БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

Факультет прикладной математики и информатики

Кафедра технологий программирования

Логистическая регрессия

Контрольная работа

Шибко Татьяны Александровны
студентки 4 курса 12 группы
специальность "прикладная информатика"

Преподаватель:

Кандидат технических наук
М.М. Лукашевич

Минск, 2024

СОДЕРЖАНИЕ

ИДЕЯ АЛГОРИТМА 3
ПЛЮСЫ И МИНУСЫ 4
ПРИМЕР НА РЕАЛЬНЫХ ДАННЫХ 5
СПИСОК ИСТОЧНИКОВ 8
ПРИЛОЖЕНИЕ 9

ИДЕЯ АЛГОРИТМА

Логистическая регрессия — это алгоритм машинного обучения, используемый для задач двоичной классификации, где целью является предсказание вероятности принадлежности наблюдения к одному из двух классов (например, да или нет, 1 или 0). Это тип регрессионного анализа, который оценивает вероятность наступления события на основе одной или нескольких независимых переменных.

Логистическая регрессия как метод статистического анализа была разработана в середине 20-го века. Хотя конкретного "изобретателя" этого метода сложно выделить, он основывается на концепциях, разработанных в области статистики и теории вероятностей.

Логистическая регрессия изначально была разработана для анализа бинарных исходов. В 1958 году Дэвид Кокс предложил метод максимального правдоподобия для оценки её параметров. Она использует логистическую (сигмоидальную) функцию, чтобы преобразовать линейное уравнение в вероятность бинарного результата.

Алгоритм обучается, подбирая коэффициенты, которые максимизируют вероятность наблюдаемых данных, и применяет их для прогнозирования.

Применение: прогноз оттока клиентов, выявление мошенничества, диагностика заболеваний (финансы, маркетинг, здравоохранение).

Вот основные термины логистической регрессии:

- **Независимые переменные**: входные характеристики, используемые для прогнозирования.
- **Зависимая переменная**: целевая переменная (0 или 1).
- **Логистическая функция**: преобразует входные данные в вероятность (от 0 до 1).
- **Шансы**: отношение вероятности события к вероятности его отсутствия.
- **Логарифм шансов (логит)**: натуральный логарифм шансов, моделируется как линейная комбинация признаков.
- **Коэффициент**: показывает вклад признаков в целевую переменную.
- **Свободный член**: логарифм шансов при нулевых значениях признаков.
- **Оценка максимального правдоподобия**: метод нахождения коэффициентов, увеличивающий вероятность наблюдаемых данных.

ПЛЮСЫ И МИНУСЫ

Плюсы логистической регрессии:

1. **Простота**: Легко реализуется, требует меньше ресурсов, подходит для небольших данных.
2. **Интерпретируемость**: Коэффициенты легко объясняют влияние признаков, что важно для задач в медицине и финансах.
3. **Работа с вероятностями**: Модель выдает вероятности, полезные для оценки рисков.
4. **Гибкость**: Подходит для мультиклассовых задач через стратегии OvA или OvO.
5. **Быстрая сходимость**: Оптимизация работает эффективно из-за выпуклой функции потерь.

Минусы логистической регрессии:

1. **Линейность**: Плохо работает с нелинейными зависимостями.
2. **Чувствительность к выбросам**: Требуется нормализация и борьба с шумом.
3. **Ограниченная мощность**: Уступает более сложным моделям в извлечении сложных паттернов.
4. **Требовательность к данным**: Нужна тщательная предобработка и устранение мультиколлинеарности.
5. **Гиперпараметры**: Регуляризация требует точной настройки.

Примеры применения:

- Кредитный скоринг.
- Диагностика заболеваний.
- Спам-фильтры.
- Анализ оттока клиентов.

Когда использовать:

- Для бинарной классификации на малых данных с линейной зависимостью.
- Если требуется объяснимость.

Когда избегать:

- Для нелинейных задач или сложных данных.
- Если данных слишком много и доступны более мощные алгоритмы.

ПРИМЕР НА РЕАЛЬНЫХ ДАННЫХ

В качестве реальных данных я решила взять датасет «Crimes Against Women in India».

