general de este trabajo es predecir los resultados de las carreras de Fórmula 1 mediante un enfoque basado en aprendizaje automático y análisis de datos. Para ello, se definieron tres objetivos específicos:

- Crear un *dataset* consolidado a partir de múltiples fuentes que recopilan datos históricos y actuales sobre las carreras de Fórmula 1.
- Enriquecer el *dataset* mediante ingeniería de características para incluir variables relevantes como condiciones climáticas, longitud del circuito y características de los pilotos y equipos.
- Ajustar modelos de predicción que permitan estimar con precisión la probabilidad de finalizar en el podio.

El primer objetivo permite obtener un panorama amplio y actualizado de los datos necesarios, superando las limitaciones de cobertura y calidad de los *datasets* existentes. El segundo se enfoca en maximizar la riqueza de los datos, creando variables clave como el porcentaje de podios por equipo, las posiciones promedio y métricas relacionadas con el clima y los circuitos. Finalmente, el tercer objetivo aborda el desarrollo de modelos predictivos optimizados que, utilizando técnicas avanzadas como *GridSearchCV* [9] y SMOTE (Synthetic Minority Over-sampling Technique) [10], permitan descubrir patrones y tendencias útiles para entender los factores determinantes del rendimiento en Fórmula 1.

3. Materiales y métodos

En esta sección se describen las principales etapas del trabajo, las cuales dan sustento a los objetivos previamente mencionados. La primera etapa involucra la construcción del *dataset*, incorporando datos desde múltiples fuentes, y el tratamiento de datos que fue realizado. La segunda etapa consta de los ajustes realizados a los modelos, las técnicas incorporadas y la evaluación de las configuraciones utilizadas.

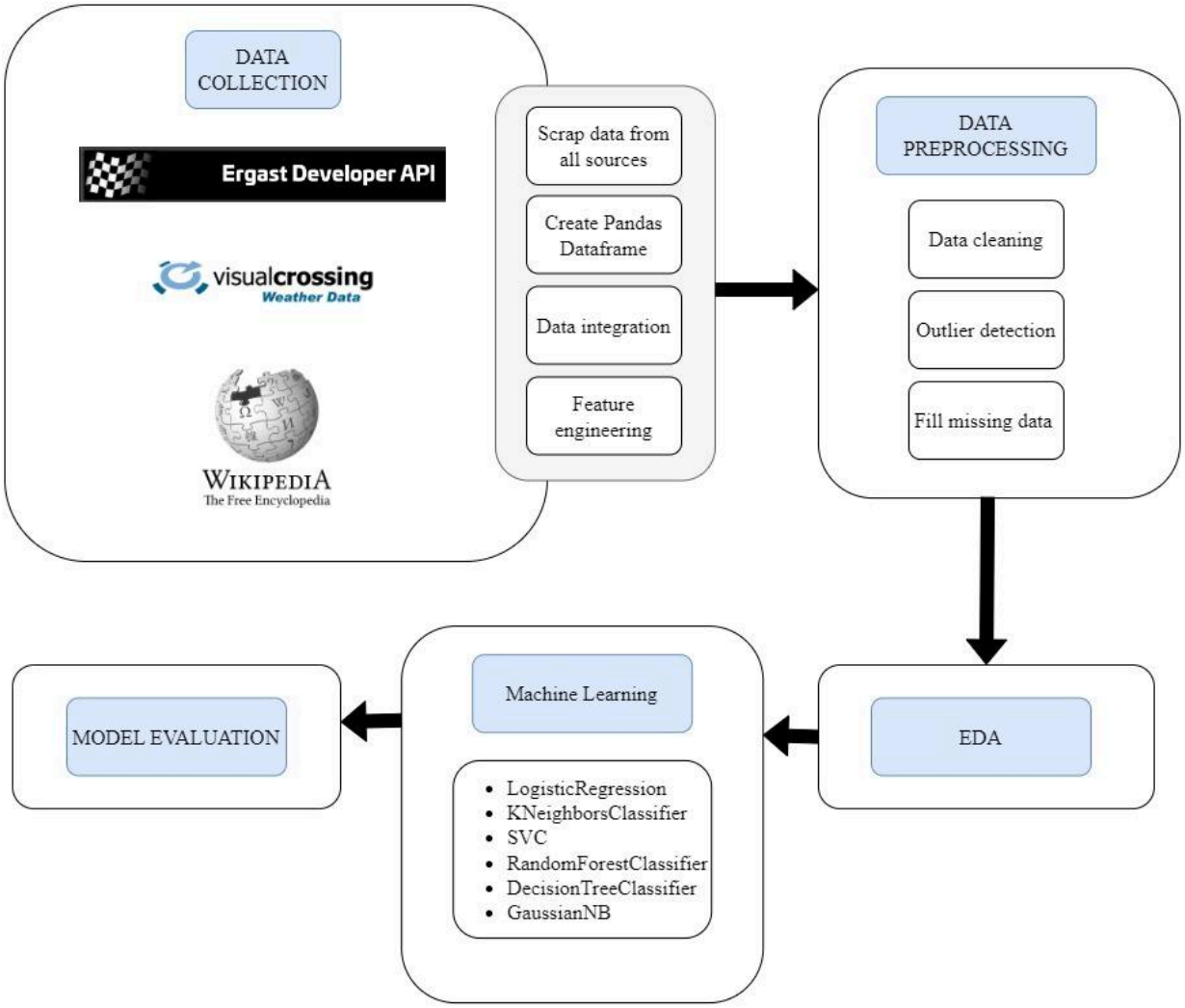


Figura 1: Esquema de la metodología

3.1. Construcción del dataset

El acceso a los datos telemétricos de la Fórmula 1 puede ser complejo debido a que de estos datos surgen las estrategias que usan las escuderías para optimizar el rendimiento del auto, entender su comportamiento en diferentes condiciones, y tomar decisiones estratégicas durante la carrera. Sin embargo, existe información de público acceso la cual puede ser recopilada de manera de poder consolidar un *dataset* que permita su posterior explotación. La estrategia implementada en este trabajo fue recopilar los datos disponibles en la API *Eargast* para conformar una primera instancia del *dataset*. Se obtuvieron los datos históricos consumiendo esta API, recopilando los datos de los años 2000-2024.

3.1.1. Recopilación de datos

El primer objetivo planteado en este trabajo indica que es necesario recopilar datos de múltiples fuentes heterogéneas, almacenarlos e integrarlos. En esta sección se brindan más detalles sobre la recolección de los datos y el origen de los mismos.

3.1.1.1. Consumo de la API Eargast

La API *Eargast*, de acceso público, proporciona un mecanismo eficiente para la consulta de datos mediante un endpoint que permite la aplicación de múltiples filtros de búsqueda. Esta herramienta abarca información desde el año 1950 hasta 2024. No obstante, para los fines de este estudio, se limitará la recuperación de datos al período comprendido entre el año 2000 y 2024.

Como parte del proceso de recuperación de datos, se obtuvieron un total de 14 conjuntos de datos, los cuales fueron integrados en un único dataset consolidado. Para llevar a cabo la integración, se utilizó como base dataset denominado *results*, que contenía un total de 9,839 registros. A partir de esta base, se incorporaron campos específicos provenientes de los restantes conjuntos de datos con el objetivo de enriquecer el dataset principal mediante la adición de nuevas variables. Los datasets recuperados se observan en la tabla 1.

Tabla 1: Datasets recuperados a partir de la API

Nombre del dataset	Cantidad de columnas
circuits_df	9
constructor_results_df	5
constructor_standings_df	7
constructors_df	5
driver_standings_df	7
drivers_df	9
lap_times_df	6
pit_stops_df	7
qualifying_df	9
races_df	18
results_df	18
seasons_df	2
sprint_results_df	16
status_df	2

Los campos presentes en cada uno de estos datasets se detallan en el Anexo A.

3.1.1.2. Integración de datos

Los datos utilizados en este estudio fueron obtenidos, en primera instancia, a partir de una única fuente: la API de *Ergast*. Aunque esta fuente proporciona una gran cantidad de información, la aplicación de técnicas avanzadas, conocimiento del dominio y enfoques creativos permite derivar nuevos atributos a partir de los datos disponibles.

Adicionalmente, para enriquecer el conjunto de datos, se integraron otras fuentes de información. En particular, se empleó la API de *VisualCrossing* para recuperar datos climáticos asociados a las carreras y se realizó un proceso de scraping HTML en la página de *Wikipedia* para extraer información complementaria sobre los circuitos.

3.1.1.2.1. Scraping HTML

Una estrategia para enriquecer el conjunto de datos consiste en incluir información relacionada con el circuito donde se desarrolla cada carrera. Específicamente, se incorporaron datos sobre la complejidad del circuito, determinados por dos características clave: la longitud del circuito y la cantidad de curvas.

Para obtener esta información, se llevó a cabo un proceso de *scraping* HTML en la página de *Wikipedia*. El conjunto de datos *Circuits* contiene las URLs correspondientes a los circuitos utilizados en la Fórmula 1, lo que permitió extraer los valores requeridos de manera automatizada.

Se añadieron dos atributos numéricos: *Length*, que representa la longitud del circuito en kilómetros, y *Turns*, que indica la cantidad de curvas por vuelta en cada circuito. Esta información complementa el análisis al proporcionar métricas relevantes sobre la complejidad estructural de los circuitos.

3.1.1.2.2. Consumo de la API VisualCrossing

Otra forma de enriquecer el *dataset* es agregar información relacionada al clima de las carreras, en el caso particular de este trabajo, información relacionada a la lluvia durante la carrera. Para esto se utilizó la API de *VisualCrossing*, la cual permite recuperar datos históricos sobre condiciones climáticas a partir de las coordenadas del lugar.

El atributo agregado al dataset fue *Rainy* el cual indica la ocurrencia de lluvia en la hora de la carrera o en instancias previas a la misma, siendo este un atributo de tipo binario.

3.1.1.2.3. Ingeniería de features

Las variables generadas a partir del dataset para enriquecer al mismo se observan en la tabla 2, los detalles de cada variable se detallan en el anexo D.

Tabla 2: Variables generadas

Variable	Tipo de dato
Top 3 finish	Binario
Porcentaje de terminar en la pole del conductor del año anterior	Numérico
Porcentaje de terminar en la pole de la escudería del año anterior	Numérico
Porcentaje de terminar en la pole del conductor desde principio de temporada	Numérico
Porcentaje de terminar en la pole de la escudería desde principio de temporada	Numérico
Posición promedio del conductor del año pasado	Numérico
Posición promedio de la escudería del año pasado	Numérico
Posición promedio del conductor desde el principio de temporada	Numérico
Posición promedio de la escudería desde el principio de temporada	Numérico
Posición del conductor en la carrera previa	Numérico
Promedio ponderado en base a cantidad de curvas	Numérico
Promedio ponderado en base a largo de pista	Numérico
Número de corredor para esa escudería	Numérico
Promedio de puntos en las últimas 10 carreras	Numérico

De esta manera, una vez integrados todos los datos el dataset que se va a utilizar para el modelado quedó conformado por los campos que se observan en la tabla 3.

