UFU CAMPUS MONTE CARMELO
Mineração de Dados
Trabalho 1

Anagê Calixto Mundim Filho

Maio 2019 - Monte Carmelo

1 Introdução

1. **Base de dados**
   A nossa base de dados consiste em um estudo feito no "National Institute of Diabetes and Digestive and Kidney Diseases", os dados foram recebidos dia 09/05/1990.

   Número total de instâncias: 768, sendo 500 casos negativos e 268 positivos. Serão 193 tuplas de teste e 577 de treino.
   Número de variáveis: 8 mais a classe.
   As variáveis consistem em:

   (a) Número de vezes grávida
   (b) Concentração plasmática de glicose
   (c) Pressão arterial diastólica (mm Hg)
   (d) Espessura da dobra da pele do tríceps (mm)
   (e) Insulina sérica de 2 horas (mu U / ml)
   (f) Índice de massa corporal (peso em kg / (altura em m)
   (g) Função de diabetes
   (h) Idade (anos)
   (i) Variável de classe (teste positivo ou teste negativo)

2. **Tratamento**
   O tratamento utilizado foi o Min-max, serve para colocar todos os dados na mesma grandeza. Foi utilizado em todos atributos tirando a classe.

3. **Medida de proximidade**
   A medida de proximidade usada para este trabalho foi a Distância Euclidiana, ela pega a distância entre os objetos, no caso a usamos para comparar cada tupla de teste com todas de treino, assim calculando todas as distâncias para pegar as K menores para aplicarmos o Knn.

4. **Algoritmo de classificação abordado.**
   O algoritmo de classificação Knn é o algoritmo usado, sendo K = 3 e K = 7, isso significa que pegaremos os 3 ou os 7 vizinhos mais proximos, e olhando a classe deles, pegando a classe que mais se repetiu entre esse range.

2 Desenvolvimento

1. **Características do algoritmo diante do problema:**
   Crio as funções que serão utilizadas, sendo elas:

   binariza classe: Serve para colocar as classes em 0 (tested-negative) e 1 (tested-positive) para ser possível usar as operações entre array.

   pega menor: Função para pegar o menor valor de uma coluna de uma array, utilizada para calcular o min max

   pega maior: Mesma ideia do pega-menor mas usada para pegar o maior valor, também usada no min-max.

   min max: Usada para normalizar as arrays para todos os valores ficarem no intervalo de 0 a 1.

   dist euclidiana: Função que calcula a distância entre 2 arrays, é usada no knn.

   knn: Aqui é onde fazemos as verificações das distâncias e ordenamos as 3 ou 7 menores para comparar e obter o resultado de qual classe pertence o teste.

   pega classe knn: Retorna a classe da posição da menor posição do vetor de treino, será usada para retornar as classes das distâncias para obter o resultado da acurácia.

   acuracia e erros: Retorna a acurácia e o total de erros.

   Depois ,li os arquivos txt, carregando os índices de teste, e treino.

Após isso é lido todos os dados da nossa base de dados numa array. Após

2. **Resultados obtidos em termos de acurácia:**
   Os resultados obtidos foram:

   (a) k=3

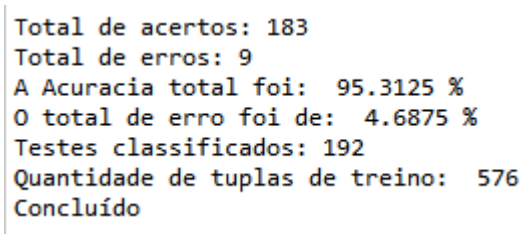

Figure 1: Resultado com o k = 3 no algoritmo knn

   (b) k=7

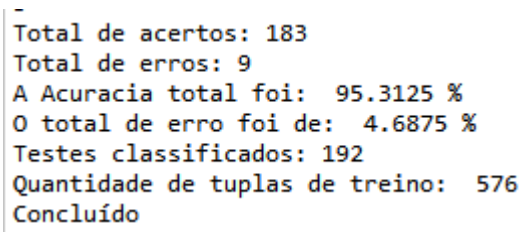

Figure 2: Resultado com o k = 7 no algoritmo knn

   Testei com k=5 e o resultado foi o mesmo do k=3 e k=7, creio que foi coincidência, pois observando o algoritmo ele trata os resultados sempre maior que a metade de k, isso faz ele sempre pegar a maioria.

3. **Estratégias para melhoria do desempenho**
   Estou aprendendo Python e com certeza este não é o código mais otimizado para o knn mas ele funciona, porém leva bastante tempo passando pela função do knn pois são muitas verificações, creio que se tivesse um melhor conhecimento sobre Python saberia usar funções para otimizar o código e os processos dentro dele.

   Após a conclusão do trabalho irei ver o que poderia ser melhorado no código para deixa-lo mais eficiente, como trabalhar com matrizes da maneira correta pelo Python, pois apliquei bastante

conhecimento que tinha em C e C++ neste trabalho pois ainda não conheço muito bem Python.

3 Conclusão

1. **Principais pontos:**A acurácia foi bem positiva com apenas aproximadamente 4 porcento de erro. A quantidade de tuplas de treino foi mais que o dobro de testes, creio que isso foi um ponto crucial para quantidade de acertos.

2. **Pontos que se destacaram para o aluno:**Um dos principais pontos foi o fato de aprender Python programando esse trabalho, sobre o projeto: Me assustou de começo a quantidade de dados e a primeira vez que o programa demorou mais de 1 minuto para compilar um código, precisei reduzir os dados para conseguir ver como estava fluindo os dados dentro do knn e conseguir concluir o código.