26022024_Card10_RELATORIO

K-Nearest_Neighbor

Em KNN, um objeto é classificado pelo "voto " de seus vizinhos, de acordo com as características que tem em comum. se K = 1(quantidade de vizinhos), o objeto é atribuído à classe desse vizinho. Variar o tamanho da amostra implica na variação da quantidade de vizinhos de diferentes classes que compartilham as mesmas características, alterando o resultado da classificação do objeto.

Limitações

- Se o dataset estiver desorganizado e não tratado, quando o classificador tentar classificar um novo elemento, ele tentará encontrar seus K-vizinhos mais próximoas mas o resultado terá baixa precisão, pois os dados estão imprecisos.
- Caso o novo elemento estiver muito distante dos data points, o KNN deixa de ser um algorítmo eficaz.
- não funciona bem com grandes datasets
- não funciona bem com um grande número de dimensões

Definindo_K

- Quanto menor de valor de K, menor a estabilidade da predição. Quanto maior o valor de K, mais estável é o resultado da predição.
- O valor de K deve ser ímpar, e o algoritmo testado várias vezes para encontrar o valor de k em que o modelo apresente a menor quantidade de erros, enquanto realiza uma predição satisfatória

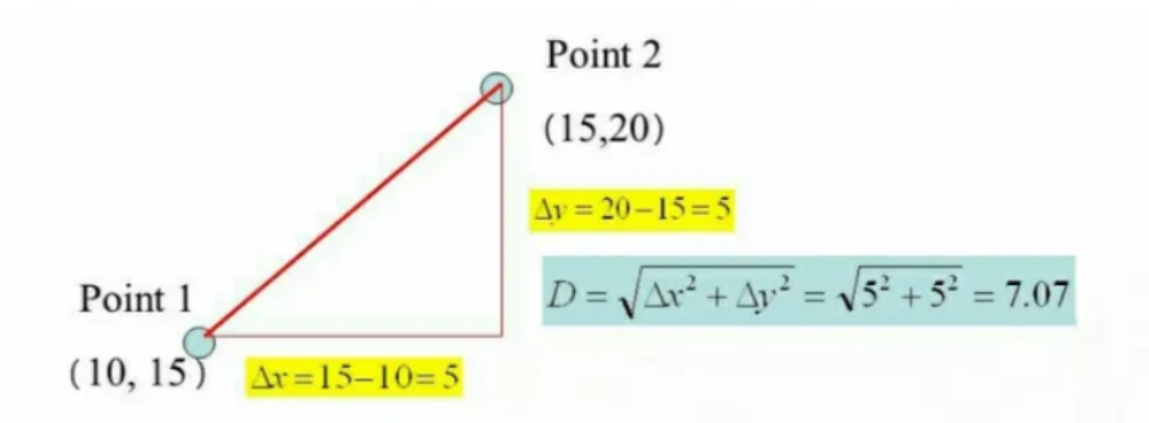
Distance_Metrics

Métricas de distância são ofator crucial para medir a distancia entre um novo dado e o conjunto de dados já treinados. Existem algumas métricas de distância que são utilizadas nos modelos de KNN:

Métrica_Euclidiana

- Mede a menor distância em linha reta entre dois pontos.
- Conhecida como L2 norm, utiliza um cálculo geométrico para medir a distância entre dois pontos bidimensionais A1(X1,Y1) e A2(X2,Y2).

$$\text{Euclidean distance} = \sqrt{\sum_{i=1}^{n} \left(x_i - y_i\right)^2}$$



Manhattan_Distance

- Mede a distância entre dois pontos em um espaço bidimensional. É a soma das diferenças absolutas entre as coordenadas dos dois pontos no plano. Manhattan Distance=|x1−x2|+|y1−y2|
- É muito utilizado em algoritmos de reconhecimento de fala e processamento de imagem em machine learning

Minowski_distance

Generalização das normas L1 e L2, é a distânia de ordem p entre dois pontos X=(x1, x2, x3,….,xn) e Y=(y1,y2,y3,…,yn)

$$D(\mathbf{x}_i, \mathbf{x}_j) = \left(\sum_{l=1}^{d} |x_{il} - x_{jl}|^{1/p} \right)^p .$$

Cosine_Distance_e_Cosine_Similarity

Dois pontos bidimensionais A1(x1,y1) e A2(x2,y2) formam vetores, e o coseno do ângulo entre esses vetores determina a sua similaridade.