Tabla 3: Atributos del dataset utilizado para el modelado.

Lista de atributos	
Atributo	Tipo de dato
year', 'round', 'circuitId', 'rainy', 'Turns', 'Length', 'driverId', 'constructorId', 'grid', 'Driver Top 3 Finish Percentage (Last Year)', 'Constructor Top 3 Finish Percentage (Last Year)', 'Driver Top 3 Finish Percentage (This Year till last race)', 'Constructor Top 3 Finish Percentage (This Year till last race)', 'Driver Avg position (Last Year)', 'Constructor Avg position (Last Year)', 'Driver Average Position (This Year till last race)', 'Constructor Average Position (This Year till last race)', 'driver_age', 'nationality_encoded', 'wins_cons', 'points_cons_champ', 'wins', 'points_driver_champ', 'laps', 'statusId', 'Weighted_Top_3_Probability', 'Weighted_Top_3_Prob_Length', 'position_previous_race', 'nro_cond_escuderia', 'raceId', 'points', 'prom_points_10'	Numérico

Top 3 Finish'	Booleano
'Date'	Categórico

3.1.2. Caracterización del Dataset

En la figura 2 se observa la distribución de la variable *target* binaria *Top 3 Finish* que indica si un conductor finaliza o no en el podio en una carrera.

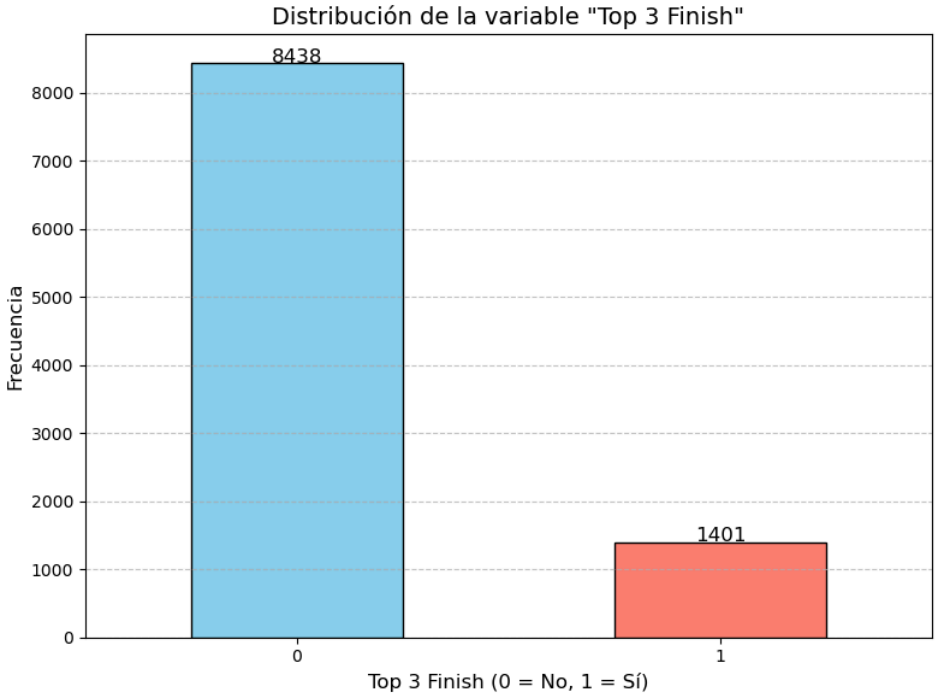


Figura 2: Distribución de la variable Top 3 Finish´

Los resultados reflejan un desbalance significativo en la variable objetivo, con una notable mayoría de observaciones correspondientes a la clase 0 (pilotos que no terminaron en el top 3) frente a la clase 1 (pilotos que sí lo hicieron).

Este desbalance es característico en problemas relacionados con eventos poco frecuentes, como la clasificación en las primeras posiciones de una carrera. Desde una perspectiva metodológica, este desbalance puede influir en el rendimiento de los modelos predictivos, especialmente en algoritmos que no manejan de manera nativa clases desbalanceadas [8]. En estos casos, las métricas de evaluación estándar como la exactitud pueden no ser adecuadas, ya que podrían estar sesgadas hacia la clase mayoritaria.

Por lo tanto, es necesario considerar estrategias para mitigar este efecto, como la utilización de técnicas de remuestreo (*oversampling*), el uso de modelos adaptados a datos desbalanceados, o la adopción de métricas de evaluación que consideren este contexto, como la sensibilidad, especificidad, y el área bajo la

curva ROC (AUC-ROC). Este análisis inicial subraya la necesidad de ajustar el diseño del experimento y la selección de modelos para abordar este desafío de manera efectiva.

3.1.2.1. Valores faltantes

En el conjunto de datos, algunos de los atributos generados contienen valores faltantes, lo que hace necesario identificar y tratar estos casos de manera adecuada para preservar la calidad del análisis. En particular, se observaron valores faltantes en los registros relacionados con la posición del conductor durante la temporada anterior o la temporada en curso, atribuibles a que el conductor es un debutante (*rookie*). Para estos casos, se imputaron los valores faltantes utilizando el promedio de las posiciones correspondientes a su escudería.

Por otra parte, el campo Posición del conductor en la carrera previa presentó 277 valores faltantes. Dado que esta variable es numérica y la proporción de datos faltantes en la variable objetivo no es significativa (2,8%), se optó por una estrategia de imputación basada en la información de los conductores que participaron en la carrera inmediatamente anterior. El procedimiento consistió en analizar la lista de corredores de una carrera con valores faltantes en este campo, identificar la lista de participantes de la carrera anterior, y determinar los conductores presentes en una carrera pero no en la siguiente. Posteriormente, los valores faltantes fueron completados utilizando las posiciones finales de los conductores que participaron únicamente en la carrera previa. Este enfoque asegura una imputación coherente con el contexto de la competición y mantiene la consistencia de los datos.

3.1.2.2. Variables correlacionadas a Pole

Es de particular interés examinar el coeficiente de correlación de *Pearson* entre cada una de las variables y la variable objetivo binaria *Top 3 Finish*, ya que este análisis proporciona una perspectiva integral de las relaciones existentes. Los resultados de dicho análisis se presentan en la Figura 3.

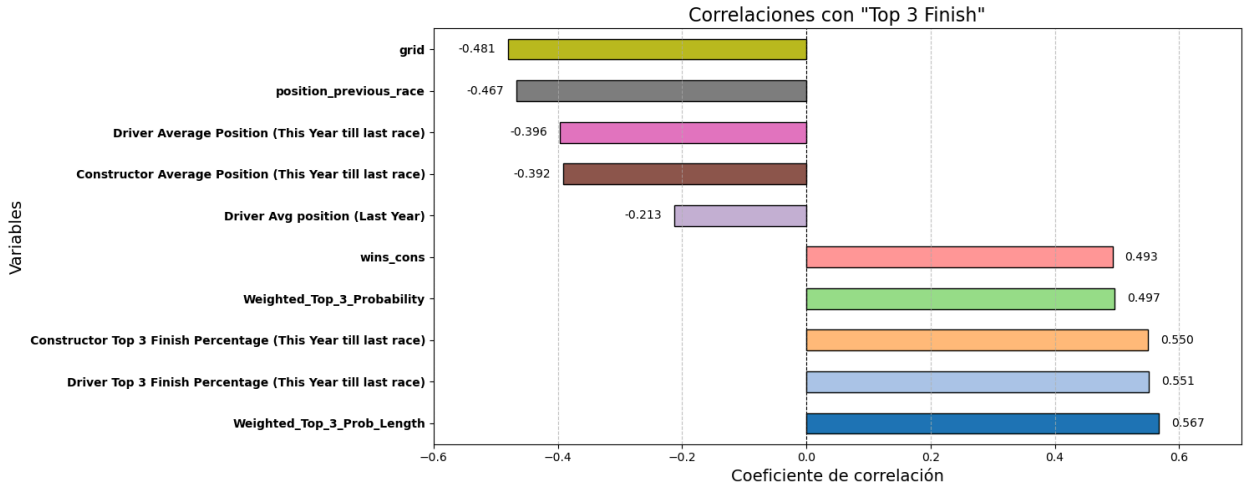


Figura 3: Correlación con la variable *Top 3 Finish*

Al considerar la totalidad de los registros, se identificaron tres variables con una correlación positiva superior a 0.5 respecto a la variable objetivo. Aunque estos valores no superan significativamente este umbral, resulta relevante destacar que dichas variables corresponden a características derivadas del proceso de *Feature Engineering* implementado previamente. En cuanto a las correlaciones negativas, no se observaron valores inferiores a -0.5. Las correlaciones negativas más cercanas a este umbral se identificaron en las variables relacionadas con la posición de largada en la carrera actual y la posición en la carrera previa, las cuales presentan valores próximos a -0.5.

3.1.2.3. Análisis de valores ruidosos

El conjunto de datos *Circuits* presentó valores ruidosos en 5 registros en el atributo *Url*, el cual era fundamental para la recuperación de datos de los circuitos desde el sitio web de *Wikipedia*. El principal problema identificado fue que los enlaces contenidos en este campo estaban desactualizados, lo que impedía la extracción correcta de información asociada a los circuitos.

Para resolver esta situación, se realizó una actualización manual de cinco URLs correspondientes a los circuitos de *Long Beach, Fair Park, Las Vegas* y *Red Bull Ring*. Esta corrección fue necesaria para garantizar la integridad de los datos utilizados en el análisis y para habilitar la correcta recuperación de la información complementaria desde *Wikipedia*.

3.2. Preprocesamiento de datos

El conjunto de datos utilizado en este trabajo originalmente contenía información en el formato proporcionado por la fuente, pero presentaba desafíos como ruido, valores faltantes y la necesidad de aplicar transformaciones como normalización y discretización. Estas operaciones se llevaron a cabo como parte de la etapa de preprocesamiento, cuyo objetivo fue mejorar la calidad de los datos para mejorar el análisis posterior.

3.2.1. Limpieza de datos

El atributo *Nationality*, de tipo cadena de texto (*string*), resulta relevante para la etapa de modelado. Por esta razón, se aplicó la técnica de *Label Encoding* a esta variable categórica, transformándola en una representación numérica mediante la asignación de un número entero único a cada nacionalidad, sin introducir un orden lógico implícito entre las categorías.

