Софийски университет „Св. Кл. Охридски”

Факултет по математика и информатика

Курсов Проект

на тема: Откриване на измами с кредитни карти

Студент: **Ивелин Бориславов Искренов** Ф.Н. **3MI0800010**

Курс: 4, Учебна година: 2024/25

Преподаватели: **проф. Иван Койчев, …,** Консултант(и) (ако има):

==========

Декларация за липса плагиатство:

- Плагиатство е да използваш, идеи, мнение или работа на друг, като претендираш, че са твои. Това е форма на преписване.
- Тази курсова работа е моя, като всички изречения, илюстрации и програми от други хора са изрично цитирани.
- Тази курсова работа или нейна версия не са представени в друг университет или друга учебна институция.
- Разбирам, че ако се установи плагиатство в работата ми ще получа оценка “Слаб”.

28.6.25 г. Подпис на студента:

Съдържание

1. Увод

С нарастването на електронните плащания и сложността на финансовите транзакции, измамите с кредитни карти представляват сериозен проблем както за банките, така и за клиентите.

Целта на проекта е да се разработят и сравнят два подхода за откриване на измами чрез:

- **`Logistic Regression`** – обучение с учител, техника за бинарна класификация, която оценява вероятността дадена транзакция да е измама.
- **`Isolation Forest`** – обучение без учител, алгоритъм за откриване на аномалии, който „изолира“ отделни наблюдения чрез изграждане на множество случайно конструирани дървета (`Isolation Trees`).

Основните задачи включват:

- Предварителна обработка и стандартизация на данните от CSV файл.
- Реализация на двата алгоритъма на C++.
- Обучение, тестване и оценка на моделите чрез метрики като `precision, recall и F1-score`.
- Провеждане на 10-кратна крос-валидация за оценка на стабилността на представянето.

---

2. Откриване на измами с кредитни карти

Откриването на измами с кредитни карти е от особена важност за предотвратяване на финансови загуби и защита на потребителите. Съществуват различни подходи за решаване на този проблем, сред които:

- **Класификационни методи:** Например *`Logistic Regression`*, които се обучават върху етикетирани данни за разделяне на нормални и измамни транзакции.
- **Методи за откриване на аномалии:** Например *`Isolation Forest`*, които не изискват предварително етикетиране и работят чрез идентифициране на необичайни (изолирани) наблюдения в данните.

Сравнителният анализ между тези подходи помага за определяне на най-подходящото решение в зависимост от естеството и баланса на данните.

---

3. Проектиране

За решаване на задачата са дефинирани следните ключови компоненти:

3.1.1 Предварителна обработка на данните:

- **Четене от CSV:** Използва се функцията `readCSV`, която зарежда данните от файл *creditcard.csv*, като се спазват ограничения за максимален брой измамни и нормални транзакции.
- **Стандартизация:** Чрез функции като `avgValue`, `standardDeviation` и `standardize` се изчисляват средните стойности, стандартните отклонения и се нормализират характеристиките.
- **Разделяне на данните:** Данните се разбиват на тренировъчен и тестови набор чрез функцията `shuffleAndSplitData`. При необходимост е реализирана и крос-валидация чрез стратифицирано разделяне (`splitDataIntoStratifiedKFolds и crossValidationTry`).

3.2 Имплементация на алгоритмите:

3.2.1 `Logistic Regression`

- **Класът `LogisticRegression`:**
  - Съдържа вектор от тегла, пристрастие, `learning rate` и брой итерации.
  - Методите включват:
    - `sigmoid`: Изчисляване на сигмоидната функция.
    - `predict`: Изчисляване на вероятността даден пример да принадлежи към положителния клас.
    - `train`: Обучение чрез градиентен спуск с актуализация на теглата и пристрастията.
    - Методи за оценка: `precision, recall и f1_score.`

