Lista 1 - Mineração de Texto

1. a) TF (Term Frequency – Frequência do termo) é uma medida de quantas vezes um termo aparece em um documento. A forma mais simples de calcular é utilizando a seguinte fórmula:

$$TF(t,d) = \frac{f_{t,d}}{\sum_{k=1}^{n} f_{k,d}},$$

Em que t refere-se ao termo e d ao documento. $f_{t,d}$ indica o número de vezes que o termo t aparece no documento d e $\sum_{k=1}^{n} f_{k,d}$ o número total de termos no documento d.

Já o IDF (Inverse Document Frequency – Frequência inversa do documento) É uma medida de quanto um termo é importante dentro de toda a coleção de documentos (corpus). Sua fórmula é a seguinte:

$IDF(t,D) = log(\frac{N}{Df(t)})$, em que D refere-se aos documentos que formam o corpus, N é o número total de documentos no corpus e $Df(t)$ o número de documentos que contêm o termo t.

A relação entre TF e IDF é:

$$TF - IDF(t,d,D) = TF(t,d)XIDF(t,D)$$

Seja a seguir o seguinte conjunto de documentos:

Corpus = {
d1: "o gato está no telhado",
d2: "o cachorro está no quintal",
d3: "o gato e o cachorro brincam juntos"
}

Calcule o TF, em seguida o IDF e por fim a relação TF-IDF, sem o uso de algoritmo.
R:

b) Refaça o exercício e elabore um algoritmo em python para resolvê-lo, depois compare os resultados.
R:

2. Seja o modelo Bag of Words (BoW) representa um documento como um vetor de frequências dos termos de um vocabulário.
   A fórmula do vetor BoW é:

$$\overline{d} = (f_{t_1,d}, f_{t_2,d}, f_{t_3,d}, \ldots, f_{t_n,d}) .$$

   Em que, $\overline{d}$ é o vetor que representa o documento d, t_i representa o termo i do vocabulário, $f_{t_i,d}$ a frequência (número de ocorrências) do termo t_i no documento d e n o tamanho do vocabulário. Seja cospus a seguir:

   Corpus = {
   d1**:** "o gato está na casa",
   d2**:** "o cachorro está no quintal",
   d3**:** "o gato e o cachorro brincam”
   }

   Elabore vetores que representem a frequência de cada termo em todos os documentos do corpus. Exemplo:

   Corpus = {
   d1:“O gato no telhado”,
   d2:“O gato em casa”,
   d3:”O rato mora na casa”
   }
   Resultado:

Termos	**d1**	**d2**	**d3**
O	1	1	1
gato	1	1	0
no	1	0	0
telhado	1	0	0
em	0	1	0
casa	0	1	1
mora	0	0	1

na	0	0	1

Logo, d1 = (1,1,1,1,0,0,0,0), d2 = (1,1,0,0,1,1,0,0) e d3 = (1,0,0,0,0,1,1,1).
R:

3. O cosseno da similaridade é uma métrica muito utilizada em mineração de textos, aprendizado de máquina e sistemas de recomendação. Ele mede o grau de similaridade entre dois vetores ao calcular o cosseno do ângulo entre eles.

   Em vez de comparar diretamente os valores absolutos dos vetores, o cosseno da similaridade avalia o quanto eles apontam para a mesma direção no espaço vetorial. Isso é útil porque dois documentos podem ter comprimentos diferentes (quantidade de palavras distintas), mas ainda assim serem muito semelhantes em conteúdo. Sua fórmula é dada por:

   $$cos(\theta) = \frac{A.B}{||A||\,||B||}$$

   Considere agora a questão 2, a partir do documento de consulta (dc:"O cachorro brinca no quintal" e dos resultados obtidos, calcule o BoW para o documento de consulta em relação ao corpus e aplique o cosseno da similaridade, por fim crie um ranking com os mais similares ao documento de consulta em ordem descrecente.
   R:

4. Considere as questões 2 e 3 e refaça usando um algoritmo em python. Sugestão: Use o paradigma funcional.
   R:

5. Refaça a questão 4 aplicando a tokenização com a biblioteca nltk, retire os stopwords e aplique a stemmização e a lematização em outra versão do algoritmo. Compare as duas versões entre si e com os resultados anteriores. Realize uma discussão dos resultados.
   R:

6. O Perceptron foi o primeiro modelo formal de rede neural, proposto por Frank Rosenblatt em 1958. Ele é composto apenas por uma camada de neurônios de saída conectados diretamente às entradas.

