

Implementado Machine Learning

Práctica 3 – Problemas de clasificación – Regresión Logística – Redes Neuronales – Evaluación de modelos

Material de lectura:

- Sebastian Raschka, Vahid Mirjalili. Python Machine Learning. Capítulos 2, 3 y 6.
- Andreas C. Müller, Sarah Guido. Introduction to Machine Learning with Python. Capítulo 2 y 5.
- Recursos online sobre algunos conceptos:
 - Optimización de hiperparámetros. AutoML. AutoSklearn: <https://towardsdatascience.com/auto-sklearn-scikit-learn-on-steroids-42abd4680e94>
 - Regresión lineal vs. Redes Neuronales: <https://www.kdnuggets.com/2021/08/3-reasons-linear-regression-instead-neural-networks.html>
 - Desbalance de clases: <https://machinelearningmastery.com/random-oversampling-and-undersampling-for-imbalanced-classification/>
- Recursos de visualización:
 - Regresión con webcam: <https://editor.p5js.org/AndreasRef/sketches/B1q7ds0wm>
 - Clasificación o Regresión simple en datasets 2D: <https://playground.tensorflow.org/>
 - <https://cs.stanford.edu/people/karpathy/convnetjs/demo/classify2d.html>
 - Red entrenada para clasificar imágenes de números: <http://scs.rverson.ca/~aharley/vis/fc/>

Para todos los ejercicios:

- En el caso de utilizar una Regresión Logística **grafique** sus parámetros con la función `AA_utils.visualizar_parametros`
- **Grafique e interprete** la matriz de confusión de la predicción sobre el set de test con la función `AA_utils.visualizar_confusion_matrix`
- **Elabore** un un reporte de métricas tanto para la predicción de train como para la de test con la función `AA_utils.print_classification_report`
- En los casos de clasificación binaria **grafique e interprete** la curva ROC con la función `AA_utils.visualizar_curva_ROC`
- En los caso de clasificación de dos variables **grafique** la función de transferencia y la frontera de decisión con las funciones `AA_utils.visualizar_funcion_transferencia_2D` y `AA_utils.visualizar_frontera_de_desicion_2D`
- En los casos de clasificación binaria de una variable **grafique** la curva logística con la función `AA_utils.visualizar_curva_logistica`

- **Clasificación binaria con 1 variable:** Analice el código de ejemplo en `ejemplo_clasificacion_binaria_una_variable_regresion_logistica.py`. El script entrena un modelo de regresión logística para clasificar tumores como malignos o benignos en base a su tamaño. El script también muestra la función sigmoidea entrenada (distribución de probabilidades). Analice cómo cambia la curva entrenada al cambiar el parámetro `class_weight`.
- **Clasificación con 2 variables:** Entrene modelos de Regresión Logística para clasificar los siguientes datasets:
 - `2D_moons.csv`
 - `2D_unbalanced.csv`

El código “`ejemplo_clasificacion_binaria_dos_variables_regresion_logistica.py`” posee un ejemplo de cómo realizar esta tarea.

- **Clasificación Binaria:** Compare modelos de Regresión Logística y Redes Neuronales para los siguientes datasets:
 - `Breast_cancer.csv`. El archivo posee diferentes mediciones sobre tumores con la información de si son malignos o benignos.
 - `Titanic.csv`. El archivo contiene información de los pasajeros que viajaban. El objetivo es crear un modelo capaz de predecir si un pasajero sobrevive o no al naufragio.
 - `Diabetes.csv`. El archivo posee 2000 registros con información de análisis clínicos de pacientes, indicando si poseen o no diabetes gestacional.

El código “`ejemplo_clasificacion_binaria_dos_variables_red_neuronal.py`” posee un ejemplo de cómo realizar esta tarea.

- **Clasificación Multiclase:** Compare modelos de Regresión Logística y Redes Neuronales para clasificar el dataset `Iris.csv`. El dataset contiene información de 3 especies de flores. El código “`ejemplo_clasificacion_multiclase_dos_variables_regresion_logistica.py`” posee un ejemplo de cómo realizar una clasificación multiclase.
- **Clasificación Multiclase de imágenes:** Analice el código en el archivo “`Regresion_Logistica_MNIST.py`”. El script entrena un modelo para clasificar imágenes de 10 dígitos escritos a mano. Analice la matriz de confusión y las métricas obtenidas para el conjunto de training y para el de testing. Mejore los resultados utilizando una Red Neuronal como clasificador.
- **Regresión con Redes Neuronales:**
 - Analice el código en el archivo “`Redes_neuro_regresion_Nafta.py`”. El script entrena una red neuronal para predecir el consumo de un vehículo en base a la velocidad a la que circula. Analice la curva entrenada y las métricas obtenidas. ¿Qué ocurre si no normaliza las variables?