3.2.2. Normalización

El análisis exploratorio de datos reveló una diversidad en las escalas de las variables numéricas consideradas para el modelado. Las estadísticas descriptivas mostraron rangos heterogéneos entre las características, como el número de vueltas completadas (*laps*) con valores en un rango de 0 a 87, y la puntuación acumulada en el campeonato de pilotos (*points_driver_champ*), que varía entre 0 y 575. Estas diferencias indican la coexistencia de magnitudes dispares que podrían influir en el rendimiento de los

algoritmos de aprendizaje automático sensibles a la escala, como los métodos basados en distancia (e.g., *Support Vector Machines*).

El análisis visual mediante *boxplots* que se observa en la figura 4 destaca aún más estas discrepancias, mostrando asimetrías significativas y rangos amplios en ciertas características. Debido a la disparidad observada en las escalas de las variables y la presencia de distribuciones asimétricas, se implementó una normalización *Min-Max* con el propósito de homogeneizar las escalas de las variables. Esta estrategia no solo permite alinear las magnitudes de las variables, sino que también facilita la convergencia y mejora la precisión de los modelos de aprendizaje automático.

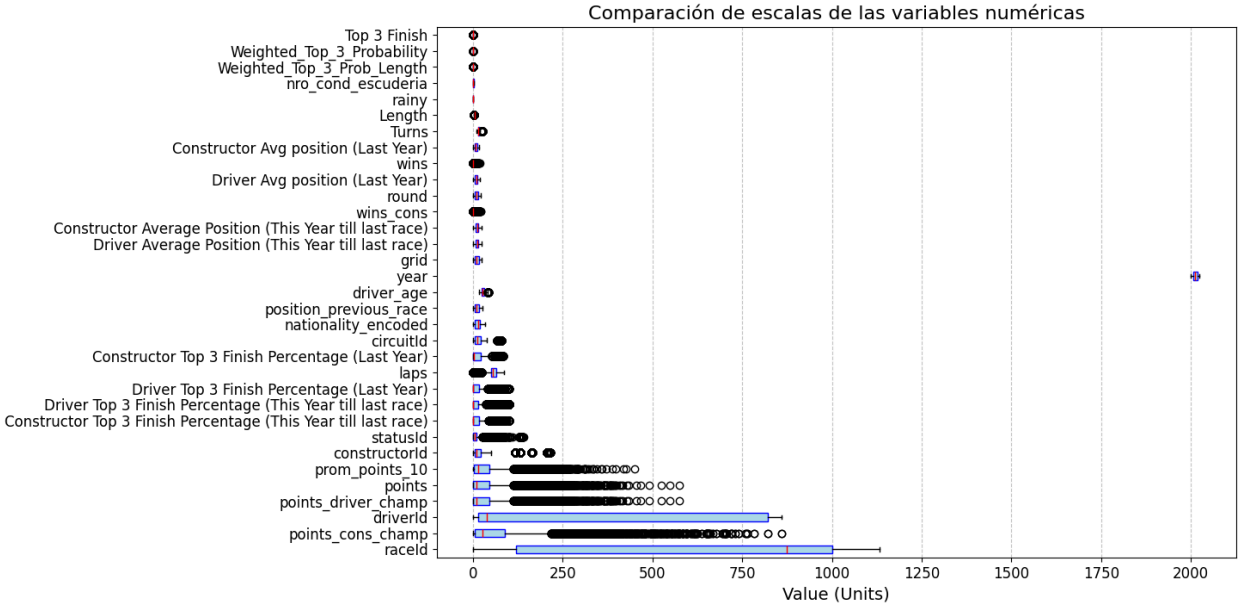


Figura 4: Comparación de las escalas de las variables numéricas

3.3. Ajustes de los modelos de aprendizaje automático

Para dividir el conjunto de datos en un set de entrenamiento y uno de test se utilizó un esquema de muestreo estratificado con una proporción 70-30 respectivamente. Esta estrategia de muestreo garantiza que la proporción de clases en la variable objetivo binaria sea la misma en ambos conjuntos. Esto evita el desequilibrio en las clases que podría afectar el rendimiento del modelo.

3.3.1. LogisticRegression, KNeighborsClassifier, SVC, RandomForestClassifier, DecisionTreeClassifier, GaussianNB

En este estudio, se compararon seis algoritmos de aprendizaje supervisado de la librería *Scikit-learn* [11] para un problema de clasificación binaria. Cada uno de estos modelos aporta un enfoque único para la clasificación, permitiendo evaluar no solo su precisión general, sino también su desempeño en términos de sensibilidad y especificidad en un problema con una variable objetivo binaria. La selección del modelo más adecuado dependerá de los patrones inherentes en los datos y las métricas de evaluación específicas del problema.

3.3.2. Cross Validation y Grid Search

En este trabajo, se utilizó una combinación de CV *(Stratified K-Fold Cross-Validation)* [12] y *Grid Search* [9] para optimizar los hiper parámetros de los modelos de aprendizaje supervisado. La validación cruzada estratificada asegura que cada partición utilizada durante el entrenamiento y validación mantenga la misma proporción de la variable objetivo que el conjunto de datos original, lo cual es crucial en problemas con clases desbalanceadas. Esto permite una evaluación más robusta del rendimiento del modelo en escenarios donde las clases minoritarias podrían ser subrepresentadas.

Por otro lado, *Grid Search* permitió explorar sistemáticamente combinaciones predefinidas de hiper parámetros para cada modelo, maximizando el área bajo la curva ROC (AUC-ROC) como métrica de evaluación. Este proceso identificó los hiper parámetros óptimos para los algoritmos mencionados en la sección previa, garantizando que cada modelo se ajusta de manera efectiva al conjunto de datos.

3.3.3. SMOTE

La variable *target* presentaba un desbalance entre clases, como puede observarse en la figura 2. Se empleó el algoritmo SMOTE [10] para abordar el desbalance de clases en la variable objetivo, que es binaria. El desbalance de clases puede llevar a que los modelos de aprendizaje automático favorezcan la clase mayoritaria, afectando negativamente su capacidad para identificar correctamente la clase minoritaria. Se aplicó SMOTE al conjunto de entrenamiento para equilibrar las proporciones de las clases, asegurando que ambas estuvieran representadas de manera equitativa durante el entrenamiento del modelo.

3.3.4. Selección de features

Para el modelado se utilizaron dos conjuntos de variables, el primero con 32 variables numéricas, y el segundo con 14 variables numéricas que surgen del análisis de variables correlacionadas que se visualiza en la figura 3. Las variables que conforman ambos conjuntos pueden encontrarse en el Anexo B.

3.4. Métricas de evaluación

A continuación, se presentan las métricas empleadas para evaluar y comparar el rendimiento de los modelos, con el propósito de identificar el más adecuado para la tarea de predicción. En particular, se utilizaron las métricas F1-Score, Accuracy y Área Bajo la Curva ROC (AUC-ROC) para la evaluación de modelos de clasificación.

Cabe destacar que el presente estudio se enfrenta a un marcado desbalance en la variable objetivo, lo cual representa un desafío significativo para la evaluación del rendimiento de los modelos. En este contexto, el Accuracy puede resultar una métrica poco representativa, dado que un modelo podría predecir sistemáticamente la clase mayoritaria y obtener un valor elevado sin reflejar un rendimiento realista. Por este motivo, se recurre al F1-Score, que resulta más adecuado para problemas con desequilibrio de clases,

ya que proporciona un balance entre la precisión y la exhaustividad (recall), permitiendo una mejor valoración del rendimiento en ambas clases.

Adicionalmente, el AUC-ROC se emplea para comparar el desempeño de los modelos de forma independiente del umbral de decisión. Esta métrica es especialmente útil en situaciones de desbalance de clases, como en el presente caso de estudio, ya que evalúa la capacidad del modelo para distinguir entre clases minoritarias y mayoritarias de manera robusta.

4. Descripción de experimentos

En esta sección se presentan los experimentos realizados con el objetivo de evaluar el rendimiento de distintos enfoques de aprendizaje automático. Los experimentos combinan diferentes conjuntos de datos con enfoques metodológicos que incluyen la aplicación de diversos algoritmos de aprendizaje y métodos de evaluación. Esta combinación permite analizar cómo varían los resultados según las configuraciones utilizadas, facilitando la comparación de desempeños en distintos escenarios. La Tabla 4 resume de manera estructurada cada uno de los experimentos, detallando sus características principales y el conjunto de datos empleado.

Tabla 4: Experimentos realizados

Experimento	Descripción	Dataset utilizado
V1	Se emplean los modelos LogisticRegression, KNeighborsClassifier, SVC, RandomForestClassifier, DecisionTreeClassifier y GaussianNB. Se aplica la técnica de normalización. Se entrena y testea con los registros correspondientes a todas las carrera entre los años 2020-2024	Conjunto de datos 1 (Anexo B)
V2	Se emplean los modelos LogisticRegression, KNeighborsClassifier, SVC, RandomForestClassifier, DecisionTreeClassifier y GaussianNB. Se aplican las técnicas de normalización y GridSearch. Se entrena y testea con los registros correspondientes a todas las carrera entre los años 2020-2024	Conjunto de datos 1 (Anexo B)
V3	Se emplean los modelos LogisticRegression, KNeighborsClassifier, SVC, RandomForestClassifier, DecisionTreeClassifier y GaussianNB. Se aplican las técnicas de normalización, GridSearch Y SMOTE. Se entrena y testea con los registros correspondientes a todas las carrera entre los años 2020-2024	Conjunto de datos 1 (Anexo B)
V4	Se emplean los modelos LogisticRegression, KNeighborsClassifier, SVC, RandomForestClassifier, DecisionTreeClassifier y GaussianNB. Se aplica la técnica de normalización. Se entrena y testea con los registros correspondientes a todas las carrera entre los años 2020-2024	Conjunto de datos 2 (Anexo B)

