Proyecto final de Inteligencia Artifical

Tendencias salariales globales de Inteligencia Artificial y Machine Learning

Integrantes:

Juan Manuel Agudelo Ortiz

María Isabel Echavarría

Jerónimo Mallarino

Curso de Inteligencia Artificial Explorador – básico

2025

Tendencias salariales globales de Inteligencia Artificial y Machine Learning

Introducción

Cuando se habla del mercado laboral en los campos de inteligencia artificial (IA) y Machine Learning (ML), se describe un entorno en constante evolución, caracterizado por una alta demanda de profesionales y una amplia gama de oportunidades. En la sociedad actual, muchas personas buscan formar parte de este mercado, atraídas no solo por las posibilidades de crecimiento profesional, sino también por el deseo de contribuir a la construcción del futuro. Así, estudiar las tendencias y las oportunidades que ofrece este campo resulta esencial para ganar una ventaja competitiva, tanto para quienes buscan ingresar como para aquellos que ya forman parte de la industria.

Este proyecto tiene como objetivo principal analizar un conjunto de datos relacionados con empleos en IA y ML, con el fin de construir modelos de regresión lineal que predigan el salario anual en dólares (USD). A través de este análisis, no solo se busca comprender mejor las dinámicas del mercado laboral en estos campos, sino también ofrecer herramientas prácticas que faciliten la toma de decisiones estratégicas en reclutamiento y planificación profesional.

Objetivos

Objetivo general: Analizar y caracterizar el mercado laboral en Inteligencia Artificial y Machine Learning mediante técnicas de análisis exploratorio y visualización de datos, con el fin de identificar los factores que más influyen en la determinación del salario y las condiciones laborales de los profesionales del sector a partir de variables como su

experiencia, nivel educativo, industria, tamaños de la empresa, y habilidades requeridas.

Objetivos específicos:

- Identificar tendencias y relaciones entre variables experiencia, educación, tamaño de la empresa, país y tipo de industria.
- Identificar cuales variables son significativas para la creación de los modelos de regresión, observando cómo se comporta el salario en términos de dichas variables.
- Aplicar técnicas de Feature Engineering de manera efectiva para asegurar que los datos de entrenamiento para los modelos estén en adecuadas condiciones.
- Generar visualizaciones interactivas que apoyen la interpretación de los resultados y la toma de decisiones informadas.
- Entrenar modelos de regresión: (Lineal, Árboles de Decisión, Random Forest y XGBoost) para predecir el salario en función de múltiples factores.

Diseño y Desarrollo

Recolección de los datos

Los datos que se van a utilizar en la creación de este modelo de regresión provienen de un Dataset obtenido del sitio web Kaggle. Este Dataset se encuentra bajo el título *“Global AI Job Market & Salary Trends 2025”.* Un Dataset conformado por 19 columnas y 15,000 registros. El autor original que publicó este Dataset informa que este está conformado por datos sintéticos, creados con propósitos educativos, simulando el

mercado de trabajos y tendencias salariales reales de inteligencia artificial y Machine Learning. El dataset incluye múltiples columnas, donde la mayoría resulta ser relevantes para la construcción de un modelo de regresión, como pueden ser los salarios, habilidades requeridas, nivel de experiencia, nivel de educación, entre otros.

	6171
job_id	AI06172
job_title	Head of AI
salary_usd	410273
salary_currency	CHF
salary_local	369246
experience_level	EX
employment_type	FT
company_location	Switzerland
company_size	L
employee_residence	Switzerland
remote_ratio	50
required_skills	Python, TensorFlow, Deep Learning
education_required	PhD
years_experience	19
industry	Education
posting_date	2024-03-01
application_deadline	2024-03-28
job_description_length	1739
benefits_score	7.000000
company_name	Machine Intelligence Group

Este es un ejemplo de un registro del Dataset, más en específico, del registro con el mayor salario en dólares estadounidenses (USD).

Entorno de desarrollo y herramientas

Para llevar a cabo todo el desarrollo del modelo, se trabajó con el lenguaje de Python, en un Jupyther Notebook, con distintas librearías, que serían:

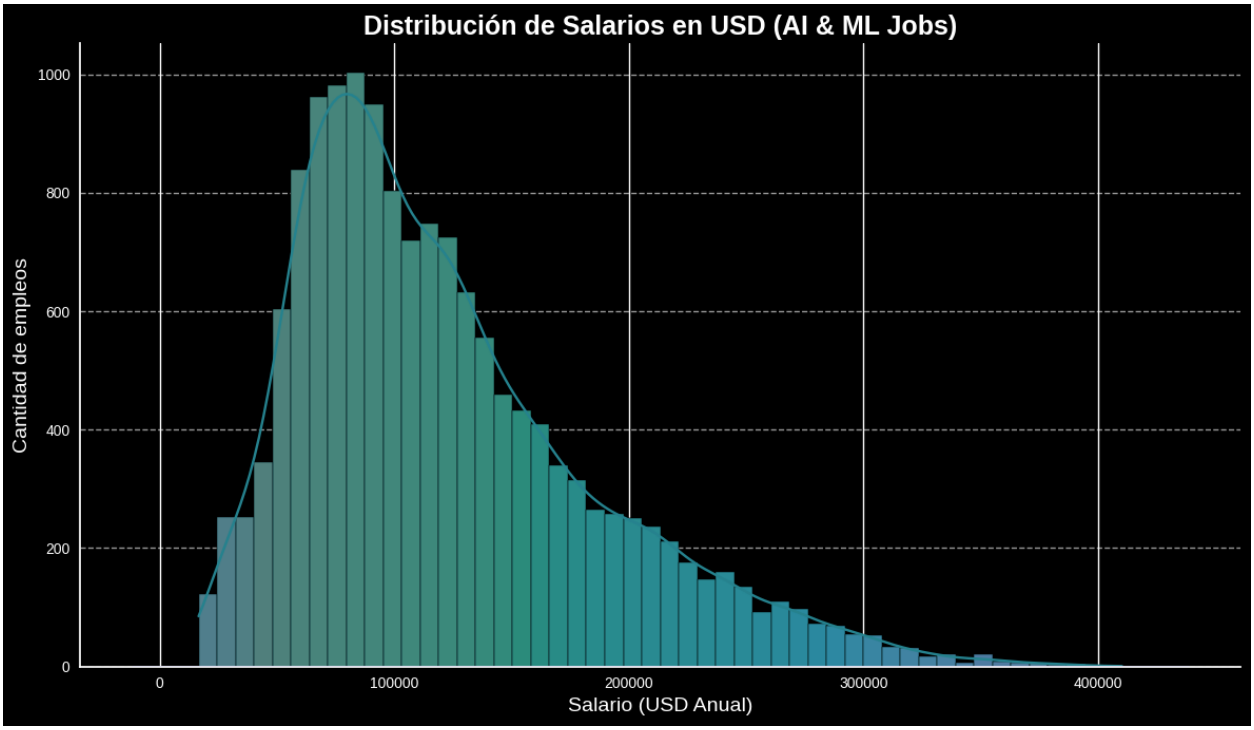
- De uso general: Pandas, Numpy, Heapq.
- Para generar gráficos interactivos y estáticos: Matplotlib, Seaborn, Plotly
- Para análisis y preparación de los datos: Scikitlearn (preprocessing, model_selection, pipeline, compose)
- Para la creación de modelos: Scikitlearn (linear_model, ensemble, tree), XGBoost.
- Para el análisis de métricas de los modelos: Scikitlearn (metrics, model_selection), yellowbrick.

