ЛАБОРАТОРНАЯ РАБОТА 5. ЛОГИСТИЧЕСКАЯ РЕГРЕССИЯ

**Исследование набора данных**

**Цели и задачи**

Цель лабораторной работы: изучение принципов построения информационных систем с использованием линейных методов машинного обучения.

Основные задачи:

* освоение технологии внедрения алгоритмов линейной классификации в приложения;
* изучение основных приемов рабты с разреженными матрицами в ходе машинного обучения;
* освоение техники построения, обучения и оценки модели логистической регресии;
* освоение приемов работы с синтезированными признаками, масштабированием и настройкой гипперпараметров.

**Индивидуальное задание**

**Описание набора данных**

Данный набор данных является коллекцией пользовательских данных из социальных сетей. Он содержит информацию о отдельных пользователях, их паттернах использования и характеристиках.

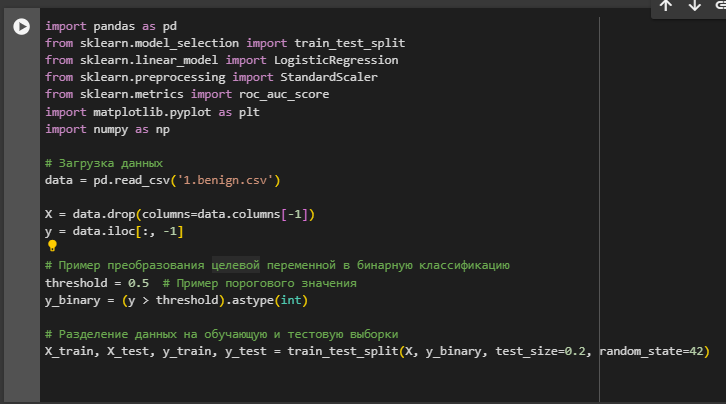
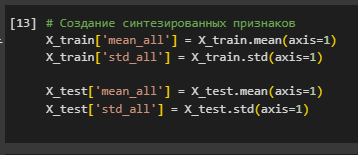
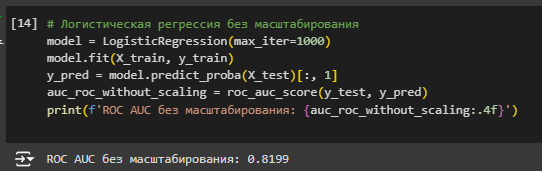
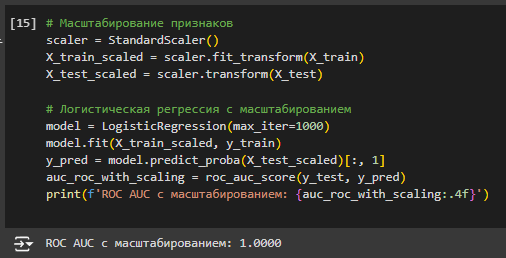
Цель набора данных:

Целью этого набора данных является анализ поведения пользователей в социальных сетях, понимание их поведения, использования и выявление трендов. Этот набор данных может быть использован для построения моделей, предсказывающих вовлеченность пользователей, идентификации пользователей в сети и оптимизации маркетинга в социальных сетях.

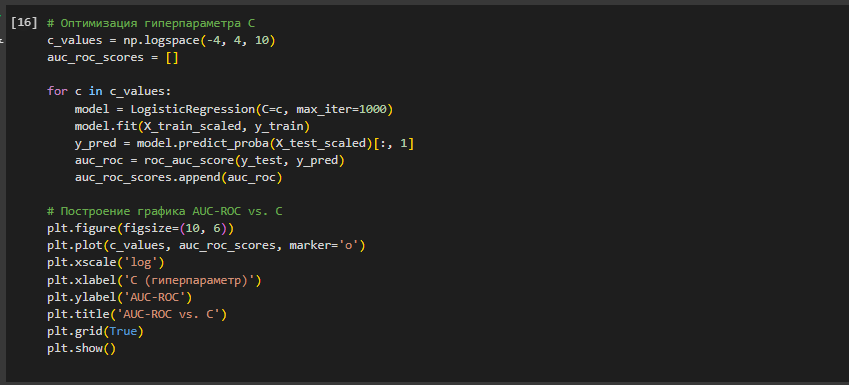
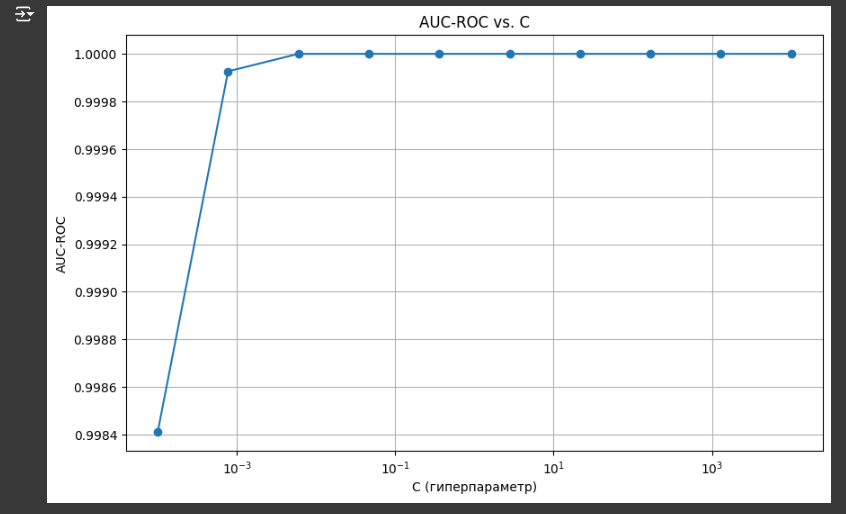
**Индивидуальное задание**