V5	Se emplean los modelos LogisticRegression, KNeighborsClassifier, SVC, RandomForestClassifier, DecisionTreeClassifier y GaussianNB. Se aplican las técnicas de normalización y GridSearch. Se entrena y testea con los registros correspondientes a todas las carrera entre los años 2020-2024	Conjunto de datos 2 (Anexo B)
V6	Se emplean los modelos LogisticRegression, KNeighborsClassifier, SVC, RandomForestClassifier, DecisionTreeClassifier y GaussianNB. Se aplican las técnicas de normalización, GridSearch Y SMOTE. Se entrena y testea con los registros correspondientes a todas las carrera entre los años 2020-2024	Conjunto de datos 2 (Anexo B)
V7	Se emplean los modelos LogisticRegression, KNeighborsClassifier, SVC, RandomForestClassifier, DecisionTreeClassifier y GaussianNB. Se aplican las técnicas de normalización, GridSearch Y SMOTE. Se entrena y testea con los registros correspondientes a todas las carrera entre los años 2000-2024	Conjunto de datos 1 (Anexo B)
V8	Se emplean los modelos LogisticRegression, KNeighborsClassifier, SVC, RandomForestClassifier, DecisionTreeClassifier y GaussianNB. Se aplican las técnicas de normalización, GridSearch Y SMOTE sobre los registros correspondientes a todas las carrera entre los años 2000-2024	Conjunto de datos 2 (Anexo B)
Robustez temporal	Se emplean los modelos LogisticRegression, KNeighborsClassifier, SVC, RandomForestClassifier, DecisionTreeClassifier y GaussianNB. Se aplican las técnicas de normalización, GridSearch Y SMOTE sobre los registros correspondientes a todas las carreras entre los años 2015-2019. Los datos son testeados desde el año 2019 hasta el 2024 para ver cómo evoluciona el rendimiento a lo largo de los años.	Conjunto de datos 1 (Anexo B)
Ventana deslizante	Se emplean los modelos Logistic Regression, Random Forest, Decision Tree, Gaussian Naive Bayes, K-Nearest Neighbors y Support Vector Classifier. Se aplican técnicas de normalización, ajuste de hiperparámetros con GridSearchCV y balanceo de clases con SMOTE. El experimento consiste en entrenar los modelos con datos de dos años consecutivos (ej. 2015-2016) y evaluarlos con los dos años siguientes (ej. 2017-2018), deslizando la ventana un año en cada iteración hasta llegar a 2022-2023. Finalmente, se grafican los resultados para analizar cómo varía el rendimiento a lo largo del tiempo y evaluar el impacto de la incorporación de datos más recientes.	Conjunto de datos 1 (Anexo B)

Simulación de rendimientos	Se emplea el modelo RandomForestClassifier. Se aplican las técnicas de normalización y GridSearch, los datos son entrenados en el periodo 2020-2022 y son testeados en el año 2023. Luego, se arma un ranking de podios para los conductores y las escuderías y se comparan los valores reales con los valores predecidos por el modelo. Se comparan los rankings usando correlación de Spearman	Conjunto de datos 1 (Anexo B)

5. Resultados obtenidos

5.1. Modelado de los años 2020-2024

El modelo construido en base 467 carreras realizadas entre los años 2020-2024 llega a alcanzar un *accuracy* del 97% y un *F1 Score* de 79% utilizando el algoritmo *RandomForestClassifier*, tomando como entrada todas las variables del *dataset*, y utilizando las técnicas de Normalizacion Min-Max, *Grid Search* y *Smote*.

Los modelos generados que se observan en la figura 5 corresponden a los experimentos mencionados en la tabla 4. Como se observa en la figura 5, *RandomForestClassifier* es el modelo más robusto en versiones iniciales, pero pierde ventaja con la reducción de variables. En general, *RandomForestClassifier* y SVC demostraron ser los modelos más robustos, adaptándose positivamente a la normalización, la reducción de variables y el ajuste de hiperparámetros mediante *Grid Search*. Por otro lado, modelos como *DecisionTreeClassifier* y *GaussianNB* exhibieron una alta sensibilidad al sobremuestreo mediante SMOTE, presentando caídas significativas en el rendimiento, probablemente debido al *overfitting* o a la introducción de ruido en los datos sintetizados. Asimismo, la reducción del número de variables tuvo un impacto negativo notable en modelos altamente dependientes de la complejidad de los datos, como *DecisionTreeClassifier*, mientras que otros, como *KNeighborsClassifier*, se mantuvieron relativamente estables. La combinación óptima de hiperparametros para cada caso puede observarse en el Anexo C.

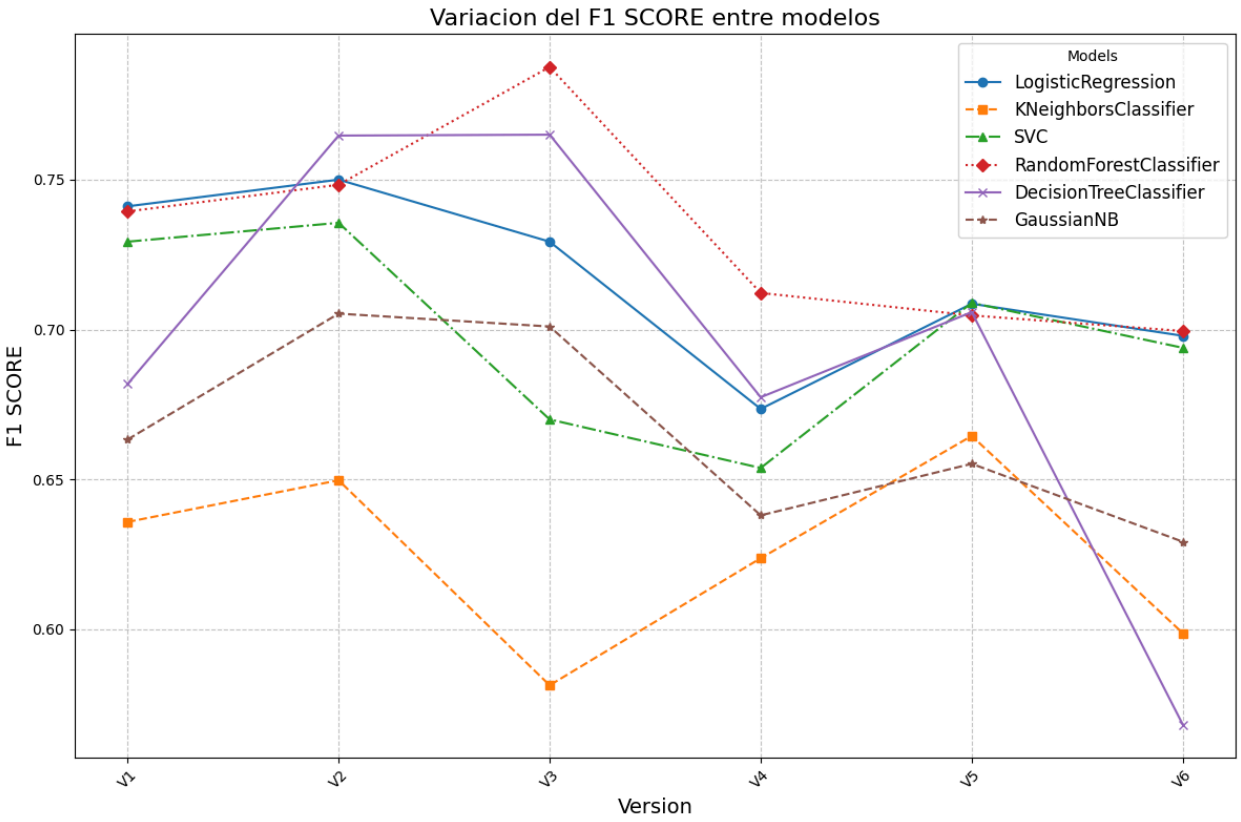


Figura 5: Variación del F1 Score entre los diferentes modelos.

5.1.1. Analisis de DecisionTreeClassifier

Mediante el algoritmo *DecisionTreeClassifier*, se obtuvo una precisión (*Accuracy*) del 93% para el experimento V3 (ver tabla 4), destacando la importancia de analizar la selección de variables que forman parte del árbol y las diferencias con respecto a las variables más correlacionadas presentadas en la Figura 3. Este análisis permite identificar discrepancias y consistencias clave entre la selección automática de características del modelo y las correlaciones iniciales observadas.

La posición de salida en la parrilla (*grid*) destaca como la variable más relevante, con un peso del 0.62 en el árbol de decisión, consistente con su alta correlación negativa con el objetivo (-0.481). Asimismo, *Weighted_Top_3_Prob_Length*, con una importancia del 0.24 en el modelo, también presenta una correlación significativa, apareciendo recurrentemente en el árbol, lo que refuerza su rol como predictor clave.

Sin embargo, ciertas variables con correlaciones moderadas, como el promedio de posiciones del constructor (*Constructor Average Position*), fueron relegadas en el modelo. Esto podría explicarse por redundancia con predictores más fuertes, como *grid*, o por una relación menos compleja con el objetivo que no contribuye significativamente a la partición del árbol.

El árbol de decisión generado que ilustra estos resultados se presenta en el Anexo E, donde se evidencia cómo el modelo prioriza variables contemporáneas con alta capacidad predictiva, mientras reduce la influencia de aquellas que, a pesar de su correlación, no optimizan el desempeño del modelo en términos de la métrica objetivo.

5.2. Modelado de los años 2000-2024

Se realizaron nuevos modelos con los datos de las carreras desde el año 2000 al 2024. Resulta interesante comparar los experimentos V3 y V7 que abarcan la totalidad de las variables del *dataset* en distintos períodos, y los experimentos V6 y V8 que abarcan un menor número de variables, pero en periodos distintos.

