Pontificia Universidad Católica de Chile
Escuela de Ingeniería
Departamento de Ciencia de la Computación

Actividad 3:

Mini proyecto - evaluación de clasificadores

Aplicaciones en Ciencia de Datos e Inteligencia Artificial

Profesor : Francisco Pérez Galarce
Ayudante : Yesenia Salinas

Fecha : 19 de noviembre de 2024

1 Introducción

Un clasificador en ciencia de datos es un algoritmo o modelo que recibe un conjunto de datos discretos y/o continuos (*descriptores*) y retorna un valor categórico (*la clase*). En esta actividad, usaremos algunos clasificadores simples, como la regresión logística y *Naive Bayes*. Para entrenar estos modelos, usted podrá seleccionar entre dos posibles bases de datos, una asociada a la clasificación de estrellas variables y otra asociada a la identificación de lesiones cutaneas.

Los clasificadores deben desarrollarse de manera que no se sobreajusten a los datos de entrenamiento y puedan obtener buenos resultados con datos no vistos. Es decir, tal como estudiamos en la actividad anterior, debemos construir modelos que generalicen adecuadamente. Para seleccionar el modelo adecuado, usted deberá presentar los resultados utilizando distintas métricas (*accuracy, recall, precision* y *F1-score*) en una estrategia de validación (*hold-out* o *cross-validation*).

2 Bases de datos

2.1 Opción 1: estrellas variables (descargar)

Las estrellas variables son objetos que cambian su brillo en el tiempo. El origen de su variabilidad puede ser originado por objetos pasando frente a ellos (planetas, asteroides, manchas solares u otra estrella) o causados por procesos físicos en su interior.

Las observaciones de estas estrellas no son constantes, ya que no siempre pueden ser observadas producto de las estaciones, condiciones climáticas, fases de la luna, entre otros factores. Esto hace que las series de tiempo resultantes no tengan un largo fijo. Además, el muestreo es distinto para cada estrella. Para poder usar la información en un clasificador automático, se calculan descriptores que resumen la información en un vector de largo fijo. Esta información puede ser de origen físico, como la periodicidad, o estadístico, como la dispersión en torno a la media.

En esta actividad usarán descriptores estadísticos ya calculados para clasificar estrellas variables.

2.2 Opción 2: Lesiones cutaneas (descargar)

El conjunto de datos PAD-UFES-20 fue recopilado junto con el Programa de asistencia dermatológica y quirúrgica de la Universidad Federal de Espírito Santo, que es un programa sin fines de lucro que proporciona tratamiento gratuito de lesiones cutáneas, en particular, a personas de bajos ingresos que no pueden permitirse un tratamiento privado.

El conjunto de datos consta de 2,298 muestras de seis tipos diferentes de lesiones cutáneas. Cada muestra consta de una imagen clínica y características clínicas, incluida la edad del paciente, la ubicación de la lesión cutánea, el diámetro de la lesión cutánea, entre otras.

Las lesiones cutáneas son: carcinoma de células basales (BCC), carcinoma de células escamosas (SCC), queratosis actínica (ACK), queratosis seborreica (SEK), enfermedad de Bowen (BOD), melanoma (MEL) y nevus (NEV). Como la enfermedad de Bowen se considera SCC, se agrupa, lo que da como resultado seis lesiones cutáneas en el conjunto de datos, tres cánceres de piel (BCC, MEL y SCC) y tres enfermedades de la piel (ACK, NEV y SEK). Todos los BCC, SCC y MEL están probados mediante biopsia. Los restantes podrán tener diagnóstico clínico según consenso de un grupo de dermatólogos. En total, aproximadamente el 58% de las muestras de este conjunto de datos están probadas mediante biopsia.

3 Instrucciones del mini proyecto

3.1 *Transformación e imputación de datos (25 puntos)*

0 ptos Abrir entorno de programación, de preferencia utilizar `Visual studio code`. Importe las librerías `pandas`, `searborn`, `matplotlib`, `numpy` y `sklearn`. Le recomendamos usar un ambiente de `conda` específico para el curso.

2 ptos Cargar la base de datos (*Gaia_NaN.csv* o *metadato.csv*). Cree una función que permita cargar la base de datos bajo diferentes condiciones. Los argumentos de esta función deben ser: (i) un string con el nombre del directorio donde se encuetre la base de datos, (ii) una variable booleana que indique si se trabajará con una muestra o con la base de datos completa y (iii) un argumento que reciba las columnas con las que se pueda trabajar en una lista. Usted puede agregar nuevos argumentos que den mayor flexibilidad a la carga de datos. Recuerde verificar el tipo de variable reconocido por `pandas`.

