Métodos Probabilísticos
para Engenharia Informática
2024/2025

Deteção de fraude em transações bancárias usando o classificador de Naive Bayes, Filtros de Bloom e MinHash

Trabalho realizado por Gonçalo Simões (119412)

Índice

Introdução

O objetivo deste trabalho é desenvolver e analisar ferramentas para detetar e estudar transações fraudulentas de forma eficiente, utilizando uma combinação de métodos probabilísticos, como o filtro de Bloom, o MinHash e o classificador de Naive-Bayes. Estas abordagens permitem lidar com grandes volumes de dados, oferecendo rapidez, economia de recursos e precisão na identificação de fraudes.

O projeto foi organizado em diferentes etapas, que abrangem a criação de filtros para identificação rápida de fraudes conhecidas até à construção de um modelo probabilístico para prever fraudes desconhecidas. As etapas principais são:

1. **Criação do Filtro Bloom**: Implementamos um filtro probabilístico para armazenar transações fraudulentas conhecidas. Esta parte permite verificar de forma eficiente se uma nova transação é potencialmente fraudulenta, reduzindo o custo de processamento. Apesar da sua eficácia, o filtro está sujeito a falsos positivos, o que exige outras estratégias para complementar a análise.

2. **Divisão dos Dados**: Para garantir uma validação confiável, os dados foram divididos em 95% para treino e 5% para teste. Esta divisão é essencial para avaliar o desempenho do sistema em dados que o programa ainda não tomou conhecimento.

3. **Transformação em Vetores Binários**: As informações das transações (como comerciante e categoria) foram convertidas em vetores binários, facilitando a comparação entre elas e permitindo que diferentes modelos explorem essas características.

4. **Classificador Naive Bayes**: Com o conjunto de treino, desenvolvemos um classificador Naive Bayes. Este modelo é baseado na "ingenuidade" de que todas as características são independentes entre si. Apesar desta suposição simplificada, o Naive Bayes é bastante eficiente em cenários onde há padrões bem definidos. Ele foi usado para prever a probabilidade de uma nova transação ser fraudulenta, mesmo quando não há um histórico claro.

5. **Uso de MinHash para Similaridade**: Para identificar padrões e agrupar transações semelhantes, utilizamos o MinHash, que reduz a complexidade dos dados e calcula similaridades com base na distância de Jaccard.

6. **Análise de Padrões e Resultados**: Combinando o Naive Bayes e as assinaturas do MinHash, investigamos as transações para encontrar possíveis esquemas de fraude relacionados a comerciantes ou categorias específicas. Além disso, avaliamos a performance geral do sistema, medindo a precisão, taxas de falsos positivos e eficiência computacional.

Classificador Naive-Bayes

A primeira etapa do projeto foi implementada com base no classificador Naive Bayes, que é um método probabilístico baseado no teorema de Bayes.

O processo começa com o carregamento de um arquivo de dados chamado `data_table.csv`, que é lido em formato de tabela. Para garantir segurança na execução, o sistema verifica se o arquivo está disponível e emite uma mensagem de erro caso contrário. Após o carregamento, o pré-processamento é realizado para preparar os dados para o treinamento do

modelo. Durante essa etapa, colunas irrelevantes como informações de clientes, comerciantes e códigos postais são removidas, e colunas categóricas, como idade, gênero e categoria, são convertidas em valores numéricos usando a função `grp2idx`.

O sistema permite ainda que o usuário visualize as classes de cada coluna, caso deseje.

Após o pré-processamento, os dados são divididos em variáveis de entrada (X) e a variável alvo (y), que indica se a transação foi fraudulenta ou não. A seguir, é realizada a divisão dos dados em conjunto de treino e teste, utilizando 70% para o treino e 30% para o teste, por meio de uma partição aleatória. O modelo Naive Bayes é então ajustado aos dados de treino, aproveitando sua simplicidade e eficiência para lidar com classificações binárias. Uma vez treinado, o modelo é testado com o conjunto de testes para realizar previsões sobre as transações.

