


 <b>Universidad Pontificia Bolivariana</b> MONTERÍA	Universidad Pontificia Bolivariana – Seccional Montería Escuela de Ingenierías y Arquitectura Programa de Ingeniería Electrónica Formato de Talleres Generales			
Fecha Actualización: 18 de febrero de 2022	Código: FTG-01	Página: 1 de 3	Versión: 1.0	

Nombre del Taller			
Decision Tree			
Curso(s)	Python aplicado a la Ingeniería		
HW No.	4	Tema(s)	Arboles de decisión
Resultados de Aprendizaje			
<ul style="list-style-type: none"><li>• <b>Comunicación:</b> comunico de manera asertiva los resultados de mis proyectos de forma oral, escrita y gráfica.</li><li>• <b>Resolución de Problemas:</b> identifico, formulo y resuelvo problemas de ingeniería Electrónica mediante la aplicación de los principios de ingeniería, ciencias y matemáticas</li></ul>			
Objetivo(s)			
<ul style="list-style-type: none"><li>• Formular algoritmos usando librerías especializadas con el lenguaje Python para la solución creativa de problemas cotidianos.</li><li>• Utilizar herramientas de programación (Spyder, GitHub Desktop) para el desarrollo de algoritmos utilizando el lenguaje Python.</li></ul>			
Descripción de la Actividad			
<div>1. Utilizar el set de datos <code>Carseats</code>, original del paquete de R <code>ISLR</code> y accesible en Python a través de <code>statsmodels.datasets.get_rdataset</code>, contiene información sobre la venta de sillas infantiles en 400 tiendas distintas. Para cada una de las 400 tiendas se han registrado 11 variables. Se pretende generar un modelo de clasificación que permita predecir si una tienda tiene ventas altas (<math>Sales &gt; 8</math>) o bajas (<math>Sales \leq 8</math>) en función de todas las variables disponibles.</div> <div>2. Para ello, por favor importar la librería</div> <div><pre>import statsmodels.api as sm</pre></div> <div>3. Utilizando los comandos a continuación, cree el DataFrame</div> <div><pre>carseats = sm.datasets.get_rdataset("Carseats", "ISLR") datos = carseats.data print(carseats.  doc  )</pre></div>			

 <b>Universidad Pontificia Bolivariana</b> <b>MONTERÍA</b>	<b>Universidad Pontificia Bolivariana – Seccional Montería</b> <b>Escuela de Ingenierías y Arquitectura</b> <b>Programa de Ingeniería Electrónica</b> <b>Formato de Talleres Generales</b>			
Fecha Actualización: 18 de febrero de 2022	Código: FTG-01	Página: 2 de 3	Versión: 1.0	

4. Como `Sales` es una variable continua y el objetivo del estudio es clasificar las tiendas según si venden mucho o poco, se debe crear una nueva variable dicotómica (0, 1) llamada `ventas_altas`.

```
datos['ventas_altas'] = np.where(datos.Sales > 8, 0, 1)
```



5. Una vez creada la nueva variable respuesta se descarta la original

```
datos = datos.drop(columns = 'Sales')
```

6. Divida sus datos en train / test  
7. Haga el árbol de decisión a partir de lo aprendido en clase.  
8. Realice predicciones con los datos de test

-----

Contenido			
Presentaciones.			
Entregable(s)			
Enlace hacia el repositorio en el que hospeda la solución al problema planteado. La solución es presentada en un archivo .py (spyder). La solución se encuentra identificada con el nombre, código, email institucional del estudiante.			
Bibliografía Recomendada			
Presentaciones de Curso (Ver Referencias en diapositivas)			
Criterios de Evaluación			
No.	Criterio	Descripción	%
1	Documentación y Legibilidad	La solución presentada se encuentra lo suficientemente documentada. Utiliza el lenguaje de escritura Spyder para comentar su solución. Los nombres de variables y recursos	20%

 <b>Universidad Pontificia Bolivariana</b> MONTERÍA	Universidad Pontificia Bolivariana – Seccional Montería Escuela de Ingenierías y Arquitectura Programa de Ingeniería Electrónica Formato de Talleres Generales			
Fecha Actualización: 18 de febrero de 2022	Código: FTG-01	Página: 3 de 3	Versión: 1.0	

		utilizados en el programa son descriptivos. Las operaciones realizadas son claras y sencillas.	
2	Estilo	El algoritmo desarrollado es creativo y eficiente.	20%
4	Solución	El algoritmo arroja los resultados esperados.	60%