Todo el análisis, Feature Engineering y creación de modelos se realizó en un cuaderno de Google Colab.

Análisis y Limpieza

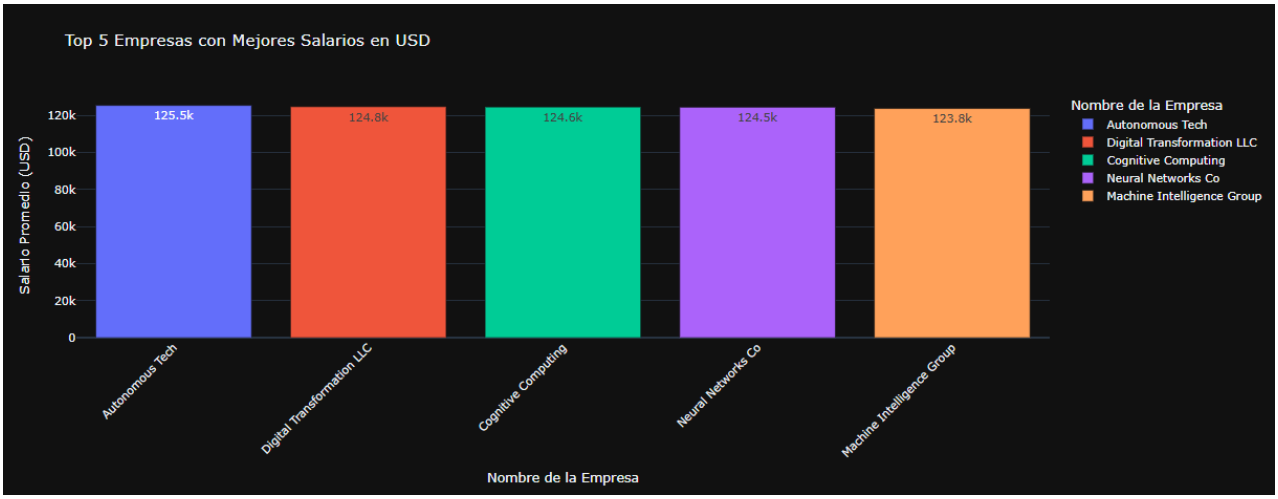
Al hacer las revisiones iniciales de la calidad de los datos, se observa como el Dataset tiene una buena calidad de datos, sin ningún valor nulo ni datos duplicados, datos poco atípicos, columnas nombradas adecuadamente y documentación apropiada que justifican los valores de cada columna.

Posteriormente, se comenzó con el análisis exploratorio de los datos, comparando variables por si solas, al igual que un análisis multivariado. Como primer paso, se definió la que sería la variable dependiente, misma que más adelante sería la que se intentaría predecir con los modelos, la cual es “salary_usd”, que representa el salario de las vacantes en dólares. Esta decisión se basa en la facilidad de comparar salarios considerando su equivalente en dólares, ya que comparar los salarios por los valores que tienen en su modela local resulta complicado e inconsistente, lo cual puede causar problemas en el ajuste de los modelos.



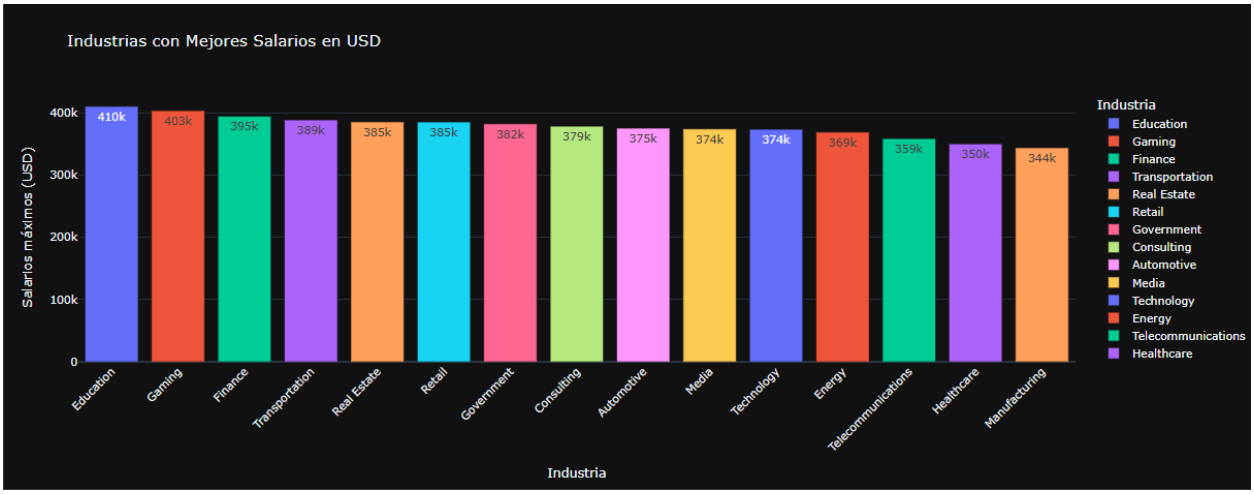
El primer paso fue analizar la variable definida como dependiente y que se quiere predecir. Se observa una distribución natural de los salarios, observando como aproximadamente el 50% de los registros se concentran entre 50,000 USD y 200,000 USD, y como decrece la cantidad de registros mientras mas se aleja de este rango, especialmente para los valores más grandes.

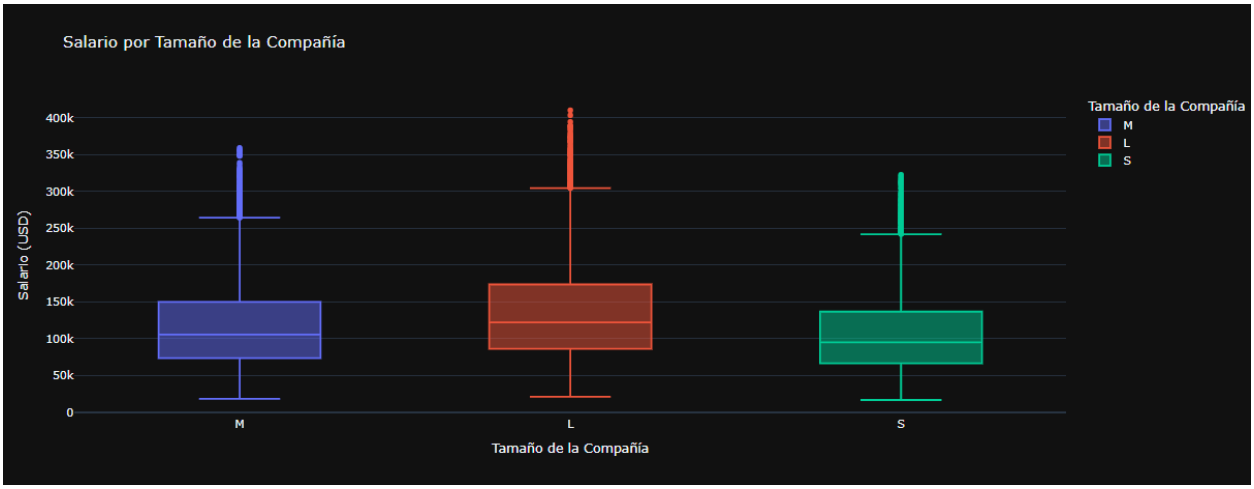
Tras realizar esto, se empezó a revisar como está distribuido este salario con respecto a otras variables, como, por ejemplo, las empresas.

