Proyecto Final del Curso

Integrantes:
- Sebastian Antonio Saldaña Rodriguez
- Valery Siccha
- Diana Llamoca
- Leonardo Mejía

Estructura del Cuaderno de nuestro Proyecto:

- Introducción
  - Presentaremos el problema general que abordará nuestro proyecto y cómo se relaciona con el campo del aprendizaje automático.
  - Explicamos el objetivo principal del estudio.

- Diseño del experimento
  - Describiremos el conjunto de datos que hemos elegido, incluyendo su origen y características generales.
  - Especificaremos el número y tipo de características presentes en el conjunto de datos (binarias, discretas, continuas, etc.).
  - Indicaremos el número de muestras en los conjuntos de entrenamiento y prueba. En caso de ser relevante, mencionaremos el número de muestras por clase.
  - Explicaremos la metodología que seguiremos en nuestro proyecto.
  - Si existen datos faltantes, detallaremos la estrategia que vamos a emplear para manejarlos.
  - Describiremos el proceso de selección y extracción de características.
  - Vamos a justificar la medida de calidad que utilizaremos para evaluar el rendimiento de los modelos.
  - Indicaremos los algoritmos que utilizaremos y la estrategia que seguiremos para ajustarlos.
  - Si es necesario vamos a ajustar hiperparámetros, explicaremos la estrategia de validación que utilizaremos.

- Experimentación y resultados
  - Evaluaremos el rendimiento de los modelos que hemos probado.
  - Realizaremos una comparación entre una línea base (si existe) y los resultados obtenidos en nuestro estudio.
  - Presentaremos los resultados de manera clara y concisa, utilizando métricas relevantes y visualizaciones si es necesario.

- Discusión
  - Interpretaremos los resultados obtenidos y proporcionaremos una explicación de su significado.

- Discutiremos las fortalezas y debilidades de nuestro sistema.
- Vamos a sugerir posibles mejoras para nuestro sistema y cómo podrían implementarse.

- Conclusiones y trabajos futuros
  - Resumimos las conclusiones principales de nuestro proyecto.
  - Proporcionaremos ideas sobre posibles trabajos futuros que podrían ampliar o mejorar el enfoque y los resultados obtenidos.

El conjunto de datos que hemos elegido es el conjunto de datos "Handwritten digit classification" (clasificación de dígitos escritos a mano), ya que este conjunto de datos se encuentra disponible en la biblioteca sklearn, lo que facilita su acceso y carga en nuestro entorno de desarrollo.

1.Introducción:

- Nuestro proyecto se centra en el problema de clasificación de dígitos escritos a mano utilizando técnicas de aprendizaje automático. Este problema es ampliamente estudiado en el campo del reconocimiento de patrones y es de gran relevancia en aplicaciones como el procesamiento de imágenes, reconocimiento óptico de caracteres y sistemas de identificación biométrica.

- El objetivo principal de este estudio es desarrollar un modelo de aprendizaje automático que pueda clasificar de manera precisa y eficiente los dígitos escritos a mano. Para lograrlo, emplearemos algoritmos de ensamble, en particular, utilizaremos la técnica de boosting (AdaBoost), el algoritmo XGBoost, así como Bosques Aleatorios y Regresión Logística.

- La importancia de este proyecto radica en la capacidad de los algoritmos de aprendizaje automático para reconocer y clasificar patrones complejos en datos no estructurados. La clasificación precisa de dígitos escritos a mano tiene aplicaciones prácticas en diversos campos, como el procesamiento de formularios, la clasificación de documentos y la detección de fraudes en firmas.

- Al aplicar los algoritmos de ensamble y evaluar su rendimiento en el conjunto de datos de dígitos escritos a mano, podremos demostrar la eficacia de estas técnicas y su utilidad en problemas de clasificación de patrones.

- A través de nuestro proyecto, buscaremos no sólo desarrollar un modelo preciso, sino también comprender y analizar la importancia de las características utilizadas, identificar los desafíos específicos asociados con este problema y explorar posibles

mejoras o extensiones para futuros trabajos en el campo del reconocimiento de dígitos escritos a mano.

