Análisis y predicción del rendimiento en la fórmula 1 mediante un data warehouse y técnicas de aprendizaje automático

Bases de datos masivas (11088)

Esteban Iarza, legajo 134875

Resumen

Este trabajo presenta el diseño e implementación de una solución analítica sobre datos históricos de la Fórmula 1 (1950–2024). A partir de 13 archivos CSV obtenidos de fuentes públicas, se construyó un Data Warehouse con esquema estrella en PostgreSQL 16 que integra 26.759 registros de carreras, clasificaciones y pit stops en tres tablas de hechos y seis dimensiones. Sobre esa base se realizó un análisis exploratorio respondiendo cuatro preguntas de investigación concretas, con 14 visualizaciones generadas. Para el modelado predictivo se formuló el problema como clasificación binaria (podio/no podio), entrenando cuatro modelos, y utilizando los algoritmos de regresión logística, random forest, SVM RBF y XGBoost [1][2], con un split temporal estricto: entrenamiento 1994–2021 y evaluación 2022–2024. La métrica principal es la correlación de Spearman entre el ranking predicho y el resultado real por carrera. Random Forest obtuvo el mejor Spearman mediano (0,680), superando el baseline de grilla sola (0,629), con ROC-AUC = 0,913 y F1 = 0,650. El trabajo se complementa con un dashboard interactivo implementado en Dash/Plotly para explorar el podio predicho versus el real por Gran Premio.

Se contrastaron los resultados contra 5 trabajos relacionados y se validó el modelo sobre la temporada 2025 (no incluida en el entrenamiento ni en el test original), donde Random Forest sostuvo su ventaja sobre el baseline pero XGBoost no.

Palabras clave: data warehouse, esquema estrella, ETL, aprendizaje automático, random forest, correlación de spearman, fórmula 1.

I. Introducción

La fórmula 1 genera una cantidad considerable de datos en cada carrera como tiempos de clasificación, posiciones de grilla, estrategias de pit stop, rendimiento por vuelta, estado de los neumáticos. Esa riqueza hace que sea un caso de estudio interesante para aplicar herramientas de análisis de datos, no solo para hacer estadísticas descriptivas sino para intentar construir algo con capacidad predictiva real. Predecir el resultado, sin embargo, no es trivial: intervienen el

desempeño del piloto, el rendimiento del auto, la estrategia de equipo y factores externos como el clima o los incidentes de carrera, todos interactuando dentro de una misma sesión.

Previo a la etapa de modelado predictivo, resultó prioritario resolver las inconsistencias presentes en las fuentes de origen. Las fuentes disponibles públicamente tienen formatos distintos, valores faltantes no estandarizados, y columnas con tipos inconsistentes. Intentar hacer análisis directamente sobre esos archivos crudos hubiera significado repetir las mismas transformaciones en cada script. Por eso el eje del trabajo es el data warehouse, donde una vez construido, el EDA y el pipeline de ML operan sobre la misma base limpia y estructurada, con vistas analíticas que evitan duplicar lógica.

Sobre predicción en F1 hay trabajos previos en distintas direcciones. Patil y Jain [3] usan análisis de correlación y componentes principales sobre 21 variables de características del auto y el piloto, con datos de 2015-2019, para identificar qué factores explican mas los puntos obtenidos por temporada, encontrando que completar vueltas, el tipo de neumáticos y la posición en pista son las más determinantes. Van Kesteren y Bergkamp [4] optan por un enfoque bayesiano que separa la habilidad del piloto de la ventaja del constructor sobre la era híbrida (2014-2021), y encuentran que el constructor explica cerca del 88% de la varianza en el resultado de la carrera, un contraste interesante con nuestro propio modelo donde la frecuencia histórica del constructor aporta solo 7% de la importancia frente al 60% que concentra la posición de clasificación y grilla. Urdhwareshe [5] usa TabNet, una arquitectura de deep learning para datos tabulares, sobre datos 2010-2023, y encuentra que el modelo a nivel piloto supera en precisión al modelo a nivel constructor. Krzysztoń y Smołka [10] entrenan un ensamble de SVM, Random Forest y Gradient Boosting sobre datos de 2023-2024, clasificando el resultado en cuatro categorías (ganador, top 3, puntos, sin puntos), y logran un F1 macro-promediado de 77,8%, a diferencia de nuestro enfoque, formulan el problema como clasificación multiclase de grano más fino, sobre un conjunto de datos considerablemente más chico y acotado a solo dos temporadas. Nuestro trabajo se diferencia de estos en que aborda el problema como ranking predictivo por carrera evaluado con Spearman, no como puntos de temporada, modelo de varianza global ni clasificación multiclase, y en que el data warehouse no es un paso previo sino un componente central del diseño.

El objetivo principal de este trabajo es predecir el ranking de pilotos por gran premio (GP) usando únicamente información disponible antes de que la carrera comience, y evaluar si ese ranking predicho se aproxima al resultado real de forma significativamente mejor que un baseline que usa solo la posición de largada.

A su vez, se incluyen los siguientes objetivos secundarios:

1- Integrar datos históricos de F1 en un DW con esquema estrella sobre postgreSQL.

2- Realizar un análisis exploratorio respondiendo preguntas de investigación concretas.

3- Calibrar y comparar modelos de aprendizaje automático para el ranking predictivo.

4- Evaluar la correlación entre el ranking predicho y el real mediante la correlación de spearman, comparando contra el estado del arte.

5- Construir un dashboard interactivo que visualice el podio predicho versus el real carrera a carrera.

II. Materiales y Métodos

1. Fuentes de datos

Los datos provienen de dos fuentes públicas. La primera es el dataset de Kaggle *Formula 1 World Championship*, con 13 archivos CSV que cubren carreras, pilotos, constructores, circuitos, clasificaciones, pit stops y tiempos de vuelta desde 1950 hasta 2020. La segunda es la API Ergast, de donde se obtuvieron los datos de las temporadas 2021–2024 mediante descarga directa de sus archivos exportados. El dataset integrado abarca 1.125 Grandes Premios, con foco en el período 1994–2024 donde la cobertura es más completa y consistente.

2. Pipeline ETL

El ETL se organizó en dos fases separadas, lo que tiene una ventaja práctica: la fase de limpieza no necesita que la base de datos esté disponible y se puede correr y depurar de forma independiente.

La **Fase 1** (`etl_clean.py`) procesa los 13 CSV originales. Las operaciones más importantes fueron estandarizar los nombres de columnas, convertir tiempos en formato string del tipo `"1:23.456"` a segundos numéricos, y reemplazar los valores centinela `\N` por nulos. Los nulos en tiempos de clasificación se imputaron con la mediana del año correspondiente, más representativa que la mediana global dado que los tiempos varían entre circuitos, los nulos en posición de clasificación se imputaron con la posición de grilla. También se generaron columnas derivadas: `is_podium` (posición ≤ 3), `is_winner` (posición = 1) y `best_quali_time_s`. El resultado son 13 CSV limpios con 26.759 registros de resultados de carrera.

