Minería de Datos

Solemne 2: Requerimientos del Proyecto 2

---

1. Problemas

Problema 1. Modelos de Regresión Lineal (Linear Regression Models)

Contexto: Predicción del rendimiento en el consumo de combustible

Se requiere predecir el rendimiento en el consumo de combustible de un automóvil en función de varias características, como, por ejemplo, el peso del automóvil, la cantidad de cilindros y la potencia medida en caballos de fuerza. El rendimiento en el consumo del combustible muestra qué tan lejos puede viajar un automóvil con una cierta cantidad de combustible. Esta propiedad puede medirse en "millas por galón" (mpg).

La base de datos a utilizar tiene el siguiente enlace de descarga directa:
https://archive.ics.uci.edu/ml/machine-learning-databases/auto-mpg/auto-mpg.data

También se publicó el archivo de la base de datos “auto-mpg.dat” en la sección de “Evaluaciones\Solemne 2”

La información sobre las variables involucradas en la base de datos se puede conseguir en el siguiente enlace:
https://archive.ics.uci.edu/dataset/9/auto+mpg

Se debe tener en cuenta que esta base de datos no tiene incorporados los nombres de sus columnas o variables. Estos nombre o identificadores pueden ser los siguientes y corresponden con el orden de las columnas de datos: 'mpg', 'cylinders', 'displacement', 'horsepower', 'weight', 'acceleration', 'model_year', 'origin', 'car_name'

Requerimientos:

Se requiere crear tres (3) modelos de regresión lineal simple y un modelo de regresión lineal múltiple que sirvan para la predicción de la variable indicada (rendimiento en el consumo de combustible).

Para esto se debe elaborar un notebook de Google Colab o un archivo .py donde se pueda codificar, ejecutar y explicar los resultados de las instrucciones del lenguaje Python que cubran las siguientes actividades:

Paso 1. Importar Bibliotecas y Cargar Datos

- Importar librerías Python.
- Cargar los datos en un dataframe.

Paso 2. Preparación de Datos:

- Determinar la cantidad total de registros y de variables (atributos).
- Determinar los tipos de datos por cada variable (atributo).
- Eliminar la columna 'car_name'.
- Sustituir los datos faltantes por la media respectiva.

Paso 3. Selección de Variables Independientes y Dependiente.

- Generar una matriz de correlación a través de un mapa de calor de todas las variables.
- Tomando como variable dependiente la columna 'mpg', determine las tres (3) variables que presenten una mayor correlación (positiva o negativa), con dicha variable dependiente (mpg).
- Elegir estas tres (3) variables como independientes y generar los conjuntos de datos (dataframe) para cada una de estas variables y para la variable dependiente.

Paso 4: Dividir Datos en Conjuntos de Entrenamiento y Prueba

- Determinar los conjuntos de entrenamiento y prueba para cada una de las variables independientes y dependiente.
- Utilice una proporción 70% entrenamiento - 30% prueba.

Paso 5: Crear los tres Modelos de Regresión Lineal Simple.

- Crear un modelo de regresión lineal simple para cada variable independiente elegida.

Paso 6: Crear el Modelo de Regresión Lineal Múltiple.

- Utilizando las tres variables independientes elegidas crear un modelo de regresión lineal múltiple.

Paso 7: Evaluar los modelos de regresión.

- Evaluar los tres modelos de regresión lineal simple y el modelo de regresión lineal múltiple utilizando las siguientes métricas: Error Absoluto Medio (MAE), Error Cuadrático Medio (MSE) y Raíz del Error Cuadrático Medio (RMSE).

Paso 8: Probar el mejor modelo de regresión.

- Elegir el mejor de los modelos de regresión (con base en las métricas de evaluación) y utilizarlo para predecir el valor de la variable dependiente.
- Genere nuevos datos para la o las variables independientes y utilice el modelo para predecir el valor de la variable dependiente.

Paso 9. Reflexiones sobre la experiencia (esta actividad no requiere instrucciones Python, pero es importante comentarla en el video)

- ¿Qué fue lo más complicado?
- ¿Cómo se resolvió?
- ¿Qué se aprendió?

Problema 2. Modelos de Clasificación: Árbol de Decisión (Decision Tree)

Contexto: Predicción de la aprobación de una solicitud de préstamo

En este caso de estudio, utilizaremos datos relacionados con históricos de solicitudes de préstamos, como por ejemplo el ingreso del solicitante, el monto del préstamo, el historial crediticio, etc., para predecir si una solicitud de préstamo será aprobada o no.

