Profesor: Dr. Oldemar Rodríguez Rojas
PF-1319 y PF-1320 Análisis de Datos II
Fecha de Entrega: Domingo 18 de septiembre a las 12 media noche
Instrucciones:

- La solución a cada tarea se debe subir en el aula virtual, no pueden ser enviadas por correo.
- Las tareas son estrictamente individuales.
- Tareas idénticas se les asignará cero puntos.
- Todas las tareas tienen el mismo valor en la nota final del curso.
- Las tareas se pueden entregar tarde, pero cada día de atraso tendrá un rebajo de 20 puntos.

Tarea Número 4

- **Pregunta 1:** [10 puntos] [**no usar** `Python`] Para la Tabla de Datos que se muestra seguidamente (la variable a predecir es `Tipo`):

Tipo	Color	Tamaño
1	Amarillo	Grande
1	Amarillo	Grande
0	Amarillo	Pequeño
1	Azul	Pequeño
0	Azul	Grande
0	Azul	Grande
0	Azul	Pequeño
1	Amarillo	Pequeño
0	Azul	Grande
1	Azul	Grande
1	Azul	Grande

  1. Calcule la información ganada usando el índice Gini para las 2 posibles divisiones (iniciando con la variable Color o iniciando con la variable Tamaño) ¿Cuál división es la mejor? ¿Por qué?
  2. De acuerdo al resultado del ítem anterior, genere el árbol que representa todas las reglas de decisión. En caso de nodos hoja donde exista un empate en las clases de la variable a predecir puede clasificar como usted desee. Puede utilizar cualquier herramienta para dibujar el árbol, inclusive puede hacerlo a mano. El dibujo debe ser completamente legible.

- **Pregunta 2:** [30 puntos] [**no usar** `Python`] Supongamos que tenemos un Árbol de Decisión con tres clases A, B, C. Se tiene que decidir cómo dividir el nodo padre:

$$N = \begin{pmatrix} A & 100 \\ B & 50 \\ C & 60 \end{pmatrix}$$

para esto hay dos posibles divisiones. La primera posible división N_1 divide el nodo N en los 2 siguientes nodos:

$$N_{1,1} = \begin{pmatrix} A & 62 \\ B & 8 \\ C & 0 \end{pmatrix}, \quad N_{1,2} = \begin{pmatrix} A & 38 \\ B & 42 \\ C & 60 \end{pmatrix}.$$

La segunda opción de división N_2 para el nodo N es la siguiente en 3 nodos:

$$N_{2,1} = \begin{pmatrix} A & 65 \\ B & 20 \\ C & 0 \end{pmatrix}, \quad N_{2,2} = \begin{pmatrix} A & 21 \\ B & 19 \\ C & 20 \end{pmatrix}, \quad N_{2,3} = \begin{pmatrix} A & 14 \\ B & 11 \\ C & 40 \end{pmatrix}.$$

1. Calcule la información ganada usando el índice de Gini para las dos posibles divisiones (N_1 en 2 nodos y N_2 en 3 nodos). ¿Cuál división es la mejor?
2. Otro índice utilizado para decidir cuál división es la mejor es la Complejidad (que también es utilizado como criterio en la poda de un árbol). Dado un índice Q que puede ser Gini, se define la Complejidad de un árbol T con nodos terminales $(t_j)_{1\leq j\leq m}$ (m = cantidad de nodos terminales) como sigue:

   $$C_\alpha(T) = \sum_{j=1}^{m} n_j Q(t_j) + \alpha m,$$

   con $\alpha \geq 0$ y donde n_j es la cardinalidad del nodo t_j. Dado un árbol grande T_L y si $T \leq T_L$ denota que T es un subárbol de T_L, entonces se define el Árbol Óptimo como sigue:

   $$\hat{T}_\alpha = \min_{T\leq T_L} C_\alpha(T).$$

   Si usamos como parámetro para la complejidad $\alpha = 25$ para cada nodo terminal y usando el índice de Gini para las dos posibles divisiones (N_1 en 2 nodos o N_2 en 3 nodos). ¿Cuál división es la mejor en el sentido de que minimiza la complejidad? ¿Para que valores de α este criterio prefiere N_1 sobre N_2?

