Realice lo siguiente en lenguaje de programación Python con Scikit-learn/Keras y plotly express:
Desarrolla un clasificador de texto con redes neuronales para identificar texto de una de veinte categorías. Para esta tarea se usará el conjunto de datos “The 20 newsgroup”
Pasos:

1. **Preprocesar el texto:** Realizar la tokenización, la eliminación de stopwords y otros pasos necesarios para preparar el texto para el análisis. Este paso es fundamental para asegurar que los datos estén en el formato adecuado.
2. **Visualizar la distribución de datos:** Utilizar histogramas y diagramas de caja (boxplots) para examinar la distribución de las características y etiquetas del dataset. Esta visualización ayuda a identificar la dispersión y la presencia de valores atípicos.
3. **Implementar una red neuronal:** Construir una red neuronal con capas densas utilizando Keras, estableciendo la arquitectura básica del modelo para la tarea de clasificación.
4. **Entrenar y ajustar el modelo:** Entrenar el modelo utilizando el dataset y ajustar múltiples hiperparametros para optimizar su rendimiento. Este proceso incluye la selección del optimizador, la tasa de aprendizaje, y otros parámetros relevantes.
5. **Visualizar las curvas de aprendizaje:** Mostrar cómo cambia el rendimiento del modelo durante el entrenamiento y la validación a lo largo de las épocas. Esta visualización es útil para identificar problemas de sobreajuste y subajuste.
6. **Evaluar el rendimiento utilizando medidas de desempeño:** Calcular y analizar métricas de desempeño como precisión, recuerdo, F1 para evaluar la eficacia del modelo en la clasificación.
7. **Mostrar la matriz de confusión:** Presentar una matriz de confusión para evaluar cómo se clasifican correcta e incorrectamente las instancias del conjunto de datos. Esto proporciona una visión clara de la precisión del modelo.
8. **Experimentar con diferentes arquitecturas de red:** Probar diversas configuraciones de arquitectura de la red neuronal y visualizar los resultados para comparar el rendimiento de las diferentes configuraciones.
9. **Realizar pruebas con k-fold cross-validation:** Implementar validación cruzada con k-folds para asegurar el rendimiento general del modelo y evitar problemas de sobreajuste.
10. **Mostrar la curva ROC y AUC:** Presentar la curva ROC (Receiver Operating Characteristics) y calcular el valor del AUC (Area Under the Curve) para evaluar el rendimiento del modelo en problemas de clasificación binaria.
11. **Escribir hallazgos:** Documentar los hallazgos y conclusiones del análisis y los experimentos realizados, proporcionando una visión comprensiva del rendimiento del modelo y las recomendaciones.

Elementos para tomar en cuenta:

1. Deberá documentar todo el proceso de desarrollo en un notebook de Google colab o Jupyter.
2. Deberá justificar el porqué de cada decisión tomada en el proceso de entrenamiento y prueba.
3. Deberá comentar en la medida de lo posible la mayor parte del código para ver si entienden los conceptos.