- Quanto cosin = 1, a distância entre os pontos é mínima e a similaridade é máxima. o inverso ocorre quando cosin = 0.

Atividade: Como o algorítmo apresentado na videoaula pode ser melhorado?

- A função Cosine foi escolhida para o modelo, mas pode não ser a melhor para a quantidade de dados avaliados
- O valor de K utilizado foi arbitrário, sem nenhum teste para verificar se este é mesmo o valor que retorna a menor quantidade de erros.

Medidas para melhoria do código:

1. Utiliar K-Fold Cross Validation para encontrar o numero ótimo para o valor de K

Testando valores para K

K = 7: Liar Liar (1997) 3.156701030927835 Aladdin (1992) 3.8127853881278537 Willy Wonka and the Chocolate Factory (1971) 3.6319018404907975 Monty Python and the Holy Grail (1974) 4.0664556962025316 Full Monty, The (1997) 3.926984126984127 George of the Jungle (1997) 2.685185185185185 Beavis and Butt-head Do America (1996) 2.7884615384615383 Avg Rating = 3.4383535437685526

K=5: Liar Liar (1997) 3.156701030927835 Aladdin (1992) 3.8127853881278537 Willy Wonka and the Chocolate Factory (1971) 3.6319018404907975 Monty Python and the Holy Grail (1974) 4.0664556962025316 Full Monty, The (1997) 3.926984126984127 Avg Rating= 3.7189656165466287

K=3: Liar Liar (1997) 3.156701030927835 Aladdin (1992) 3.8127853881278537 Willy Wonka and the Chocolate Factory (1971) 3.6319018404907975 Avg Rating = 3.5337960865154954

aparentemente, alterar o valor de K altera apenas o valor médio das avaliações, mas os vizinhos mais próximos se mantém os mesmos, apenas limitados ao numero de K.

Dimensionality_Reduction

- **Objetivo:** Diminuir o número de características (dimensões) em um conjunto de dados enquanto preserva o máximo de informação possível.
- **Benefícios:**
  - Melhora a eficiência computacional (processamento e treinamento mais rápidos)
  - Reduz os requisitos de armazenamento
  - Pode melhorar o desempenho do modelo, evitando a "maldição da dimensionalidade"
- **Técnicas:**
  - Seleção de características: Subconjunto de características relevantes
  - Extração de características: Criação de novas características a partir de características existentes (por exemplo, PCA)

Análise de Componentes Principais (PCA):

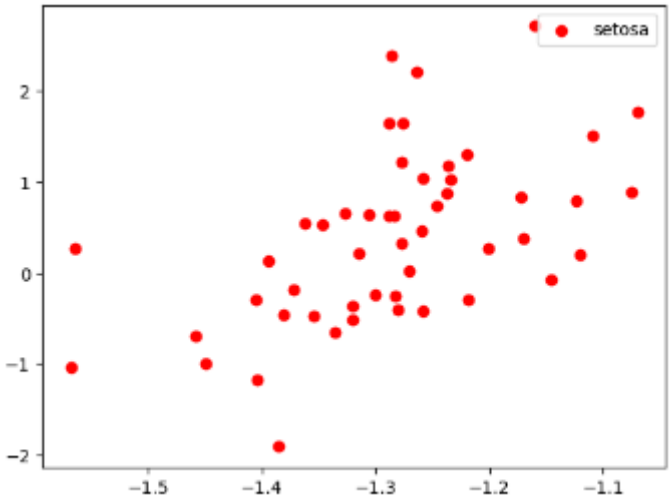
- **Redução de dimensionalidade linear:** Preserva apenas as relações lineares entre as características
- **Componentes:**
  - Componentes principais (PCs): Ordenados pela quantidade de variância que capturam