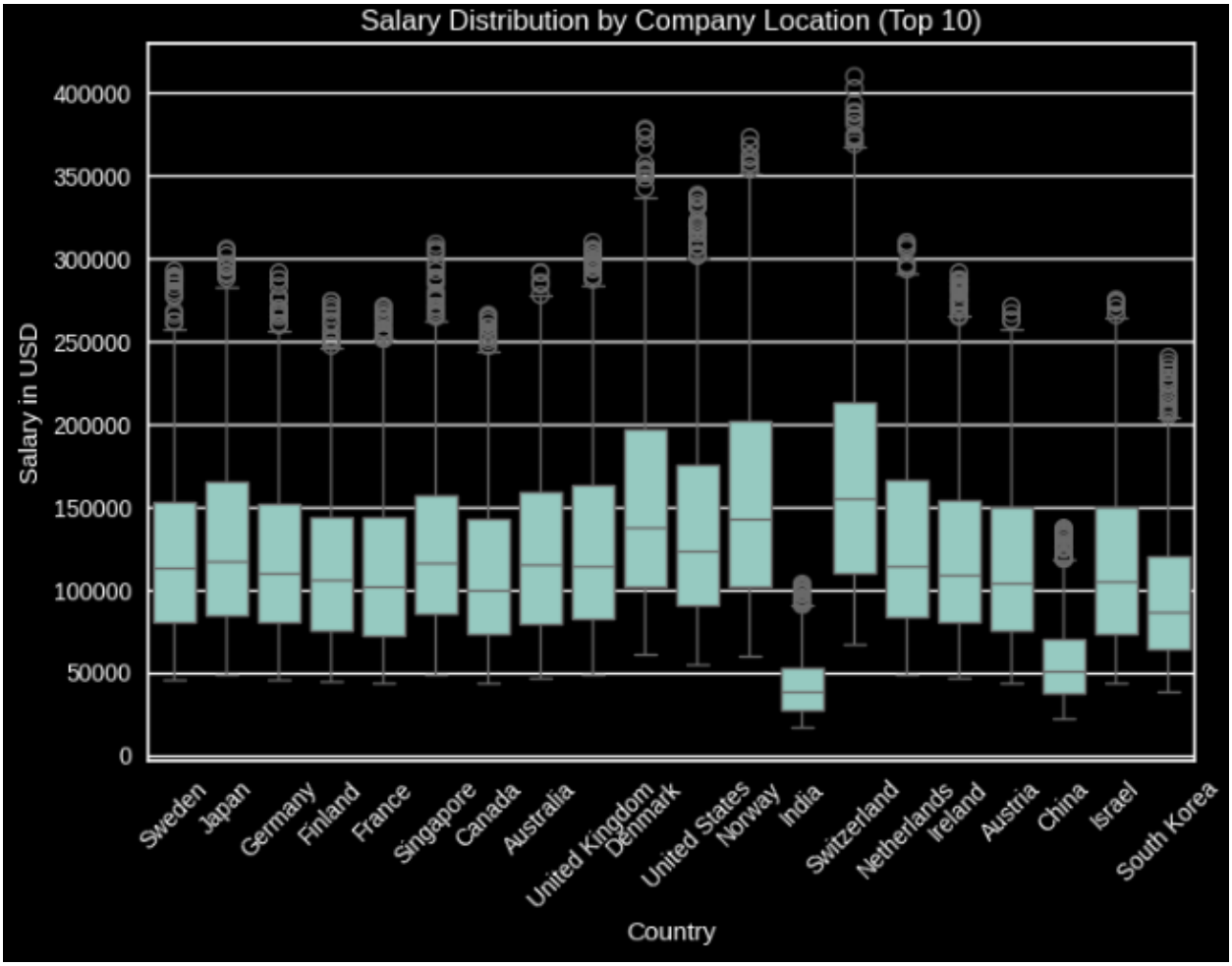


Nuevamente, se observan las cinco empresas que ofrecen el mejor salario en promedio.

Este análisis se realiza de esta manera con el fin de observar como se comporta el salario en términos de otras variables, para luego realizar un proceso de selección de cuales variables se van a tener en cuenta para realizar los modelos de regresión, tras finalizar el análisis. Se adjuntan gráficos adicionales incluidos en el análisis.

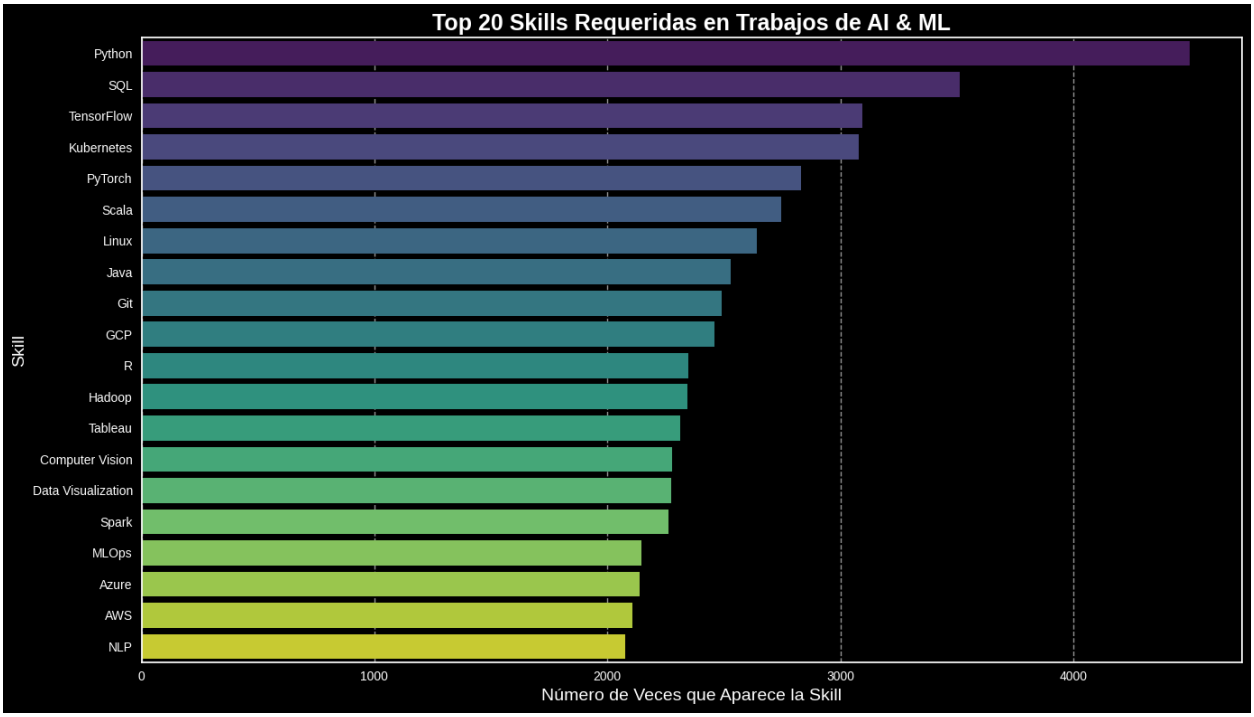






En el Dataset, existe una columna que menciona habilidades requeridas para la vacante a la que se desea aplicar; esta columna contiene varias habilidades distintas