Для начала я просто загрузила датасет и ознакомилась с его содержанием.

```
[4] import pandas as pd
    from google.colab import files

[2] uploaded = files.upload()

    Выбрать файлы  CrimesO...nData.csv
    • CrimesOnWomenData.csv(text/csv) - 32551 bytes, last modified: 16.09.2024 - 100% done
    Saving CrimesOnWomenData.csv to CrimesOnWomenData.csv

    data = pd.read_csv("CrimesOnWomenData.csv")
    print(data.head())

       Unnamed: 0            State  Year  Rape   K&A   DD   AoW   AoM    DV  WT
    0           0   ANDHRA PRADESH  2001   871   765  420  3544  2271  5791   7
    1           1  ARUNACHAL PRADESH  2001    33    55    0    78     3    11   0
    2           2            ASSAM  2001   817  1070   59   850     4  1248   0
    3           3            BIHAR  2001   888   518  859   562    21  1558  83
    4           4      CHHATTISGARH  2001   959   171   70  1763   161   840   0
```

Задача: Построить модель для предсказания высокого уровня преступности против женщин в зависимости от данных по различным типам преступлений.
Целевая переменная: создать бинарную метку High_Crime:

- 1 (высокий уровень преступности) — если общее число преступлений (AoW) превышает медиану,
- 0 (низкий уровень) — если общее число преступлений меньше или равно медиане.

Модель должна определить, как различные виды преступлений (например, Rape, K&A, DV, и др.) влияют на вероятность высокой преступности.

Код будет представлен в приложении. Сейчас я кратко поясню результаты, которые получила.

```
# Создаем целевую переменную: высокий уровень преступности
data['High_Crime'] = (data['AoW'] > data['AoW'].median()).astype(int)

# Проверяем распределение нового столбца
print(data['High_Crime'].value_counts())
```

```
High_Crime
1    368
0    368
Name: count, dtype: int64
```

Данные показывают, что классы целевой переменной High_Crime сбалансированы:

- 1 (высокий уровень преступности): 368 записей.

- 0 (низкий уровень преступности): 368 записей.

Сбалансированность важна, так как она предотвращает перекос модели в сторону одного из классов и обеспечивает корректные оценки метрик.

```
# Инициализация и обучение модели логистической регрессии
model = LogisticRegression()
model.fit(X_train, y_train)

print("Коэффициенты модели:", model.coef_)
print("Перехват (Intercept):", model.intercept_)
```

```
Коэффициенты модели: [[ 3.58087078  0.15260406 -0.90559583  0.88869524  1.8943469   0.22168762]]
Перехват (Intercept): [1.82152296]
```

Каждый коэффициент показывает вклад соответствующего признака в вероятность высокой преступности. Например:

- Признак с коэффициентом **3.58** (вероятно, Rape) оказывает наибольшее влияние.
- Отрицательный коэффициент (**-0.90**, вероятно, DD) указывает на то, что увеличение этого признака снижает вероятность высокого уровня преступности.

Intercept (перехват) = **1.82** — базовая вероятность высокого уровня преступности без учета признаков.

```
# Предсказание на тестовых данных
y_pred = model.predict(X_test)

print("Точность модели:", accuracy_score(y_test, y_pred))
print("Отчет классификации:\n", classification_report(y_test, y_pred))
print("Матрица ошибок:\n", confusion_matrix(y_test, y_pred))
```

```
Точность модели: 0.8648648648648649
Отчет классификации:
              precision    recall  f1-score   support

           0       0.89      0.89      0.89        88
           1       0.83      0.83      0.83        60

    accuracy                           0.86       148
   macro avg       0.86      0.86      0.86       148
weighted avg       0.86      0.86      0.86       148

Матрица ошибок:
 [[78 10]
 [10 50]]
```

Precision (Точность):

- Для класса 0 (низкий уровень преступности): **0.89** — из всех случаев, предсказанных как низкий уровень, 89% верны.
- Для класса 1 (высокий уровень преступности): **0.83** — из всех случаев, предсказанных как высокий уровень, 83% верны.

Recall (Полнота):

- Для класса 0: **0.89** — из всех реальных случаев низкого уровня, 89% правильно предсказаны.
- Для класса 1: **0.83** — из всех реальных случаев высокого уровня, 83% правильно предсказаны.

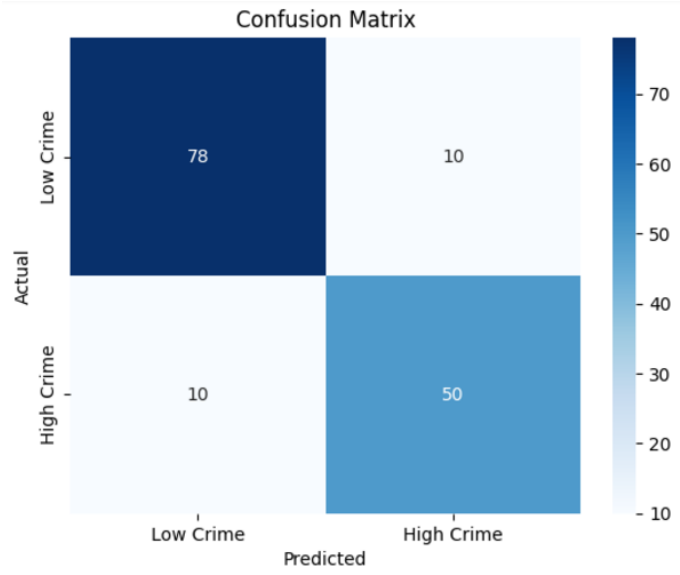
F1-score:

- Среднее значение точности и полноты для каждого класса. Чем выше, тем лучше баланс между точностью и полнотой.