La **Fase 2** (`etl_load_dw.py`) toma esos CSV limpios y los carga al esquema estrella de PostgreSQL usando SQLAlchemy con inserción por lotes. En esta fase se generan las claves surrogadas incrementales para cada dimensión, lo que desacopla el DW de los identificadores del sistema fuente.

3. Diseño del Data Warehouse

Para el data warehouse optamos por un esquema estrella siguiendo el enfoque dimensional de Kimball y Ross [6], en lugar de un esquema normalizado. La razón es que la mayoría de las consultas que necesitábamos hacer cruzan resultados de carrera con información del piloto, circuito y temporada al mismo tiempo. Con un esquema normalizado esas consultas requerirían varios JOINs encadenados en cada script. Con el esquema estrella, esos JOINs están

preconstruidos en las vistas analíticas y cada componente del sistema los usa directamente sin reescribir lógica.

El DW quedó implementado en PostgreSQL 16 bajo el schema `f1_dw` con tres tablas de hechos y seis dimensiones:

Tabla de hechos	**Grano**
FactRaceResults	Un piloto en una carrera
FactQualifying	Un piloto en una sesión de clasificación
FactPitStops	Un pit stop

Dimensión	**Qué describe**
DimDriver	Piloto (nombre, nacionalidad)
DimConstructor	Equipo y país
DimCircuit	Circuito, país, localidad
DimRace	Carrera, temporada, ronda, fecha
DimDate	Calendario (año, mes, trimestre)
DimStatus	Estado de finalización de la carrera

Sobre este esquema se crearon cinco vistas analíticas. La más importante es `vw_race_analysis`, que es la fuente para el EDA y el pipeline de ML. Las otras cuatro cubren rendimiento por constructor y temporada, carrera completa por piloto, estadísticas por circuito, y análisis de pit stops.

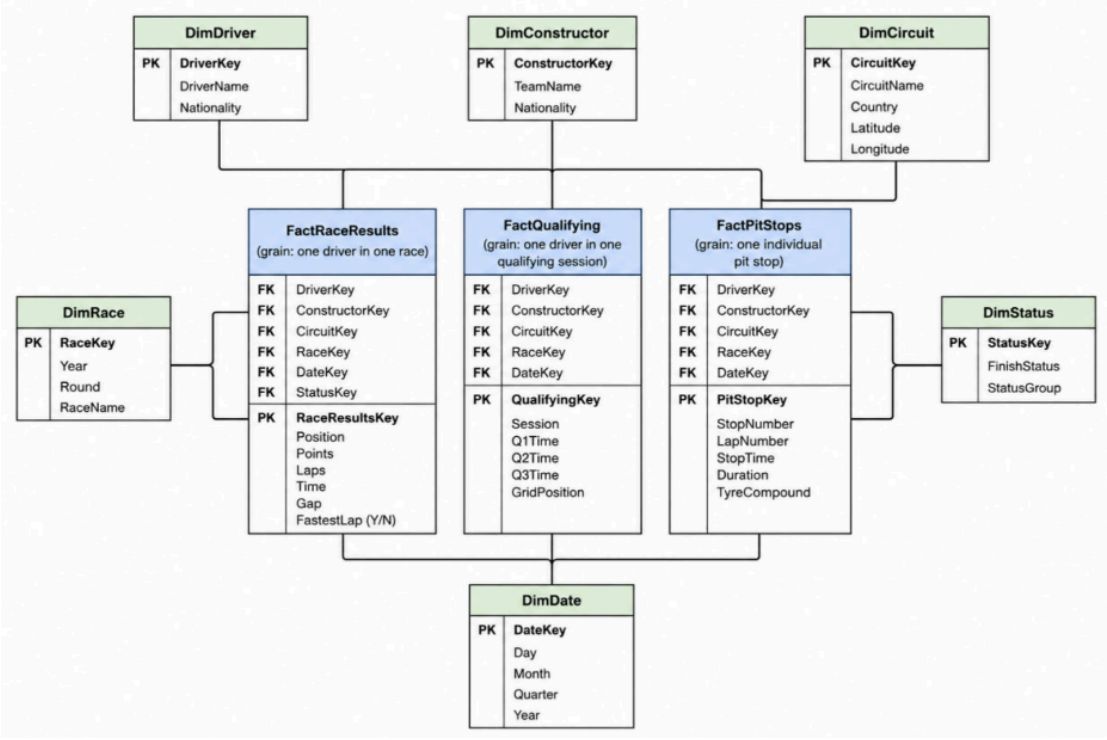


Figura 1: Diagrama del esquema estrella del data warehouse

4. Análisis Exploratorio de Datos

El EDA corre sobre `vw_race_analysis` y responde cuatro preguntas sobre el período 1994–2024.

La primera es qué factores influyen más en llegar al podio, analizando la tasa de conversión a podio según la posición de grilla de largada.

La segunda es cómo evolucionó el dominio de equipos y pilotos por décadas, identificando los constructores con más victorias en cada una.

La tercera es qué tan determinante es la posición de clasificación, calculando la correlación de Spearman entre grilla y posición de llegada, tanto sobre la era moderna completa (1994–2024) como específicamente sobre el período que después se usa como test set (2022–2024).

La cuarta es cómo impactan los pit stops en el resultado, analizando la relación entre cantidad de paradas y posición final, separando por era reglamentaria (antes y después de la prohibición del reabastecimiento en 2010, y del cambio de reglamento en 2022).

5. Feature Engineering y diseño del experimento

La variable objetivo es `is_podium`: vale 1 si el piloto termina en el top 3 y 0 en caso contrario. Formulamos el problema como clasificación probabilística para ranking: el modelo

estima P(podio) para cada piloto y el ranking de la carrera se construye ordenando por esa probabilidad de mayor a menor.

Usamos exclusivamente variables pre-carrera. Las variables post-carrera como posición de llegada, puntos obtenidos, vueltas completadas, tiempo de carrera, estado de finalización, fueron excluidas porque son el resultado de la carrera, no una variable de entrada. Incluirlas hubiera producido un modelo con métricas excelentes pero inútil en la práctica.