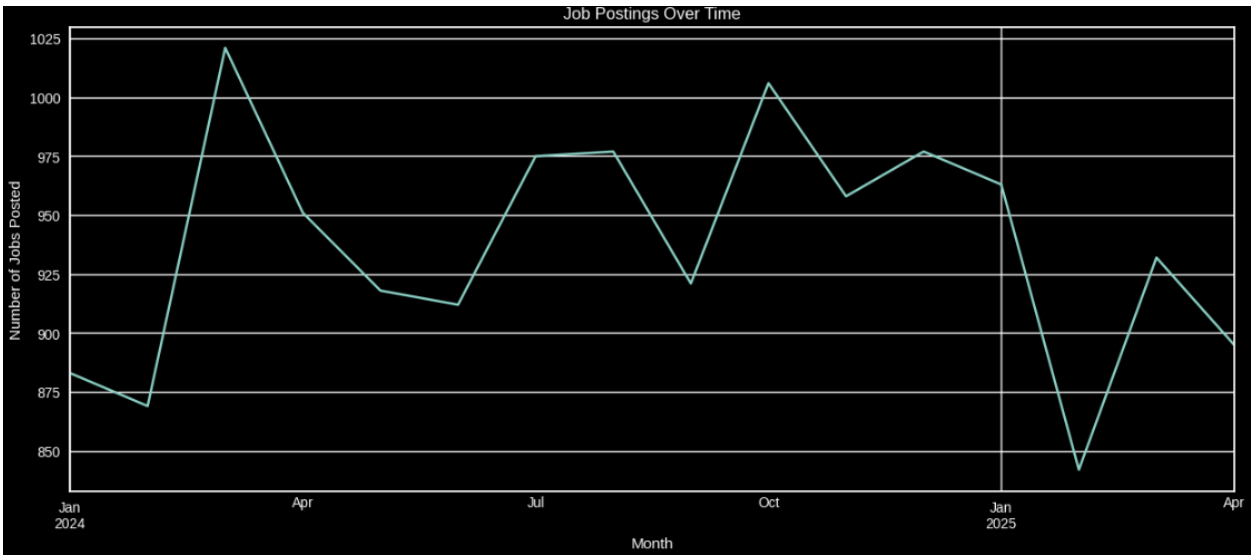
que varían desde lenguajes de programación, hasta Frameworks y plataformas como tal. Para realizar un grafico donde es pudiera analizar cuales eran las habilidades que más se solicitan, hizo falta aplicar distintos métodos para separar estas habilidades para cada registro en sus propias columnas, en una copia del DataFrame donde se venía trabajando.



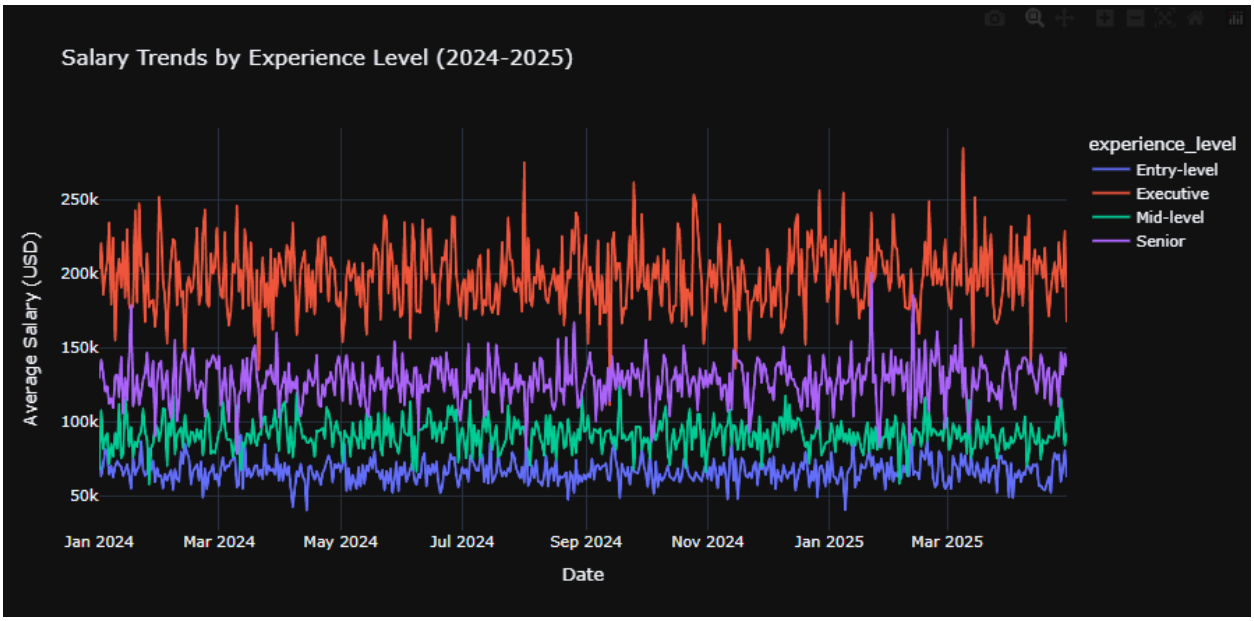
Este grafico da a entender cuántas veces se solicitan las distintas habilidades en los registros del Dataset. Y aunque estas habilidades no sean una columna esencial para considerar en el entrenamiento de un modelo, ayuda también con el entendimiento del mercado y conocer aquellos conocimientos claves que alguien que quiera adentrarse en este mundo debe adquirir.

Para realizar análisis multivariados, se empezó a comparar dos variables en relación con el tiempo, como ver cuantas vacantes se publican según el mes del año.

Un análisis que permite ver si considerar épocas del año es viable o si solo resultaría en problemas de ajuste para el modelo.



De igual forma, para terminar de analizar el comportamiento del salario, se buscó que tan consistente eran los salarios, según el nivel de experiencia requerido, a través del tiempo que cubren los registros del Dataset.



En este gráfico, no se puede observar una “línea” consistente de cuanto es el salario, debido a los altibajos que atraviesa cada rango salarial, alcanzando tanto como los salarios de mas nivel de experiencia, y viceversa.

Feature Engineering

Como primer paso, se seleccionaron cuales variables eran verdaderamente significativas para el modelo:

'experience_level': Muestra cual es el nivel de experiencia requerido para ser considerado en la vacante, incluye: “Entry Level”, “Mid Level”, “Senior”, “Executive”.

'employment_type': Define que tipo de contratación tiene la vacante, variando entre: “Free Lance”, “Part Time”, “Full Time”, “Contract”.

'company_location': Es el nombre de donde está ubicada la compañía que publica la vacante, incluye aproximada 20 paises, y es un dato valioso para predicciones donde se quieran ver vacantes extranjeras, por ejemplo.

'company_size': Expresa el tamaño de la compañía en 3 rangos posibles de empleados: “Small” para menos de 50 empleados, “Medium” para entre 50 y 250 empleados y “Large” para más de 250 empleados.