3.2.2 `Isolation Forest`

- **Структурата `IsolationTree`:**
  - Представлява индивидуално дърво, използвано за изолация на наблюденията.
  - Член-данни:
    - `feature_index`: Индекс на избраната за разделяне характеристика (избира се случайно).
    - `threshold`: Праг за разделяне, определен на базата на минимална и максимална стойност на избраната характеристика.
    - `left и right`: Указатели към лявото и дясното поддърво.
    - n: Брой на примерите в текущия възел.
    - H: Хармонично число, изчислено като log(n) + 0.57721, използвано за нормализация.
    - `isLeaf`: Булева променлива, която указва дали възелът е лист.
  - **Методът `build_tree`:**

        - Рекурсивно изгражда дървото, като избира на всяка стъпка случайна характеристика и праг за разделяне.
        - Спирането на рекурсията се осъществява при достигане на максимална дълбочина (`MAX_DEPTH`) или когато броят на примерите е под минимално зададената стойност (`MIN_SPLIT_SAMPLES`).
- **Структурата `IsolationForest`:**
    - Съдържа вектор от указатели към изградени дървета и брой на дърветата (`TREE_COUNT`), които формират гората.
    - **Методът `fit`:**
        - За всяко дърво се извиква `build_tree` върху тренировъчния набор и се запазва изчисленото хармонично число за нормализация.
    - **Методът `anomaly_score`:**
        - За даден пример се изчислява средната дълбочина на изолация в всички дървета.
        - Използва се нормализиращ фактор, изчислен въз основа на хармоничното число.
    - **Методът `predict`:**
        - Ако аномалната оценка е по-голяма от зададен праг (например 0.50), примерът се класифицира като измама ('1'), в противен случай – като нормален ('0').
    - Допълнителни методи за оценка включват изчисляване на точност, `precision, recall и F1-score` върху даден набор от данни.
    - **Крос-валидация:**
        - Функциите `splitDataIntoStratifiedKFolds` и `crossValidationTry` реализират стратифицирано разделяне на данните на *k* групи (`folds`) и провеждане на 10-кратна крос-валидация, която да оцени стабилността на модела.

---

4. Реализация, тестване/експерименти

4.1 Използвани технологии, платформи и библиотеки

- **Език за програмиране:** C++
- **Стандартна библиотека на C++ (STL):**
  Използвани са библиотеки `като <iostream>, <vector>, <random>, <cmath>, <fstream>, <sstream>, <algorithm> и <numeric>`, които подпомагат:
    - Работа с файлове и потоци
    - Манипулация на данни (вектори и контейнери)
    - Изчислителни и статистически операции
- **Компилатор:** g++ или друг съвместим C++ компилатор
- **Интегрирана среда за разработка (IDE):** `Visual Studio`

4.2 Реализация и експерименти с `Logistic Regression`

Имплементация:

- **Класът `LogisticRegression`** съдържа методи за:
  - Изчисляване на сигмоидната функция чрез `sigmoid`.
  - Предсказване на вероятността даден пример да е положителен чрез метода `predict`.
  - Обучение на модела чрез градиентен спуск в метода `train`, като се актуализират теглата и пристрастията.
  - Оценка на представянето с помощта на метриките `precision`, `recall` и `F1_score`.

Обработка на данните:

- Данните се четат от CSV файл с функцията `readCSV`, след което се стандартизират.
- Функцията `shuffleAndSplitData` разделя данните на тренировъчен 80% и тестови набор.

Изпълнение:

- Програмата стартира от `main()`, където се зареждат данните, се стандартизират и разделят.
- Моделът се обучава върху тренировъчния набор и след това се оценява чрез изчисляване на `recall` и `F1-score` върху тренировъчните и тестовите данни.
- Резултатите се отпечатват в конзолата.

4.3 Реализация и тестване с `Isolation Forest`

Имплементация:

- **Структура `Isolation tree`:**
  - Отговаря за изграждането на индивидуално изолационно дърво.
  - При избора на разделяща характеристика се избира случайно (например чрез rand() % 29 за 29 характеристики).
  - **Методът `build_tree`:**
    - Определя минималната и максималната стойност на избраната характеристика.
    - Генерира случайно число, използвано за изчисляване на праг (`threshold`).
    - Разделя данните на две групи (ляво и дясно) според това дали стойността е по-малка или по-голяма от прага.

