Semestral AD2021

Aprendizaje automático
Práctica 3 “Clasificadores k-NN y regresión logística”

La práctica se deberá realizar en equipos de 2 personas.

Datasets

1. Dataset 1: Credit Card Default Data (`Default.txt`). Este conjunto de datos contiene 10,000 instancias donde el objeto es predecir cuáles clientes van a incumplir con la deuda de su tarjeta de crédito. Este dataset cuenta con el siguiente formato.
    a. `ID:` número de la entrada (descartar).
    b. `default:` un factor con niveles `No` y `Yes` indicando si el cliente va a incumplir su deuda. Este es la variable de salida del dataset.
    c. `student:` un factor con niveles `No` y `Yes` indicando si el cliente es un estudiante. Actúa como atributo.
    d. `balance:` el saldo promedio que le queda al cliente en su tarjeta de crédito después de realizar su pago mensuale. Actúa como atributo.
    e. `income:` ingreso del cliente. Actúa como atributo.
2. Dataset 2: Identificación de género (`genero.txt`). Este conjunto contiene 10,000 instancias donde el objeto es predecir el género de una persona, es decir, `Male` o `Female`. Este dataset cuenta con el siguiente formato.
    a. `Gender:` género de la persona con dos niveles: `Male` o `Female`.
    b. `Height:` altura de la persona.
    c. `Weight:` peso de la persona.

A continuación se presentan los procedimientos para los tres métodos de clasificación de interés en esta práctica. El procedimiento debe ser realizado para los dos datasets. Los datasets deberán ser partidos de forma aleatoria en una proporción 80%-20% para generar el training set y test set, respectivamente. Estos conjuntos deberán ser ocupados por los tres clasificadores sin cambiarlos. Por lo tanto, una única vez se hará la partición del dataset en el training set y test set y luego estos serán compartidos por los tres clasificadores. En todos los casos se hará uso de una validación simple.

Procedimiento de clasificación con k-NN

1. Investigar el uso del método k-NN en `Scikit-Learn`, es decir, la función `NearestNeighbors()`.

2. Entrenar el modelo. En la función `NearestNeighbors()` usar `brute` como el valor del parámetro `algorithm` y para el parámetro `n_neighbors` emplear los valores: 1, 2, 3, 5, 10, 15, 20, 50, 75, 100. El uso de los 10 valores de `n_neighbors` implica que se generarán 10 modelos.
3. Calcular la precisión de los modelos (obtener la tasa de precisión) e investigar cómo obtener la matriz de confusión con `Scikit-Learn`.
4. Con base en los valores de las tasas de precisión se debe generar una gráfica donde muestre k (el número de vecinos) contra la tasa de precisión obtenida. La intención es que se observe cómo la precisión de k-NN está en función del número de vecinos empleados. Generar una gráfica por cada dataset. Unir los puntos de la tasa de precisión para que se vea la tendencia de la tasa de precisión.
5. Discutir en el reporte todas las observaciones realizadas en esta experimentación.
6. Sólo para el caso del dataset de identificación de género se debe realizar lo siguiente.
    a. Con base en el test set que se utilizó para generar los modelos, graficar las instancias que lo constituyen, es decir graficar `Height vs Weight`. Cada punto debe ser identificado de acuerdo con la clase que tiene asociada. Color rojo para `Female` y azul para `Male`.
    b. Realizar este tipo de graficación ahora con base en lo que predijo el modelo con la k que produjo la mejor tasa de precisión .

Procedimiento para regresión logística

1. Investigar el uso de la regresión logística en `Scikit-Learn`, es decir, la función `LogisticRegression()`.
2. Entrenar el modelo.
3. Calcular la precisión de los modelos (obtener la tasa de precisión) y la matriz de confusión con `Scikit-Learn`.
4. Sólo para el caso del dataset de identificación de género se debe realizar lo siguiente.
    a. Con base en el test set que se utilizó para generar los modelos, graficar las instancias que lo constituyen, es decir graficar `Height vs Weight`. Cada punto debe ser identificado de acuerdo con la clase que tiene asociada. Color rojo para `Female` y azul para `Male`.
    b. Realizar este tipo de graficación ahora con base en lo que predijo el modelo.

Comparación de clasificadores

1. Generar una discusión del comportamiento y precisión de los dos clasificadores con base en los resultados de la tasa de precisión.
2. Comparar las gráficas `Height vs Weight` donde se observa cómo clasificaron las instancias del test set por parte de los 2 clasificadores. Discutir el comportamiento de los clasificadores.
3. Ocupar las matrices de confusión creadas para los tres clasificadores para gráficar sus puntos asociados (True positive rate, False positive rate) en el espacio ROC. Con

base en el análisis del espacio ROC determinar quién es el mejor clasificador. Dar argumentos que respalden sus observaciones.

Reporte de la práctica

Emplear el formato de práctica dado por el profesor y seguir las instrucciones mostradas. El archivo que se subirá a *Canvas* deberá estar estrictamente en formato PDF y deberá ser nombrado como `report.pdf`.

Usar el lenguaje `Python` para desarrollar la práctica. **Únicamente será aceptado este lenguaje para la generación de los programas**. Además, es forzoso el uso de `Scikit-learn`. Entregar el programa con extensión `.py` debidamente comentado. Entregar un archivo `README.txt` donde se exponga cómo ejecutar el programa (indicar los parámetros en caso de necesitarlos) y un ejemplo para cómo ejecutar el programa y producir así los resultados reportados.

Entrega global

Tanto el reporte y el programa deberán ser empaquetados en un archivo `.ZIP` y nombrarlo: `practice3.zip`. **Cualquier falta a las instrucciones pedidas implicará la anulación de la práctica para todos los integrantes del equipo.**