'remote_ratio': Expresa valores de porcentajes el como es la modalidad de trabajo de la vacante, donde 0 significa completamente presencial, 50 significa hibrido y 100 significa completamente remoto.

'education_required': Muestra la educación requerida por la vacante, variando desde un graduado de bachillerato hasta un PhD.

'years_experience': Representa el número sencillo de cuantos años de experiencia solicita la vacante.

'industry': Clasifica las vacantes según el sector de la industria en el que está operando la empresa.

Tras elegir estas variables y descartar otras que desde un principio se podía asumir que no aportarían nada a la predicción, como la ID de la vacante o el nombre de la empresa, se comenzaron a cuantificar variables con pocos valores únicos, para luego poder realizar un análisis de la correlación de las variables definitivas. Este resultado se consiguió utilizando funciones como *".map()"* que permite darle un valor especifico manualmente a cada valor único, misma funcionalidad que permite darle mas o menos peso a un valor, apoyando mucho al adecuado entrenamiento del modelo.

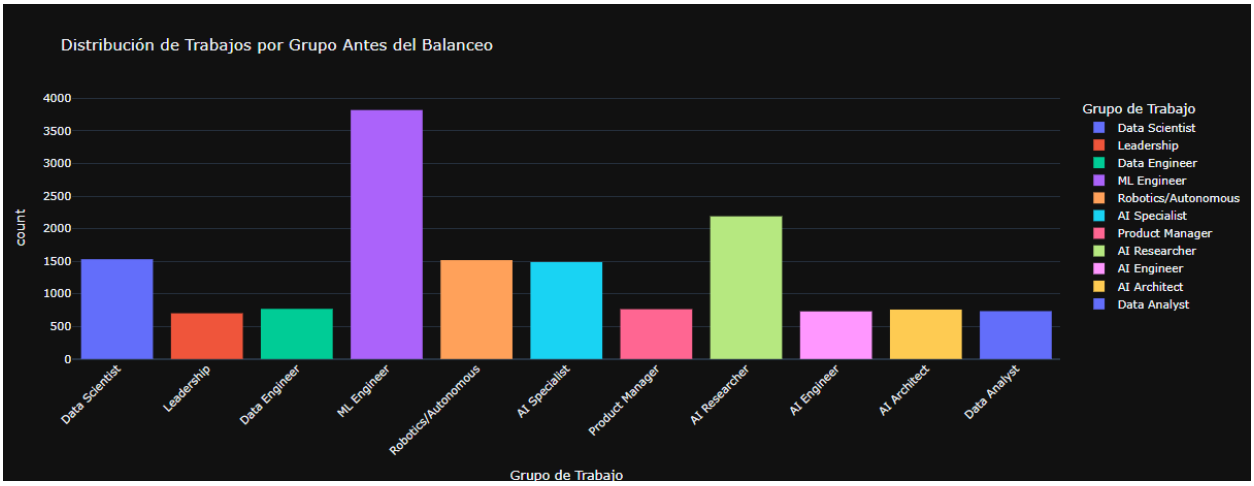
Tras esto, se tomó la columna de *"job_group",* que cuenta con muchos valores únicos en el Dataset original, y se le realizó una reclasificación manual, donde se agrupan títulos de trabajo que se muestran como distintos pero que, esencialmente, son similares, como agrupar distintos títulos de ingenieros, en un solo grupo.

Con estas nuevas columnas, se pasó a integrarlas todas en un DataFrame reducido, donde ya se han eliminado las columnas que no se van a considerar en el entrenamiento del modelo, y también las columnas originales, mismas que sirvieron de base para generar las nuevas columnas codificadas.

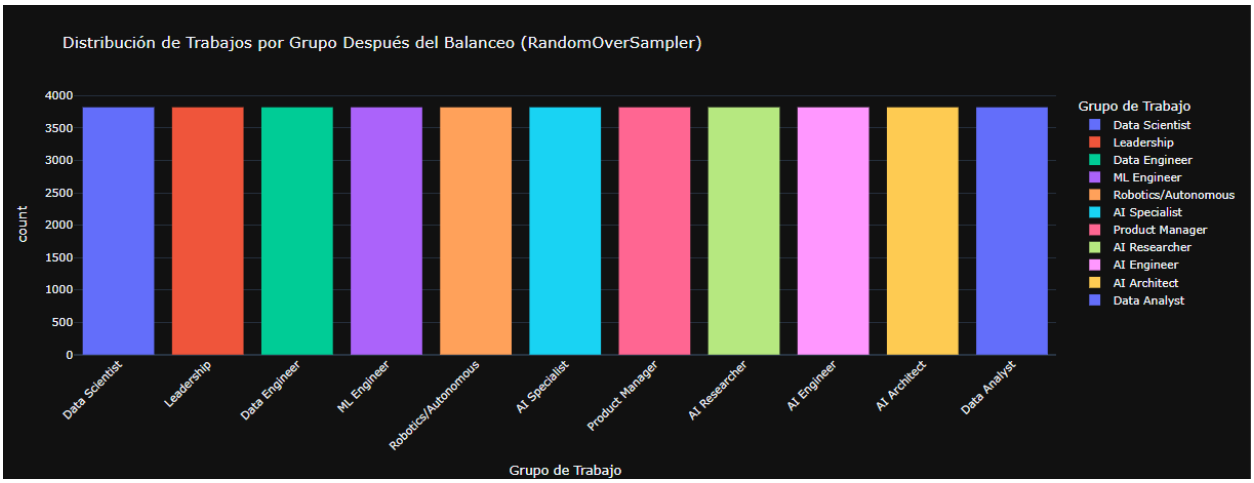
Durante la aplicación inicial de estas técnicas, se consideró aplicarles codificación a todas las variables con One-Hot Encoding, buscando hacer toda la codificación y creación de nuevas columnas de manera automática, pero se observaron

mejoras nulas y aplicaciones de código que no era fácil de comprender o de analizar posteriormente, por lo que se prefirió seguir adelante sin aplicar esta codificación, aunque aún queda mención de esta en el cuaderno con el resto del código.

