Universidade Federal do Ceará, Departamento de Estatística e Matemática Aplicada
Introdução à Mineração de Dados (CC0101), período 2019.2
Professor: Tibérius O. Bonates (`tb@ufc.br`).

Atividade 3 – Algoritmo de vizinhos mais próximos com esquema de pesos baseados em distância.

1 Descrição

Modificar, com base no material discutido em sala de aula e no livro-texto, o código Python para o algoritmo k-NN, fornecido via SIGAA (arquivo `knn.py` do dia 12/setembro/2019). A modificação a ser feita deve ser aquela que atribui um peso a cada um dos vizinhos mais próximos. A previsão do algoritmo é a classe de maior peso total dentre os k vizinhos mais próximos.

O peso a ser utilizado deve ser o inverso da distância a cada vizinho mais próximo (equação (1)).

Seja x o que exemplo que desejamos classificar e seja $V(x) = \{x^{i_1}, x^{i_2}, \ldots, x^{i_k}\}$ o conjunto dos k vizinhos mais próximos de x no conjunto de treinamento. Vamos definir $C(x)$ como o conjunto das classes que encontramos dentre os k vizinhos mais próximos de x, isto é, $C(x) = \bigcup_{j=1}^{k} \{y^{i_j}\}$. Esse conjunto $C(x)$ é o conjunto de classes "candidatas" a serem a classe prevista para x, já que são as classes existentes na vizinhança de x.

Seu código deve calcular, para cada classe em $c \in C(x)$, o peso total dos exemplos em $V(x)$ que são da classe c, isto é:

$$P(c) = \sum_{j=1}^{k} \left(\frac{1}{d(x, x^{i_j})} \right) \cdot I(c, y^{i_j}), \quad \forall c \in C(x), \tag{1}$$

onde $I(a, b) = 1$ se $a = b$, $I(a, b) = 0$ se $a \neq b$, e $d(x, x^{i_j})$ é a distância euclidiana entre x e x^{i_j}.

Perceba que há um problema com a equação (1): se o ponto x for igual a algum exemplo x^t do conjunto de treinamento, então temos uma divisão por zero e a fórmula deixa de fazer sentido. Para evitar este problema, se esta situação ocorrer, seu código deve prever a classe de x como sendo a classe do conjunto de treinamento que está associada ao exemplo x^t, isto é, y^t. Se, por acaso, existirem dois ou mais exemplos iguais a x no conjunto de treinamento, seu código deve prever a classe de x como sendo a classe mais frequente dentre estes exemplos iguais a x.

Certifique-se de que seu código funciona corretamente para diferentes valores de k, inclusive para k igual ao número de exemplos no conjunto de treinamento.

2 Entrega

- Pontuação máxima pela atividade: 1,5 (um ponto e meio).
- Entregáveis: código-fonte em Python.
- Data de entrega: 06 (seis) de outubro de 2019, 23:59:59, via SIGAA. **Não** via e-mail.

3 Mais detalhes

- A atividade pode ser feita individualmente ou em dupla. Os nomes dos participantes da equipe **devem estar no código-fonte**, e **não** em mensagem via SIGAA.
- Se for necessário submeter múltiplos arquivos, compacte-os em um único arquivo (ZIP).
- Lembre-se de utilizar a opção de "cadastrar grupo" no SIGAA para poder enviar seu material.