- **Pregunta 3:** [30 puntos] Realice las siguientes demostraciones que quedaron pendientes en la presentación teórica de Árboles de Decisión:

1. (Filmina 19) Pruebe que $\Delta\text{Imp}(v) \geq 0$ y que $\Delta\text{Imp}(v) = 0$ si y solo si $p(s|v) = p(s|v_i) = p(s|v_d)$ para $s = 1, \ldots, r$.
2. (Filmina 20) Sea la función $g : [0,1]^r \longrightarrow [0, \infty[$ definida por $g(x_1, \ldots, x_r) = \sum_{i\neq j}^{r} x_i x_j$, con $\sum_{i=1}^{r} x_i = 1$. Pruebe que la función g es una función de impureza (pruebe únicamente las propiedades a) y c).
3. (Filminas 22 y 23) Sea v un nodo de $A_{\max}$; v_i y v_d sus hijos izquierdo y derecho respectivamente, $n_d = |v_d|$, $n_i = |v_i|$, $n_{sd} = |E_s \cap v_d|$, $n_{si} = |E_s \cap v_i|$, $p_i = \frac{|v_i|}{|v|}$ y $p_d = \frac{|v_d|}{|v|}$. Pruebe que:

a) $\Delta\text{Imp}(v) = \dfrac{1}{|v|^2 n_i n_d} \displaystyle\sum_{s=1}^{r} (n_d n_{si} - n_i n_{sd})^2$.

b) El nodo v es puro $\Longleftrightarrow \exists\, h \in \{1, \ldots, r\}$ tal que $v \subseteq E_h$.

4. (Filmina 29) Si un objeto que es seleccionado al azar, cae en el nodo v_t del árbol $A_{\max}$ y es clasificado en la clase a priori "i", el costo esperado (estimado) de mala clasificación, dado el nodo v_t, se define como:

$$c(v_t) = c(t) = \frac{1}{|v_t|} \min_{i=1,\ldots,r} \sum_{j=1}^{r} c(i|j)|E_j \cap v_t|.$$

Si se asumen costos unitarios, es decir, $c(i|j) = 1$ para todo $i \neq j$, pruebe que:

$$c(t) = 1 - \frac{1}{|v_t|} \max_{j=1,\ldots,r} |E_j \cap v_t|.$$

5. (Filmina 30)

a) Pruebe que $p(v_t)c(v_t) \geq p(v_i)c(v_i) + p(v_d)c(v_d)$.

b) ¿Es Falso o Verdadero? que la igualdad puede ocurrir aún cuando los nodos hijos v_i y v_d contengan una mezcla de objetos de distintas clases a priori.

6. (Filmina 31) El Costo-Complejidad de una rama A_v de $A_{\max}$ se define como:

$$c_\alpha(A_v) = \sum_{u_t \in \widetilde{A}_v} c_\alpha(u_t),$$

Pruebe que:

$$c_\alpha(A_v) = \alpha|\widetilde{A}_v| + \sum_{u_t \in \widetilde{A}_v} p(u_t)c(u_t),$$

7. (Filmina 33) Pruebe que un nodo v es preferido a la rama A_v si:

$$\alpha \geq \frac{C(v) - C(A_v)}{|\widetilde{A}_v| - 1}.$$

- **Pregunta 4:** [10 puntos] En este ejercicio vamos a usar la tabla de datos `raisin.csv`, que contiene es resultado de un sistema de visión artificial para distinguir entre dos variedades diferentes de pasas (Kecimen y Besni) cultivadas en Turquía. Estas imágenes se sometieron a varios pasos de preprocesamiento y se realizaron 7 operaciones de extracción de características morfológicas utilizando técnicas de procesamiento de imágenes.

  El conjunto de datos tiene 900 filas y 8 columnas las cuales se explican a continuación.