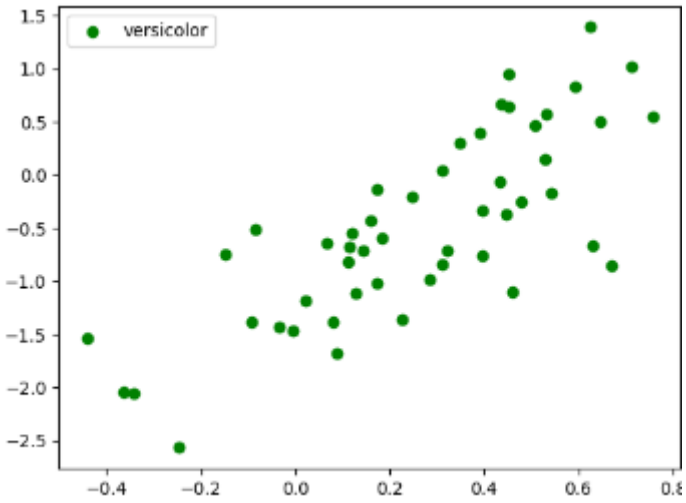
    - Cada PC é uma combinação linear das características originais
- **Aplicações:**
    - Visualização
    - Detecção de anomalias
    - Engenharia de features
    - Pré-processamento para vários modelos de aprendizado de máquina

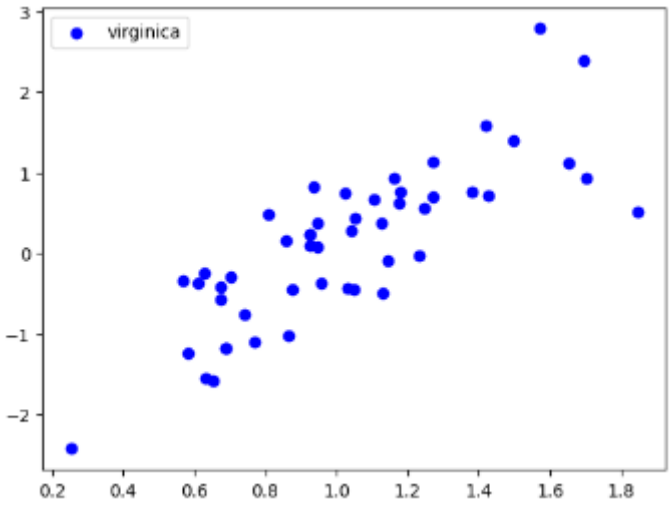
Algoritmo de PCA com IRIS dataset

Redução de 4 para 2 dimensões:

```
from sklearn.datasets import load_iris # importa a função loaf_iris
from sklearn.decomposition import PCA #importa a biblioteca PCA
import pylab as pl
from itertools import cycle

iris = load_iris()

numSamples, numFeatures = iris.data.shape

print(numSamples)
print(numFeatures)
print(list(iris.target_names))

X = iris.data
pca = PCA(n_components=2, whiten=True).fit(X)
X_pca = pca.transform(X)

print (pca.components_)

print(pca.explained_variance_ratio_)
print (sum(pca.explained_variance_ratio_))
pl.show()
```







O código acima pega a matriz 150 x 4 do iris_dataset, onde as 4 colunas representam características de 150 espécimes da flor iris. Em seguida, faz a redução das 4 dimensões que representam as características utilizando PCA, para 2 dimensões, a fim de possibilitar a plotagem dos dados. Este processo garante que a representação do dataset em duas dimensões mantenha a maior parte das informações mais

importantes contidas no dataset original, o tornando mais simples de se visualizar.

