OdontoVision

Sabrina
Matheus Duarte Oliveira
Luis

OdontoVision: Detecção de Fraudes em Planos Odontológicos com IA - Documentação de Atualização

Introdução

O projeto OdontoVision busca implementar uma solução de **inteligência artificial** para a **detecção automática de fraudes** em sinistros odontológicos. Fraudes são práticas recorrentes e prejudiciais aos planos de saúde, onde procedimentos são reivindicados de maneira injustificada, elevando os custos e impactando negativamente a todos os clientes. O projeto visa utilizar técnicas de **machine learning** para identificar padrões de fraude de maneira precisa e automatizada, com a aplicação de um modelo baseado em **Random Forest** como o componente principal da solução.

Desde a primeira entrega, o projeto passou por diversas atualizações e melhorias, tanto na qualidade dos dados quanto nas técnicas de modelagem, o que resultou em uma arquitetura mais robusta e eficiente. Este documento descreve as diferenças entre a etapa inicial e o estado atual do projeto, detalha as ferramentas utilizadas e explica como os conceitos de IA estão sendo empregados na detecção de fraudes.

1. Diferenças entre a Etapa Atual e a Proposta Inicial

Na etapa inicial, o projeto focava em um protótipo de um modelo de machine learning simples, utilizando um dataset gerado artificialmente com uma proporção desequilibrada de fraudes e não fraudes. O modelo escolhido foi o **Random Forest**, implementado com uma configuração padrão para validar a viabilidade da abordagem.

Na etapa atual, o projeto evoluiu consideravelmente em termos de **sofisticação técnica e refinamento dos dados**. As principais mudanças são:

- **Balanceamento dos Dados**: No protótipo inicial, a desproporção entre as classes (fraude e não fraude) limitava a capacidade do modelo de aprender padrões de fraude. A nova abordagem utiliza o método **SMOTE (Synthetic Minority Over-sampling Technique)** para balancear o dataset, gerando amostras sintéticas para a classe minoritária (fraude). Isso permite que o modelo tenha uma representação mais equilibrada das duas classes e aprenda melhor os padrões de fraude.
- **Ajuste Fino de Hiperparâmetros**: Em vez de confiar em configurações padrão, o modelo agora passa por uma etapa de otimização de hiperparâmetros utilizando **GridSearchCV**, uma ferramenta poderosa para identificar a melhor combinação de parâmetros. Essa etapa é fundamental para que o Random Forest opere de forma ideal, maximizando a capacidade de detecção de fraudes.
- **Reponderação das Classes no Modelo**: Para garantir que o modelo valorize a detecção da classe de fraudes, a configuração `class_weight='balanced'` foi incluída no Random Forest. Isso permite que o modelo atribua maior peso aos exemplos de fraude, ajudando a reduzir falsos negativos.

Essas atualizações resultaram em um modelo mais eficiente e ajustado às necessidades do projeto, com melhor capacidade de detectar fraudes e menor propensão a ignorar casos fraudulentos.

2. Ferramentas e Bibliotecas Utilizadas no Desenvolvimento

Para alcançar os resultados atuais, o projeto se apoia em um conjunto robusto de bibliotecas Python e ferramentas de machine learning. Abaixo, detalhamos as principais:

- **Pandas**: Biblioteca essencial para a manipulação e análise de dados. Utilizada para carregar, explorar e preparar os dados antes de enviá-los para o modelo. O Pandas permite uma manipulação eficiente de tabelas e séries, facilitando a preparação de variáveis preditoras (features) e variáveis-alvo.
- **Numpy**: Complementa o Pandas com funcionalidades matemáticas e manipulação de arrays, ajudando a realizar operações de pré-processamento e manipulação de dados numéricos.
- **Scikit-learn**: Biblioteca central para o desenvolvimento do modelo de machine learning. A Scikit-learn oferece implementações robustas para:
    - **Random Forest**: Algoritmo de machine learning escolhido para a tarefa de classificação de fraudes.
    - **GridSearchCV**: Módulo de busca em grade que permite ajustar hiperparâmetros do Random Forest de maneira automática e eficiente.
    - **Divisão de Dados**: Funções como `train_test_split` permitem separar os dados em conjuntos de treinamento e teste, fundamentais para avaliar o desempenho do modelo.
    - **Métricas de Avaliação**: Ferramentas para calcular acurácia, precisão, recall, F1-score e gerar a matriz de confusão, permitindo uma análise detalhada do desempenho do modelo.
- **Imbalanced-learn (imblearn)**: Biblioteca especializada para lidar com datasets desbalanceados. Utilizada para aplicar o **SMOTE**, que cria amostras sintéticas da classe minoritária, ajudando o modelo a aprender melhor com exemplos de fraudes e reduzir o viés em favor da classe majoritária (não fraude).