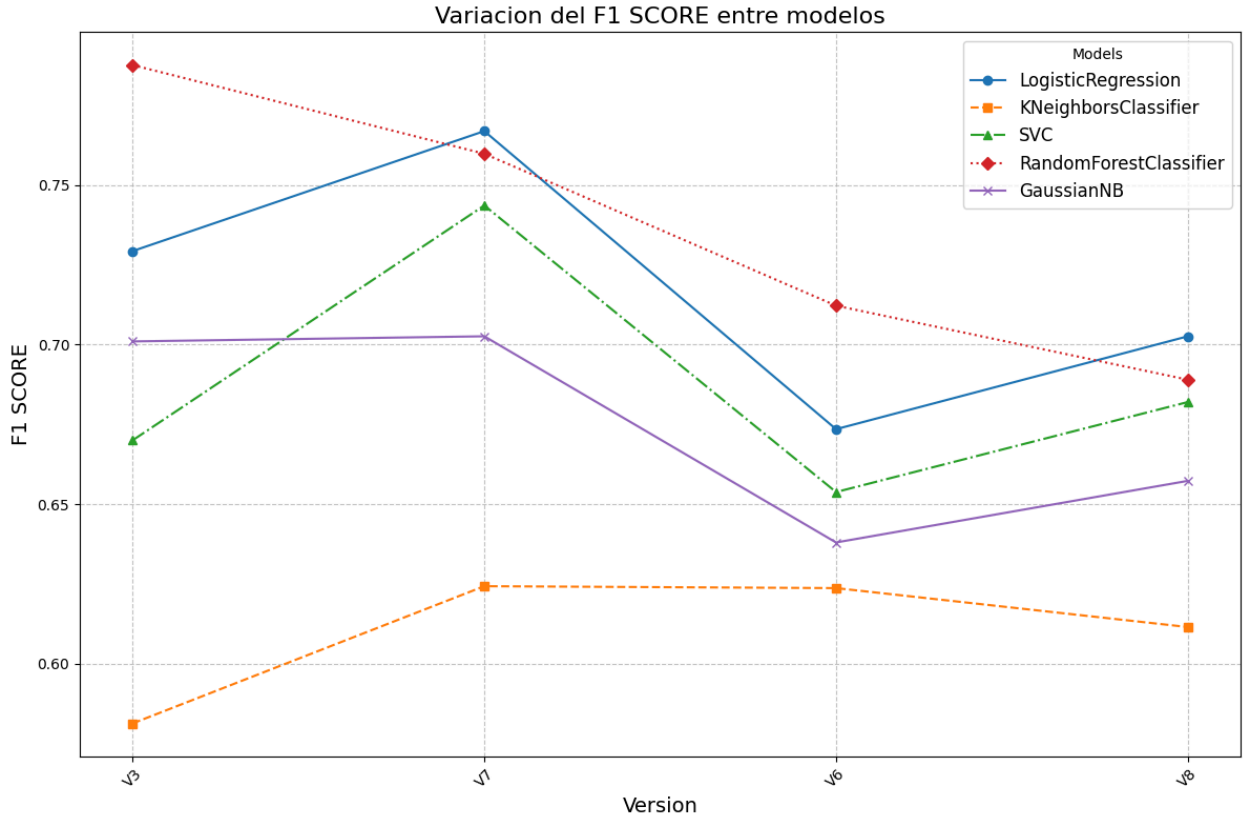


Figura 6: Comparación de modelos en diferentes periodos

La comparación de las configuraciones que se observa en la figura 6 evidencia que el uso de todas las variables (V3 y V7) permite un mejor desempeño general en términos de *F1-Score*, especialmente en modelos robustos como *RandomForestClassifier* y SVC. Además, la extensión del rango temporal (V7 y V8) aporta una mejora marginal en ciertos modelos, aunque su beneficio parece ser dependiente del modelo. Por otro lado, la reducción de variables (V6 y V8) tiende a limitar el desempeño, destacando la importancia de un enfoque cuidadoso en la selección de variables, especialmente cuando se utiliza un rango temporal limitado.

5.3. Evaluación de robustez temporal

En este trabajo, se llevó a cabo un experimento para evaluar la robustez temporal de un modelo predictivo aplicado al ámbito de las carreras de fórmula 1. El modelo fue entrenado utilizando datos históricos de carreras correspondientes al período 2015-2019, considerando todas las variables disponibles en el conjunto de datos. Posteriormente, su desempeño fue evaluado en datos de carreras desde la primera ronda del año 2019 hasta la última ronda del año 2023.

La Figura 7 presenta el *F1-score* suavizado mediante LOESS [13], graficado a lo largo de las rondas de cada año.Este método de suavizado se aplica con fines de visualización, dado que facilita la identificación de patrones subyacentes al reducir la variabilidad de los datos observados, resultando particularmente util en el análisis de series temporales.

Los resultados permiten identificar dos patrones principales:

- **Mejoras intraanuales:** Dentro de cada año evaluado, el desempeño del modelo tiende a mejorar progresivamente a medida que avanzan las rondas. Este comportamiento podría atribuirse a la mayor homogeneidad en las características de las carreras dentro de un mismo año, facilitando la capacidad del modelo para realizar predicciones más precisas.

- **Degradación interanual:** Se evidencia una disminución progresiva del *F1-score* en los años sucesivos, particularmente al evaluar el modelo en datos correspondientes a períodos cada vez más alejados del rango temporal utilizado para el entrenamiento (2015-2019), demostrando una pérdida de capacidad de generalización del modelo en escenarios futuros.

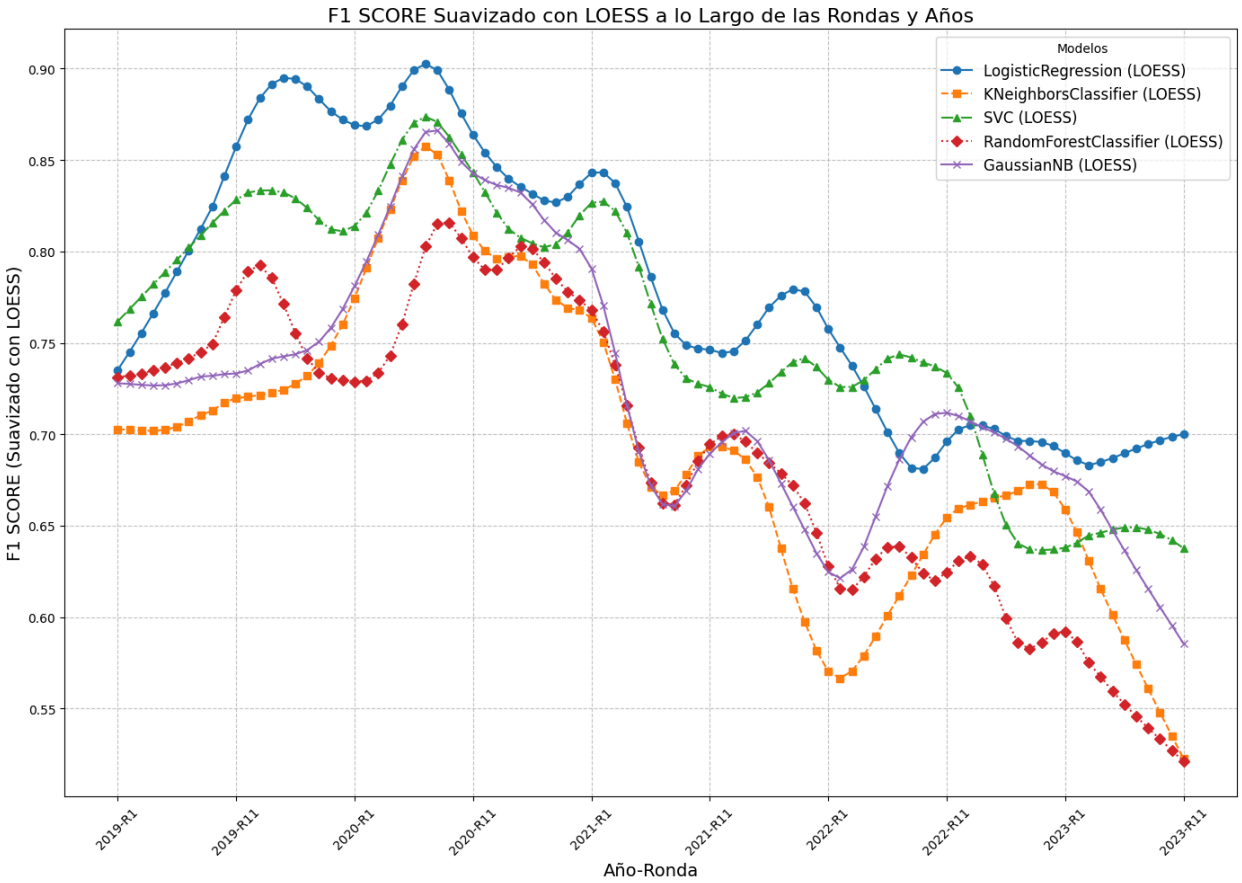


Figura 7: F1 Score de los modelos para cada año(Suavizado)

5.4. Ventana deslizante

En el contexto de este trabajo, la aplicación de la técnica de ventana deslizante permite evaluar la robustez temporal y la capacidad de adaptación de los modelos predictivos a lo largo del tiempo, complementando el análisis presentado en la Sección 4.3. Mientras que dicha sección se enfoca en la degradación del rendimiento de un modelo entrenado en un periodo fijo (2015-2019), la ventana deslizante introduce una dinámica más flexible al actualizar continuamente el conjunto de entrenamiento y evaluación. Esto permite observar cómo el rendimiento de los modelos evoluciona al enfrentar nuevos patrones de datos.

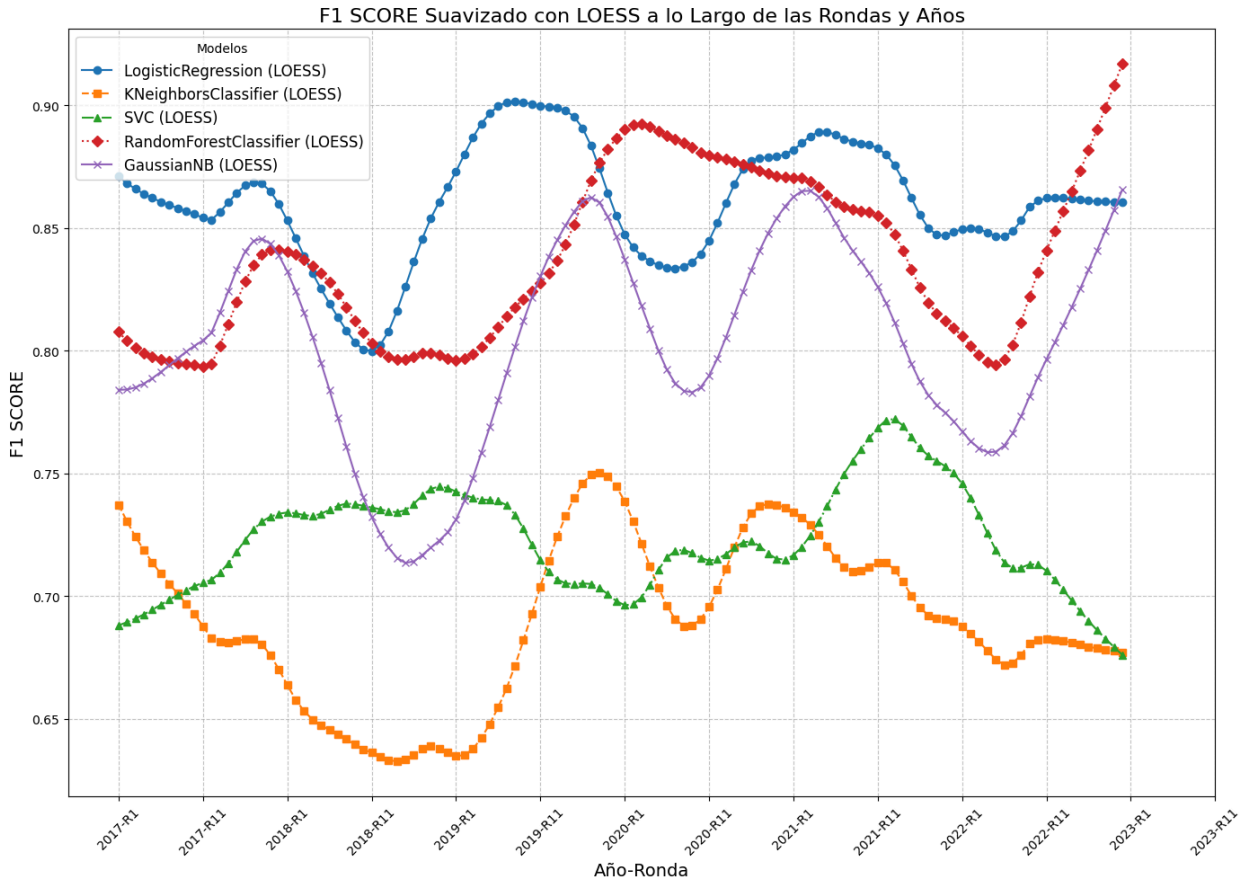


Figura 8: Evolución del F1 Score de los Modelos utilizando Ventana Deslizante (2018-2023)

En este experimento de ventana deslizante, se evaluó el rendimiento de varios modelos de clasificación (Logistic Regression, Random Forest, Decision Tree, Gaussian Naive Bayes, K-Nearest Neighbors y Support Vector Classifier) simulando escenarios del mundo real donde los datos evolucionan con el tiempo. Para cada iteración, el conjunto de entrenamiento se desplazó un año, comenzando con el entrenamiento en los datos de 2015-2016 y la prueba en el periodo 2017-2018. En la siguiente iteración, se entrenó con los datos de 2016-2017 y se probó con 2018-2019, continuando este proceso hasta el rango de testeo 2022-2023. Este experimento permite observar cómo se degrada o mantiene el rendimiento del modelo cuando se enfrenta a datos que están más alejados en el tiempo en relación al conjunto de entrenamiento.

