Tarea 3 EL4106 - Semestre Otoño 2021

Profesor: Javier Ruiz del Solar
Auxiliar: Patricio Loncomilla

Fecha enunciado: Jueves 29-04-2021
Plazo entrega tarea: Miércoles 12-05-2021

El objetivo de esta tarea es hacer selección de características, para un problema de clasificación de diabetes. Se utilizará el conjunto de datos: *Pima Indians Diabetes Database.* Este conjunto tiene 8 características, con 268 muestras positivas y 500 negativas. La base de datos estará disponible en u-cursos.

En este conjunto, la obtención de la mayoría de las características requiere realizar exámenes médicos a personas, por lo cual reducir su número es importante.

Se les pide usar scikit-learn para analizar los datos, y encontrar un subconjunto reducido de características que permita resolver el problema.

Se pide:

1) Pruebas
   a) Explicar brevemente cuáles son las características contenidas en el dataset
   b) Leer el archivo de diabetes usando pandas. Se debe reemplazar las etiquetas de la clase por los valores 0 y 1. Además se debe dividir en entrenamiento (60%), validación (20%) y prueba (20%) usando train_test_split
   c) Preprocesar las características usando un StandardScaler
   d) Realizar una clasificación inicial usando un clasificador svm lineal. Se debe usar una grilla sobre el hiperparámetro C. En esta tarea, en el grid search se debe usar el conjunto de validación para evaluar la calidad de los hiperparámetros. Además, se debe indicar el tiempo de entrenamiento.
   e) Generar una matriz de confusión normalizada para el clasificador inicial, calculando además el *accuracy* (el promedio de la diagonal), usando el conjunto de validación. Se debe usar sns.heatmap( ) para mostrar la matriz de confusión.
   f) Realizar selección de características usando SelectFromModel (un clasificador de tipo *wrapper*), usando el mejor clasificador encontrado en el punto anterior (el que contiene los mejores hiperparámetros). Indique cuáles son las características seleccionadas
   g) Entrenar un segundo clasificador con el conjunto de características reducido, indicando el tiempo de entrenamiento.
   h) Generar una matriz de confusión normalizada para el clasificador con características reducidas, calculando además el *accuracy*, usando el conjunto de validación. Se debe usar sns.heatmap( ) para mostrar la matriz de confusión.
   i) Repetir los pasos (g) y (h) usando un clasificador svm lineal, eligiendo las 4 mejores características encontradas por SelectKBest (método de tipo filtro)
   j) Repetir los pasos (g) y (h) usando un clasificador svm lineal, eligiendo las 2 mejores características encontradas por SelectKBest
   k) Repetir el paso (d) y (e) usando un clasificador RandomForest, con una profundidad de 3 y usando una grilla sobre el hiperparámetro n_estimators
   l) Repetir los pasos (g) y (h) usando un clasificador RandomForest, usando el selector de características SelectFromModel.
   m) Repetir los pasos (g) y (h) usando un clasificador RandomForest, eligiendo las 4 mejores características encontradas por SelectKBest
   n) Repetir los pasos (g) y (h) usando un clasificador RandomForest, eligiendo las 2 mejores características encontradas por SelectKBest
   o) Evalúe el mejor svm y el mejor random forest encontrados con características reducidas sobre el conjunto de prueba, indicando sus matrices de confusión normalizadas y sus *accuracies*

2) Análisis
   a) Indique qué tipo de clasificador (svm lineal o random forest) entrega en general mejores resultados
   b) Indique el efecto de reducir características sobre el *accuracy* obtenido. Considere el número de características seleccionadas v/s el *accuracy* obtenido.
   c) Según los resultados de los dos puntos anteriores, analice la utilidad de la reducción de características para el conjunto de datos usado.

Para evaluar la grilla sobre el conjunto de validación, averigüe sobre el uso de PredefinedSplit de scikit-learn. Note que el mejor clasificador encontrado sobre la grilla debe ser reentrenado con los datos del conjunto de entrenamiento.

Los hiperparámetros que se deben considerar son:

Para svm lineal: C = [0.0001, 0.001, 0.1].
Para random forest: max_depth = 3, y n_estimators = [50, 100, 150, 200, 250]

Los informes y códigos deben ser subidos a u-cursos a más tardar a las 23:59 del día miércoles 12 de mayo.

Se recuerda que la estructura esperada para los informes se encuentra en u-cursos. Además, los informes deben estar en formato PDF, y los códigos relevantes para cada sección deben ser agregados como texto (no como imágenes). En el caso en que se quiera entregar un notebook como informe, éste se debe entregar en PDF, debe cumplir la estructura pedida para los informes, debe incluir todos los experimentos pedidos y no debe contener warnings que puedan dificultar la corrección del informe.

Además, si el notebook no se ejecuta directamente celda por celda (su ejecución no es directa), se debe adjuntar un archivo README.

Importante: La evaluación de la tarea considerará la inclusión de los resultados de los pasos pedidos en el informe, la calidad de los experimentos realizados y de su análisis, la inclusión de las partes importantes del código en el informe, así como la forma, prolijidad y calidad del mismo.

Nota: Los informes en PDF deben ser subidos a la plataforma turnitin.