- En conclusión, el objetivo principal de este estudio es aplicar técnicas de aprendizaje automático, específicamente algoritmos de ensamble, para clasificar de manera precisa los dígitos escritos a mano. Al lograr esto, esperamos contribuir al campo del reconocimiento de patrones y demostrar la utilidad de estos algoritmos en problemas de clasificación de imágenes y datos no estructurados.

2.Diseño del experimento:

Conjunto de datos:

- El conjunto de datos que hemos elegido para este proyecto es el conjunto de dígitos escritos a mano, que está disponible en la biblioteca sklearn. Este conjunto de datos es ampliamente utilizado en el campo del reconocimiento de patrones y contiene imágenes de dígitos escritos a mano en escala de grises, representadas como matrices numéricas.

Características del conjunto de datos:

- Cada imagen en el conjunto de datos es una matriz de 8x8 píxeles, lo que resulta en un total de 64 características. Cada píxel está representado por un valor entero en el rango de 0 a 16, que indica la intensidad de gris. Por lo tanto, las características son continuas y representan la información de los píxeles de la imagen.

Muestras en los conjuntos de entrenamiento y prueba:

- El conjunto de datos de dígitos escritos a mano se divide en conjuntos de entrenamiento y prueba. El conjunto de entrenamiento contiene 1437 muestras, mientras que el conjunto de prueba tiene 360 muestras. Utilizaremos estas muestras para entrenar nuestros modelos y evaluar su rendimiento respectivamente.
- No hay información sobre un desequilibrio significativo en la distribución de las muestras por clase, lo que implica que cada clase tiene una cantidad razonable de muestras para el entrenamiento y la evaluación de los modelos.

Metodología:

- En este proyecto, seguiremos una metodología estándar para el aprendizaje automático. Comenzaremos con la exploración y preprocesamiento de los datos, donde analizaremos su distribución, realizaremos visualizaciones y aplicaremos técnicas de normalización si es necesario. A continuación, seleccionaremos y extraeremos características relevantes de las imágenes utilizando técnicas como el análisis de componentes principales (PCA) o el análisis de discriminante lineal (LDA) para reducir la dimensionalidad de los datos y mejorar el rendimiento de los modelos.

- Una vez realizado el preprocesamiento, implementaremos los algoritmos de ensamble mencionados: Bosques Aleatorios (Random Forests), AdaBoost y

XGBoost. Estos algoritmos se caracterizan por combinar múltiples modelos más simples para mejorar la precisión y generalización del modelo final.

- Para cada algoritmo, entrenaremos y ajustaremos los modelos utilizando el conjunto de datos preparado. Durante esta etapa, ajustaremos los hiperparámetros correspondientes para obtener el mejor rendimiento posible de cada modelo.

- Una vez entrenados los modelos, evaluaremos su rendimiento utilizando métricas relevantes, como la precisión, y compararemos los resultados obtenidos con los diferentes algoritmos de ensamble. También realizaremos una comparación con una línea base, en este caso utilizando la Regresión Logística, para determinar la mejora o el desempeño relativo de los modelos de ensamble.

Manejo de datos faltantes:

- En este conjunto de datos específico de dígitos escritos a mano, no se espera que haya datos faltantes, ya que todas las imágenes están completas y no hay información adicional asociada a cada muestra que pueda faltar.

Selección y extracción de características:

- Se utilizará una técnica de selección de características, como PCA o LDA, para reducir la dimensionalidad del conjunto de datos y seleccionar las características más relevantes para el problema de clasificación de dígitos escritos a mano.

Medida de calidad:

- Utilizaremos la precisión (accuracy) como medida de calidad para evaluar el rendimiento de nuestros modelos. La precisión nos permite medir qué tan bien clasifican los modelos los dígitos escritos a mano correctamente en relación con el total de muestras.

Algoritmos y ajuste:

- En este proyecto, utilizaremos los algoritmos de ensamble Boosting (AdaBoost), XGBoost y Bosques Aleatorios. Estos algoritmos son adecuados para problemas de clasificación y nos permiten combinar múltiples clasificadores débiles para construir clasificadores más robustos.

- Ajustaremos los hiperparámetros de cada algoritmo utilizando validación cruzada para encontrar la configuración óptima que maximice el rendimiento en el conjunto de prueba. Este proceso nos ayudará a encontrar los valores adecuados de los hiperparámetros, como el número de estimadores, la tasa de aprendizaje y la profundidad del árbol, para cada algoritmo.

