Universidad Tecnológica Centroamericana
Facultad de Ingeniería

CC414 - Sistemas Inteligentes
Docente: Kenny Dávila, PhD

Tarea #2 (3% Puntos Oro)

Para completar esta tarea es requerido usar **Python 3** y la librería **Scikit-Learn**. También se recomienda el uso de las librerías **Numpy** para manejar los datos y Matplotlib para la creación de plots con los datos resultantes.

Parte 1. Clustering (1.0)

Se le entregan 3 diferentes datasets (datos_1.csv, datos_2.csv, datos_3.csv). El objetivo de este ejercicio es experimentar con diferentes algoritmos de clustering y comparar sus resultados en cada uno de ellos. En particular se le pide utilizar los algoritmos K-means, Agglomerative Clustering ("Ward"), y DBScan. Nótese que diferentes algoritmos de clustering requieren el uso de diferentes parámetros para obtener mejores resultados:

a) Para K-means, se le pide experimentar con diferentes valores de K, con $1 \leq K \leq 5$. En total son 5 posibles configuraciones que debe probar con este algoritmo.
b) Para Clustering jerárquico o aglomerativo, se le pide usar "Ward", y deberá experimentar con valores de K ($1 \leq K \leq 5$), y con diferentes valores en el umbral de distancia: 0.25, 0.50, 0.75, 1.0 y 1.5. Nótese que si usa umbral de distancia entonces el valor K debe ser None y viceversa ya que solamente se puede usar un número fijo de clústeres o una distancia máxima, pero no ambos criterios al mismo tiempo. En total son 10 posibles configuraciones que debe probar con este algoritmo (5 valores K + 5 umbrales de distancia).
c) Para DBScan, debe probar con diferentes valores de distancia entre vecinos (eps={0.25, 0.35,0.5}), y diferentes valores del mínimo de muestras por vecindario (min_samples={5,10,15}). En total son 9 posibles configuraciones ({3 valores eps} X {3 valores min_samples}) que debe probar con este algoritmo.

Para cada configuración de cada algoritmo, se le pide generar un scatter plot (puede usar matplotlib), donde se pueda observar los resultados del algoritmo de clustering. Su objetivo es analizar dichos resultados visualmente y reportar cual considera que fue la combinación de parámetros que produjo los resultados más satisfactorios (por cada algoritmo sobre cada uno de los datasets). En Total, deberá proveer 9 scatter plots (3 algoritmos x 3 datasets).

Posteriormente, se le pide proveer un breve análisis sobre cada uno de los datasets provistos. ¿Qué tipo de clustering considera que funciona mejor en cada dataset y por qué? ¿Cuántas clases reales cree que se usaron para generar los datos en cada dataset?

Parte 2. K-NN (1.0)

Se desea crear un programa que pueda clasificar películas de manera automática en 4 categorías generales: Horror, Acción, Comedia, y Drama. Para este propósito, se consideran diferentes atributos de cada película como ser:

1. **Animada.** Determina si la película es animada (ya sea 3D o estilo caricatura) o no.
2. **Basada en Libro.** Determina si la película está basada en algún libro.
3. **Clasificación.** Indica la audiencia que puede ver la película, donde A es apta para todo público, B es adolescentes y adultos, y C es solamente adultos.
4. **Desenlace Feliz.** Indica si la historia tiene un desenlace feliz o no.
5. **Duración.** Describe el rango de la longitud de la película en minutos y puede ser: 30-80, 80-120, o 120+.
6. **Narración.** Describe que tipo de narración se usa para contar la historia y puede ser: lineal, mosaico o circular.
7. **Origen.** Determina si la película esta basada en una historia de origen real o ficticia.
8. **Saga.** Indica si se trata de una película que es parte de una saga o no.
9. **Tiempo.** Describe el tiempo o era en que se desarrolla la mayor parte de la historia y puede ser era contemporánea, futura o pasado.
10. **Trama.** Describe la complejidad de la trama, y puede ser simple o compleja.