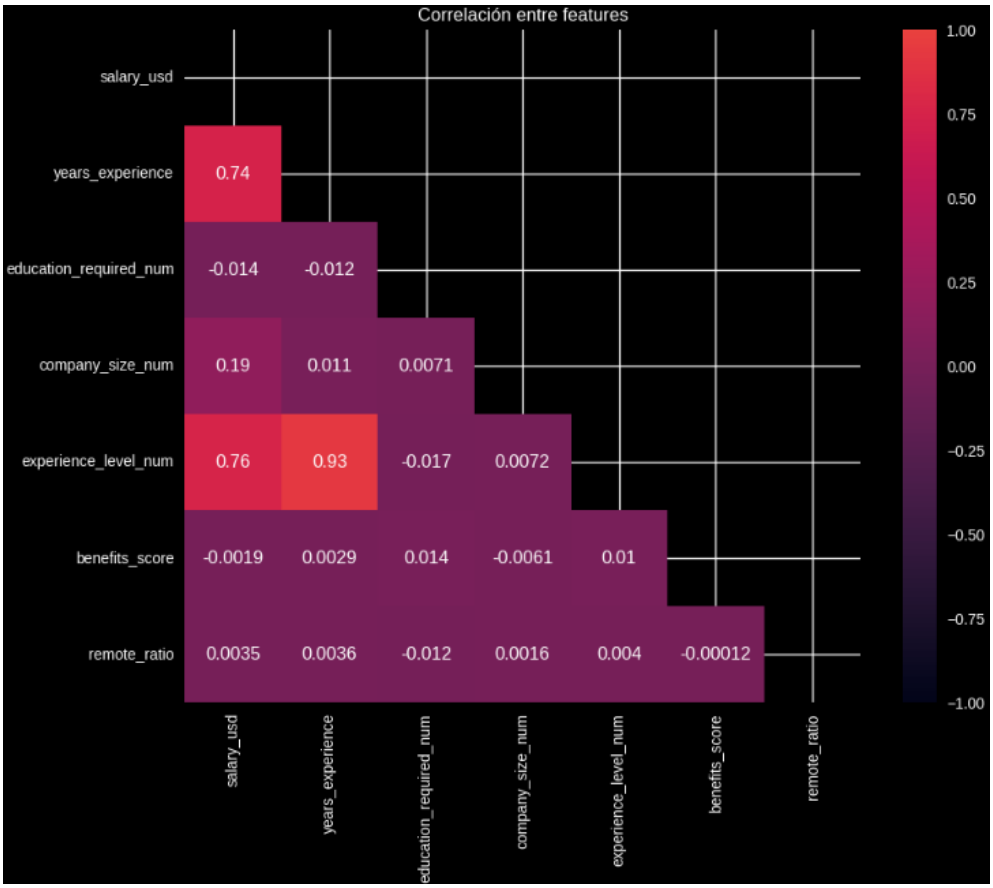
Finalmente, como ultima aplicación, se dirigió el enfoque nuevamente a la columna de *"job_group",* que tras realizar la recategorización de los valores en cada registro, se generó un desbalance significativo entre cada categoría, mismo que podría afectar negativamente al modelo, teniendo efectos donde le modelo puede mostrar una preferencia muy marcada por este grupo.



Por tanto, se decidió aplicar un balanceo a estos valores, para asegurar la integridad del modelo y evitar comportamientos inesperados tras pasar por el entrenamiento de este. Este balanceo se realizó utilizando la técnica SMOTE, que con pocas líneas de código utilizando métodos de la librería RandomOverSampler, realiza un balanceo optimo, dando como resultado los siguientes valores.



Con todos estos pasos hechos, finalmente se puede revisar que tan fuerte es la correlación entre estas nuevas variables, esto creando una matriz de correlación, que después se adaptó en un mapa de calor, que permite una interpretación y análisis de la correlación más sencillo.



Con este gráfico, se observa que la correlación entre estas variables es muy débil o casi inexistente, a excepción de variables donde su correlación es casi natural, como el nivel de experiencia y los años de experiencia. Más allá de estas dos, la otra relación técnicamente fuerte se presenta entre el salario en USD y el nivel de experiencia, como es algo que ya se ha observado en gráficos anteriores, y es algo que hay que tener en cuenta cuando se vayan a realizar predicciones con los modelos.

Entrenamiento y ajuste del modelo

Una vez terminó el proceso de aplicación de las técnicas de Feature Engineering, se procedió a separar las variables dependientes de la dependiente, mismas que se almacenan por separado para poder utilizarlas adecuadamente a la hora de crear los parámetros de los modelos.

```
X = jobs_market_copy_1[['experience_level', 'employment_type', 'company_location',
                        'company_size', 'remote_ratio', 'education_required',
                        'years_experience', 'industry']]

y = jobs_market_copy_1['salary_usd']
```

Ahora, aún hace falta cuantificar unas pocas variables, por lo que en este caso si se prefirió aplicar One Hot Encoder para hacer este proceso automáticamente, con las variables que aún no eran completamente numéricas.

Hecho esto, se definen los modelos que se van a entrenar, los cuales serían Linear Regression, Decision Tree, Random Forest y XGBoost. Se prefirieron estos tres para tener distintas técnicas y decidir cuales verdaderamente son la mejor opción para esta predicción en particular. Para entrenar, predecir y evaluar cada modelo a la vez, se introdujeron cada uno en un diccionario, mismo que luego utilizando ciclos, se les

puede aplicar su respectiva función “.fit()”, la cual es la función maestra para ajustar y entrenar los modelos. La metodología preferida fue un 80/20: 80% de entrenamiento y 20% prueba.

Y una vez termine ese proceso, directamente pasar a mostrar los valores de R^2, MAE y RMSE, métricas clave para determinar el modelo es capaz de predecir adecuadamente los valores con los que se entrenó, y además mantener una buena precisión al hacerlo.

```
------------------------------------------------------------
Model: Linear Regression
R²: 85.289%
MAE: 17865
RMSE: 24280
------------------------------------------------------------
------------------------------------------------------------
Model: Decision Tree
R²: 77.372%
MAE: 20424
RMSE: 30113
------------------------------------------------------------
------------------------------------------------------------
Model: Random Forest
R²: 86.819%
MAE: 16116
RMSE: 22982
------------------------------------------------------------
------------------------------------------------------------
Model: XGBoost
R²: 86.771%
MAE: 16122
RMSE: 23025
------------------------------------------------------------
```

Hay que considerar que, cada métrica tiene un resultado esperado. Siempre se busca que el R^2, en su resultado real, esté lo más cerca posible a 1, pero no en 1 o mayor a 1. La métrica de MAE y RMSE deben ser lo más bajas posible, en ambos casos.

Cada modelo arrojó resultados muy distintos, pero se puede observar que Decision Tree, acorde a las métricas, resulta ser el que peor resultados debería arrojar cuando se intente hacer una predicción, mientras que XGBoost y Random Forest se encuentran bastante parejos en cuanto a sus métricas se refiere, pero las diferencias de cada modelo saldrán a la luz una vez se pase a realizar predicciones y demás pruebas con dichos modelos.

Pruebas y predicciones

Con los modelos entrenados, se crearon pequeñas líneas de código interactivas que permiten crear una predicción con distintos valores que se pueden ingresar manualmente, uno para cada variable que fue considerada e incluida en el entrenamiento del modelo. Para que luego, una vez se ejecute la línea de código, arroje los resultados predichos por cada modelo. Esto se observa de la siguiente manera:

Para comparar realmente el funcionamiento de esta estrategia de una forma visual, tomamos los datos específicos de un registro y los evaluamos en el modelo como se ve en la imagen anterior.

