**Contenido:**

1. **Introducción:**
   1. **Objetivos.**
   2. **Preguntas de negocio.**
   3. **Plan de acción modelo predictivo y análisis exploratorio.**
2. **Análisis exploratorio.**
3. **ETL de los datos**
4. **Modelo predictivo con red neuronal.**
   1. **Escogencia hiperparametros por bibliografía.**
   2. **Escogencia hiperparametros por Iteración y estructura final red neuronal.**
   3. **Resultados.**
5. **Despliegue**
6. **Conclusiones**
7. **Bibliografía**

**1.1**

**Objetivo global:** Desarrollar una plataforma de analítica de datos para ayudar a rectores y administradores escolares en Colombia a entender, visualizar y predecir el desempeño de los estudiantes en las pruebas Saber 11.

1.2

1. ¿Cuáles son los principales factores que impactan en el rendimiento académico de los estudiantes en las pruebas Saber 11?

2. ¿Cómo los temas socioeconómicos y demográficos juegan un papel sumamente importante en los resultados de las pruebas Saber 11?

1.3.

En primer lugar, para llevar a cabo el análisis exploratorio, debido a la variedad de los factores de influencia, se seleccionan por categoría los factores de influencia. Partiendo de un análisis más enfocado a las instituciones, luego una caracterización socioeconómica y culminando con un análisis ya más enfocado a los resultados en las pruebas y esos factores que pueden llegan a ser tan determinantes para estos puntajes.

Para el modelo predictivo se entrenaron cinco redes neuronales para predecir los resultados por competencias académicas (ingles, matemáticas, Ciencias sociales, Ciencias naturales, Lectura critica) de estudiantes, basándose en los datos sociales, personales y del colegio sobre los estudiantes recolectados de la página de datos públicos del gobierno nacional de Colombia ://www.datos.gov.co/Educaci-n/Resultados-nicos-Saber-11/kgxf-xxbe. Cada red neuronal se configuró con un solo output numérico correspondiente a una de las competencias académicas, permitiendo así estimar de manera individualizada y precisa el desempeño en cada área

**2. Análisis exploratorio.**

En el análisis exploratorio se buscó cuantificar y dar datos de interés para poder tomar y entender el contexto en el cual se hace el proyecto. Partimos de que los datos son tomados para la presentación de las pruebas saber 11 en el año 2019 para primer y segundo semestre de este año. Los datos que esta base contempla son 131.461 después de haber eliminado filas vacías y y también se dejan un total de 30 variables explicativas. La ciudad elegida en el estudio es Bogotá, esto a razón de la gran variedad de población que sabíamos se podía abarcar con este proyecto. Con el solo análisis de los colegios se vislumbran cifras sumamente llamativas, como el hecho de que solo 0,16% de los colegios en la muestra elegida eran Bilingües, un aspecto fundamental para la formación de ciudadanos globales. En la parte socioeconómica, se evidencia que los estratos 1, 2 y 3 son el grueso de la muestra sumando entre todos cerca de un 89% de la población. Para enfocarnos en lo que sería nuestra variable de respuesta que es a evaluación por competencia como análisis inicial se mira el promedio por competencia global, siendo lectura crítica la materia mejor puntuada con 56.3 y sociales y ciudadanas la peor calificada con 51.1. Estos análisis se profundizan de mejor manera mezclando varios factores y visualizando su comportamiento con gráficas que permitieran mostrar su relación.

**3. ETL de los Datos.**

Primero, la base de datos se filtró utilizando la página https://www.datos.gov.co/Educaci-n/Resultados-nicos-Saber-11/kgxf-xxbe, limitando los datos a colegios ubicados en Bogotá y pertenecientes al periodo 20191 a 20192. Una vez obtenidos los datos, se configuró una instancia EC2 con Ubuntu en AWS y se cargaron los datos ahí (Anexos: Foto 1). Posteriormente, se transfirieron a una base de datos RDS en AWS. Para esto se creó una base de datos de nombre “icfes” y se le añadió la tabla llamada “resultados”. A esta tabla se le insertaron los datos ya filtrados como se menciona anteriormente. (Anexos: Foto 2). Finalmente, para su uso en el análisis, se extrajeron de la base de datos y se convirtieron en un DataFrame en un archivo .py. Este proceso aseguró que los datos fueran accesibles y utilizables de manera eficiente para el proyecto.

