Universidad Tecnológica Centroamericana
Facultad de Ingeniería

CC414 - Sistemas Inteligentes
Docente: Kenny Dávila, PhD

Tarea #3 (2% 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 ("Single Linkage"), y DBScan. Notese 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 "Single Linkage", 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 numero fijo de clusters 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 DB Scan, 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 6 posibles configuraciones 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 (0.25%)

El objetivo de este este ejercicio es utilizar la implementación de K-NN provista en la librería Scikit-Learn. Debe re-utilizar los datos de clasificación binaria (Windows vs. Linux) del primer mini-proyecto 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. Se le pide usar los datos de entrenamiento ("os_training_data.csv") para entrenar diferentes modelos del K-NN, con los valores de K = {1, 3, 5, 7, 9, 11, 13, 15}. Para cada valor K, deberá reportar el accuracy, recall, precisión, F1-score (con clase positiva=Windows), y tiempo total de predicción sobre los datos de prueba ("os_testing_data.csv"). Puede usar una sola tabla para reportar dichos resultados.

En comparación con los resultados obtenidos con Arboles de Decisión, ¿considera que K-NN funcionó mejor o peor para este problema?

Parte 3. Regresión Lineal (0.5%)

En esta parte debe aplicar regresión lineal para predecir el Puntaje total (de 0 a 5 puntos) que un usuario le podría dar a una película que no ha visto. Para esto, se le da un dataset con una serie de atributos como ser: rating global, genero favorito, casting, advertising y longitud. El rating global se refiere a un puntaje que otros usuarios le han dado a la película (de 1 a 5 estrellas). El genero favorito se refiere a si el usuario gusta de este género de películas (0=disgusto total, 1=genero favorito). El casting se refiere a que tanto le gustan los actores principales de esta película al usuario (0=no le gusta ninguno, 10=le gustan todos). El advertising se refiere a que tan buena y precisa ha sido la publicidad hecha a la película (0=mala publicidad incluyendo trailers engañosos, 1 = buena publicidad muy atractiva). Finalmente, longitud es la duración de la película (entre 60 a 180 minutos).

Usando regresión lineal, usted deberá encontrar los diferentes pesos que se le pueden dar a cada atributo (más el intercepto) a fin de predecir el puntaje final otorgado por el usuario. Se le pide experimentar con dos tipos de regresión lineal: mínimos cuadrados y Lasso con regularización L1. Ambos métodos deben ser entrenados usando el dataset de entrenamiento ("datos_4_train.csv"), y su rendimiento deberá ser evaluado usando el dataset de pruebas ("datos_4_test.csv"). Deberá calcular métrica de evaluación promedio del error al cuadrado (MSE), y también debe imprimir los pesos que se obtiene para cada atributo en cada uno de los regresores. Deberá repetir estos pasos usando la versión original de la data y la versión normalizada de la data (usando la clase StandardScaler de Scikit-learn).

Se le pide contestar las siguientes preguntas:

1. ¿Cuáles son los valores de MSE del modelo de regresión por mínimos cuadrados usando tanto los datos originales como también los normalizados? (reporte tanto en datos de entrenamiento como en datos de pruebas)
2. ¿Cuáles son los valores de MSE del modelo de regresión de Lasso usando tanto los datos originales como también los normalizados? (reporte tanto en datos de entrenamiento como en datos de pruebas)

3. ¿Cuál de los dos modelos funciona mejor a su criterio y por qué?
4. En base a datos no normalizados, ¿Cómo ordenaría la importancia de cada uno de los 5 atributos en la decisión final?
5. En base a datos normalizados, ¿Cómo ordenaría la importancia de cada uno de los 5 atributos en la decisión final? ¿hay cambios con respecto al punto anterior? ¿Por qué si o porque no?

Parte 4. Regresión Logística (0.25%)

El objetivo de este este ejercicio es utilizar la implementación de Regresión Logística provista en la librería Scikit-Learn. Debe re-utilizar los datos de clasificación binaria (Windows vs. Linux) igual que en la parte 2 y evaluar el rendimiento del clasificador basado en Regresión Logística sobre dichos datos. Nótese que será necesario que convierta los atributos binarios en números (0 y 1s) antes de poder usar la regresión logística sobre estos datos. Se le pide usar los datos de entrenamiento ("os_training_data.csv") para entrenar un modelo de regresión logística. Deberá reportar el Accuracy, Recall, Precisión, F1-score (con clase positiva=Windows), y tiempo total de predicción sobre los datos de prueba ("os_testing_data.csv").

En comparación con los resultados obtenidos con Arboles de Decisión y K-NN, ¿considera que la regresión Logística funcionó mejor o peor para este problema?

Reporte los pesos aprendidos para cada atributo por la regresión logística. ¿Qué atributos tienen mayor peso absoluto y que atributos tienen menor peso absoluto? ¿Es esto consistente con sus hallazgos basados en arboles de decisión y entropía?

Fechas Importantes

1. Asignado: 25 de Noviembre del 2020.
2. Entrega: 2 de Diciembre del 2020.

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**.