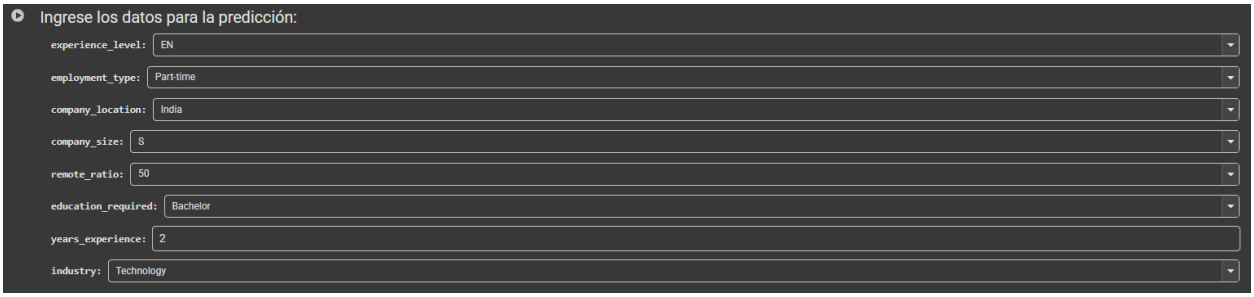
Se usó el registro número 12333:

```
salary_usd                  112203
experience_level                MI
employment_type                 FL
company_location           Ireland
company_size                    L
remote_ratio                   50
education_required            PhD
years_experience                2
industry                    Media
benefits_score                8.6
education_required_num          4
company_size_num                3
experience_level_num            2
job_group              ML Engineer
Name: 12333, dtype: object
```

Al evaluar estos datos en el modelo, podemos comparar los salarios. El salario del registro 12333 es de $112,203 dólares anuales, mientras que el salario final que sugieren los modelos es de $95,661 dólares anuales. Gracias a esto podemos verificar que el conjunto de modelos funciona y arroja resultados bastante acercados.

- Experimento:

Para verificar el comportamiento del modelo es importante hacer experimentos y comparar. Esta vez, se ingresaron los datos que se creen pueden llevar a un salario bajo al modelo:



Como se ha visto durante el análisis exploratorio de datos, un nivelo de experiencia laboral bajo (Entry), un puesto de trabajo en India, en una compañía pequeña, con poca experiencia requerida en la industria de la tecnología, son variables que se han visto estar relacionadas con salarios bajos.

El resultado del salario que predice el modelo es el siguiente:

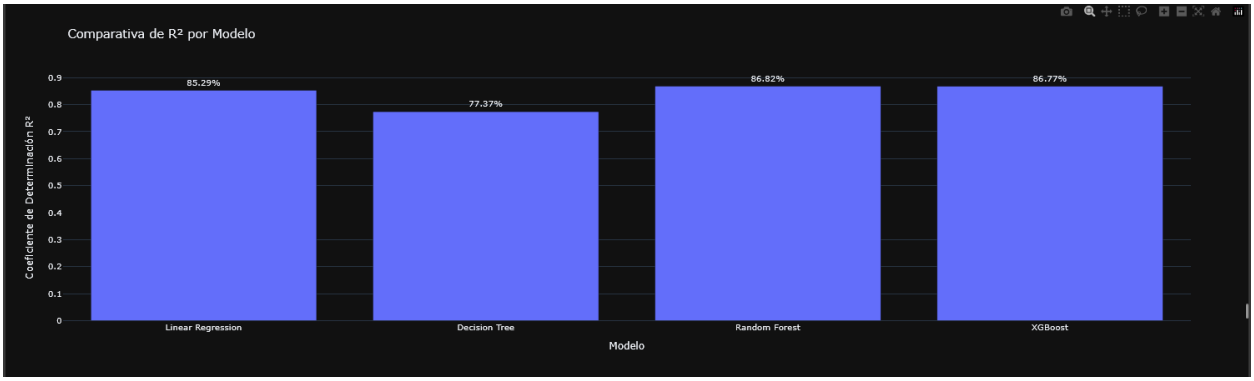
```
Predicciones de salario (USD):
- Linear Regression: -22186.01
- Decision Tree: 17942.00
- Random Forest: 24054.37
- XGBoost: 19371.90

------------------------------
Promedio de las 3 predicciones más altas: 20456.09 USD
```

Aunque el resultado final es de $20,456 dólares anuales, gracias a tomar el promedio de las tres mayores predicciones, es necesario darle atención al salario negativo que arrojó la regresión lineal. En este caso es debido al comportamiento del modelo que se pueden generar estos tipos de valores exagerados o absurdos.
Para poder solucionar esto se requiere de tomar solo las predicciones que sean salarios negativos y darle esa condición al modelo.

Estadísticas de métricas

- R^2: Es un coeficiente que corresponde a la medición de la proporción de variabilidad de la variable dependiente (target). Lo que permite ver que tan bien se ajusta el modelo a los datos de muestra.



En la gráfica anterior se observa el comportamiento de este coeficiente. A partir de aquí se puede asumir que el XGBoost es el modelo que más precisión demuestra. Adicional, el Random Forest es también preciso y aunque tiene un coeficiente parecido a la regresión lineal, ambos tienen comportamientos muy distintos que se evidencian en el análisis anterior.

- MAE**:** Es un indicador que corresponde a la medición del error absoluto medio entre los valores reales y los valores predichos por el modelo. Mide, en promedio, cuánto se equivoca el modelo al predecir. Mientras más bajo sea el MAE, mejor será la precisión del modelo.
- RMSE: La raíz cuadrada del error cuadrático medio. Es un indicador que mide la magnitud promedio del error, penalizando más fuertemente los errores grandes. Un RMSE bajo indica que las predicciones del modelo se ajustan mejor a los valores reales.

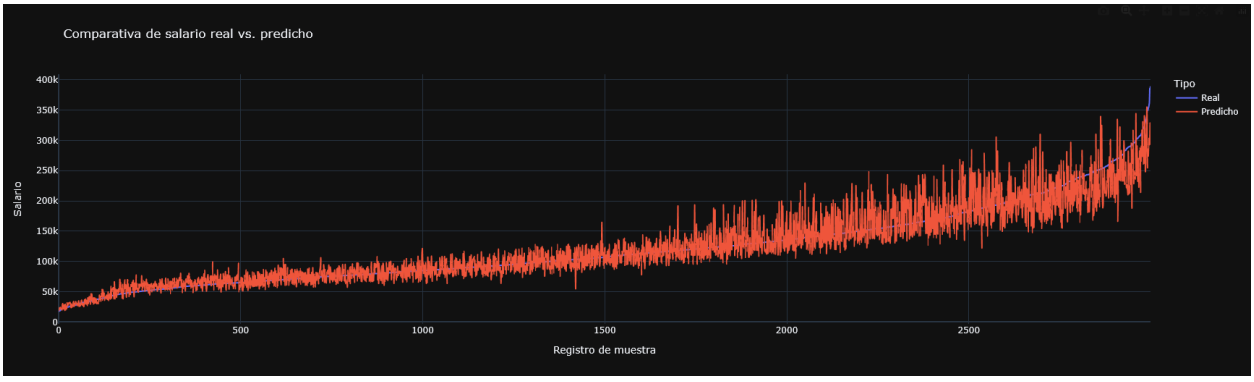
En la gráfica anterior se puede analizar el nivel de error de cada uno, identificando qué tan cerca están las predicciones de los valores reales. Un RMSE más bajo señala que el modelo no solo tiene menos errores promedio, sino que también penaliza más los errores grandes, por lo que muestra qué tan sensibles son los modelos a desviaciones importantes.

En este caso, se observa que los modelos Random Forest y XGBoost presentan los valores más bajos de MAE y RMSE, lo que sugiere que tienen un mejor desempeño predictivo, con errores más pequeños y consistentes.

Gráficos de comportamiento

- XGBoost

Siguiendo con el análisis del comportamiento de los modelos. Se usaron gráficas escalables (o interactivas) con la librería plotly.express que facilitan la visualización de este comportamiento.