Los resultados observados en la figura 8 muestran que Logistic Regression y Random Forest presentan una alta estabilidad en el F1 Score a lo largo del tiempo, lo que indica una fuerte capacidad de generalización frente a cambios en la distribución de los datos. En contraste, modelos como K-Nearest Neighbors y Support Vector Classifier exhiben una mayor sensibilidad a la variabilidad temporal, con caídas significativas en su rendimiento. El enfoque de ventana deslizante permite evaluar la capacidad de adaptación del modelo en entornos dinámicos, como lo es la predicción de resultados en Fórmula 1.

5.5. Simulación de rendimientos para el año 2023

La correlación de Spearman es utilizada como métrica para evaluar la capacidad de un modelo de clasificación en la predicción del ranking de conductores y escuderías con más podios, enfocándose en el orden relativo de los resultados y minimizando el impacto de errores en valores absolutos. Para este experimento, se empleó el algoritmo *Random Forest Classifier*, que alcanzó un *F1 Score* cercano al 80% tras ser entrenado con datos del período 2020-2022.

En particular, se analizó la predicción de la cantidad de *poles* obtenidas por las escuderías. La Figura 9 presenta los resultados de este análisis, destacando una correlación de *Spearman* de 0.89, lo cual indica una correlación positiva muy fuerte. Este valor refleja que el ranking predicho por el modelo muestra un alto nivel de concordancia con el ranking real, validando así la capacidad del modelo para capturar de manera efectiva el orden relativo de las escuderías con mejor desempeño.

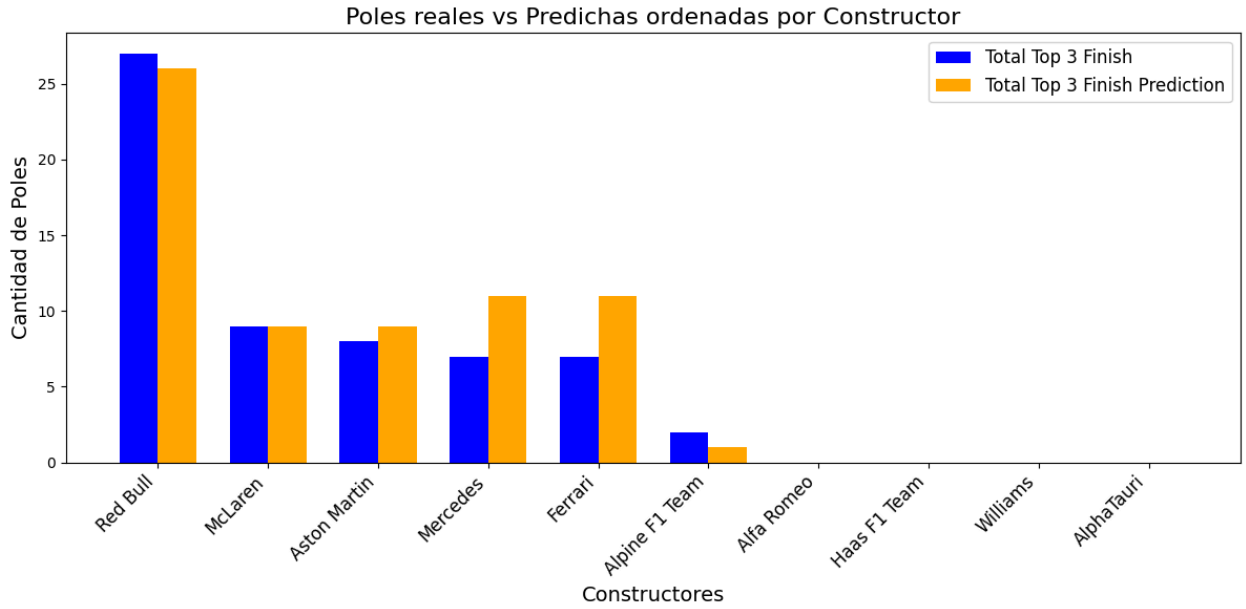


Figura 9: Ranking de conductores para el año 2023

La figura 10 muestra la comparación entre las *poles* reales y las predichas por el modelo para cada conductor, ordenados de mayor a menor cantidad de *poles*. La correlación de Spearman calculada entre ambos rankings es de 0.93, lo que indica una fuerte concordancia positiva entre el ranking real y el predicho. Este resultado evidencia que el modelo es altamente efectivo para capturar el orden relativo de los conductores en términos de *poles*, incluso si existen ligeras discrepancias en los valores absolutos. En particular, se observa que los conductores con mayor cantidad de *poles* reales, como Verstappen y Alonso, también ocupan los primeros lugares en el ranking predicho, reflejando la capacidad del modelo para identificar correctamente a los conductores más destacados.

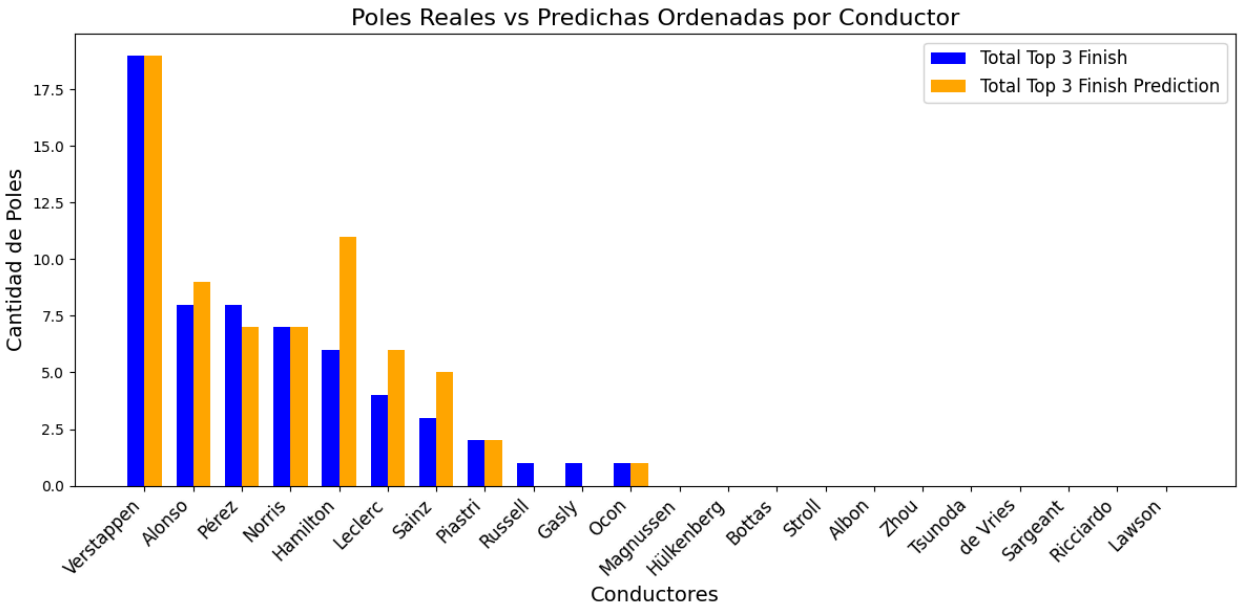


Figura 10: Ranking de conductores para el año 2023.

6. Discusión

El desarrollo de este trabajo subraya tanto los logros como las limitaciones inherentes al proceso de predicción en carreras de Fórmula 1. Uno de los principales desafíos fue la construcción del *dataset*, que involucró la integración de datos provenientes de múltiples fuentes heterogéneas. A pesar de las dificultades asociadas a la calidad y cobertura de los datos originales, la ingeniería de características permitió enriquecer el conjunto de datos, generando variables predictoras clave que mejoraron significativamente el desempeño de los modelos.

En cuanto a los resultados obtenidos, el uso de modelos de aprendizaje automático como *RandomForestClassifier* y *LogisticRegression* evidenció su capacidad para capturar patrones relevantes en los datos, alcanzando un AUC-ROC de hasta 0.97 en los mejores escenarios. Sin embargo, se identificaron limitaciones relacionadas con el manejo de clases desbalanceadas, especialmente en configuraciones con datos históricos más extensos, donde la técnica SMOTE, aunque efectiva para equilibrar las clases, mostró sensibilidad a la introducción de ruido en algunos modelos.

El análisis de robustez temporal reveló la degradación del rendimiento del modelo cuando se lo aplicó a períodos temporalmente alejados del conjunto de entrenamiento. Este hallazgo destaca la importancia de considerar dinámicas temporales en el diseño de modelos predictivos, como el reentrenamiento periódico. Para profundizar en este aspecto, se implementó un experimento de ventana deslizante que permitió evaluar la estabilidad del rendimiento de los modelos a lo largo de los años. Los resultados mostraron que algunos modelos, como Logistic Regression y Random Forest, mantienen un rendimiento robusto pese al paso del tiempo, mientras que otros, como K-Nearest Neighbors y Support Vector Classifier, presentan una mayor sensibilidad temporal con caídas significativas en su desempeño.

Por último, los resultados basados en la correlación de *Spearman* (hasta 0.93 en rankings de conductores) demuestran la efectividad del enfoque para predecir el orden relativo en términos de cantidad de podios, aunque persisten ligeras discrepancias en los valores observados. Esto valida el potencial del modelo para tareas relacionadas con el análisis estratégico en Fórmula 1, pero también invita a explorar métodos más sofisticados para mejorar la precisión global y la generalización en escenarios futuros.

En conclusión, este trabajo presenta una solución robusta para predecir podios en carreras de Fórmula 1, destacando el impacto de la ingeniería de datos y las técnicas avanzadas de aprendizaje automático. No obstante, las limitaciones identificadas abren oportunidades para investigaciones futuras que exploren enfoques más adaptativos y enfoques más integrales para capturar dinámicas complejas y evolutivas en este contexto.