  - `Area`: El número de píxeles dentro de los límites de la pasa..
  - `MajorAxisLength`: La longitud del eje principal, que es la línea más larga que se puede dibujar en la pasa.
  - `MinorAxisLength`: La longitud del eje pequeño, que es la línea más corta que se puede dibujar en la pasa.

- `Eccentricityl`: Una medida de la excentricidad de la elipse, que tiene los mismos momentos que las pasas.
- `ConvexArea`: El número de píxeles de la capa convexa más pequeña de la región formada por la pasa.
- `Extent`: La proporción de la región formada por la pasa al total de píxeles en el cuadro delimitador.
- `Perimeter`: Mide el entorno calculando la distancia entre los límites de la pasa y los píxeles que la rodean.
- `Class`: Tipo de pasa Kecimen y Besni (Variable a predecir).

Realice lo siguiente:

1. Utilice el método de Árboles de Decisión en **Python** para generar un modelo predictivo para la tabla `raisin.csv` usando el 80 % de los datos para la tabla aprendizaje y un 20 % para la tabla testing. Use los parámetros por defecto.
2. Calcule los índices de precisión e interpréte los resultados.
3. Explique el funcionamiento de los parámetros `max_depth` y `min_samples_split`. Genere un modelo de árboles de decisión variando los valores para estos parámetros e intente mejorar los resultados obtenidos del modelo anterior.
4. Grafique el árbol del mejor modelo generado anteriormente e interprete al menos dos reglas que se puedan extraer del mismo. Si es necesario pode el árbol para que las reglas sean legibles.
5. Construya un `DataFrame` que compare el modelo de árboles construido arriba con el mejor modelo construido en la tarea anterior para la tabla `raisin.csv`. Para esto en cada una de las filas debe aparecer un modelo predictivo y en las columnas aparezcan los índices *Precisión Global, Error Global, Precisión Positiva (PP) y Precisión Negativa (PN)*. ¿Cuál de los modelos es mejor para estos datos? Guarde los datos de este DataFrame, ya que se irá modificando en próximas tareas.

- **Pregunta 5:** [10 puntos] En este ejercicio usaremos la tabla de datos `abandono_clientes.csv`, que contiene los detalles de los clientes de un banco.

  La tabla contiene 11 columnas (variables), las cuales se explican a continuación.

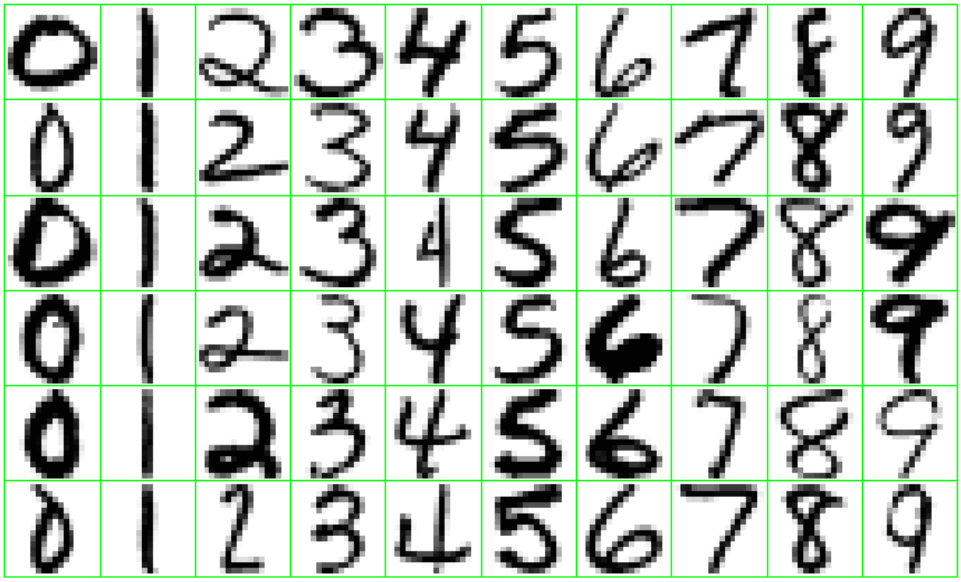
  - `CreditScore:` Indica el puntaje de crédito.
  - `Geography:` País al que pertenece.
  - `Gender:` Género del empleado.
  - `Age:` Edad del empleado.
  - `Tenure:` El tiempo del vínculo con la empresa.
  - `Balance:` La cantidad que les queda.
  - `NumOfProducts:` Los productos que posee.
  - `HasCrCard:` Tienen tarjeta de crédito o no.
  - `IsActiveMember:` Es un miembro activo o no.