Feature	**Qué representa**
grid_position	Posición de largada en la grilla
quali_position	Posición en clasificación (si falta, se usa grid_position)
best_quali_time_s	Mejor tiempo en qualifying en segundos
started_top3	Indicador: salida desde posición 1 a 3
started_top10	Indicador: salida desde posición 1 a 10
grid_quali_diff	Diferencia entre grilla y posición de clasificación
round	Número de carrera en la temporada
constructor_name_freq	Frecuencia histórica del constructor (proxy de calidad)
driver_nationality_freq	Frecuencia histórica de la nacionalidad del piloto
circuit_country_freq	Frecuencia histórica del país del circuito

Para el split optamos por separación temporal y no aleatoria, entrenamiento en 1994–2021 (10.909 registros) y evaluación en 2022–2024 (1.319 registros, 68 carreras). Un split aleatorio mezclaría datos de distintas eras y produciría métricas artificialmente optimistas. La separación

temporal simula el escenario real: un modelo siempre se evalúa sobre datos que no existían cuando se entrenó. El período 2022–2024 coincide además con el reglamento de efecto suelo, que modificó la jerarquía de equipos, lo que hace el test deliberadamente más exigente.

El dataset tiene desbalance de clases: la clase positiva (podio) representa el 14% del entrenamiento y el 15,5% del test. Se usó `class_weight='balanced'` en Regresión Logística, Random Forest y SVM, y `scale_pos_weight` en XGBoost. Todos los features numéricos se normalizaron con StandardScaler.

6. Modelos de aprendizaje automático

Entrenamos cuatro modelos cubriendo un espectro razonable de enfoques: lineal (Regresión Logística), ensambles de árboles (Random Forest [1], XGBoost [2]) y métodos kernel (SVM RBF).

Modelo	**Hiperparámetros principales**
Regresión Logística	C=1.0, solver lbfgs, class_weight balanced
Random Forest	300 árboles, max_depth 8, min_samples_leaf 10
SVM RBF	C=1.0, kernel rbf, gamma scale, probability True
XGBoost	300 estimadores, max_depth 5, learning_rate 0,05

La implementación usó scikit-learn [7] para los primeros tres modelos y la librería nativa de XGBoost para el cuarto.

7. Métricas de evaluación

Reportamos cuatro métricas sobre el test set. El **ROC-AUC** evalúa la capacidad discriminativa independientemente del umbral. El **F1-Score** es la media armónica entre precisión y recall, más representativa que el accuracy en contextos de desbalance de clases, un clasificador que siempre predice "no podio" alcanzaría 84,5% de accuracy sin aprender nada útil. La **correlación de Spearman mediana por carrera** es la métrica principal, para cada Gran Premio se rankean los pilotos según su probabilidad predicha y se calcula el Spearman de ese ranking con el ranking real de llegada, se reporta la mediana sobre las 68 carreras del período. El baseline de comparación es Spearman = 0,629, calculado usando solo la grilla como predictor sobre el mismo test set.

8. Dashboard interactivo

Para la construcción de la interfaz visual, se evaluaron alternativas analíticas como Tableau, Power BI y Python Dash, tomando como referencia el estudio comparativo "Automation of Data Analysis in Formula 1" [8]. Si bien las tres herramientas son eficaces para el análisis de datos, la decisión de implementar el dashboard en Dash se fundamenta en su capacidad para operar dentro del mismo ecosistema de programación. A diferencia de las soluciones de Business Intelligence tradicionales, Dash permite una integración nativa y directa con los modelos predictivos entrenados mediante scikit-learn y XGBoost. Esta arquitectura garantiza que la aplicación web pueda consumir eficientemente las estimaciones probabilísticas generadas por el pipeline de machine learning, manteniendo un flujo de trabajo centralizado y coherente sin requerir capas adicionales para la exportación o ingesta de los resultados.

El dashboard se desarrolló en Dash 2.18 con Plotly y dash-bootstrap-components, corriendo en localhost:8050 dentro del contenedor Docker. Las funcionalidades principales son la selección de Gran Premio por temporada y ronda, la visualización del podio predicho versus el real con las probabilidades de cada piloto, la evolución del Spearman carrera a carrera, y el análisis histórico de rendimiento por piloto y constructor.

III. Resultados

Como resultado de los ajustes descriptos en la sección anterior, a continuación se presentan los resultados del desarrollo. La sección se divide en cuatro partes: los resultados del análisis exploratorio, los resultados del modelado predictivo, contrastados contra los trabajos relacionados, una validación prospectiva sobre la temporada 2025, y el producto de visualización derivado de todo lo anterior, el dashboard interactivo.

1. Resultados del Análisis Exploratorio

El EDA confirmó que la posición de grilla es el predictor individual más relevante, aunque con variabilidad importante según el período. La correlación de Spearman mediana entre grilla y resultado sobre la era moderna es 0,788 (media: 0,735), un resultado consistente con un estudio independiente de Weissbock y Mills [9], que reportan 0,777 para la era turbo-híbrida (2014-2022) con datos y período propios. Sobre el test set 2022–2024 específicamente, el Spearman baja a 0,629, una caída que el estudio de Weissbock y Mills no llega a capturar, porque su corte no incluye el cambio reglamentario de 2022.

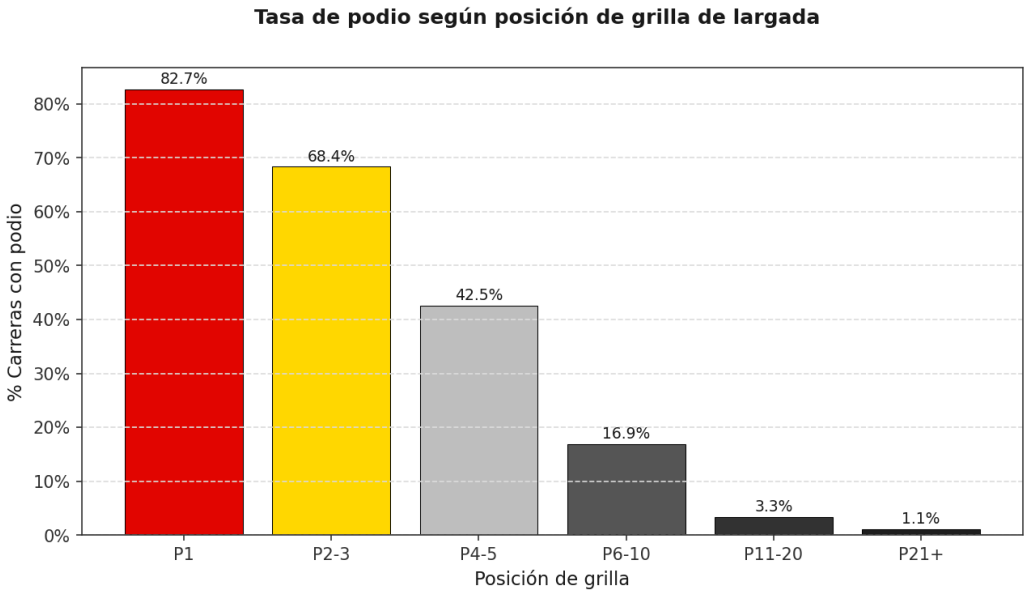


Figura 2: Distribución de podios por posición de grilla. La caída es marcadamente no lineal: del 82,7% de conversión a podio desde la pole se pasa a 68,4% en la segunda fila (P2-3), y a partir de P6-10 la tasa cae por debajo del 17%, estabilizándose cerca de cero desde P11 en adelante, la curva que motivó los features started_top3 y started_top10 del modelo, y la misma asimetría que explica por qué la posición de grilla es tan determinante.

