Prueba de aptitud y conocimientos

Esta prueba busca evaluar los conocimientos del aspirante en programación y análisis de datos, por lo que puede ser realizado utilizando R o Python.

Instrucciones generales:

- Utilice gráficos cuando sea posible.
- Evite utilizar lazos de repetición.
- Evite utilizar rutas absolutas que limiten la reproducción de su análisis.
- Use nombres descriptivos para las variables que utiliza.
- Comente su código de manera adecuada y comente el resultado de su análisis.
- Las actividades y tipos de actividades deben ser datos categóricos.
- Debe evidenciar uso de diferentes tipos de estructuras de datos de acuerdo a como sea apropiado para los datos utilizados.
- Cuando se requiere: información general, descripción, resúmenes de los datos obtenidos o para resolver preguntas sobre los datos puede utilizar: medidas de estadística descriptiva, histogramas, correlaciones, gráficos, etc.

Tema 1. Manipulación de datos

Descripción del conjunto de datos:

1. El conjunto de datos a utilizar será el de "Reconocimiento de actividades humanas y transiciones de postura usando Smartphones Samsung", disponible en el Repositorio de Machine Learning de UCI: http://archive.ics.uci.edu/ml/datasets/Smartphone-Based+Recognition+of+Human+Activities+and+Postural+Transitions
2. Este conjunto de datos contiene muestras 10929 muestras, cada una con 561 features/características/atributos/variables, tomadas de 30 sujetos que participaron en el estudio. Los 561 features corresponden a datos de sensores que caracterizan una actividad.
3. El archivo features.txt tiene los nombres de las 561 variables.
4. Los datos de los sujetos están en los archivos: subject_id_train.txt, subject_id_test.txt
5. Los datos de los sensores están en los archivos: x_train.txt, x_test.txt
6. Los datos de la actividad reconocida están en los archivos: y_train.txt, y_test.txt

Actividades:

1. Descargar el dataset de la web, extraer y leer los datos, considerando que no nos interesa la data en la carpeta RawData.
2. Cargar los datos de entrenamiento y testeo en una sola estructura de datos que tenga asociado los nombres propios de los features, el sujeto y la actividad reconocida (esta estructura tendrá 563 columnas).
3. Muestre estadísticas descriptivas que resuman información sobre los datos.

4. ¿Cuántas muestras existen en este dataset para cada posible actividad reconocida?
5. ¿Cuántas actividades fueron reconocidas para cada sujeto?
6. ¿Cuáles y cuantas fueron las actividades reconocidas para cada sujeto?
7. Cree una nueva estructura de datos para almacenar sólo los valores máximos leídos por cada uno de los sensores, descríbala brevemente.
8. ¿Cuál es la relación de cada uno de estos features entre ellos y entre la actividad que fue reconocida?
9. Cree dos subconjuntos de datos adicionales en base al tipo de actividad: estática (SITTING, STANDING, LYING) y dinámica (WALKING, WALKING_UPSTAIRS, WALKING_DOWNSTAIRS) y describa brevemente.
10. ¿Cuál es la relación entre los datos recopilados por los sensores y los tipos de actividad (estática y dinámica)?

Tema 2. Análisis de series de tiempo

Para este tema se requiere el desarrollo de un script que pueda ser utilizado para evaluar y comparar automáticamente el desempeño de varios métodos de modelamiento y pronóstico de series de tiempo.

Actividades:

1. El script debe ser capaz de leer una serie de tiempo con la estructura del archivo serie.dat. La serie debe ser univariada, mensual, con al menos 60 meses, sin encabezado, con los decimales separados por punto. Para el desarrollo puede utilizar el archivo serie.dat subido a la plataforma como referencia. La serie en el archivo son datos mensuales registrados desde enero de 1946.
2. Deberá particionar la serie en TRAIN (primeros 80% de los puntos) y TEST (últimos 20% de los puntos).
3. Se consideraran los siguientes modelos: naive, snaive, drift, STL (con método naive), Holt-Winters (aditivo y multiplicativo) y ARIMA (en R puede utilizar unicamente el modelo ARIMA obtenido de aplicar la función auto.arima). En cada caso puede decidir usar una transformación de Box-Cox o no. ***Si se desea añadir modelos adicionales, siéntase libre de hacerlo.***
4. Cada modelo deberá ser ajustado a los datos TRAIN y se debe calcular el RMSE utilizando los datos TEST.
5. La tabla (dataframe) comparativa con los RMSE de todos los modelos debe ser grabada en un archivo resultados.csv por el script. Ordenar los resultados de mejor a peor desempeño antes de grabarlos.

Entregables:

- Código fuente R o Python que permita reproducir el análisis.
- PDF o HTML del análisis comentado con gráficos y resultados procesados