7. Trabajos futuros

El presente trabajo constituye una base sólida para el análisis predictivo en el ámbito de la Fórmula 1, sin embargo, se identificaron múltiples oportunidades para continuar y enriquecer esta línea de investigación.

En primer lugar, se plantea la posibilidad de emplear *Lasso Regression*[4] como método de selección de características. Esta técnica podría ayudar a identificar y seleccionar automáticamente las variables más relevantes del *dataset*, reduciendo el ruido y mejorando el rendimiento del modelo. La implementación de esta metodología permitiría evaluar cómo se comporta el modelo predictivo al limitarse a un subconjunto optimizado de variables.

Además, se visualiza la incorporación de la API *OpenF1*[5] para generar nuevas variables relacionadas con los compuestos de neumáticos y las condiciones climáticas específicas de cada carrera. Estas características adicionales podrían reflejar mejor las dinámicas de las competencias, considerando factores estratégicos que actualmente no se encuentran representados en el *dataset*, como la degradación de los neumáticos en función de las características del circuito y del clima.

Otro aspecto de interés sería explorar el impacto de diferentes técnicas de enriquecimiento de datos en el rendimiento del modelo. Por ejemplo, el uso de datos telemétricos y estrategias avanzadas de simulación para integrar escenarios hipotéticos en el análisis podría proporcionar una perspectiva única sobre las decisiones estratégicas que influyen en los resultados.

Finalmente, se sugiere evaluar el desempeño de modelos más sofisticados, como redes neuronales profundas y enfoques de aprendizaje reforzado, para abordar el problema de manera no lineal e integrando temporalidades. Esto permitiría no solo predecir podios, sino también simular el comportamiento de los pilotos y escuderías a lo largo de una temporada completa.

El conjunto de datos utilizado en este estudio, que incluye información detallada de las carreras de Fórmula 1 entre los años 2000 y 2024, ha sido puesto a disposición del público en Kaggle[6]. Investigadores

[4] *https://scikit-learn.org/1.5/modules/generated/sklearn.linear_model.Lasso.html*
[5] *https://openf1.org*
[6] *https://www.kaggle.com*

y profesionales pueden acceder al dataset para fines de replicación, análisis adicionales o para el desarrollo de nuevos modelos relacionados con el rendimiento en carreras de Fórmula 1. El conjunto de datos incluye documentación detallada sobre las fuentes de datos y cómo trabajar con el *dataset.*

El conjunto de datos se encuentra disponible aquí: F1 Races 2000-2024 Dataset en Kaggle.

Referencias

1. García Villalón, M. J. (2022). *Organización, análisis de datos e inteligencia artificial para la predicción de resultados en Fórmula 1* [Trabajo de Fin de Grado, Universidad de Sevilla]. Repositorio idUS.
2. Masaracchia, A., Daoud, A., & Reymond, J. (2022). Formula 1: An analysis of driver performance using machine learning.
3. Van Kesteren, E.-J., & Bergkamp, T. (2022). Bayesian Analysis of Formula One Race Results: Disentangling Driver Skill and Constructor Advantage.
4. Fry, J., Brighton, T., & Fanzon, S. (2023). Faster Identification of Faster Formula 1 Drivers via Time-Rank Duality.
5. Weiss, T., & Behl, M. (2020). DeepRacing: Parameterized Trajectories for Autonomous Racing.
6. Sicoie, H. (2020). Machine Learning Framework for Formula 1 Race Winner and Championship Standings Predictor [Tesis de maestría, Universidad de Tilburg].
7. Patil, A., Jain, N., Agrahari, R., Hossari, M., Orlandi, F., & Dev, S. (2023). A Data-Driven Analysis of Formula 1 Car Races Outcome. En Artificial Intelligence and Cognitive Science (pp. 134–146).
8. Abella Miravet, B. (2021). Mejora de las predicciones en muestras desbalanceadas [Trabajo de Fin de Grado, Universidad Autónoma de Madrid]. Repositorio UAM.
9. Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., & Duchesnay, É. (2011). Scikit-learn: Machine learning in Python. Journal of Machine Learning Research, 12, 2825–2830.
10. Chawla, N. V., Bowyer, K. W., Hall, L. O., & Kegelmeyer, W. P. (2002). SMOTE: synthetic minority over-sampling technique. Journal of Artificial Intelligence Research, 16, 321–357.
11. Scikit-learn: Machine Learning in Python, Pedregosa et al., JMLR 12, pp. 2825-2830, 2011.
12. Widodo, Slamet & Brawijaya, Herlambang & Samudi, Samudi. (2022). Stratified K-fold cross validation optimization on machine learning for prediction. Sinkron. 7. 2407-2414. 10.33395/sinkron.v7i4.11792.
13. Cleveland, W. S., & Devlin, S. J. (1988). Locally weighted regression: An approach to regression analysis by local fitting. Journal of the American Statistical Association, 83(403), 596–610.

A. Anexo: Variables de los datasets obtenidos de la API Eargast

Circuits:

Atributo	Tipo de Dato	Cant. de Instancias	Valores Faltantes	Porc. Faltantes	Valores Únicos
circuitId	Numerica	77	0	0	77
circuitRef	Categorica	77	0	0	77
name	Categorica	77	0	0	77
location	Categorica	77	0	0	75
country	Categorica	77	0	0	35
lat	Numerica	77	0	0	77
lng	Numerica	77	0	0	77
alt	Numerica	77	0	0	66
url	Categorica	77	0	0	77

Constructor_results:

Atributo	Tipo de Dato	Cant. de Instancias	Valores Faltantes	Porc. Faltantes	Valores Únicos
constructorResultsId	Numerica	12,505	0	0	12,505
raceId	Numerica	12,505	0	0	1,048
constructorId	Numerica	12,505	0	0	175
points	Numerica	12,505	0	0	60
status	Categorica	12,505	0	0	2

Constructor_standings:

Atributo	Tipo de Dato	Cant. de Instancias	Valores Faltantes	Porc. Faltantes	Valores Únicos
constructorStandingsId	Numerica	13,271	0	0	13,271
raceId	Numerica	13,271	0	0	1,049
constructorId	Numerica	13,271	0	0	160
points	Numerica	13,271	0	0	557
position	Numerica	13,271	0	0	22
positionText	Categorica	13,271	0	0	23
wins	Numerica	13,271	0	0	22

Constructors:

Atributo	Tipo de Dato	Cant. de Instancias	Valores Faltantes	Porc. Faltantes	Valores Únicos

constructorId	Numerica	212	0	0	212
constructorRef	Categorica	212	0	0	212
name	Categorica	212	0	0	212
nationality	Categorica	212	0	0	24
url	Categorica	212	0	0	175

Driver_standings:

Atributo	Tipo de Dato	Cant. de Instancias	Valores Faltantes	Porc. Faltantes	Valores Únicos
driverStandingsId	Numerica	34,595	0	0	34,595
raceId	Numerica	34,595	0	0	1,113
driverId	Numerica	34,595	0	0	852
points	Numerica	34,595	0	0	429
position	Numerica	34,595	0	0	108
positionText	Categorica	34,595	0	0	109
wins	Numerica	34,595	0	0	20

Drivers:

Atributo	Tipo de Dato	Cant. de Instancias	Valores Faltantes	Porc. Faltantes	Valores Únicos
driverId	Numerica	859	0	0	859
driverRef	Categorica	859	0	0	859
number	Categorica	859	0	0	47
code	Categorica	859	0	0	97
forename	Categorica	859	0	0	478
surname	Categorica	859	0	0	800
dob	Categorica	859	0	0	841
nationality	Categorica	859	0	0	42
url	Categorica	859	0	0	859

Lap_times:

Atributo	Tipo de Dato	Cant. de Instancias	Valores Faltantes	Porc. Faltantes	Valores Únicos
raceId	Numerica	575,029	0	0	531
driverId	Numerica	575,029	0	0	141
lap	Numerica	575,029	0	0	87
position	Numerica	575,029	0	0	24
time	Categorica	575,029	0	0	75,514

milliseconds	Numerica	575,029	0	0	75,514

Pit_Stops:

Atributo	Tipo de Dato	Cant. de Instancias	Valores Faltantes	Porc. Faltantes	Valores Únicos
raceId	Numerica	10,990	0	0	273
driverId	Numerica	10,990	0	0	74
stop	Numerica	10,990	0	0	14
lap	Numerica	10,990	0	0	74
time	Categorica	10,990	0	0	7,983
duration	Categorica	10,990	0	0	7,434
milliseconds	Numerica	10,990	0	0	7,434

Qualifying:

Atributo	Tipo de Dato	Cant. de Instancias	Valores Faltantes	Porc. Faltantes	Valores Únicos
qualifyId	Numerica	10,254	0	0	10,254
raceId	Numerica	10,254	0	0	482
driverId	Numerica	10,254	0	0	170
constructorId	Numerica	10,254	0	0	47
number	Numerica	10,254	0	0	55
position	Numerica	10,254	0	0	28
q1	Categorica	10,254	0	0	8,955
q2	Categorica	10,241	13	0.13	5,320
q3	Categorica	10,227	27	0.26	3,374

Races:

Atributo	Tipo de Dato	Cant. de Instancias	Valores Faltantes	Porc. Faltantes	Valores Únicos
raceId	Numerica	1,125	0	0	1,125
year	Numerica	1,125	0	0	75
round	Numerica	1,125	0	0	24
circuitId	Numerica	1,125	0	0	77
name	Categorica	1,125	0	0	54
date	Categorica	1,125	0	0	1,125
time	Categorica	1,125	0	0	35
url	Categorica	1,125	0	0	1,125
fp1_date	Categorica	1,125	0	0	91
fp1_time	Categorica	1,125	0	0	21

fp2_date	Categorica	1,125	0	0	91
fp2_time	Categorica	1,125	0	0	20
fp3_date	Categorica	1,125	0	0	73
fp3_time	Categorica	1,125	0	0	19
quali_date	Categorica	1,125	0	0	91
quali_time	Categorica	1,125	0	0	16
sprint_date	Categorica	1,125	0	0	19
sprint_time	Categorica	1,125	0	0	13

Results:

Atributo	Tipo de Dato	Cant. de Instancias	Valores Faltantes	Porc. Faltantes	Valores Únicos
resultId	Numerica	26,519	0	0	26,519
raceId	Numerica	26,519	0	0	1,113
driverId	Numerica	26,519	0	0	859
constructorId	Numerica	26,519	0	0	211
number	Categorica	26,519	0	0	130
grid	Numerica	26,519	0	0	35
position	Categorica	26,519	0	0	34
positionText	Categorica	26,519	0	0	39
positionOrder	Numerica	26,519	0	0	39
points	Numerica	26,519	0	0	39
laps	Numerica	26,519	0	0	172
time	Categorica	26,519	0	0	7,272
milliseconds	Categorica	26,519	0	0	7,493
fastestLap	Categorica	26,519	0	0	81
rank	Categorica	26,519	0	0	26
fastestLapTime	Categorica	26,519	0	0	7,298
fastestLapSpeed	Categorica	26,519	0	0	7,514
statusId	Numerica	26,519	0	0	137