**4. Modelo Predictivo de Clasificación con Red Neuronal.**

La selección adecuada de los hiperparámetros es esencial para el rendimiento y la capacidad de generalización de un modelo de red neuronal. Los hiperparámetros, como la función de activación en cada capa, la función de pérdida y el tipo de optimizador, se decidieron basándonos en bibliografía. Por otro lado, la profundidad de la red y el ancho de la red fueron probados de manera iterativa ya que su efectividad depende de la base de datos específica que se esté utilizando. Se eligieron aquellos que minimizaron la perdida (MSE) y la métrica MAE. A continuación, se describe el proceso para seleccionar los hiperparámetros:

**Hiperparametros escogidos por bibliografía:**

**Función de activación:** Se seleccionó el método ReLU como función de activación para todas las capas menos la de salida debido a su probada eficacia y amplia aceptación en el entrenamiento de redes neuronales profundas. Aunque investigaciones recientes, como la realizada por Ramachandran, Zoph y Le (2017), han propuesto nuevas funciones de activación, como Swish, que muestran mejoras marginales de menos de 1% en la precisión de clasificación en comparación con ReLU, este último, según estos mismos autores, sigue siendo la opción preferida y más utilizada en la práctica para modelos de clasificación binaria. Para el proyecto, se utilizó la función de activación softplus en la capa de salida debido a su capacidad para manejar salidas que deben ser siempre positivas. Esta función de respuesta es diferenciable y se asemejan estrechamente a la función identidad para valores positivos del predictor de regresión, implicando un modelo cuasi-aditivo (Wiemann,2023). Contrastamos nuestro enfoque con la comúnmente utilizada función de respuesta exponencial. En este proyecto, como el output no podía ser menor a 0, la función softplus fue la elección adecuada (Wiemann et al ,2023).

**Función de perdida y metrica:** Se utilizó el MeanSquaredError (MSE) como función de pérdida porque penaliza fuertemente los errores grandes al calcular la media de los cuadrados de las diferencias entre los valores predichos y los reales. Esto incentiva al modelo a minimizar los errores grandes durante el entrenamiento. Por otro lado, se empleó el mean\_absolute\_error (MAE) como métrica porque proporciona una interpretación más intuitiva de la magnitud promedio de los errores, calculando la media de las diferencias absolutas. MAE pondera todos los errores por igual y es menos sensible a los valores atípicos, ofreciendo una visión complementaria del rendimiento del modelo (Brownlee,2020).

**Optimizador:** El optimizador Adam se utilizó porque es conocido por ser eficiente y efectivo en el entrenamiento de redes neuronales. Adam combina las ventajas del método de gradiente estocástico y de los métodos de promedio de momentos adaptativos. Proporciona una tasa de aprendizaje adaptativa para cada parámetro, ajustándola según las estimaciones de primer y segundo orden de los momentos del gradiente. Esto significa que Adam puede converger rápidamente y manejar de manera efectiva problemas con grandes cantidades de datos y parámetros (Doshi, 2019).