Se tienen las siguientes características o variables predictoras:

- Gender: Género del solicitante (Masculino/Femenino).
- Married: Estado civil del solicitante (Casado/Soltero).
- Dependents: Número de dependientes.
- Education: Nivel de educación del solicitante (Graduado/No Graduado).
- Self_Employed: Si el solicitante es autónomo o no.
- ApplicantIncome: Ingreso del solicitante.
- CoapplicantIncome: Ingreso del co-solicitante.
- LoanAmount: Monto del préstamo solicitado.
- Loan_Amount_Term: Plazo del préstamo en meses.
- Credit_History: Historial crediticio del solicitante (1 - Sí, 0 - No).
- Property_Area: Área de propiedad del solicitante (Rural, Semiurban, Urbano).

En este caso la etiqueta o variable objetivo es la siguiente:

- Loan_Status: Estado de aprobación del préstamo (Y - Sí, N - No).

La base de datos y la información sobre las variables involucradas se puede conseguir en el siguiente enlace:
https://www.kaggle.com/datasets/devzohaib/eligibility-prediction-for-loan/data

También se publicó el archivo de la base de datos “Loan_Data.csv” en la sección de Evaluaciones\Solemne 2.

Requerimientos:

Se requiere crear un modelo de clasificación, basado en un árbol de decisión, que sirva para la predicción de la variable indicada (aprobación de solicitud de préstamo).

Para esto se debe elaborar un notebook de Google Colab o un archivo .py donde se pueda codificar, ejecutar y explicar los resultados de las instrucciones del lenguaje Python que cubran las siguientes actividades:

Paso 1. Importar Bibliotecas y Cargar Datos

- Importar librerías Python.
- Cargar los datos en un dataframe.

Paso 2. Preparación de Datos:

- Determinar la cantidad total de registros y de variables (atributos).
- Determinar los tipos de datos por cada variable (atributo).
- Eliminar la columna 'Loan_ID'.
- Si se da el caso de datos faltantes en las variables numéricas sustituirlos por la media respectiva.
- Si se da el caso de datos faltantes en las variables categóricas sustituirlos por la moda respectiva.
- Convertir a valores numéricos los datos de las variables categóricas, es decir, numerizar las variables categóricas.

Paso 3. Separar los conjuntos de variables predictoras (características) y de la variable objetivo (etiqueta).

Paso 4. Dividir los datos en conjuntos de entrenamiento y prueba.

- Utilice una proporción 70% entrenamiento - 30% prueba.

Paso 5. Crear el modelo de árbol de decisión preliminar

- Utilizar los siguientes parámetros y argumentos para crear el modelo preliminar:
  - criterion="entropy"
  - max_depth=10
  - random_state=42
  - Es decir, se tendría la siguiente instrucción:

  modelo = DecisionTreeClassifier(criterion="entropy", max_depth = 10, random_state=42)

- En caso de que los valores de la variable objetivo estén desbalanceados, utilizar el siguiente parámetro y argumento:
  - class_weight='balanced'
  - Es decir, se tendría la siguiente instrucción:

  modelo = DecisionTreeClassifier(criterion="entropy",max_depth = 10,
  class_weight='balanced', random_state=42)

Paso 6. Evaluar el modelo de predicción preliminar

- Calcular e interpretar la matriz de confusión
- Calcular e interpretar la métrica Exactitud (Accuracy)
- Calcular e interpretar la métrica Precisión (Precision)
- Calcular e interpretar la métrica Sensibilidad (Recall)

Paso 7. Optimizar el modelo de árbol de decisión

- Determinar la profundidad máxima óptima.
- Crear el árbol de decisión con base al valor óptimo para el hiperparámetro profundidad máxima.

Paso 8. Evaluar el modelo de predicción optimizado generado en el paso 7

- Calcular e interpretar la matriz de confusión
- Calcular e interpretar la métrica Exactitud (Accuracy)
- Calcular e interpretar la métrica Precisión (Precision)

- Calcular e interpretar la métrica Sensibilidad (Recall)
- ¿Cuál de los dos modelos es mejor: el preliminar o el optimizado?

Paso 9. Visualizar el árbol de decisión.

Paso 10. Probar el modelo con un conjunto de datos.

- Crear un conjunto de datos para las variables predictoras y determinar si el modelo recomienda otorgar o no otorgar el préstamo al cliente.

Paso 11. Reflexiones sobre la experiencia (esta actividad no requiere instrucciones Python, pero es importante comentarla en el video)

- ¿Qué fue lo más complicado?
- ¿Cómo se resolvió?
- ¿Qué se aprendió?