        - Рекурсивно изгражда поддърветата, докато не се достигне максимална дълбочина (`MAX_DEPTH`) или броят на примерите стане по-малък от зададения праг (`MIN_SPLIT_SAMPLES`).
    - **Методът `isolation_depth`:**
        - Изчислява дълбочината на изолация на даден пример, като връща колко нива трябва да се премине, за да се изолира наблюдението.
- **Структура IsolationForest:**
    - Съдържа вектор от указатели към множество изолационни дървета (например 150 дървета – TREE_COUNT).
    - **Методът `fit`:**
        - За всяко дърво се извиква build_tree, използвайки тренировъчния набор, и се съхранява хармоничното число *H* (изчислено като log(n) + 0.57721), което се използва при нормализиране на аномалните оценки.
    - **Методът `anomaly_score`:**
        - За даден пример се изчислява средната дълбочина на изолация през всички дървета.
        - С помощта на нормализиращ фактор се изчислява аномалната оценка.
    - **Методът `predict`:**
        - Ако аномалната оценка надвишава прага (например 0.50), примерът се класифицира като измама ('1'); в противен случай – като нормален ('0').
    - Допълнителни функции за оценка включват изчисляване на точност, `precision`, `recall` и `F1-score` върху зададени данни.

Обработка на данните и крос-валидация:

- Данните се зареждат от CSV файл чрез функцията `readCSV`, както при Logistic Regression.
- Функцията `shuffleAndSplitData` разделя данните на тренировъчен и тестови набор.
- За по-детайлна оценка на представянето е реализирана и 10-кратна крос-валидация чрез функциите `splitDataIntoStratifiedKFolds` и `crossValidationTry`, която осигурява стратифицирано разделяне на данните и изчисляване на средна точност и стандартно отклонение.

Изпълнение:

- В main() се зареждат и разделят данните, след което се изгражда моделът чрез създаване на обект от тип `IsolationForest` с предварително зададен брой дървета.
- Моделът се обучава върху тренировъчния набор чрез извикване на fit.
- След това се изчисляват метриките `recall` и `F1-score` както за тренировъчния, така и за тестовия набор, като резултатите се отпечатват в конзолата.

- Възможно е допълнително изпълнение на 10-кратна крос-валидация при потребителски избор.

---

5. Заключение

В рамките на проекта са реализирани два различни подхода за откриване на измами с кредитни карти:

- **`Logistic Regression`** – подход, базиран на супервизирано обучение, който позволява добра интерпретируемост и оценка чрез стандартни метрики.
- **`Isolation Forest`** – метод за откриване на аномалии, който чрез изграждането на множество случайни дървета изолира потенциално измамни транзакции.

Основни изводи:

- **`Logistic Regression`:**
  - Предимства: Ясно разделение на класовете, лесна интерпретация на коефициентите. Бърз при много количество данни.
  - Ограничения: Изисква наличието на етикетирани данни и може да бъде чувствителен към несбалансирани класове.
- **`Isolation Forest`:**
  - Предимства: Не се изисква предварително етикетиране, добро представяне при небалансирани данни, сравнително ниска сложност при изчисления.
  - Ограничения: Изборът на оптимални параметри (като максимална дълбочина и брой дървета) е критичен за качеството на класификацията. Бавен при по-големи количества данни.

Проектът предоставя възможност за бъдещи подобрения чрез комбиниране на двата подхода или интегриране на допълнителни техники за регуларизация и оптимизация на моделите.

---

6. Използвана литература

1. GeeksforGeeks. Retrieved from https://www.geeksforgeeks.org/.
2. Medium. Retrieved from https://medium.com/.
3. Онлайн ресурси и статии, свързани с Logistic Regression и откриване на измами с кредитни карти.
4. Допълнителни материали, свързани с алгоритъма Isolation Forest.