

Questão 01:

Coded Bias explora os impactos sociais da inteligência artificial, com foco nos vieses presentes na tecnologia de reconhecimento facial. O documentário acompanha a pesquisadora Joy Buolamwini, do MIT, que descobre que os algoritmos não identificam corretamente rostos negros e femininos devido à falta de diversidade nos dados usados para treiná-los.

A partir dessa constatação, o filme investiga como essas vieses afetam diferentes áreas, desde recrutamento e saúde a segurança pública. Ele destaca como sistemas automatizados podem perpetuar desigualdades e reforça a necessidade de regulação e ética no desenvolvimento dessas tecnologias.

O documentário também mostra exemplos de uso da IA ao redor do mundo, incluindo a vigilância na China e o reconhecimento facial em Londres. Ao longo da narrativa, especialistas e ativistas discutem os desafios e riscos da IA, defendendo uma abordagem mais transparente e inclusiva para evitar que essas ferramentas tecnológicas aprofundem discriminações já existentes.

Questão 02:

1) Classificação:

A classificação é um tipo de problema de aprendizado supervisionado onde o objetivo é prever uma categoria ou classe a partir dos dados de entrada. O modelo aprende a mapear as entradas para classes pré-definidas.

Exemplo: Um sistema de reconhecimento facial que identifica se uma pessoa está com máscara ou sem máscara em uma imagem.

2) Regressão:

A regressão também é um problema de aprendizado supervisionado, mas, ao invés de prever uma classe, o objetivo é estimar um valor numérico contínuo.

Exemplo: Previsão do consumo de energia elétrica em uma cidade com base na temperatura, hora do dia e histórico de consumo.

3) Agrupamento:

O agrupamento é um tipo de aprendizado não supervisionado, onde o modelo identifica padrões e separa os dados em grupos semelhantes sem rótulos pré-definidos.

Exemplo: Agrupamento de músicas semelhantes em um serviço de streaming para criar playlists automáticas baseadas no estilo musical dos usuários.

4) Associação:

A associação é um problema de aprendizado não supervisionado onde o objetivo é identificar padrões ou relações frequentes entre diferentes variáveis nos dados.

Exemplo: Em um site de compras, descobrir que clientes que compram celulares frequentemente compram capinhas e fones de ouvido, sugerindo esses produtos como recomendação.

Questão 03:

- 1) A árvore de decisão é gerada a partir da análise dos atributos disponíveis, buscando aquele que melhor separa os dados de acordo com o critério desejado. Esse processo envolve a medição da entropia e do ganho de informação de cada atributo, para escolher o mais relevante na tomada de decisão. O atributo na raiz da árvore é aquele que possui o maior ganho de informação, ou seja, aquele que melhor separa os dados em relação ao critério de decisão. Ele é o primeiro critério avaliado para classificar uma nova entrada.

- 2) Com uma árvore de decisão, você pode realizar classificações (separar os dados em categorias) e regressões (prever valores contínuos). Ela pode ser utilizada para inferir regras de decisão a partir dos dados, permitindo interpretações diretas e explicáveis sobre como as decisões são tomadas, também pode ser usada para

reduzir a dimensionalidade dos dados e identificar os atributos mais relevantes para um determinado problema.

3) Vantagens:

Simple de entender e interpretar, mesmo sem conhecimento técnico profundo. Pode lidar com dados categóricos e numéricos. Requer pouca preparação dos dados em comparação com outros algoritmos. Funciona bem com grandes conjuntos de dados e pode ser ajustada para evitar sobreajuste (overfitting) através de poda.

Desvantagens:

Pode gerar árvores muito grandes e complexas, dificultando a interpretação e a generalização. Sensível a pequenas variações nos dados, o que pode levar a diferentes árvores para o mesmo problema. Pode sofrer de sobreajuste se não for aplicada a poda corretamente. Não funciona bem para relações lineares simples, onde modelos como regressão linear podem ser mais eficientes.

- Código em Python para calcular o ganho de Informação e o nível da árvore

```
import numpy as np
import pandas as pd
from collections import Counter

class DataSheet:
    def __init__(self, file_path):
        self.df = pd.read_csv(file_path)
        self.columns_list = self.df.columns.tolist()

    def calculate_entropy(self, column_name):
        values_series = self.df[column_name]
        frequency_count = Counter(values_series)
        total_elements = len(values_series)
        entropy_value = 0.0

        for count in frequency_count.values():
            probability = count / total_elements
            entropy_value += probability * np.log2(probability)

        return -entropy_value

    def information_gain(self, column_name):
        total_rows = len(self.df)
        overall_entropy = self.calculate_entropy('Conclusao')
        value_counts = self.df[column_name].value_counts()
        conditional_entropy = 0.0

        for val in value_counts.index:
            subset = self.df[self.df[column_name] == val]
            proportion = len(subset) / total_rows
            conditional_entropy += proportion *
self.calculate_entropy('Conclusao')

        return overall_entropy - conditional_entropy

    def root_node(self):
        info_gains = {col: self.information_gain(col) for col in
self.columns_list}
        return max(info_gains, key=info_gains.get)

    def second_level_node(self):
        root = self.root_node()
        info_gains = {col: self.information_gain(col) for col in
self.columns_list}
        info_gains.pop(root)
        return max(info_gains, key=info_gains.get)

    def display_tree(self):
        root = self.root_node()
        second_level = self.second_level_node()
        print(f"Root Node: {root}")
        print(f"Second Level Node: {second_level}")

if __name__ == "__main__":
    sheet = DataSheet('restaurante.csv')
```

```
for col in sheet.columns_list:
    print(f"Information Gain ({col}):
{sheet.information_gain(col)}")
sheet.display_tree()
```

Resultado da Execução:

- Ganho de informação (Alternativo): 0,0
- Ganho de informação (Bar): 0,0
- Ganho de informação (SexSab): 0,0
- Ganho de informação (Fome): 0,0
- Ganho de informação (Cliente): $1,11 \times 10^{-16}$
- Ganho de informação (Preço): 0,0
- Ganho de informação (Chuva): 0,0
- Ganho de informação (Res): 0,0
- Ganho de informação (Tipo): $1,11 \times 10^{-16}$
- Ganho de informação (Tempo): $1,11 \times 10^{-16}$
- Ganho de informação (Conclusão): 0,0

Interpretação:

Com base nos cálculos realizados, o atributo selecionado como raiz da árvore de decisão é **Cliente**. Em seguida, o segundo atributo escolhido para o próximo nível é **Tipo**.