El objetivo de este este ejercicio es utilizar la implementación de K-NN provista en la librería Scikit-Learn. Debe utilizar los datos de clasificación en 4 categorías (Horror, Acción, Comedia y Drama) y evaluar el rendimiento del clasificador K-NN sobre dichos datos. Nótese que será necesario que convierta los atributos binarios en números (0 y 1s) antes de poder usar el K-NN sobre estos datos. Otros atributos con mas de 2 valores necesitan codificarse usando múltiples atributos binarios. Se le pide usar los datos de entrenamiento (“genero_peliculas_training.csv”) para entrenar diferentes modelos del K-NN, con los valores de K = {1, 3, 5, 7, 9, 11, 13, 15}. Luego, usando los datos de prueba (“genero_peliculas_testing.csv”), deberá reportar las siguientes métricas de evaluación para cada valor K: Accuracy total; recall, precisión y F1-score por clase (una vs las demas); promedio para todas las clases; y también tiempo total de predicción. Puede usar una sola tabla para reportar dichos resultados.

Parte 3. Arboles de Decisión (1.0)

El objetivo de este ejercicio es utilizar la implementación de “Decision Tree Classifier” provista en la librería Scikit-Learn. Se usarán los mismos datos de clasificación de películas descritos en la segunda parte. Sin embargo, el árbol de decisión trabaja mejor con atributos categóricos y por lo tanto deben utilizarse en su formato original, sin codificación de ningún tipo. Se le pide usar los datos de entrenamiento (“genero_peliculas_training.csv”) para entrenar diferentes modelos del árbol de decisión usando diferentes combinaciones de los criterios disponibles (“Gini” y

"entropy") y max_depth (2, 3, 4, 5 y None). Se espera que se explore un total de 10 configuraciones (2 criterios x 5 profundidades máximas). Otros parámetros como splitter, max_features, min_impurity_decrease y max_leaf_nodes entre otros deben dejarse en sus valores por defecto. Igual que en la segunda parte, deberá utilizar los datos de prueba ("genero_peliculas_testing.csv") para calcular y reportar las siguientes métricas de evaluación para cada configuración de arbol: Accuracy total; recall, precisión y F1-score por clase (una vs las demás); promedio para todas las clases; y también tiempo total de predicción. Puede usar una sola tabla para reportar dichos resultados.

Adicionalmente, para obtener crédito total de esta parte deberá contestar explícitamente las siguientes preguntas de análisis:

1. ¿Cuáles fueron los 3 atributos más importantes según el árbol de decisión entrenado? y un breve análisis en sus propias palabras de porque consideran que estos atributos fueron seleccionados por el algoritmo.
2. ¿Cuáles fueron los 3 atributos menos importantes? y ¿por qué considera que no fueron tan útiles durante la clasificación?
3. Sorpresas. Detalle si los resultados anteriores fueron consistentes o no con sus propias expectativas sobre el tema.
4. En comparación con los resultados obtenidos con K-NN, ¿considera que el árbol de decisión funcionó mejor o peor para este problema?

Otras políticas

1. Esta tarea deberá trabajarse y entregarse **individual** o en **parejas.**
2. La entrega será **un solo archivo comprimido (.zip o .rar)**. Dentro de dicho archivo debe contener la guía completada **en formato PDF**. También debe los scripts de Python que se usaron para contestar cada punto.
3. Si se hace en parejas, ambas personas deben subir el mismo archivo.
4. El **plagio** será penalizado de manera severa.
5. Los estudiantes que entreguen una tarea 100% original recibirán una nota parcial a pesar de errores existentes. En cambio, los estudiantes que presenten tareas que contenga material plagiado recibirán 0% automáticamente independientemente de la calidad.
6. Tareas entregadas después de la fecha indicada solamente podrán recibir la mitad de la calificación final. Por esta razón, es posible que **un trabajo incompleto pero entregado a tiempo termine recibiendo mejor calificación que uno completo entregado un minuto tarde**.