Lista 2 IA

Henrique Oliveira da Cunha Franco

Questão 4

O índice de Gini é uma métrica utilizada no algoritmo CART (Árvores de Decisão) para medir a impureza de um nó em termos de classificação. A ideia é quantificar o quão misturados os dados estão dentro de um nó. Um índice de Gini de 0 indica pureza máxima, ou seja, todos os exemplos em um nó pertencem à mesma classe. Quanto maior o valor de Gini, mais impuro ou misturado é o nó.

No contexto de árvores de decisão, o objetivo é criar divisões que minimizem a impureza de Gini, ou seja, que resultem em nós o mais puros possível.

Fórmula do Índice de Gini: Para um nó com k classes, o índice de Gini é dado por:

$$\text{Gini} = 1 - \sum_{i=1}^{k} p_i^2$$

Onde p_i é a proporção de instâncias da classe i no nó. A fórmula reflete a soma das probabilidades quadradas de cada classe, subtraída de 1.

Funcionamento no CART:

1. **Inicialização**: O CART começa com todos os dados em um único nó.
2. **Divisão**: Para cada possível divisão, o índice de Gini é calculado para os nós resultantes.
3. **Seleção da Melhor Divisão**: A divisão que minimiza a impureza de Gini é escolhida.
4. **Recursão**: O processo continua recursivamente, dividindo os nós até alcançar um critério de parada, como profundidade máxima ou número mínimo de exemplos por nó.

O uso do Gini visa criar uma árvore que separa bem as classes, maximizando a pureza em cada divisão.

Questão 5

Parte 1 - Balanceamento

O balanceamento de dados refere-se à análise da distribuição das instâncias de cada classe dentro de uma base de dados. Dizemos que um conjunto de dados está desbalanceado quando uma classe apresenta uma quantidade significativamente maior de instâncias em comparação às demais. Esse desbalanceamento pode ser prejudicial, especialmente para algoritmos de aprendizado de máquina, que costumam favorecer a classificação de novos dados como pertencentes à classe majoritária.

Para mitigar esse problema, existem três principais estratégias:

1. **Ajuste do tamanho do conjunto de dados:** Nesta abordagem, é possível aumentar a quantidade de instâncias da classe minoritária (*oversampling*) ou reduzir a da classe majoritária (*undersampling*). O *oversampling* pode introduzir o risco de gerar dados artificiais que não representam a realidade ou de provocar *overfitting*, situação em que o modelo se ajusta excessivamente aos dados de treinamento. Por outro lado, o *undersampling* pode acarretar na perda de informações valiosas, resultando em *underfitting*, ou seja, um desempenho inadequado do modelo ao classificar novos dados.

2. **Atribuição de diferentes custos de classificação:** Esta técnica atribui pesos maiores às penalidades por erros de classificação das classes minoritárias. Ao fazer isso, o modelo é ajustado de forma a tratar essas classes com mais atenção, equilibrando a tendência natural de favorecer a classe majoritária.

3. **Indução de modelos específicos para cada classe:** Nesta abordagem, são gerados dois modelos de classificação distintos: um para a classe majoritária e outro para a minoritária. Dessa forma, o problema de desbalanceamento é isolado, permitindo que cada modelo seja treinado de forma independente e adaptada a sua respectiva classe.

Parte 2 - Dados Ausentes

A ausência de dados em uma base pode ocorrer devido a diversos fatores, como falhas nos equipamentos de coleta, problemas na transmissão e armazenamento das informações, ou erros humanos no preenchimento dos dados. Dados ausentes representam um desafio para muitos modelos de classificação, que podem ter dificuldades em processá-los adequadamente.

Existem algumas abordagens comuns para lidar com esse problema:

1. **Eliminação das instâncias com dados ausentes:** Essa estratégia é utilizada, em geral, quando o atributo ausente é fundamental, como aquele que determina a classe da instância. No entanto, ela é desaconselhada em

cenários onde poucos atributos estão ausentes, quando a ausência de dados varia entre instâncias, ou quando a exclusão de instâncias resulta em um conjunto de dados reduzido demais.

2. **Preenchimento manual de valores ausentes:** Apesar de eficaz, essa abordagem pode ser inviável quando há um grande volume de dados ausentes, tornando o processo manual impraticável.

Parte 3 - Dados Inconsistentes e Redundantes

Dados inconsistentes são aqueles que apresentam valores conflitantes em seus atributos, frequentemente resultantes da integração de diferentes bases de dados. Já os dados redundantes podem se manifestar de duas maneiras: como instâncias redundantes ou atributos redundantes.

1. **Instâncias redundantes:** São registros duplicados ou excessivamente semelhantes dentro de uma mesma base de dados.

2. **Atributos redundantes:** Refere-se a atributos que podem ser inferidos a partir de outros já existentes, tornando-os desnecessários.

A redundância pode surgir devido a falhas nos processos de coleta, entrada, armazenamento, integração ou transmissão de dados.