El análisis de dominancia por constructores cuantifica algo visible en la historia: en todas las temporadas post-1990 un equipo concentra entre el 25% y el 70% de las victorias. Esto tiene implicancias directas para el modelado: la variable constructor_name_freq captura parte de esa dinámica, pero no las transiciones entre eras.

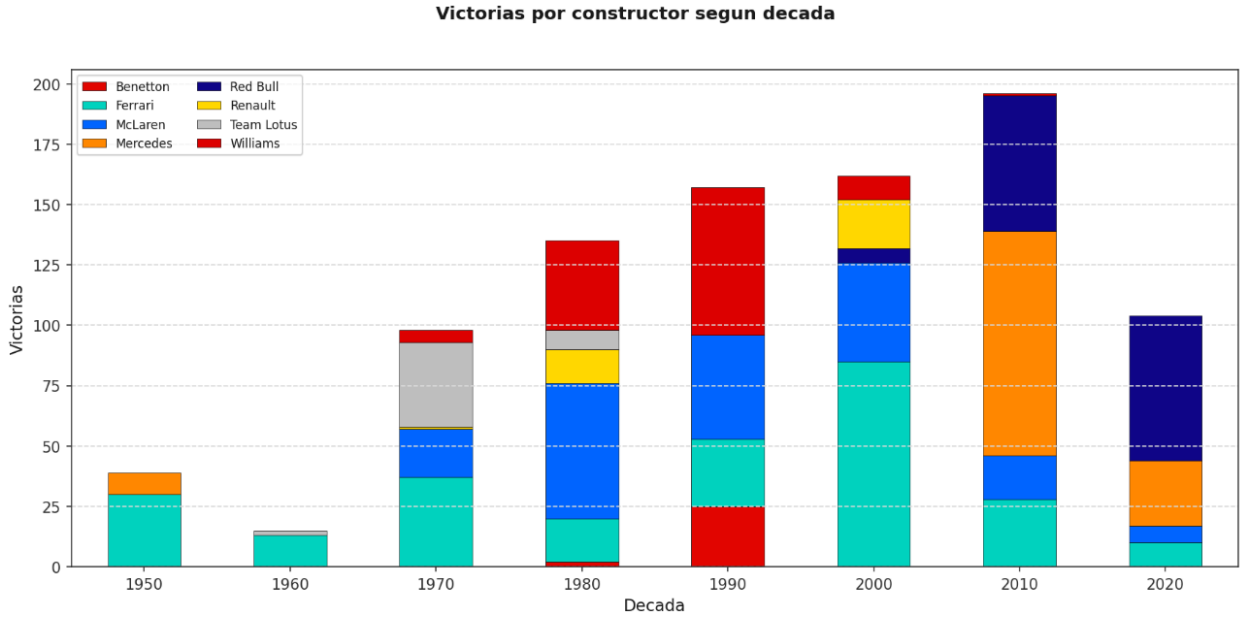


Figura 3: Evolución histórica de victorias por constructor por década. Se observan las cinco eras de dominio identificadas: Williams/McLaren en los 90, Ferrari en la década de 2000, McLaren/Red Bull entre 2007-2013, Mercedes desde 2014 hasta 2021, y Red Bull desde 2022 — el corte que coincide con el inicio de nuestro propio test set.

El análisis de pit stops mostró que en carreras con condiciones variables, lluvia, safety car prolongado, la correlación grilla-resultado cae notablemente. En estos escenarios de alta variabilidad, el rendimiento predictivo del modelo decae, al carecer de información contextual pre-carrera.

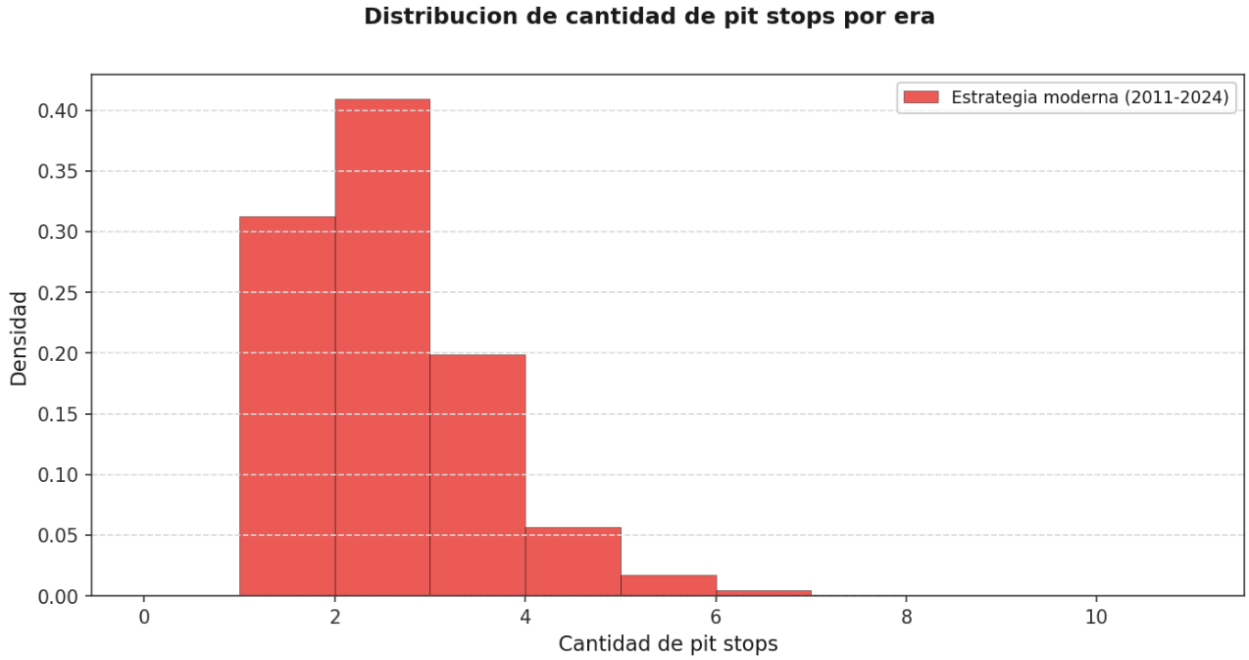


Figura 4: Distribución de pit stops por era reglamentaria. El cambio de régimen es visible en dos quiebres: la prohibición del reabastecimiento en 2010, y el nuevo reglamento de 2022, cada uno desplazando la distribución de cantidad de paradas.

