Introducción a la Inteligencia Artificial – LCC – 2023
Trabajo Practico Aprendizaje Automatizado en Python

Ejercicio 1

Descripción del Dataset

El dataset "German Credit Data" contiene información sobre diversas características financieras y personales de solicitantes de crédito, como historial de crédito, ingresos, estado civil, número de dependientes, entre otros. Cada instancia del dataset está etiquetada como "bueno" o "malo" según el riesgo crediticio asociado. El objetivo del conjunto de datos es poder predecir el riesgo crediticio de nuevos solicitantes basándose en las características proporcionadas.

El mismo se encuentra publicado en el UCI Machine Learning Repository bajo el siguiente link. Allí se puede encontrar una descripción detallada de cada uno de sus atributos y posibles valores.

Para simplificar el trabajo, el dataset se encuentra disponible, además, en el archivo credit_data.csv. Las primeras 20 columnas contienen los atributos de cada ejemplo y la última contiene la clase a la que pertenece cada fila.

1) Preparación de los datos

a) Implementar una función que lea el archivo y devuelva un objeto con los atributos:

- data: Tiene que contener los atributos de los datos
- target: Tiene que contener la clase a la que pertenece cada dato
- feature_names: Tiene que ser una lista con el nombre de los 20 atributos
- target_names: Tiene que ser una lista con el nombre de las clases

b) Para poder entrenar cualquier modelo de aprendizaje automatizado es necesario transformar los valores de los atributos en tipos numéricos. Por lo tanto, se requiere implementar una función o celda de código que tome la variable devuelta por la función del punto anterior y transforme los atributos que no son de tipo numérico en números.

Pista: Una forma de transformar los tipos de datos no numéricos en numéricos es utilizando la clase OrdinalEncoder de SciKit-Learn.

c) Dividir el conjunto de datos en entrenamiento y evaluación utilizando la función train_test_split de SciKit-Learn. El conjunto de evaluación debe tener el 10% de las muestras totales del dataset. Se debe configurar el parámetro random_state en 0.

2) Entrenamiento de modelo

a) Entrenar un árbol de decisión sobre el conjunto de entrenamiento, con los parámetros por defecto que trae el árbol. Utilizar la métrica accuracy para medir el modelo entrenado sobre

el conjunto de entrenamiento y evaluación. ¿Qué efectos observa sobre ese árbol? ¿Cómo los explica?

b) Utilizar la clase GridSearchCV para encontrar la mejor combinación de parámetros de parada para el modelo empleando la métrica accuracy.

c) Sobre el modelo entrenado en el punto (a), realizar poda por niveles y graficar cómo varía el valor de accuracy sobre el conjunto de entrenamiento y evaluación para cada nivel de poda. ¿Cuál considera que es el nivel de poda óptimo y por qué?

3) Evaluación de modelo

a) Teniendo en cuenta el contexto del problema que se pretende resolver: ¿Cuál diría que es la mejor métrica para medir el funcionamiento de los modelos entrenados y por qué?

b) Utilizar esta métrica para medir la performance del modelo entrenado en el punto (2b) y (2c). Repetir los experimentos realizados en dichos puntos y evaluar si cambian los parámetros de configuración o el nivel de poda para el mejor árbol.

Ejercicio 2

Descripción del dataset

El dataset Fashion MNIST es un conjunto de 70,000 imágenes en escala de grises, divididas en 60,000 imágenes de entrenamiento y 10,000 imágenes de prueba. Cada imagen tiene una resolución de 28x28 píxeles y está etiquetada con una de las 10 clases de prendas posibles:

- 0 - Remera
- 1 - Pantalón largo
- 2 - Pullover
- 3 - Vestido
- 4 - Abrigo
- 5 - Sandalia
- 6 - Camisa
- 7 - Zapatilla
- 8 - Mochila
- 9 - Botin

Este conjunto de datos está disponible en varios lugares en internet. En este caso, se presenta como dos archivos: fashion-mnist_train.csv y fashion-mnist_test.csv. Estos contienen los datos y etiquetas de entrenamiento y evaluación respectivamente. Cada fila cuenta con 784 columnas de atributos con los valores de cada uno de los píxeles de las imágenes. Las etiquetas para cada imagen se encuentran en la última columna de cada archivo.

Para descargar el dataset dentro de colab es necesario copiar las siguientes líneas dentro de una celda y ejecutarlas:

```
!gdown 1KqgzDAGLvJxyEn7ewy11J1jXcyuoAo_w

!gdown 1RFeimal-QVwIvFlIeHB4NzzSb6uFVgYX
```

1) Realizar un análisis del mismo indicando: cantidad de muestras, cantidad de clases, y cantidad de muestras por clase. Grafique una muestra de las imágenes para cada clase.

2) Entrenamiento de red neuronal

a) Crear redes neuronales con las siguientes configuraciones y entrenarlas utilizando el conjunto de datos de entrenamiento durante 40 épocas:

- Sin capa oculta
- Con dos capas ocultas: una de 100 neuronas y otra de 50 neuronas
- Con 6 capas ocultas: tres de 100 neuronas seguidas de otras tres de 50 neuronas

Utilizar un learning rate de 0.0001, batch_size de 16, optimizador tipo adam y funciones de activación tipo ReLU en todas las capas.

Realizar gráficas de la evolución del accuracy a lo largo del entrenamiento y comparar los valores finales de la misma sobre el conjunto de entrenamiento y validación. ¿Qué conclusiones saca para cada una de las situaciones?

b) Crear una red neuronal con una capa oculta de 50 neuronas y función de activación tipo ReLU. Entrenar distintos modelos utilizando los siguientes valores de learning rate: 10, 1, 0.1, 0.01, 0.001, 0.0001 y 0.00001. Graficar cómo varía el accuracy sobre el conjunto de entrenamiento y evaluación para cada entrenamiento. ¿Cómo explica el comportamiento de la red neuronal en este rango de valores de learning rate?

Informe

Realizar un informe donde se muestre la resolución de los ejercicios del enunciado y los distintos experimentos realizados en cada caso, junto con las conclusiones extraídas de cada punto. Deben incluir las gráficas y valores obtenidos al final de cada experimento.

La entrega de este informe debe estar acompañada de los archivos de Colab donde se hayan ejecutado todos los experimentos mencionados con sus resultados visibles.