Parte 4 - Conversão Simbólica-Numérica

Algoritmos de aprendizado de máquina que trabalham exclusivamente com dados numéricos requerem a conversão de atributos nominais para valores numéricos. Essa conversão pode ser realizada de diferentes formas, dependendo da natureza do atributo:

1. **Atributo binário:** Quando o atributo possui apenas duas possibilidades, como "sim" ou "não", basta atribuir valores como 1 e 0 para cada uma dessas opções.

2. **Atributo ordinal:** Neste caso, os valores são ordenados de forma lógica e convertidos para números inteiros, conforme sua posição relativa. Por exemplo, para uma coluna com os valores "primeiro", "segundo" e "terceiro", as correspondências seriam 0, 1 e 2, respectivamente.

3. **Atributo nominal com poucas categorias:** Quando o atributo nominal possui um número reduzido de categorias, é possível realizar a binarização. Cada categoria torna-se uma nova coluna e, para cada instância, a coluna correspondente ao valor presente será marcada com 1, enquanto as demais serão 0.

4. **Atributo nominal com muitas categorias:** Quando o atributo possui muitas possibilidades, como o nome de um país, é preferível uma codificação mais eficiente. Em vez de criar centenas de colunas para representar todas as possibilidades, pode-se adotar uma abordagem que atribua valores baseados em características específicas do atributo, otimizando o processo de conversão.

Parte 5 - Conversão Numérico-Simbólica

Algoritmos de aprendizado de máquina que trabalham exclusivamente com dados qualitativos exigem a conversão de atributos numéricos em categorias simbólicas. Para isso, utiliza-se a técnica de discretização, que consiste em dividir os valores numéricos em diferentes intervalos, possibilitando sua utilização em classificações.

A discretização pode ser realizada de forma supervisionada ou não supervisionada. Em geral, a discretização supervisionada tende a produzir melhores resultados, pois permite a definição de intervalos considerando informações externas à base de dados.

Existem várias abordagens para a discretização dos dados, como a criação de intervalos de tamanho uniforme ou de intervalos contendo o mesmo número de elementos. Além disso, pode-se realizar a escolha dos pontos de corte através de uma análise da entropia, maximizando a eficiência dos intervalos para a tarefa de classificação.

Parte 6 - Transformação de Atributos Numéricos

Em alguns casos, os valores numéricos podem estar representados de maneira que dificulte sua interpretação por algoritmos de aprendizado de máquina. Para contornar esse problema, é possível realizar transformações nos dados numéricos da base.

A principal técnica de transformação é a normalização, cujo objetivo é reduzir a amplitude entre os valores máximos e mínimos, evitando que certos valores tenham influência desproporcional. Existem duas formas comuns de normalização: por reescala e por padronização.

1. **Normalização por reescala:** Esta técnica estabelece novos limites mínimos e máximos para os atributos numéricos, recalculando cada valor com base nesses limites, o que permite que todos os valores se ajustem a um intervalo predefinido.

2. **Normalização por padronização:** Nessa abordagem, os valores não possuem limites mínimos ou máximos definidos, mas a distância entre eles é ajustada. A padronização transforma o conjunto de valores para que tenha média 0 e desvio padrão 1, garantindo uma distribuição mais uniforme dos dados.

Parte 7 - Redução de Dimensionalidade

A maldição da dimensionalidade é um problema que ocorre quando a adição de novos atributos a uma base de dados aumenta exponencialmente o número de combinações possíveis de instâncias, tornando o processamento e a análise dos dados significativamente mais complexos. Para mitigar esse efeito, é possível combinar ou eliminar atributos irrelevantes, o que melhora o desempenho dos modelos de aprendizado de máquina (ML) e facilita a interpretação dos resultados.

Existem duas principais técnicas de redução de dimensionalidade:

1. **Agregação de atributos:** Nesta técnica, novos atributos são gerados pela combinação de atributos existentes. Embora eficaz, pode resultar na perda de informações importantes, dependendo do contexto.

2. **Seleção de atributos:** A seleção de atributos consiste em manter apenas os atributos mais relevantes, descartando os demais. Entre suas vantagens estão a redução de custos, melhor organização, eliminação de ruído, simplificação dos dados e maior facilidade de visualização.

Para avaliar a qualidade e o desempenho de um subconjunto de atributos, existem três abordagens principais:

1. **Método embutido:** O próprio algoritmo de aprendizado de máquina incorpora um mecanismo interno para realizar a seleção de atributos durante o treinamento.

2. **Método baseado em filtro:** Utiliza-se um filtro no pré-processamento para selecionar os atributos relevantes, de maneira independente do algoritmo de ML. Suas vantagens incluem baixo custo computacional, eficiência com grandes volumes de dados e independência do algoritmo.

3. **Método baseado em *wrapper*:** Neste método, o algoritmo de ML é tratado como uma caixa-preta, e diferentes subconjuntos de atributos são testados para identificar qual proporciona o melhor desempenho. Embora simples e eficiente, essa abordagem pode ser computacionalmente custosa.