Seasons:

Atributo	Tipo de Dato	Cant. de Instancias	Valores Faltantes	Porc. Faltantes	Valores Únicos
year	Numerica	75	0	0	75
url	Categorica	75	0	0	75

Sprint_results:

Atributo	Tipo de Dato	Cant. de Instancias	Valores Faltantes	Porc. Faltantes	Valores Únicos
resultId	Numerica	300	0	0	300
raceId	Numerica	300	0	0	15
driverId	Numerica	300	0	0	29
constructorId	Numerica	300	0	0	12
number	Numerica	300	0	0	30
grid	Numerica	300	0	0	21
position	Categorica	300	0	0	21
positionText	Categorica	300	0	0	23
positionOrder	Numerica	300	0	0	20
points	Numerica	300	0	0	9
laps	Numerica	300	0	0	14
time	Categorica	300	0	0	282
milliseconds	Categorica	300	0	0	282
fastestLap	Categorica	300	0	0	24
fastestLapTime	Categorica	300	0	0	291
statusId	Numerica	300	0	0	7

Status:

Atributo	Tipo de Dato	Cant. de Instancias	Valores Faltantes	Porc. Faltantes	Valores Únicos
statusId	Numerica	139	0	0	139
status	Categorica	139	0	0	139

B. Anexo: Conjuntos de variables utilizadas para el modelado

Conjunto de datos 1: 'year', 'round', 'circuitId', 'date', 'rainy', 'Turns', 'Length', 'driverId', 'constructorId', 'grid', 'Driver Top 3 Finish Percentage (Last Year)', 'Constructor Top 3 Finish Percentage (Last Year)', 'Driver Top 3 Finish Percentage (This Year till last race)', 'Constructor Top 3 Finish Percentage (This Year till last race)', 'Driver Avg position (Last Year)', 'Constructor Avg position (Last Year)', 'Driver Average Position (This Year till last race)', 'Constructor Average Position (This Year till last race)', 'driver_age', 'nationality_encoded', 'wins_cons', 'points_cons_champ', 'wins', 'points_driver_champ', 'laps', 'statusId', 'Weighted_Top_3_Probability', 'Weighted_Top_3_Prob_Length', 'position_previous_race', 'nro_cond_escuderia', 'raceId', 'points', 'prom_points_10', 'Top 3 Finish'.

Conjunto de datos 2: 'driverId', 'raceId', 'year', 'circuitId', 'constructorId', 'grid', 'round', 'position_previous_race', 'wins_cons', 'Weighted_Top_3_Probability', 'Driver Top 3 Finish Percentage (This Year till last race)', 'Constructor Top 3 Finish Percentage (This Year till last race)', 'Weighted_Top_3_Prob_Length', 'Top 3 Finish'.

C. Anexo: Hiper Parametrización de los modelos

V1:
Model: LogisticRegression
Best parameters for LogisticRegression: {'C': 100, 'random_state': 42}

Model: KNeighborsClassifier
Best parameters for KNeighborsClassifier: {'n_neighbors': 20}

Model: SVC
Best parameters for SVC: {'C': 10, 'gamma': 'scale', 'kernel': 'linear', 'probability': True, 'random_state': 42}

Model: RandomForestClassifier
Best parameters for RandomForestClassifier: {'max_depth': 10, 'n_estimators': 100, 'random_state': 42}

Model: DecisionTreeClassifier
Best parameters for DecisionTreeClassifier: {'max_depth': 5, 'random_state': 42}

Model: GaussianNB
Best parameters for GaussianNB: {'var_smoothing': 1e-06}

V2:
Model: LogisticRegression
Best parameters for LogisticRegression: {'C': 100, 'random_state': 42}

Model: KNeighborsClassifier
Best parameters for KNeighborsClassifier: {'n_neighbors': 20}

Model: SVC
Best parameters for SVC: {'C': 10, 'gamma': 'scale', 'kernel': 'linear', 'probability': True, 'random_state': 42}

Model: RandomForestClassifier
Best parameters for RandomForestClassifier: {'max_depth': 10, 'n_estimators': 100, 'random_state': 42}

Model: DecisionTreeClassifier
Best parameters for DecisionTreeClassifier: {'max_depth': 5, 'random_state': 42}

Model: GaussianNB
Best parameters for GaussianNB: {'var_smoothing': 1e-06}

V3:
Model: LogisticRegression

Best parameters for LogisticRegression: {'C': 100, 'random_state': 42}

Model: KNeighborsClassifier
Best parameters for KNeighborsClassifier: {'n_neighbors': 10}

Model: SVC
Best parameters for SVC: {'C': 10, 'gamma': 'scale', 'kernel': 'rbf', 'probability': True, 'random_state': 42}

Model: RandomForestClassifier
Best parameters for RandomForestClassifier: {'max_depth': 20, 'n_estimators': 100, 'random_state': 42}

Model: DecisionTreeClassifier
Best parameters for DecisionTreeClassifier: {'max_depth': 5, 'random_state': 42}

Model: GaussianNB
Best parameters for GaussianNB: {'var_smoothing': 1e-05}

V4:
Model: LogisticRegression
Best parameters for LogisticRegression: {'C': 10, 'random_state': 42}

Model: KNeighborsClassifier
Best parameters for KNeighborsClassifier: {'n_neighbors': 7}

Model: SVC
Best parameters for SVC: {'C': 10, 'gamma': 'scale', 'kernel': 'rbf', 'probability': True, 'random_state': 42}

Model: RandomForestClassifier
Best parameters for RandomForestClassifier: {'max_depth': 20, 'n_estimators': 100, 'random_state': 42}

Model: DecisionTreeClassifier
Best parameters for DecisionTreeClassifier: {'max_depth': 5, 'random_state': 42}

Model: GaussianNB
Best parameters for GaussianNB: {'var_smoothing': 0.01}

V5:
Model: LogisticRegression
Best parameters for LogisticRegression: {'C': 100, 'random_state': 42}

Model: KNeighborsClassifier
Best parameters for KNeighborsClassifier: {'n_neighbors': 20}

Model: SVC

Best parameters for SVC: {'C': 10, 'gamma': 'scale', 'kernel': 'linear', 'probability': True, 'random_state': 42}

Model: RandomForestClassifier
Best parameters for RandomForestClassifier: {'max_depth': 10, 'n_estimators': 100, 'random_state': 42}

Model: DecisionTreeClassifier
Best parameters for DecisionTreeClassifier: {'max_depth': 5, 'random_state': 42}

Model: GaussianNB
Best parameters for GaussianNB: {'var_smoothing': 0.1}

V6:
Model: LogisticRegression
Best parameters for LogisticRegression: {'C': 10, 'random_state': 42}

Model: KNeighborsClassifier
Best parameters for KNeighborsClassifier: {'n_neighbors': 20}

Model: SVC
Best parameters for SVC: {'C': 1, 'gamma': 'scale', 'kernel': 'linear', 'probability': True, 'random_state': 42}

Model: RandomForestClassifier
Best parameters for RandomForestClassifier: {'max_depth': None, 'n_estimators': 100, 'random_state': 42}

Model: DecisionTreeClassifier
Best parameters for DecisionTreeClassifier: {'max_depth': 5, 'random_state': 42}

Model: GaussianNB
Best parameters for GaussianNB: {'var_smoothing': 0.1}

D. Anexo: Feature Engineering

Top 3 finish: variable binaria que indica si el corredor terminó en la pole en esa carrera, este dato sale de la posición final del mismo en esa carrera.

Porcentaje de terminar en la pole del conductor del año anterior: Dato numérico que resulta de promediar la cantidad de poles del conductor en todas las carreras que disputó el año anterior.

Porcentaje de terminar en la pole de la escudería del año anterior: Dato numérico que resulta de promediar la cantidad de poles de la escudería en todas las carreras que disputó el año anterior.

*Porcentaje de terminar en la pole del conductor desde principio de temporada:*Dato numérico que resulta de promediar la cantidad de poles del conductor en todas las carreras que disputó ese año hasta esa carrera.

Porcentaje de terminar en la pole de la escudería desde principio de temporada: Dato numérico que resulta de promediar la cantidad de poles de la escudería en todas las carreras que disputó ese año hasta esa carrera.

Posición promedio del conductor del año pasado: Dato numérico que se calcula promediando las posiciones finales obtenidas por el conductor en todas las carreras de la temporada anterior.

Posición promedio de la escudería del año pasado: Dato numérico que resulta del promedio de las posiciones finales obtenidas por todos los conductores de la escudería en las carreras de la temporada anterior.

Posición promedio del conductor desde el principio de temporada: Dato numérico que se calcula promediando las posiciones finales obtenidas por el conductor en todas las carreras disputadas hasta ese momento en la temporada actual.

Posición promedio de la escudería desde el principio de temporada: Dato numérico que resulta de promediar las posiciones finales de todos los conductores de la escudería en las carreras disputadas hasta ese momento en la temporada actual.

Posición del conductor en la carrera previa: Dato numérico que indica la posición final del conductor en la carrera inmediatamente anterior en la misma temporada.
Promedio ponderado en base a cantidad de curvas: Este promedio ponderado indica que tan consistentemente un piloto obtiene buenos resultados en el podio, ajustando por la dificultad de los circuitos (representada por las curvas). Los pilotos que tienen buenos resultados en circuitos más técnicos (con más curvas) tendrán un valor más alto.

Promedio ponderado en base a largo de pista: Este promedio ponderado indica que tan consistentemente un piloto obtiene buenos resultados en el podio, ajustando por la dificultad de los circuitos (representada por largo de pista). Los pilotos que tienen buenos resultados en circuitos más técnicos (mas largos) tendrán un valor más alto.

Número de corredor para esa escudería: En el contexto de la Fórmula 1, cada escudería cuenta con dos conductores. Si bien la Federación Internacional del Automóvil (FIA)[7] no establece un criterio formal para distinguir entre el piloto principal y el piloto secundario, en la práctica, el piloto principal suele ser aquel con la mayor remuneración económica. Este patrón se asocia frecuentemente con una mayor experiencia, dado que los pilotos más longevos tienden a recibir mayores salarios. En este trabajo, se adopta la edad como criterio operativo para identificar al piloto principal de cada escudería en cada carrera.

[7] https://www.fia.com

Promedio de puntos en los últimas 10 carreras: Para cada conductor, toma el promedio de puntos obtenidos de las últimas 10 carreras (o si corrió 5 por ejemplo, toma las últimas 5), es una forma de mostrar tendencia.

E. Anexo: Árbol generado por el algoritmo DecisionTreeClassifier