2. Resultados del Modelado Predictivo

Los cuatro modelos se evaluaron sobre el test set 2022–2024 (1.319 registros, 68 Grandes Premios):

Modelo	**ROC-AUC**	**F1**	**Spearman**	**Supera baseline**
Regresión Logística	0,917	0,634	0,654	Sí
Random Forest	**0,913**	**0,650**	**0,680**	**Sí**
SVM RBF	0,888	0,630	0,499	No

XGBoost	0,909	0,641	0,655	Sí
Baseline (grilla sola)	—	—	0,629	—

Random Forest fue el mejor modelo en F1, accuracy y Spearman mediano, con una ventaja de +0,051 sobre el baseline. Un hallazgo contraintuitivo fue que la Regresión Logística superó a XGBoost en ROC-AUC (0,917 vs. 0,909), pese a que XGBoost suele ser más expresivo. Una explicación posible es que la relación entre las variables pre-carrera y la probabilidad de podio tiene una componente lineal importante: la posición de clasificación y la grilla son predictores directos sin interacciones complejas que justifiquen la capacidad adicional de representación de XGBoost.

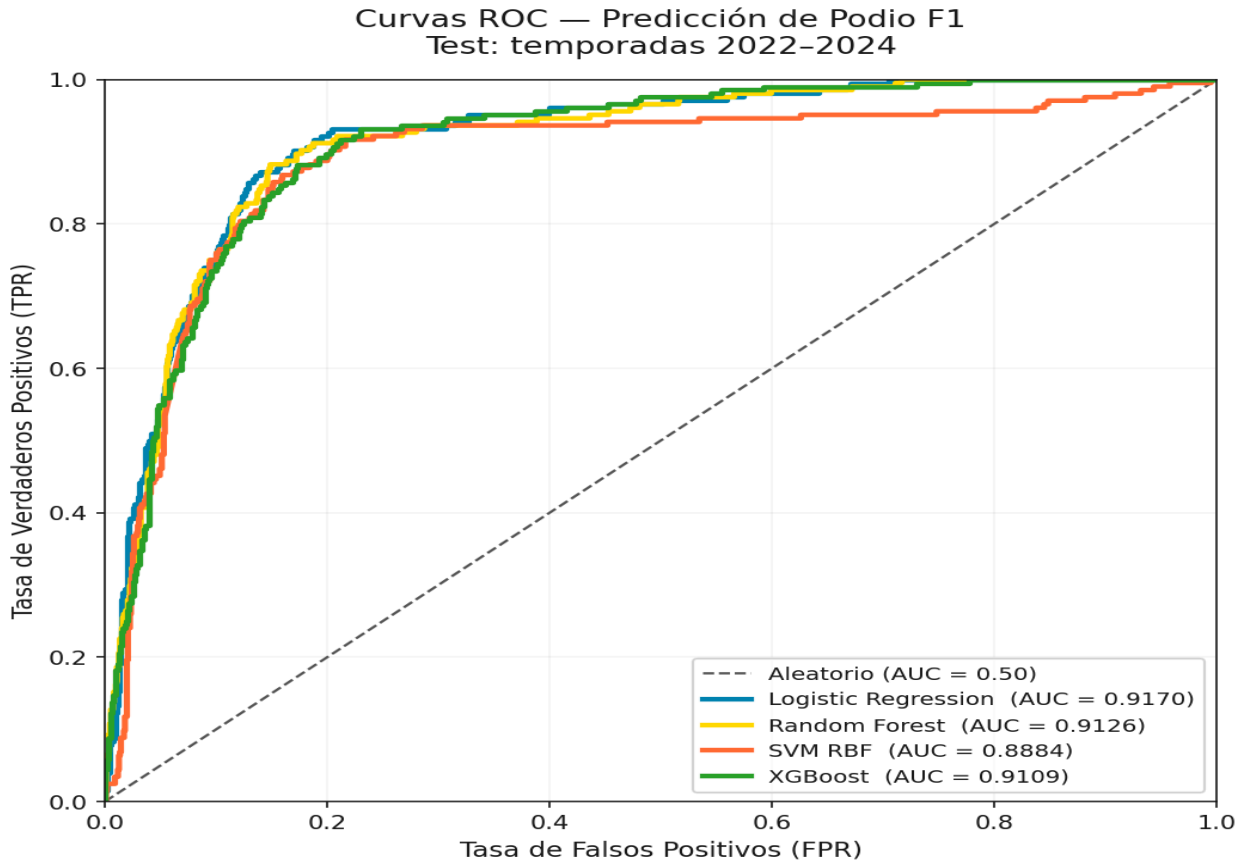


Figura 5: Curvas ROC de los cuatro modelos. Las curvas de Regresión Logística, Random Forest y XGBoost son prácticamente indistinguibles en gran parte del rango, consistente con que los tres capturan una señal similar; SVM RBF se separa levemente por debajo.

SVM RBF no superó el baseline en Spearman (0,499 vs. 0,629). Aunque su ROC-AUC es competitivo, el ranking que produce por carrera es menos coherente que el de los otros modelos. A eso se suma que tardó considerablemente más en entrenar. Dado que tampoco es el mejor en la métrica principal, no hay razón para preferirlo.