**Hiperparametros escogidos iterativamente ejecutando el modelo:**

En primer lugar, los datos se dividieron 30% los datos de testeo, 21% los datos de validación y el restante para entrenamiento. El conjunto de entrenamiento, siendo el más grande, se utilizó para ajustar los parámetros del modelo de redes neuronales. El conjunto de validación se empleó para ajustar los hiperparámetros del modelo y evaluar su rendimiento durante el entrenamiento, asegurando que el modelo no se sobreajustara. Por último, el conjunto de prueba se reservó exclusivamente para evaluar el rendimiento final del modelo después de que se ha entrenado y ajustado

**Profundidad, ancho de la red y ajuste del modelo:**

Después de una evaluación iterativa, se determinó que la mejor configuración para el modelo de predicción de puntajes en el ICFES de inglés fue utilizando una única capa oculta con 500 neuronas. Como se observa en la Grafica 1 El modelo entrenado con esta configuración logró un valor de pérdida (MSE) de 82.65 y un error absoluto medio (MAE) de 6.99 en los datos de prueba. Este desempeño se logró tras 100 épocas de entrenamiento. Durante la evaluación del modelo con los datos de validación, se observó un caso significativo de sobreajuste. Como se observa en la Grafica 2 y Grafica 3 los resultados en validación rondaban los 80 de MSE y menor a 20 para los datos de entrenamiento, al mismo tiempo, el MAE llego a 2 en la época 100 para los datos de entrenamiento y estuvo fluctuando en 6 para los de validación. Esto motivó a la reducción de la cantidad de datos de épocas de entrenamiento y la implementación de regularización L2 con un parámetro de 0.01 en las capas densas del modelo. Sin embargo, esto no presento un gran cambio en los resultados del modelo. Por lo tanto, se redujo el número de épocas de entrenamiento de 100 a 20 y se implementó Dropout con una tasa de retención del 70% (parámetro 0.3), en lugar de la regularización L2. Como se puede observar en la Gráfica 4 y en la Grafica 5, se redujo la brecha de los resultados de MAE y del MSE en los 20 periodos respecto de los datos de validación contra los de entrenamiento. Los datos de validación se mantuvieron constantes en un MSE de 75, mientras que los datos de entrenamiento llegaron a un MSE de 50. Por otro lado, el MAE se mantuvo entre 6 y 8 para los datos de validación y menos de 6 para los de entrenamiento.

Para los demás modelos, se realizo el mismo procedimiento y se obtuvieron resultados muy similares a el modelo de predicción de puntajes en el ICFES de inglés. Cada configuración incluye una sola capa oculta con un número específico de neuronas, y se evaluaron utilizando las métricas de error absoluto medio (MAE) y error cuadrático medio (MSE), así como el número de épocas necesarias para entrenar el modelo.

**Resultados:**

Como se observa en la Tabla 1, para Matemáticas, la mejor combinación fue una capa con 500 neuronas, con un MAE de 6.9733 y un MSE de 83.980 después de 24 épocas. En Ciencias Sociales y Ciencias Naturales, la configuración óptima fue una capa con 200 neuronas, con MAE de 6.506 y 6.465, y MSE de 81.546 y 80.400 respectivamente, utilizando 25 y 30 épocas. Finalmente, para Lectura Crítica, la mejor combinación fue una capa con 500 neuronas, con un MAE de 6.121 y un MSE de 78.309 después de 26 épocas.

**Despliegue:**

Para el despliegue del Dash, se cargó a una instancia en EC2 ubuntu así como los modelos serializados que lo respaldan. Una vez ahí, se instalaron las dependencias de librerías necesarias y se desplego desde la nube a una página pública. Se desplegaron dos tableros interactivos. El primero “Version2proyecto3” es el tablero en el que se muestran análisis descriptivos de las variables de interés. El segundo “Prueba\_final” es la herramienta para predecir el resultado en las competencias del ICFES. Estos se desplegaron en el EC2 de Ubuntu (Anexos: Foto 4) La dirección de la página donde se desplego es la siguiente: <http://34.206.126.100:8060/>. Esta Ip es **elástica**.