Estrategia de validación a emplear para el ajuste de hiperparámetros si fuese necesario:

- Para ajustar los hiperparámetros de los algoritmos de ensamble (AdaBoost, XGBoost y Bosques Aleatorios), utilizaremos la técnica de validación cruzada. En particular, emplearemos la validación cruzada estratificada con k-folds.

- En este enfoque, dividiremos el conjunto de entrenamiento en k pliegues (folds) de tamaño aproximadamente igual. Luego, realizaremos k iteraciones, donde en cada iteración utilizaremos k-1 pliegues para entrenar el modelo y evaluaremos su rendimiento en el pliegue restante. Esto nos proporcionará una estimación robusta del rendimiento del modelo en diferentes subconjuntos de datos.

- Utilizaremos la métrica de precisión (accuracy) como medida de calidad para evaluar el rendimiento de los modelos en cada iteración de la validación cruzada. La precisión nos indicará la proporción de muestras clasificadas correctamente en relación con el total de muestras.

3.Experimentación y resultados:

Link de Google Colab(Lo haremos en Jupyter Notebook):

- https://colab.research.google.com/drive/1HUhYSbN9dp0det6TosckfHZIaLBu6JQT?usp=sharing

Evaluación del rendimiento de los modelos:

- Modelo de Bosques Aleatorios: Precisión = 0.9722
- Modelo de AdaBoost: Precisión = 0.2194
- Modelo de XGBoost: Precisión = 0.9694

Comparación con la línea base (Regresión Logística):

- Diferencia de precisión con Bosques Aleatorios: 0.0
- Diferencia de precisión con AdaBoost: -0.7528
- Diferencia de precisión con XGBoost: -0.0028

Presentación de resultados:

A. Precisión de los modelos:

- ☐ El modelo de Bosques Aleatorios y el modelo de Regresión Logística tienen una precisión similar, ambos con una precisión de aproximadamente 0.9722.
- ☐ El modelo de AdaBoost muestra una precisión considerablemente inferior, con un valor de aproximadamente 0.2194.
- ☐ El modelo de XGBoost presenta una precisión ligeramente inferior al modelo de Regresión Logística, con un valor de aproximadamente 0.9694.

B. Comparación con la línea base:

- ☐ En general, observamos que los modelos de Bosques Aleatorios y Regresión Logística obtuvieron una precisión similar, ambos con un valor de 0.9722. Por otro lado, el modelo de AdaBoost mostró un rendimiento significativamente

inferior, con una precisión de solo 0.2194. El modelo de XGBoost también demostró un buen desempeño con una precisión de 0.9694.

- ☐ Al comparar los modelos con la línea base de Regresión Logística, observamos que los Bosques Aleatorios no presentaron una diferencia significativa en términos de precisión, mientras que tanto AdaBoost como XGBoost mostraron una disminución en la precisión en comparación con la línea base. La diferencia de precisión con Bosques Aleatorios fue de 0.0, con AdaBoost fue de -0.7528 y con XGBoost fue de -0.0028.

  En conclusión:

- ☐ En comparación con la línea base (modelo de Regresión Logística), el modelo de Bosques Aleatorios muestra una precisión igual, lo que indica que ambos modelos obtuvieron resultados similares en este conjunto de datos.

- ☐ El modelo de AdaBoost tiene una diferencia de precisión significativa con respecto a la línea base, mostrando un rendimiento inferior.

- ☐ El modelo de XGBoost tiene una diferencia de precisión mínima en comparación con la línea base, mostrando un rendimiento similar.

4.Discusión:

Interpretación de los resultados:

- Los resultados obtenidos indican que tanto los Bosques Aleatorios como la Regresión Logística lograron una alta precisión en la clasificación de dígitos escritos a mano, con una precisión del 97.22% en ambos casos. Esto demuestra que ambos modelos son efectivos para este problema en particular.

- Por otro lado, los modelos de AdaBoost y XGBoost mostraron una precisión significativamente más baja, con valores de 21.94% y 96.94%, respectivamente. Esto sugiere que estos modelos no son tan adecuados para la clasificación de dígitos escritos a mano en comparación con los Bosques Aleatorios y la Regresión Logística.