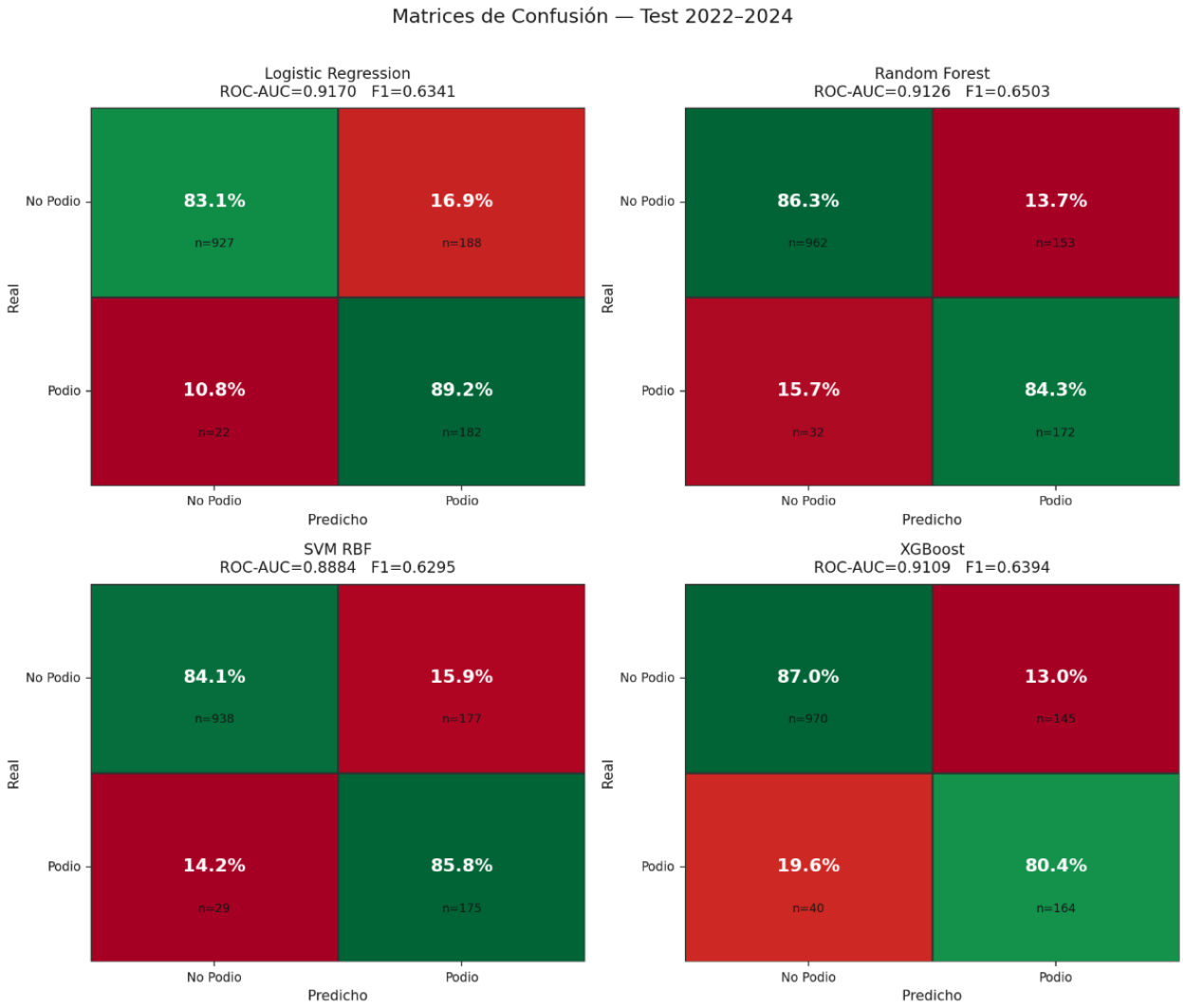


Figura 6: Matrices de confusión. Random Forest y XGBoost muestran la mayor tasa de detección de podios reales (84,3% y 80,4% respectivamente), mientras que SVM RBF, pese a acertar razonablemente en la clase mayoritaria, es el que más confunde la clase minoritaria (podio).

En cuanto a la importancia de features en Random Forest, la posición de clasificación concentra el 60% de la importancia combinada entre `quali_position` (0,32) y `grid_position` (0,27), lo que es consistente con el EDA. La frecuencia histórica del constructor aporta un 7%, capturando algo de la dinámica de dominio de equipo.

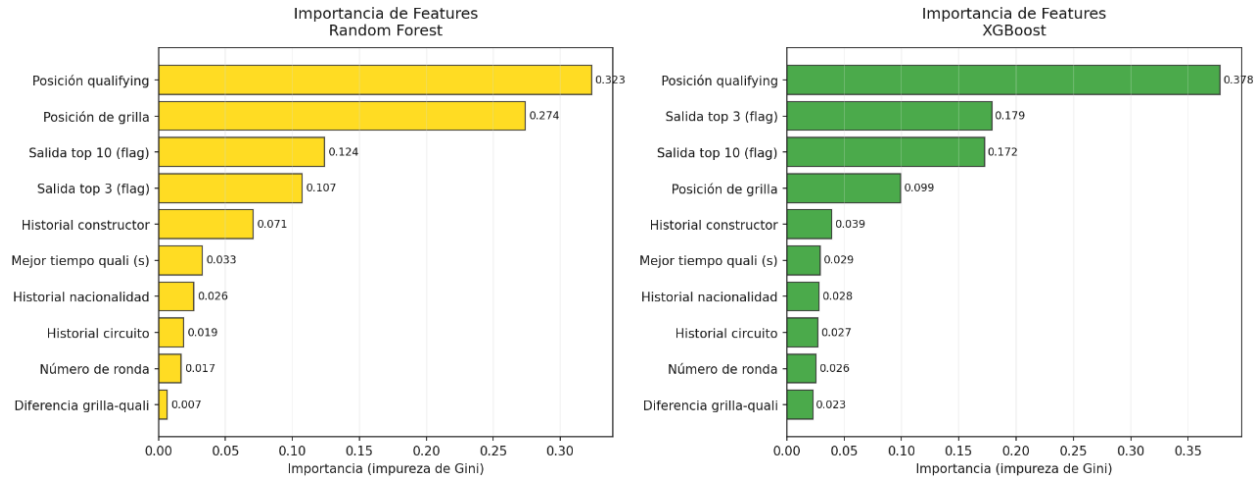


Figura 7: Importancia de features (Random Forest). Las dos variables de clasificación/grilla dominan ampliamente sobre el resto; el resto de las variables (constructor, nacionalidad, circuito, ronda) aporta en conjunto menos del 15% de la importancia total.

La principal limitación del modelo es que fue entrenado en la era Mercedes y evaluado durante la transición al dominio Red Bull. La variable `constructor_name_freq` captura historial acumulado, no el estado actual del equipo, lo que puede llevar a subestimar equipos emergentes en el período de test. El modelo tampoco incorpora variables de contexto de carrera como condiciones climáticas, compuesto de neumático, que requerirían fuentes adicionales.

Además, el uso de class_weight='balanced' para compensar el desbalance de clases distorsiona la calibración de las probabilidades predichas, P(podio) queda sistemáticamente inflado respecto a la frecuencia real, aunque esto no afecta la métrica principal: Spearman mide únicamente el orden relativo dentro de cada carrera, invariante a esa transformación.