Os resultados são avaliados utilizando uma matriz de confusão, que fornece detalhes sobre o desempenho do modelo em termos de verdadeiros positivos, verdadeiros negativos, falsos positivos e falsos negativos. Esta matriz é visualizada graficamente e os valores individuais são apresentados ao usuário. Para uma avaliação detalhada, métricas como accuracy, precision, recall, F1, taxa de falsos positivos e taxa de falsos negativos são calculadas e exibidas. A accuracy mede o percentual de classificações corretas, enquanto a precisão e o recall avaliam a qualidade do modelo em identificar fraudes reais e evitar alarmes falsos. O F1 combina precisão e recall, enquanto as taxas de falsos positivos e falsos negativos ajudam a entender os erros cometidos pelo modelo.

Por fim é gerado um histograma para mostrar a distribuição das transações entre fraudulentas e não fraudulentas. Durante os testes, foram realizadas verificações para garantir a consistência do pré-processamento e a estabilidade dos resultados em diferentes rodadas de treinamento e teste. A matriz de confusão foi usada para ajustar o modelo e melhorar o seu desempenho. Por fim, o sistema demonstrou ser eficaz, permitindo a detecção de fraudes com uma abordagem estruturada e interativa.

Resumidamente, o classificador Naive-Bayes calcula rapidamente através da Regra de Bayes se uma transação é realmente fraude ou não.

Filtro de Bloom

Primeiro, os parâmetros do filtro são calculados para uma taxa de erro de 1%, e o filtro é inicializado ou carregado, caso já exista, junto com transações conhecidas salvas. O sistema verifica a integridade dos dados, remove linhas incompletas e divide os dados em 95% para treino e 5% para teste. IDs únicos das transações são gerados com base nos atributos da idade, do género e um indicador de fraude (`IF` ou `NF`). Apenas fraudes são adicionadas ao filtro, e os IDs são salvos para futuras verificações em dois ficheiros.

No teste, novos IDs são gerados aleatoriamente para avaliar o filtro. As transações conhecidas são corretamente identificadas, enquanto transações desconhecidas são analisadas como "não conhecidas" ou "possivelmente conhecidas" (falsos positivos). IDs também podem ser verificados manualmente, incluindo uma análise de similaridade para identificar fraudes potenciais.

Por fim, o desempenho é avaliado por métricas como verdadeiros positivos (TP), falsos negativos (FN), falsos positivos (FP) e verdadeiros negativos (TN). As taxas teóricas e empíricas de falsos positivos são exibidas, confirmando a eficácia do filtro Bloom para detectar fraudes.

MinHash

A principal finalidade do MinHash no projeto foi medir a similaridade entre transações com base nas suas características categóricas, como idade, género, categoria e indicador de fraude. A similaridade foi avaliada utilizando a distância de Jaccard e as assinaturas de MinHash foram utilizadas como uma aproximação eficiente.

Foi realizado o pré-processamento dos dados, estes foram extraídos e transformados em conjuntos de valores únicos, apenas foram utilizadas as categorias consideradas relevantes. Dados contínuos, como o valor da transação, foram normalizados para o intervalo de [0,1].

Após este primeiro passo, foi realizado o cálculo das assinaturas do MinHash, para cada transação é gerada uma assinatura MinHash, composta por um valor pré definido de Hash, neste caso 200. Estas assinaturas foram criadas através de funções Hash otimizadas, foram utilizadas duas para uma maior

robustez. O que permitiu reduzir conjuntos potencialmente grandes em representações compactas e comparáveis.

Numa terceira instância foi realizado o cálculo da similaridade, ou seja, a similaridade entre as duas transações foi avaliada comparando as assinaturas do MinHash. A proporção de valores idênticos entre as assinaturas foi usada como uma estimativa da similaridade de Jaccard.

Partindo dos valores calculados, o programa apresenta uma mensagem consoante o valor da distância obtido e caso ela seja pequena o suficiente, diz se as transações podem ser suspeitas caso uma delas seja fraude.

Como executar o programa principal?

Para executar o programa principal basta correr o programa main.m disponível no repositório, este programa compila os 3 módulos usados de uma só vez e busca automaticamente todos os ficheiros de dados usados e gerados pelas diferentes funções ao longo das diferentes implementações.