   Estrutura:

   Entradas: $x_1, x_2, \ldots, x_n$

   Pesos: $w_1, w, \ldots, w_n$

   Soma ponderada:

$z = \sum_{i=1}^{n} x_i . w_i + b,$

Em que o b é o bias, usado para ajustar a rede.
Função de ativação: degrau, sigmóide.
Elabore o desenho da arquitetura da rede neural utilizando grafos.

7. Considere a questão 5 e crie os seguintes algoritmos em python:
    a) Perceptron de camada única com função de ativação degrau.
    b) Perceptron de camada única com função de ativação sigmóide.

8. Para superar as limitações do Perceptron simples, foram desenvolvidas as redes neurais de múltiplas camadas (MLPs). Elas possuem:
    - Camada de entrada (recebe os dados)
    - Camadas ocultas (processam as informações)
    - Camada de saída (gera o resultado final)

    Funcionamento:

    Cada neurônio em uma camada aplica a mesma ideia do Perceptron (pesos + soma + função de ativação), mas a saída de uma camada serve como entrada para a próxima.

    A fórmula geral para um neurônio em uma MLP é:

    $$a^{(l)} = f\left(\sum_{i=1}^{n} x_i^{(l-1)} . w_i^{(l)} + b^{(l)}\right),$$

    Em que,
    $a^{(l)}$ = ativação (saída) do neurônio na camada l,
    $w_i^{(l)}$ = peso da conexão,
    $x_i^{(l-1)}$ = saída da camada anterior,
    f = função de ativação não linear (sigmóide, ReLU, tanh etc.).

    Logo, resolva o que se pede:
    Elabore o desenho da arquitetura da rede neural utilizando grafos contendo 3 camadas.

9. Considere a questão 7 e elabore os seguintes algoritmos em python:
    a) Perceptron de camada única com função de ativação sigmóide.
    b) Perceptron de camada única com função de ativação de tangente hiperbólica.
    c) Perceptron de camada única com função de ativação ReLU.
    d) Perceptron de camada única com função de ativação ReLU exponencial.

10. Seja o seguinte corpus:

Documento	Texto	Sentimento
d1	"O filme foi incrível, adorei cada cena."	Positivo
d2	"O produto chegou quebrado e me decepcionou."	Negativo
d3	"Excelente atendimento, fiquei muito satisfeito."	Positivo
d4	"A comida estava fria e sem sabor."	Negativo
d5	"Gostei bastante da qualidade e do design."	Positivo
d6	"O serviço é péssimo, nunca mais volto."	Negativo
d7	"Uma experiência maravilhosa, recomendo a todos."	Positivo
d8	"O aplicativo trava o tempo todo, é horrível."	Negativo
d9	"Achei o hotel confortável e bem localizado."	Positivo
d10	"O atendimento foi lento e desorganizado."	Negativo

Aplique o procedimento de limpeza e preparação dos textos, o treinamento da rede neural e realize a classificação de sentimentos dos textos.
R:

Formulário:

1. **Função degrau (Heaviside Step Function):**

   $$f(x) = \{0,\ x < 0\ e\ 1,\ se\ x \geq 0\}$$

   Usada no perceptron original, mas não é diferenciável (difícil para backpropagation).

2. **Função sigmóide (Logística):**

   $$f(x) = \frac{1}{1+e^{-x}}$$

   Intervalo: (0,1)
   Boa para probabilidades, mas sofre com desvanecimento do gradiente.

3. **Função Tangente Hiperbólica (tanh):**

   $$f(x) = tanh(x) = \frac{e^{x}-e^{-x}}{e^{x}+e^{-x}}$$

   Intervalo: (−1,1)
   É uma versão “escalada” da sigmoide, centrada em 0.

4. **Função ReLU (Rectified Linear Unit):**

   $$f(x) = \{0,\ se\ x < 0\ e\ x,\ se\ x \geq 0\}$$

   Simples, eficiente e evita saturação como na sigmoide/tanh.
   Problema: neurônios mortos quando x<0x < 0x<0 por muito tempo.

5. **Função ReLU exponencial:**

   $$f(x) = \left\{\alpha(e^{x} - 1),\ se\ x < 0\ e\ x,\ se\ x \geq 0\right\}$$

   Para x<0, suaviza a curva com exponencial.
   Reduz o problema dos neurônios mortos.

6. **Softmax:**
   Dada uma entrada vetorial,
   $z = [z_1, z_2, \ldots, z_n]$,

com K classes possíveis, a **Softmax** transforma esses valores em probabilidades:

$$S(z_i) = \frac{e^{z_i}}{\sum_{j=1}^{k} e^{z_j}}$$

Para $i = 1, 2, \ldots, k$