Para contextualizar estos resultados frente a la literatura, la siguiente tabla resume los trabajos relacionados junto con nuestro propio resultado, siendo explícitos sobre qué tan comparable es cada uno:

Trabajo	Tarea	Métrica	Valor	¿Comparable con nuestro Spearman 0,680?
Este trabajo (Random Forest)	Ranking predictivo por carrera	Spearman mediano	0,680 (baseline: 0,629)	-
Patil y Jain [3]	Puntos de temporada (PCA/correlación)	Varianza explicada por factor	N/D	No
Van Kesteren y	Descomposición	% varianza	88%	No, mide ajuste

Bergkamp [4]	piloto/constructor	explicada	(constructor)	del modelo sobre todo el dataset, no generalización en un test set separado
Urdhwareshe [5]	Regresión de posición exacta (TabNet)	R^2 / RMSE	R^2=0,75 / RMSE=2,87	No, regresión continua, no ranking/clasificación. dataset de 4.637 filas
Weissbock y Mills [9]	Correlación grilla-resultado por era (descriptivo)	Spearman	0,777 (era 2014-2022)	Parcial, misma métrica, pero puramente descriptivo, no evalúa un modelo en un test set separado
Krzysztoń y Smołka [10]	Clasificación multiclase (4 categorías)	F1 macro-promediado	77,8%	No, multiclase de grano más fino, dataset de solo 2 temporadas (1.635 filas)

Ninguno de estos trabajos resulta directamente comparable número a número: difieren en la tarea, en la métrica, o en ambas cosas a la vez. La comparación que sí es válida es cualitativa: nuestro resultado confirma, en línea con Patil y Jain [3] y Urdhwareshe [5], que cuando la posición de grilla es un predictor tan fuerte, el margen de mejora de un modelo de ML es real pero acotado.

3. Validación prospectiva: temporada 2025

La *Figura 8*, presentada en la sección anterior, ya mostraba que el Spearman fluctúa carrera a carrera dentro del propio test set 2022-2024, cabía preguntarse si el modelo sostiene su desempeño al avanzar en el tiempo, sobre datos que no existían cuando se definieron el split y los hiperparámetros. Para responder esto, se evaluaron los 4 modelos ya entrenados, sin reentrenar nada, sobre la temporada 2025 completa (24 carreras, 479 registros), obtenida de la API Jolpica-F1, sucesora de Ergast tras su discontinuación a fines de 2024.

El baseline de grilla sola en 2025 fue 0,708, notablemente más alto que el de 2022-2024 (0,629), señal de que la temporada 2025 resultó más predecible por la grilla, consistente con que los

equipos ya tuvieron tres años para converger bajo el reglamento de efecto suelo introducido en 2022.

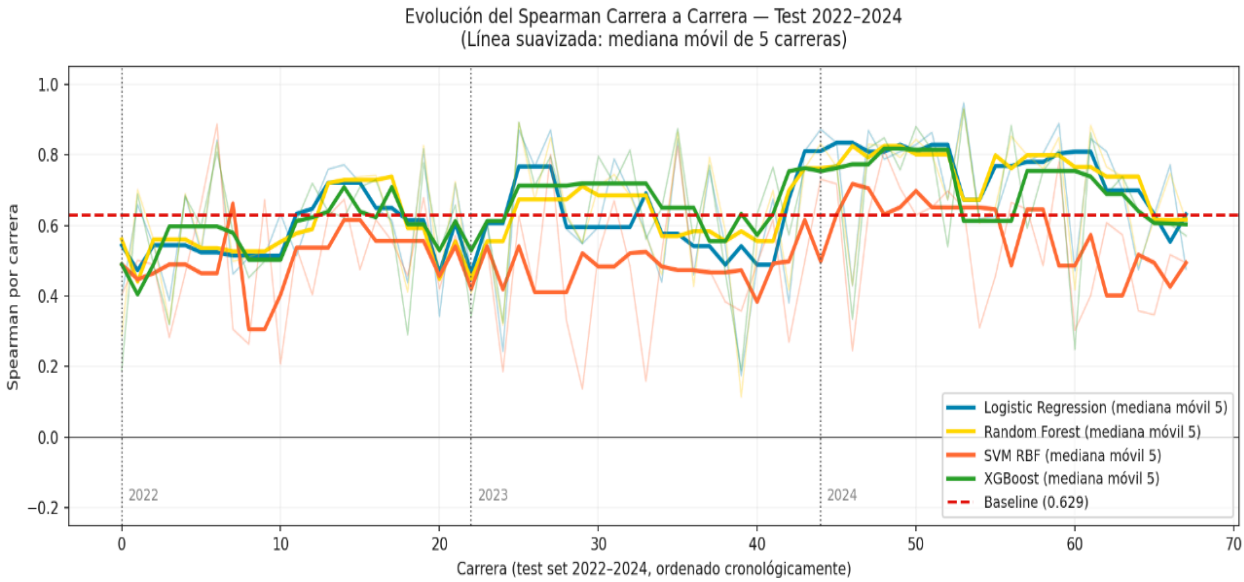


Figura 8: Evolución temporal del Spearman por temporada (test 2022-2024). La variabilidad carrera a carrera, con tramos que superan holgadamente el baseline y otros que caen por debajo, motiva la validación prospectiva sobre la temporada 2025 que se presenta a continuación.

Modelo	Spearman 2025	Spearman 2022-2024	¿Supera baseline 2025 (0,708)?
Regresión logística	0,725	0,654	Sí
Random Forest	0,720	0,680	Sí
SVM RBF	0,538	0,499	No
XGBoost	0,668	0,640	No

Los cuatro modelos mejoraron su Spearman crudo respecto a 2022-2024, pero como el baseline subió todavía más, XGBoost pasó de superar el baseline en el test original a no superarlo en 2025, un resultado que templa el entusiasmo sobre la mejora reportada y refuerza la importancia de comparar siempre contra el baseline específico del período evaluado, no contra un número fijo. Random Forest, en cambio, sostiene su ventaja sobre el baseline en ambos períodos, reforzando su elección como modelo principal.

Como cierre de la etapa de resultados, se agrega una herramienta de visualización analítica que expone estos hallazgos de forma interactiva: el dashboard.

4. Dashboard interactivo

El dashboard permite navegar los resultados del modelo carrera por carrera. La vista principal muestra el podio predicho, piloto, probabilidad, posición de grilla, versus el podio real para cada gran premio del test set. Una segunda vista muestra la evolución del spearman a lo largo de cada temporada, lo que permite ver en qué carreras el modelo funcionó mejor y peor.