Матрица ошибок

- Визуализация правильных и ошибочных классификаций:
  - **78:** верно классифицировано как низкий уровень преступности.
  - **50:** верно классифицировано как высокий уровень преступности.
  - **10:** ошибок, где модель предсказала низкий уровень, но в реальности был высокий.
  - **10:** ошибок, где модель предсказала высокий уровень, но в реальности был низкий.

Вывод:



Модель демонстрирует высокую точность (86.48%) и сбалансированные метрики (precision/recall), что делает её полезной для предсказания уровня преступности. Однако 20 ошибок из 148 (в матрице ошибок) говорят о возможности улучшения модели, например, добавлением новых признаков или использованием более сложных моделей.

СПИСОК ИСТОЧНИКОВ

1. Пошаговое руководство по обнаружению мошенничества с использованием логистической регрессии Python: комплексный подход / [Электронный ресурс] // uproger.com: [сайт]. — URL: https://uproger.com/rukovodstvo-po-obnaruzheniyu-moshennichestva-python/ (дата обращения: 28.11.2024).

2. Logistic Regression in Machine Learning / [Электронный ресурс] // geeksforgeeks.org : [сайт]. — URL: https://www.geeksforgeeks.org/understanding-logistic-regression/ (дата обращения: 28.11.2024).

3. Что такое логистическая регрессия? / [Электронный ресурс] // https://aws.amazon.com/ : [сайт]. — URL: https://clck.ru/3EtBwF (дата обращения: 28.11.2024).

4. Data Science Team Логистическая регрессия / Data Science Team [Электронный ресурс] // https://datascience.eu/ : [сайт]. — URL: https://clck.ru/3EtCBz (дата обращения: 28.11.2024).

5. Advantages and Disadvantages of Logistic Regression / [Электронный ресурс] // geeksforgeeks.org : [сайт]. — URL: https://translated.turbopages.org/proxy_u/en-ru.ru.da47a257-67484b3e-1a9c128a-74722d776562/https/www.geeksforgeeks.org/advantages-and-disadvantages-of-logistic-regression/ (дата обращения: 28.11.2024).

6. Кокс, Дэвид (статистик) / [Электронный ресурс] // Википедия : [сайт]. — URL: https://clck.ru/3EtCVi (дата обращения: 28.11.2024).

7. Crimes Against Women in India (2001-2021) / [Электронный ресурс] // https://www.kaggle.com/ : [сайт]. — URL: https://www.kaggle.com/datasets/balajivaraprasad/crimes-against-women-in-india-2001-2021/data (дата обращения: 28.11.2024).

ПРИЛОЖЕНИЕ

```
import pandas as pd
import numpy as np
from google.colab import files
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import accuracy_score, classification_report, confusion_matrix
from sklearn.preprocessing import StandardScaler
import matplotlib.pyplot as plt
import seaborn as sns

uploaded = files.upload()
data = pd.read_csv("CrimesOnWomenData.csv")
print(data.head())

# Создание целевой переменной: высокий уровень преступности
data['High_Crime'] = (data['AoW'] > data['AoW'].median()).astype(int)

features = ['Rape', 'K&A', 'DD', 'AoM', 'DV', 'WT']
X = data[features]
y = data['High_Crime']
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

# Нормализация данных
scaler = StandardScaler()
X_train = scaler.fit_transform(X_train)
X_test = scaler.transform(X_test)

# Обучение модели логистической регрессии
model = LogisticRegression()
model.fit(X_train, y_train)

print("Коэффициенты модели:", model.coef_)
print("Перехват (Intercept):", model.intercept_)

y_pred = model.predict(X_test)

print("Точность модели:", accuracy_score(y_test, y_pred))
print("Отчет классификации:\n", classification_report(y_test, y_pred))
print("Матрица ошибок:\n", confusion_matrix(y_test, y_pred))

cm = confusion_matrix(y_test, y_pred)
sns.heatmap(cm, annot=True, fmt='d', cmap='Blues', xticklabels=['Low Crime', 'High Crime'],
yticklabels=['Low Crime', 'High Crime'])
plt.xlabel('Predicted')
plt.ylabel('Actual')
plt.title('Confusion Matrix')
plt.show()
```