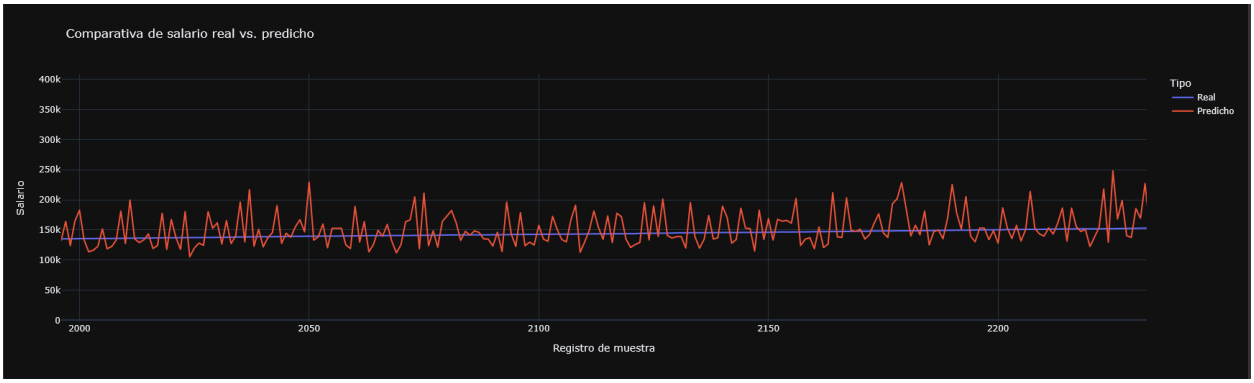
En la gráfica anterior se muestra la comparación entre los valores reales de salario y los valores predichos por el modelo XGBoost.

Se observa cómo la línea de predicciones sigue de forma cercana la tendencia de los

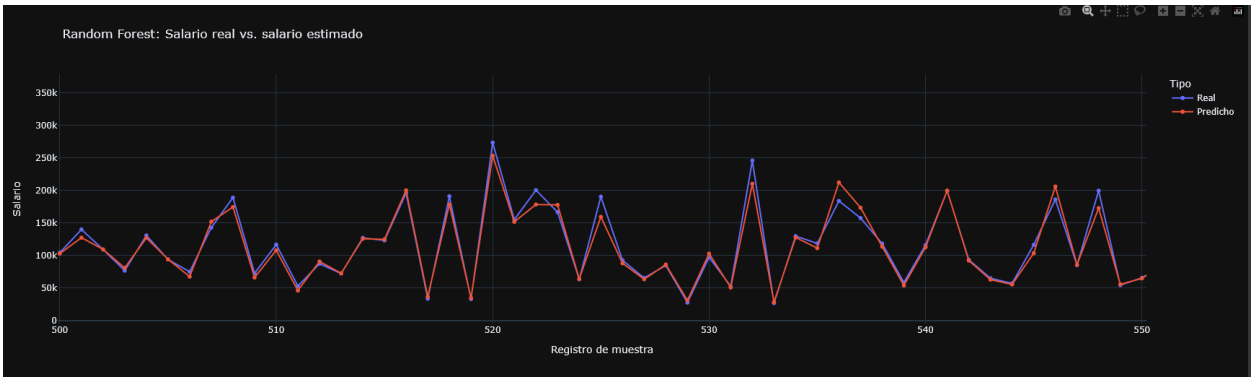
valores reales, lo que indica que el modelo logra capturar adecuadamente el comportamiento de los datos de muestra.

Cuantas más juntas estén las líneas, menor es la diferencia entre el salario real y el salario predicho, lo que refuerza la calidad del modelo. Sin embargo, también se pueden notar pequeñas fluctuaciones o picos, que representan los registros donde el modelo presenta un mayor error de predicción, algo normal en modelos de regresión aplicada a datos reales.

Aquí se puede ver más de cerca el funcionamiento de esta relación:



- Random Forest



En la gráfica se presenta una muestra reducida de registros que compara el salario real con el salario estimado por el modelo Random Forest.

Se observa cómo la línea de predicciones (roja) sigue la forma de la línea real (azul), mostrando que el modelo es capaz de aproximar correctamente los valores reales para la mayoría de los casos. También se aprecian algunos picos y caídas donde la línea roja se desvía, lo que indica casos con mayor error de predicción, comunes en datos reales con variabilidad alta.

Conclusión

Al finalizar el análisis y la construcción de modelos predictivos, se han alcanzado los objetivos propuestos para este proyecto, obteniendo una comprensión profunda del mercado laboral en Inteligencia Artificial y Machine Learning.

- Se confirmó que el nivel de experiencia y los años de experiencia son los factores con la correlación más fuerte y directa con el salario en dólares (USD). Otras variables como la ubicación de la empresa, la industria y el tamaño de la compañía también influyen, aunque en menor medida.
- De los cuatro modelos de regresión entrenados, XGBoost y Random Forest demostraron ser los más precisos y fiables para predecir salarios. Ambos alcanzaron un coeficiente de determinación (R2) cercano al 87% y obtuvieron los valores más bajos de Error Absoluto Medio (MAE) y Raíz del Error Cuadrático Medio (RMSE).
- El modelo de Regresión Lineal, aunque presentó un R2 aceptable, demostró ser propenso a generar predicciones ilógicas (como salarios negativos) bajo ciertas combinaciones de variables. El modelo de Árbol de Decisión fue el de menor rendimiento general.

- Las técnicas de Feature Engineering, como la recategorización manual de los títulos de trabajo y el balanceo de clases con la técnica SMOTE, fueron fundamentales para mejorar la calidad de los datos y, en consecuencia, el rendimiento y la fiabilidad de los modelos predictivos.
- Las pruebas realizadas, comparando predicciones con registros reales del dataset, verificaron que los modelos son capaces de generar estimaciones salariales cercanas a los valores reales, capturando adecuadamente las tendencias del mercado.

Recomendaciones

Basado en los hallazgos del análisis exploratorio de datos y el comportamiento de los modelos, se pueden ofrecer las siguientes recomendaciones para profesionales y aspirantes en el campo de la IA y el ML:

- Priorizar la Experiencia Práctica: Dado que la experiencia es el factor más influyente en el salario, se recomienda a quienes inician su carrera buscar activamente oportunidades para construir un historial sólido que les permita acceder a roles mejor remunerados en el futuro.
- Desarrollar Habilidades de Alta Demanda: El análisis de habilidades requeridas es claro: Python es la competencia fundamental. Para maximizar la competitividad, es crucial complementar el dominio de Python con habilidades en SQL, y frameworks de Machine Learning como TensorFlow y PyTorch.
- Considerar la Ubicación Geográfica: La ubicación de la empresa es un factor determinante en la compensación. Países como Estados Unidos y Suiza tienden

a ofrecer salarios promedio más elevados en comparación con otros. Profesionales con flexibilidad geográfica pueden aprovechar estas diferencias del mercado.

Referencia

GitHub del proyecto:

https://github.com/M4ll4/Proyecto-Grupo-3-IA.git

Bibliografía

- Dataset (Kaggle)

Sajjad, B. (2025). Global AI Job Market & Salary Trends 2025 [Data set].

https://www.kaggle.com/datasets/bismasajjad/global-ai-job-market-and-salary-trends-2025

- Colab (Jupyter Notebook)

https://colab.research.google.com/drive/1S6uA-1e2xJlJqRTlADmKH45M_jTO4QD0?usp=sharing