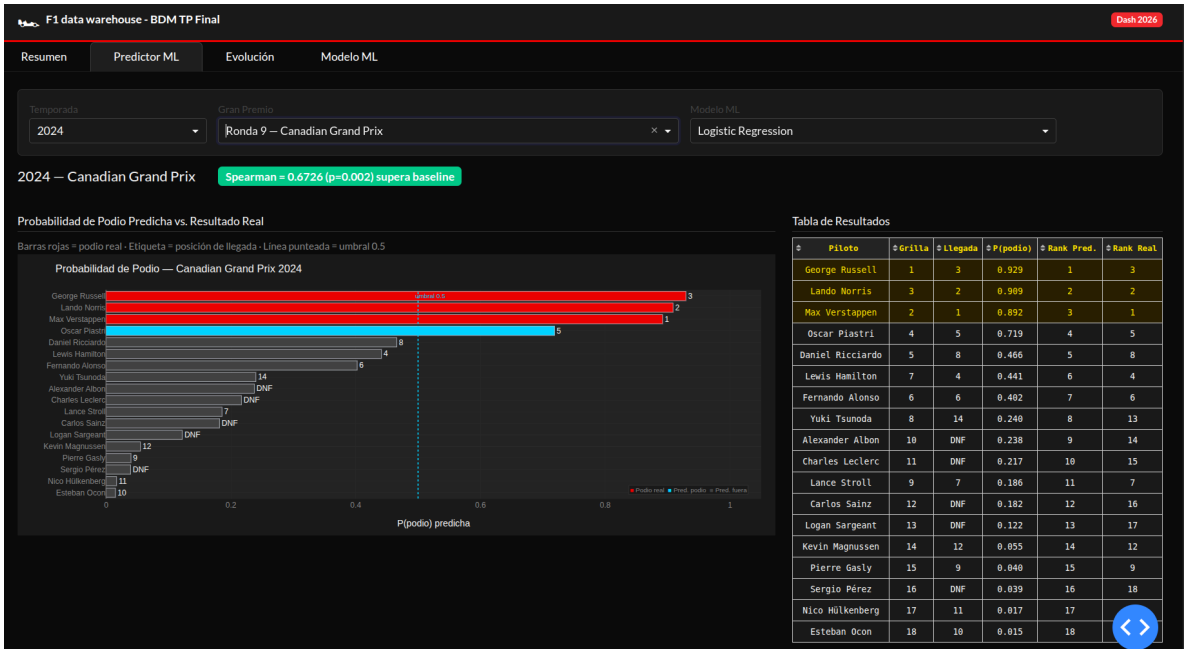


Figura 9: Dashboard: vista podio predicho vs. real, GP Canadá 2024, Random Forest

IV. Conclusiones

En cuanto al modelado, random forest logró un spearman mediano de 0,680 sobre el test 2022–2024, superando en +0,051 al baseline de grilla sola (0,629). Esta métrica, la correlación de Spearman entre el ranking predicho y el de llegada, calculada carrera por carrera y resumida como mediana sobre las 68 carreras del test set, funcionó como una validación directa del objetivo principal: a diferencia de ROC-AUC o F1, que evalúan la clasificación binaria podio/no podio, el spearman mide que tan bien se ordena la carrera completa, que es la que en definitiva se buscaba predecir. La mejora es real pero moderada, lo que es consistente con lo que reportan Patil y Jain [3] y Urdhwareshe [5], cuando la posición de grilla es un predictor tan fuerte, los features adicionales tienen un margen limitado. Que la regresión logística haya superado a XGBoost en ROC-AUC refuerza esa lectura, la relación entre las variables pre-carrera y el podio parece tener un componente lineal considerable.

El rigor del split temporal garantizó una evaluación realista, al evaluar sobre 2022–2024, un período con reglamento nuevo y distinta distribución de fuerzas, el resultado es más representativo de lo que el modelo enfrentaría en producción que un split aleatorio.

El EDA aportó una base necesaria para todo lo que vino después: confirmó que la posición de grilla es, por lejos, el predictor individual más fuerte de llegar al podio (82,7% de conversión desde la pole, contra 21,5% de tasa promedio), y esa asimetría fue la que motivó diseñar el baseline de ML usando solo la grilla. También mostró que el dominio de equipos es una característica estructural de la Fórmula 1, ningún equipo ganó menos del 25% de las carreras en ninguna temporada post-1990, lo que anticipó el rol limitado que terminó teniendo constructor_name_freq como feature. La forma no lineal de la curva de podio por posición de grilla, con una caída empinada seguida de una meseta, motivó directamente dos de los features del modelo (started_top3, started_top10), mostrando que el EDA no fue solo una etapa exploratoria previa, sino que determinó decisiones concretas de diseño en la etapa de modelado.

El trabajo construyó una solución analítica completa sobre datos históricos de la fórmula 1, desde la integración de fuentes heterogéneas hasta la evaluación de modelos predictivos y la visualización interactiva. La decisión de centralizar todo en un data warehouse resultó acertada, las vistas analíticas sobre PostgreSQL simplificaron el EDA y el pipeline de ML, evitaron duplicar transformaciones, y garantizaron consistencia entre los componentes del sistema. Esa base es la que hizo posible resolver el objetivo principal: predecir ranking y compararlo contra el baseline, sin repetir lógica de limpieza en cada etapa.

Como trabajo futuro, las dos líneas más prometedoras serían incorporar variables de contexto de carrera como condiciones climáticas o compuesto de neumático inicial, y explorar modelos de ranking directo como LambdaRank en lugar de inferir el ranking desde probabilidades de un clasificador binario.

V. Referencias

[1] Breiman, L. (2001). Random Forests. *Machine Learning*, 45(1), 5–32.

[2] Chen, T., y Guestrin, C. (2016). XGBoost: A scalable tree boosting system. En *Proc. 22nd ACM SIGKDD* (pp. 785–794). ACM.

[3] Patil, A., y Jain, N. (2022). A data-driven analysis of Formula 1 car races outcome. En *Proc. AICS 2022*. Springer.

[4] van Kesteren, E.-J., y Bergkamp, T. (2022). Bayesian analysis of Formula One race results: Disentangling driver skill and constructor advantage. arXiv preprint.

[5] Urdhwareshe, A. (2025). The use of machine learning in predicting Formula 1 race outcomes. Preprint, Sciety.

[6] Kimball, R., y Ross, M. (2013). *The Data Warehouse Toolkit* (3.ª ed.). Wiley.

[7] Pedregosa, F. et al. (2011). Scikit-learn: Machine learning in Python. *JMLR*, 12, 2825–2830.

[8] *Automation of Data Analysis in Formula 1* (2023). Cal Poly Digital Commons.

[9] Weissbock, J., y Mills, S. (2025). Evaluating the Predictive Power of Qualifying Performance in Formula One Grand Prix. arXiv preprint arXiv:2507.10966.

[10] Krzysztoń, S. A., y Smołka, J. (2026). Application of machine learning for predicting Formula 1 race results. Journal of Computer Sciences Institute, 38, 81–86.