- `EstimatedSalary`: Salario estimado.
- `Exited`: Indica si el cliente se queda o se va.

Realice lo siguiente:

1. Cargue en **Python** la tabla de datos `raisin.csv`.
2. Use el método de **Árboles de Decisión** en **Python** para generar un modelo predictivo para la tabla `abandono_clientes.csv` usando el 75 % de los datos para la tabla aprendizaje y un 25 % para la tabla testing.Use los mejores parámetros que pueda identificar.
3. Grafique el árbol generado e interprete al menos dos reglas que se puedan extraer del mismo. Si es necesario pode el árbol para que las reglas sean legibles.
4. Genere de nuevo un modelo predictivo con el método de Árboles de decisión pero esta vez utilice selección de 6 variables. ¿Mejora el resultado respecto al modelo generado con todas las variables?
5. Construya un `DataFrame` que compare el mejor modelo de árboles construido arriba con el mejor modelo construido en la tarea anterior para la tabla `abandono_clientes.csv`. Para esto en cada una de las filas debe aparecer un modelo predictivo y en las columnas aparezcan los índices *Precisión Global, Error Global, Precisión Positiva (PP) y Precisión Negativa (PN)*. ¿Cuál de los modelos es mejor para estos datos? Guarde los datos de este DataFrame, ya que se irá modificando en próximas tareas.

- **Pregunta 6:** [10 puntos] En este ejercicio vamos a predecir números escritos a mano (Hand Written Digit Recognition), la tabla de aprendizaje está en el archivo `ZipDataTrainCod.csv` y la tabla de testing está en el archivo `ZipDataTestCod.csv`. En la figura siguiente se ilustran los datos:



Los datos de este ejemplo vienen de los códigos postales escritos a mano en sobres del correo postal de EE.UU. Las imágenes son de 16×16 en escala de grises, cada pixel va de intensidad de -1 a 1 (de blanco a negro). Las imágenes se han normalizado para tener aproximadamente

el mismo tamaño y orientación. La tarea consiste en predecir, a partir de la matriz de 16×16 de intensidades de cada pixel, la identidad de cada imagen $(0, 1, \ldots, 9)$ de forma rápida y precisa. Si es lo suficientemente precisa, el algoritmo resultante se utiliza como parte de un procedimiento de selección automática para sobres. Este es un problema de clasificación para el cual la tasa de error debe mantenerse muy baja para evitar la mala dirección de correo. La columna 1 tiene la variable a predecir `Número` codificada como sigue: 0=‘cero’; 1=‘uno’; 2=‘dos’; 3=‘tres’; 4=‘cuatro’; 5=‘cinco’;6=‘seis’; 7=‘siete’; 8=‘ocho’ y 9=‘nueve’, las demás columnas son las variables predictivas, además cada fila de la tabla representa un bloque 16×16 por lo que la matriz tiene 256 variables predictoras.

1. Usando el método de Árboles de Decisión genere un modelo predictivo para la tabla de aprendizaje.
2. Con la tabla de testing calcule la matriz de confusión, la precisión global, el error global y la precisión en cada unos de los dígitos. ¿Son buenos los resultados?
3. Construya un `DataFrame` que compare el mejor modelo de árboles construido arriba con el mejor modelo construido en la tarea anterior para estos datos. Para esto en cada una de las filas debe aparecer un modelo predictivo y en las columnas aparezcan los índices *Precisión Global, Error Global, Precisión Positiva (PP) y Precisión Negativa (PN)*. ¿Cuál de los modelos es mejor para estos datos? Guarde los datos de este DataFrame, ya que se irá modificando en próximas tareas.