**Conclusiones:**

El desarrollo de una plataforma de analítica de datos ha permitido a rectores y administradores escolares en Colombia entender, visualizar y predecir el desempeño de los estudiantes en las pruebas Saber 11. A través del análisis exploratorio de los datos sociales, personales y escolares recolectados de fuentes públicas, se identificaron los principales factores que impactan el rendimiento estudiantil. Este sistema se complementa con una herramienta interactiva que despliega visualizaciones por colegio, evaluando resultados en competencias del ICFES y destacando los factores más influyentes, proporcionando así una herramienta integral para mejorar la gestión educativa y el apoyo a los estudiantes. Además, se realizaron cinco redes neuronales para predecir los resultados por competencias académicas de estudiantes en las pruebas Saber 11 permitiendo identificar tempranamente a estudiantes en riesgo de bajo rendimiento académico.

Grafica 1: Mejor combinación (Verde relleno) de profundidad y ancho de red según el error absoluto medio (MAE)

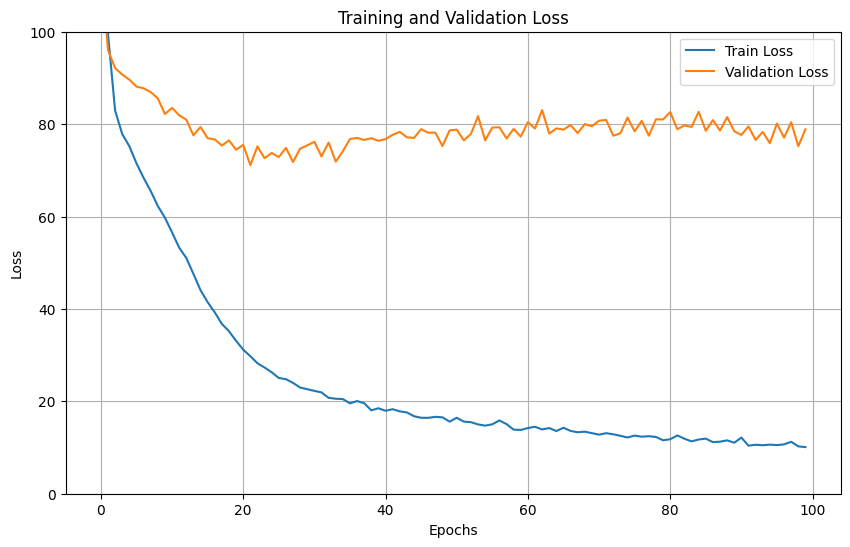
Gráfico, Gráfico de barras

Descripción generada automáticamente

Fuente: Elaboración propia

Grafica 2: Perdida (MSE) a lo largo de 100 épocas para entrenamiento Grafica 3: MAE de entrenamiento y validación en 100 épocas

y validación

Gráfico, Gráfico de líneas

Descripción generada automáticamente

Fuente: Elaboración propia Fuente: Elaboración propia

Tabla 1: Mejor combinación por MAE Y MSE, y numero de epocas de entrenamiento

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Competencia** | **Mejor combinación** | **MAE** | **MSE** | **Epocas** |
| **Matemáticas** | **1 capa, 500 neuronas** | **6.9985** | **80.4204** | **24** |
| **Ciencias Sociales** | **1 capa, 200 neuronas** | **7.6546** | **94.687** | **25** |
| **Ciencias Naturales** | **1 capa, 200 neuronas** | **6.4281** | **66.8379** | **20** |
| **Lectura Critica** | **1 capa, 500 neuronas** | **6.5301** | **69.1332** | **20** |

Fuente: Elaboración propia.