Problema 3. Modelos de Clasificación: Red Neuronal (Neural Network)

Contexto: Predicción de enfermedad cardíaca

El objetivo, en este caso de estudio, es predecir si un paciente tiene una enfermedad cardíaca en función de varias características médicas. Este desafío se refiere a un problema de clasificación binaria porque el objetivo tiene dos posibles valores: presencia de enfermedad cardíaca (1) o ausencia de enfermedad cardíaca (0). Esta predicción es importante en el campo médico para identificar a los pacientes que están en riesgo de enfermedades cardíacas y para tomar medidas preventivas.

Se tienen las siguientes características o variables predictoras:

- age (Age of the patient in years)
- sex (Male/Female)
- cp chest pain type ([typical angina, atypical angina, non-anginal, asymptomatic])
- trestbps resting blood pressure (resting blood pressure (in mm Hg on admission to the hospital))
- chol (serum cholesterol in mg/dl)
- fbs (if fasting blood sugar > 120 mg/dl)
- restecg resting electrocardiographic results ([normal, stt abnormality, lv hypertrophy])
- thalach maximum heart rate achieved
- exang exercise-induced angina (True/ False)
- oldpeak ST depression induced by exercise relative to rest
- slope the slope of the peak exercise ST segment
- ca number of major vessels (0-3) colored by fluoroscopy
- thal [normal; fixed defect; reversible defect]

En este caso la etiqueta o variable objetivo es la siguiente:

- target the predicted attribute

La base de datos y la información sobre las variables involucradas se puede conseguir en el siguiente enlace:
https://www.kaggle.com/datasets/thisishusseinali/uci-heart-disease-data?select=heart_disease_data.csv

También se publicó el archivo de la base de datos “heart_disease_data.csv” en la sección de Evaluaciones\Solemne 2.

Requerimientos:

Se requiere crear un modelo de clasificación, basado en una red neuronal, que sirva para la predicción de la variable indicada (presencia de enfermedad cardíaca).

Para esto se debe elaborar un notebook de Google Colab o un archivo .py donde se pueda codificar, ejecutar y explicar los resultados de las instrucciones del lenguaje Python que cubran las siguientes actividades:

Paso 1: Importar Bibliotecas y Cargar Datos

Paso 2: Preparación de Datos (Preprocesamiento)

- Manejar datos faltantes para características (variables predictoras) numéricas y categóricas.
- Numerizar características (variables predictoras) categóricas.

Paso 3. Separar los conjuntos de variables predictoras (características) y de la variable objetivo (etiqueta).

- De ser necesario, estandarizar las características (variables predictoras) numéricas.

Paso 4. Dividir los datos en conjuntos de entrenamiento y prueba.

Paso 5: Construir el modelo óptimo de red neuronal utilizando validación cruzada.

Paso 6: Generar el gráfico de la mejor performance de la red neuronal.

Paso 7: Evaluar el modelo con las métricas pertinentes.

Paso 8: Realizar predicciones con nuevos datos.

Paso 9. Reflexiones sobre la experiencia (esta actividad no requiere instrucciones Python, pero es importante comentarla en el video)

- ¿Qué fue lo más complicado?
- ¿Cómo se resolvió?
- ¿Qué se aprendió?

2. Requerimiento general: Presentación digital (ppt):

Cada grupo debe crear una presentación (PPT) que le permita explicar los resultados de sus soluciones para los problemas planteados.

La **estructura sugerida** para esta presentación es la siguiente:

- Slide 1: Logo de la Universidad, facultad, carrera, nombre de la asignatura, título de la presentación: “Aprendizaje Supervisado: Modelos de Regresión y Clasificación”, nombre de los integrantes del grupo de trabajo, ciudad, fecha.
- Slide 2: Redacte una breve reseña del contenido del trabajo, explicando en qué consiste el aprendizaje supervisado.
- Slide 3 y 4: Redacte una breve explicación de los algoritmos utilizados en cada una de las soluciones planteadas (Regresión Lineal Simple, Regresión Lineal Múltiple, Árbol de Decisión, Red Neuronal).
- Slide 5: Problema 1: defina el contexto, la variable dependiente y las variables independientes utilizadas, así como los nombres de los modelos predictivos creados (Regresión Lineal Simple y Regresión Lineal Múltiple)
- Slide 6: Gráfica de la matriz de correlación, basada en un mapa de calor, que justifique la selección de las variables independientes.
- Slide 7: Tabla de coeficientes de los modelos de regresión lineal simple y del modelo de regresión lineal múltiple.
- Slide 8: Tabla resumen de métricas de evaluación de los modelos de regresión.
- Slide 9: Problema 2: defina el contexto, la etiqueta (variable objetivo) y las características (variables predictoras) utilizadas, así como el nombre del modelo predictivo creado (Árbol de Decisión).
- Slide 10: Explicación de la validación cruzada utilizada en el problema 2.
- Slide 11: Tabla resumen de métricas de evaluación del modelo optimizado de árbol de decisión.
- Slide 12: Gráfico del modelo optimizado de árbol de decisión.
- Slide 13: Problema 3: defina el contexto, la etiqueta (variable objetivo) y las características (variables predictoras) utilizadas, así como el nombre del modelo predictivo creado (Red Neuronal).
- Slide 14: Explicación de la validación cruzada utilizada en el problema 3.
- Slide 15: Gráfico de la mejor performance de la red neuronal.
- Slide 16: Conclusiones problema 1: ¿qué fue lo más complicado?, ¿cómo se resolvió?, ¿qué se aprendió?
- Slide 17: Conclusiones problema 2: ¿qué fue lo más complicado?, ¿cómo se resolvió?, ¿qué se aprendió?

- Slide 18: Conclusiones problema 3: ¿qué fue lo más complicado?, ¿cómo se resolvió?, ¿qué se aprendió?
- Slide 19: Referencias bibliográficas.

Nota: de necesitar más slides puede incluirlas en la presentación.

3. Requerimiento general: Video

Cada grupo debe crear un video que muestre la explicación de las soluciones a cada uno de los problemas planteados. En este sentido, los integrantes del grupo se apoyarán tanto en la presentación digital como en los notebooks de Google Colab o archivos .py creados para cada problema.

Contenidos del Video:

- Presentación del equipo y del proyecto: señale el nombre del proyecto ("Aprendizaje Supervisado: Modelos de Regresión y Clasificación"), y los nombres de cada uno de los integrantes del grupo de trabajo.
- Explicación de la solución del problema 1 apoyándose en la presentación y/o en el notebook o archivo .py correspondiente.
- Explicación de la solución del problema 2 apoyándose en la presentación y/o en el notebook o archivo .py correspondiente.
- Explicación de la solución del problema 3 apoyándose en la presentación y/o en el notebook o archivo .py correspondiente.
- Conclusiones, apoyándose en la presentación digital.

Condiciones del Video:

- El video a desarrollar debe tener una duración mínima de 5 y máxima de 20 minutos.
- Todos los estudiantes del grupo deben participar en el video. Al momento de participar se les debe escuchar su voz y ver al menos el rostro. No olvide que su participación en el video debe aportar a la explicación del contenido.
- La calidad del video debe considerar el uso de imágenes claras y nítidas, sonido apropiado y contenidos que aporten claridad de la explicación. Además, debe estar organizado para ayudar a comprender los contenidos solicitados.
- El video puede ser desarrollado con las herramientas que al grupo de estudiantes les parezcan apropiadas.
- El grupo de estudiantes entregará un enlace al video para su revisión en línea, y éste deberá presentarse en un formato que no amerite una instalación de software especial.
- Se recomienda el uso de la plataforma Zoom institucional para la grabación de este video.

4. Entregable:

Carpeta comprimida con:

- Documento PDF con los nombres y RUT de los integrantes del grupo y el enlace al video explicativo de las soluciones a los problemas de la Solemne 2.
- Notebook de Google Colab o archivo .py con los códigos de la solución al Problema 1.
- Notebook de Google Colab o archivo .py con los códigos de la solución al Problema 2.
- Notebook de Google Colab o archivo .py con los códigos de la solución al Problema 3.
- Presentación digital (ppt).

La identificación de la carpeta comprimida debe ser la siguiente:
MD_Solemne_02_ApellidoIntegrante1_ApellidoIntegrante2_ ApellidoIntegrante3_...

Este entregable puede ser publicad por uno solo de los integrantes del grupo de trabajo.

5. Aspectos administrativos:

- Modalidad: en grupos de mínimo dos (2) y máximo cinco (5) estudiantes.
- Lugar: Sección de “Evaluaciones” del aula virtual habilitado para tal fin.
- Entregable: Carpeta comprimida.
- Fecha tope: domingo 02-06-2024, 23:59 horas.

Docente Ismael Moreno Flores
i.morenoflores@uandresbello.edu