Fortalezas y debilidades del sistema:

Fortalezas:

- Los Bosques Aleatorios y la Regresión Logística lograron una alta precisión, lo que indica que son capaces de clasificar de manera precisa los dígitos escritos a mano.

- La Regresión Logística se ejecuta rápidamente y es fácil de interpretar, lo que la hace adecuada para aplicaciones en tiempo real.
- Los Bosques Aleatorios son robustos frente a datos ruidosos y tienen la capacidad de manejar características irrelevantes.

Debilidades:

- El modelo de AdaBoost mostró una precisión significativamente más baja, lo que indica que puede no ser adecuado para este problema en particular.
- El modelo de XGBoost mostró una precisión ligeramente más baja que los Bosques Aleatorios y la Regresión Logística, lo que sugiere que puede haber margen de mejora.

Sugerencias de mejora:

- Para el modelo de AdaBoost, se pueden explorar diferentes hiperparámetros y técnicas de ajuste para mejorar su rendimiento. Además, es posible que se deba considerar una selección más exhaustiva de características relevantes para mejorar la precisión.

- En cuanto al modelo de XGBoost, se puede ajustar el valor de los hiperparámetros y realizar una validación cruzada para encontrar una configuración óptima que mejore su precisión.

- Además, se podría considerar la exploración de otros modelos de ensamble, como Gradient Boosting o Stacking, para evaluar su rendimiento en la clasificación de dígitos escritos a mano, asimismo sería beneficioso explorar técnicas avanzadas de preprocesamiento de datos, como el aumento de datos y la normalización adicional, para mejorar aún más el rendimiento de los modelos.

5.Conclusiones y trabajos futuros:

Conclusiones:

- En este estudio, se ha desarrollado y evaluado un modelo de aprendizaje automático para la clasificación de dígitos escritos a mano utilizando técnicas de ensamble como Bosques Aleatorios, AdaBoost, XGBoost y Regresión Logística.

- Se demostró que tanto Bosques Aleatorios como Regresión Logística lograron una alta precisión en la clasificación de dígitos escritos a mano, con una precisión del 97.22% en ambos casos.

- Por otro lado, los modelos de AdaBoost y XGBoost mostraron una precisión más baja, lo que indica que pueden requerir ajustes adicionales para mejorar su rendimiento en este problema específico.

- Se identificaron fortalezas y debilidades en el sistema de clasificación propuesto. Los Bosques Aleatorios fueron robustos frente a datos ruidosos y la Regresión Logística se destacó por su velocidad de ejecución y facilidad de interpretación.

Trabajos Futuros:

- Como trabajos futuros, se sugiere explorar otras técnicas de ensamble, como Gradient Boosting o Stacking, para evaluar su rendimiento en la clasificación de dígitos escritos a mano y compararlos con los modelos utilizados en este estudio.

- También se recomienda investigar técnicas avanzadas de preprocesamiento de datos, como el aumento de datos y la normalización adicional, para mejorar aún más el rendimiento de los modelos.

- Finalmente, sería interesante analizar el impacto de diferentes conjuntos de características en el rendimiento de los modelos y explorar técnicas de extracción de características más avanzadas.Asimismo, se puede considerar la aplicación de técnicas de optimización de hiperparámetros para ajustar los modelos y obtener un rendimiento aún mejor.

Referencias:

Scikit-learn: https://scikit-learn.org/

XGBoost: https://xgboost.readthedocs.io/

Matplotlib: https://matplotlib.org/

Conjunto de datos MNIST: http://yann.lecun.com/exdb/mnist/

Conjunto de datos de dígitos escritos a mano de Scikit-learn: https://scikit-learn.org/stable/modules/generated/sklearn.datasets.load_digits.html

Ensemble Methods en Scikit-learn:

https://scikit-learn.org/stable/modules/ensemble.html

Logistic Regression en Scikit-learn:

https://scikit-learn.org/stable/modules/generated/sklearn.linear_model.LogisticRegression.html

Random Forests en Scikit-learn:

https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestClassifier.html

AdaBoost en Scikit-learn:

https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.AdaBoostClassifier.html
XGBoost: Documentación oficial de XGBoost (https://xgboost.readthedocs.io/)

XGBoost: Documentación oficial de XGBoost

https://xgboost.readthedocs.io/