Grafica 4: Perdida (MSE) a lo largo de 20 épocas para entrenamiento Grafica 5: MAE de entrenamiento y validación en 20 épocas

Y validación

Gráfico, Gráfico de líneas

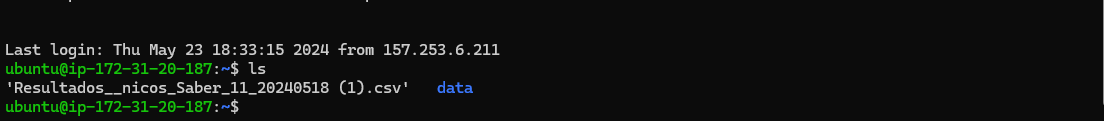
Descripción generada automáticamenteGráfico, Gráfico de líneas

Descripción generada automáticamente

Fuente: Elaboración propia Fuente: Elaboración propia

**Anexos:**

**Foto 1: Datos subidos a la instancia EC2**

****

**Foto 2: Datos subidos a una base RDS, base de datos “icfes” y tabla llamada “resultados”**

**Texto

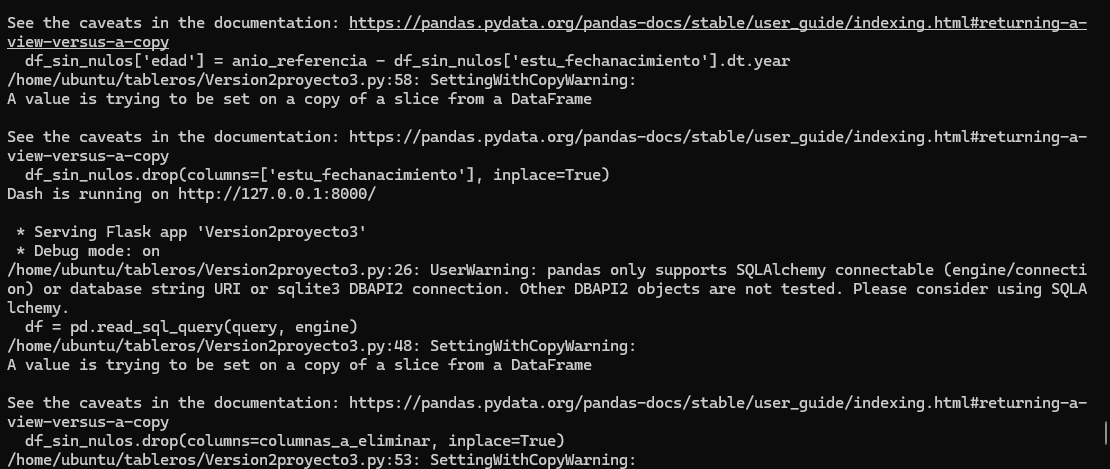
Descripción generada automáticamente**

**Foto 3: Llamado base de datos traídos con SQL a el archivo local .py de limpieza de datos**

Texto

Descripción generada automáticamente

**Foto 4: Despliegue tableros interactivos en EC2 ubuntu**

****

**Repositorio GitHub: https://github.com/Juanxtron/Proyecto\_final\_github.git**

**Bibliografía:**

* Ramachandran, P., Zoph, B., & Le, Q. V. (2017). *Searching for Activation Functions*. Disponible en: <https://arxiv.org/abs/1710.05941>
* Wiemann, P.F.V., Kneib, T. & Hambuckers, J (2023). *Using the softplus function to construct alternative link functions in generalized linear models and beyond. Stat Papers* .https://doi.org/10.1007/s00362-023-01509-x. Disponible en: https://link.springer.com/article/10.1007/s00362-023-01509-x
* Brownlee, J. (2020). *How to Choose Loss Functions When Training Deep Learning Neural Networks*. Disponible en: <https://machinelearningmastery.com/how-to-choose-loss-functions-when-training-deep-learning-neural-networks/>
* Doshi, S. (2019). *Various Optimization Algorithms For Training Neural Network*. Disponible en: https://towardsdatascience.com/optimizers-for-training-neural-network-59450d71caf6
* Wei, J. L. (2019). *How to build your first Neural Network to predict house prices with Keras*. Disponible en: https://www.freecodecamp.org/news/how-to-build-your-first-neural-network-to-predict-house-prices-with-keras-f8db83049159/