Data Warehouse

- Um grande banco de dados centralizado que contém informação sobre multiplas fontes
- Utilizado por grandes organizações para análises de negócios
- Manipulado por ferramentas como SQP ou Tableu
- Departamentos inteiros são deddicados a manter o banco de dados

ETL: Extract, Transform, Load

- ETL e ELT se refere a como os dados são importados para a warehouse. Etapas: 1. Dados brutos são importados dos sistemas operacionais 2. Transformação dos dados para a estrutura necessária 3. Os dados são carregados na warehouse, já com a estrutura necessária A etapa de transformação pode causar problemas quando se lida com uma grande quantidade de dados, e fdiferentes técnicas são utilizadas para lidar com esse problema, como a HIVE, Hadoop, armazenamento distribuído, e outras medidas para se garantir a escalabilidade do sistema.

Q-Learning

O Q-learning é um algoritmo de aprendizado por reforço que permite que um agente aprenda a tomar decisões em um ambiente sequencial para maximizar a recompensa a longo prazo. O agente interage com o ambiente, realizando ações e observando os novos estados e recompensas. A função Q mapeia estados-ações para valorese é atualizada a cada interação, levando em consideração a recompensa recebida, o fator de desconto e o valor máximo da função Q para o próximo estado. O processo se repete até que o agente tenha um resultado satisfatório.

Funcionamento:

1. **Inicialização:** A função Q é inicializada com valores arbitrários.

2. **Experiência:** O agente interage com o ambiente, realizando ações e observando os novos estados e recompensas.
3. **Atualização do valor de Q:** A função Q é atualizada para cada estado-ação executado em uma iteração do algoritmo, e indica as ações futuras com maior probabilidade de recompensa
4. **Vantagens:**

- Algoritmo simples e eficiente.
- Não requer um modelo do ambiente.
- Pode ser aplicado a uma ampla variedade de problemas.

Desvantagens:

- Pode ser lento para convergir em ambientes grandes e complexos.
- Pode ser sensível à escolha da taxa de aprendizado e do fator de desconto.

Aplicações:

- Robótica
- Jogos
- Controle de processos
- Finanças

Confusion Matrix

Uma matriz de confusão é uma ferramenta poderosa para avaliar o desempenho de modelos de classificação, especialmente em áreas como machine learning e inteligência artificial. Ela fornece uma visão abrangente da performance do modelo, ajudando a identificar pontos fortes e fracos.

- **Verdadeiros Positivos (VP):** Imagens de gatos corretamente classificadas como gatos. **Verdadeiros Negativos (VN):** Imagens de cachorros corretamente classificadas como cachorros.
- **Falsos Positivos (FP):** Imagens de cachorros incorretamente classificadas como gatos. **Falsos Negativos (FN):** Imagens de gatos

incorretamente classificadas como cachorros.

Com base nesses valores, é possível calcular diversas métricas, como:

- **Precisão:** Quantas das imagens que o modelo classificou como positivas realmente são positivas?
- **Revocação:** Quantas das imagens que realmente são positivas o modelo classificou como positivas?
- **Acurácia:** Qual a porcentagem geral de imagens que o modelo classificou corretamente? A matriz de confusão vai além da acurácia, revelando detalhes importantes sobre o desempenho do modelo. Ela ajuda a identificar erros de classificação em casos de um modelo com múltiplas classes, e os viéses do modelo.

Métricas de Avaliação em Classificação:

Precisão (Precision): Indica a proporção de previsões positivas que realmente são positivas. Em outras palavras, mede a exatidão das previsões positivas.

Revocação (Recall): Indica a proporção de instâncias positivas que foram corretamente identificadas pelo modelo. Mede a capacidade do modelo de encontrar todas as ocorrências positivas.

F1-Score: Uma medida que combina precisão e revocação. Dá igual peso aos dois, e é útil quando as classes do problema são balanceadas.

Curva ROC (Receiver Operating Characteristic): Um gráfico que mostra a relação entre a taxa de Verdadeiros Positivos (TPR) e a taxa de Falsos Positivos (FPR) em diferentes limiares de classificação. É útil para visualizar o desempenho do modelo em todo o espectro de possíveis classificações.

Área Sob a Curva ROC (AUC): Um valor entre 0 e 1 que representa a probabilidade do modelo classificar corretamente uma instância aleatória positiva versus uma negativa. Quanto mais próximo de 1, melhor o desempenho do modelo.