

3ª Sessão prática

Objectivo:

Com esta aula prática pretende-se que o aluno se familiarize com a formulação de Bayes calculando uma árvore de classificação Bayesiana com base na situação apresentada. A resolução do problema deverá ser efetuada com base na linguagem Python.

Procedimento:

1 – Com base no ficheiro de dados fornecido pelo docente construa a tabela de probabilidades condicionais que lhe servirá de referência ao desenvolvimento do classificador Bayesiano que irá desenvolver nos passos seguintes. Deverá obter uma *dataframe pandas* com os dados contidos no ficheiro Excel.

	Problema	Fumo	Falta de força	Dificuldade em pegar
0	0	1	0	0
1	0	1	0	0
2	0	1	0	0
3	0	1	0	0
4	0	1	0	0
...
254	2	0	0	1
255	2	0	1	1
256	2	0	0	1
257	2	0	0	0
258	2	0	0	0

259 rows x 4 columns

Sugestão: utilizar a função `read_excel` da biblioteca Pandas

2 – Nos dados resultantes da leitura anterior a coluna *Problema* é uma string, o que poderá não dar muito jeito para o restante processamento. Transforme o seu conteúdo para valores numéricos fazendo as correspondências:

Injetores – 1

Pistons – 2

Problema Electrico – 3

Sugestão: fazer um *apply* sobre a coluna *Problema* que faça a transformação pretendida. Tenha em conta a possibilidade de repetir a transformação sobre esta coluna sem dar erro.

3 – Construa a tabela de probabilidades das avarias (Injetores, Pistons e Problema Electrico) e das probabilidades condicionadas dos sintomas (Fumo, Falta de força e Dificuldade em pegar). Deverá obter uma tabela com os seguintes valores:

Avaria	P(D)	P(Fumo Problema)	P(Falta de força Problema)	P(Dificuldade em pegar Problema)
Injetores	24,32%	77,78%	25,40%	31,75%
Pistons	41,70%	59,26%	92,59%	25,93%
Electrico	33,98%	20,45%	13,64%	78,41%

Sugestão: criar listas de valores (ou listas de listas) e calcular as diversas probabilidades de forma organizada por avaria ou/e por sintoma.

`Lista=[]` cria uma lista vazia

`Lista.append(value)` acrescenta um valor ao final da lista

`dataFrame[Coluna].sum()` dá a soma dos valores de uma coluna

`dataFrame[Coluna].count()` dá o número de valores de uma coluna

4 – Construir a árvore de decisão *Bayseana* que permite o diagnóstico *bayseano* considerando todos os sintomas e todas as possíveis avarias. Deverá chegar a uma árvore com três níveis (um por cada sintoma) e 8 folhas contendo cada nó e cada folha a probabilidade de cada uma das avarias.

Sugestões:

- a) Criar uma função que recebendo a lista das probabilidades condicionais e a lista das probabilidade das avarias calcule a probabilidade de uma dada avaria dado um sintoma
- b) utilizar a capacidade de processamento de listas do Python para criar listas de probabilidades das avarias dados os sintomas:

`Lista=[func(x) for x in range(10)]` cria uma lista com os valores de `func(x)` variando `x` entre 0 e 9

5 – Verifique que os resultados finais (terceiro nível) não dependem da ordem dos sintomas.

6 – Com base na árvore anteriormente calculada e sabendo que o perito questiona sempre os diversos sintomas na ordem Fumo-Falta de força-Dificuldade em pegar e que toma uma decisão sempre que umas das hipóteses de avaria excede os 80%, responda às seguintes questões:

- considerando todas as possíveis situações de avarias e sintomas igualmente prováveis, qual o número de perguntas esperado para obter um diagnóstico?
- nas mesmas condições, qual a probabilidade de o perito não chegar a um diagnóstico conclusivo?
- as respostas anteriores dependem da ordem pela qual são interrogados os sintomas?