2 ptos Genere un diagnóstico de estadística descriptiva y de datos faltantes. Cree una función que permita realizar el diagnóstico de forma flexible, la función debe retornar, media, desviación estándar, valores perdidos por descriptor, valor máximo y valor mínimo. Usted puede usar funciones internas de otras librerías. Cada uno de los estadísticos debe ser un argumento booleano en la función y solo cuando se indique `True` este se calculará. Los descriptores para los cuales se calcularán estos descriptores también deben ser un argumento de la función.

3 ptos Impute los datos perdidos con el método de su elección. Genere una función que reciba una lista de descriptores, el dataframe original y una lista con las estrategias de imputación de cada descriptor. La función debe retornar la nueva base de datos imputada. ¿Cómo cambió la distribución de los datos con la imputación realizada?

2 ptos Genera 2 gráficos diferentes (ejemplo: boxplot, scatter plot, histogramas, gráfico de torta, etc.) que entreguen información relevante para el modelamiento del problema (ejemplo: correlaciones evidentes, datos atípicos, patrones no lineales de relaciones, etc). Debe explicar tanto la elección de cada gráfico como la información otenida a partir de ellos.

2 ptos Cree una función que permita hacer scatter plots y/o box plots para dos descriptores datos. La función debe recibir como argumento las dos variables, y el tipo de gráfico que se desea obtener. La función debe recibir como argumento la decisión de visualizar o guardar los gráficos realizados. Usted puede agregar más argumentos para obtener visualizaciones más personalizadas. Usando dicha función, genere visualizaciones para 5 de los descriptores de la base de datos entregada.

3 ptos Aplique normalización z o escalamiento a los datos. Genere una función que permita aplicar estas transformaciones a los datos, como argumento se debe indicar qué tipo de estrategia se usará para cada descriptor. La función debe retornar el dataframe modificado.

3 ptos Genere sets de entrenamiento y testeo, con separación estratificada. Genere una función que aplique este procesamiento. No olvide fijar la semilla aleatoria para poder replicar los resultados.

8 ptos Consolide todas las funciones en una clase. Esta clase tendrá por nombre `preprocesamiento`. Algunos de los parámetros que se usan en las funciones antes creadas pueden ser entregadas en la inicialización de la clase. Agregue una función que aplique todo el procesamiento, denomine a esta función `ejecutar_procesamiento`.

3.2 *Entrenamiento de modelos (35 puntos)*

10 ptos Ajuste los clasificadores naive Bayes (desde `sklearn.naive_bayes.GaussianNB`) y regresión logística (desde `sklearn.linear_model.LogisticRegression`). Genere una función con nombre `clasificador` que reciba como argumento: (i) el tipo de clasificador que desea ajustar, (ii) el nombre de la dirección donde se guardara el modelo y (iii) los datos de entrenamiento. La función solo debe ajustar y guardar el modelo.

10 ptos Cree una función que tenga por nombre `evaluar_rendimiento`, esta función debe recibir la dirección del modelo, los datos que desea evaluar (entrenamiento o test) y el tipo de análisis. Los análisis posibles son: (i) mostrar la matriz de confusión y (ii) mostrar las métricas de evaluación (*accuracy, recall, precision* y *F1-score*).

15 ptos Use estas funciones para probar distintos modelos, explore los siguientes argumentos en la regresión logística: `penalty, C, class_weight, l1_ratio`. En naive Bayes modifique: `priors` de acuerdo a la descripción de la librería. Entregue un análisis de los resultados y seleccione un modelo. También puede aplicar procedimientos para seleccionar los descriptores que se incluyen en el modelo final.

3.3 Entrega

- La actividad deberá entregarse en un archivo comprimido donde incluya archivos `Jupyter notebook` (.ipynb), `Python` (.py) y otros archivos para gestionar parámetros (ejemplo: `.yaml, .json`). El archivo comprimido debe subirse a la plataforma del curso y subirse a su repositorio del curso en `Github`[a]. No subir el archivo con los datos originales.
- La actividad debe realizarse en los grupos ya formados para el curso. Se recomienda tener instalado el complemento `Live share`.
- La actividad debe ser subida a la plataforma antes del martes 26 a las 21:59 P.M.

[a] https://github.com/