- **Matplotlib e Seaborn**: Bibliotecas de visualização usadas para criar gráficos e analisar o desempenho do modelo. A matriz de confusão e os gráficos de importância de features são exemplos de visualizações que auxiliam na interpretação dos resultados.

3. Funcionamento dos Recursos/Ferramentas no Contexto do Projeto

Cada uma das bibliotecas acima contribui de maneira única para a implementação do modelo de detecção de fraudes. Abaixo, explicamos como elas se integram ao projeto:

- **Pandas e Numpy** são utilizadas nas etapas iniciais de pré-processamento, permitindo que a equipe manipule os dados com flexibilidade. A limpeza dos dados e o tratamento de valores ausentes, por exemplo, são realizados com o Pandas.
- **Imbalanced-learn (SMOTE)** é empregado após o pré-processamento e antes do treinamento. Com o SMOTE, criamos novas amostras de fraudes baseadas nas existentes, permitindo que o modelo tenha mais exemplos de fraude para aprender e, assim, aumentando a sensibilidade do modelo a casos fraudulentos.
- **Scikit-learn (Random Forest e GridSearchCV)**: O Random Forest é o algoritmo de machine learning central do projeto, e o GridSearchCV permite ajustar seus hiperparâmetros de forma otimizada. O processo é:
    - Dividir o dataset em treino e teste.
    - Aplicar o SMOTE ao conjunto de treinamento para balanceamento.
    - Usar o GridSearchCV para encontrar a melhor combinação de hiperparâmetros para o Random Forest, maximizando a métrica de **recall**.
    - Treinar o modelo otimizado e avaliar seu desempenho no conjunto de teste, com base em métricas como precisão, recall e F1-score.
- **Visualização com Matplotlib e Seaborn**: Após o treinamento, usamos essas bibliotecas para visualizar os resultados, com a matriz de confusão sendo um dos gráficos

principais. Isso nos permite avaliar visualmente o desempenho e verificar onde o modelo está acertando ou falhando na detecção de fraudes.

4. Aplicação de Conceitos de Machine Learning e IA

O projeto OdontoVision aplica conceitos fundamentais de **Machine Learning supervisionado**. Abaixo, detalhamos como esses conceitos são integrados:

- **Aprendizado Supervisionado**: O Random Forest é um modelo supervisionado que aprende a partir de exemplos rotulados (fraude e não fraude). Esse aprendizado supervisionado é crucial para que o modelo consiga generalizar e detectar fraudes em novos dados de sinistros odontológicos.
- **Balanceamento de Classes**: O uso de SMOTE e da reponderação com `class_weight='balanced'` no Random Forest são estratégias de balanceamento. Essas técnicas garantem que o modelo aprenda de maneira justa, considerando a importância de detectar fraudes em um contexto onde os dados são desbalanceados.
- **Otimização de Hiperparâmetros com GridSearchCV**: Ajustar hiperparâmetros é uma técnica de refinamento em Machine Learning. Através do GridSearchCV, testamos diferentes configurações do Random Forest para encontrar a que melhor maximiza o recall para fraudes, reduzindo a chance de falsos negativos.

Esses conceitos e técnicas trabalham juntos para criar um modelo que seja ao mesmo tempo preciso e robusto na detecção de fraudes odontológicas, alinhado com o objetivo do projeto.

Link para pitch de apresentação do projeto:

https://youtu.be/aOV7Q72TriQ