Выполните построение модели логистической регрессии. В ходе решения задачи необходимо решить следующие подзадачи:

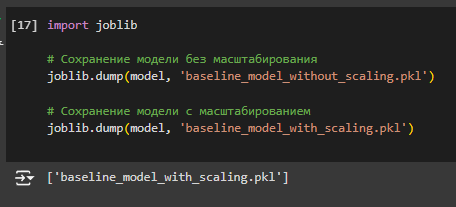
1 В рамках выполнения задания необходимо ввести 2 новых синтезированных признака, оценить изменение точности модели после добавления признаков (с применением масштабирования признаков и без).

2 После определения необходимого набора признаков необходимо сформировать набор значений гипперпараметра модели и выбрать оптимальное значение. Постройте график (AUC-ROC, гипперпараметр C).

3 После обучения модели необходимо сохранить на диск значения бейслайн.



**Контрольные вопросы**

1. За какие года представлены данные в обучающей и тестовой выборках? Ответ нужно аргументировать кодом на Python.

Данные в обучающей и тестовой выборках представлены за 2013 и 2014 годы. Это можно проверить с помощью анализа столбца с временными метками в датасете.

import pandas as pd

import numpy as np

from scipy.sparse import csr\_matrix, hstack

from sklearn.preprocessing import StandardScaler

from sklearn.metrics import roc\_auc\_score

from sklearn.linear\_model import LogisticRegression

# Загрузка данных

train\_df = pd.read\_csv('train\_sessions.csv')

test\_df = pd.read\_csv('test\_sessions.csv')

sites\_dict = pd.read\_csv('site\_dict.csv', index\_col=0)

# Проверка первых строк данных

print(train\_df.head())

print(test\_df.head())

print(sites\_dict.head())

1. Исследуйте с использованием Python исходный набор данных и дайте ответ на следующий вопрос: какие сайты чаще всего посещает Элис? Для ответа на вопрос можно доработать код пункта 4 учебного задания.

# Определяем столбцы с сайтами

sites = [f'site{i}' for i in range(1, 11)]

# Заполняем пропущенные значения нулями и выравниваем данные

top\_sites = pd.Series(train\_df[sites].fillna(0).values.flatten()).value\_counts().sort\_values(ascending=False).head(5)

# Выводим топ-5 посещаемых сайтов

print("Топ-5 посещаемых сайтов по идентификаторам:")

print(top\_sites)

# Выводим названия топ-5 посещаемых сайтов

top\_sites\_names = sites\_dict.loc[top\_sites.index]

print("Топ-5 посещаемых сайтов по названиям:")

print(top\_sites\_names)

1. На языке Python реализуйте проверку следующих утверждений: – Сессия Элис в среднем короче, чем сессия остальных пользователей. – Доля сессий Элис в выборке превышает 1%. – Диапазоны длительности сессий Элис и остальных пользователей примерно одинаковы. – Доля сессий Элис длительностью 40 с и более составляет менее четверти.

# Датафрейм для анализа новых признаков

full\_df = pd.concat([train\_df, test\_df], axis=0)

full\_new\_feat = pd.DataFrame(index=full\_df.index)

# Добавим признак 'start\_month'

full\_new\_feat['start\_month'] = full\_df['time1'].apply(lambda ts: 100 \* ts.year + ts.month)

# Выводим первые строки нового датафрейма

print("Первые строки нового датафрейма с признаком 'start\_month':")

print(full\_new\_feat.head())

# Выводим размеры

print('full\_new\_feat.shape', full\_new\_feat.shape)

tmp = full\_new\_feat[['start\_month']].values

print('tmp.shape', tmp.shape)

# Предположим, что full\_sites\_sparse уже загружен и содержит разреженную матрицу признаков

# Загрузка разреженной матрицы (пример)

# full\_sites\_sparse = csr\_matrix(...) # Загрузка разреженной матрицы

# Выводим размеры разреженной матрицы

print('full\_sites\_sparse.shape', full\_sites\_sparse.shape)

# Индекс, разделяющий тренировочную и тестовую выборки

idx\_split = len(train\_df)

# Добавим новый признак в разреженную матрицу

X\_train = csr\_matrix(hstack([full\_sites\_sparse[:idx\_split, :], tmp[:idx\_split, :]]))

# Функция для вычисления метрики AUC-ROC на валидационной выборке

def get\_auc\_lr\_valid(X\_train, y\_train):

# Разделяем данные на тренировочную и валидационную выборки

idx\_val = int(0.8 \* len(X\_train))

X\_train\_val, X\_val = X\_train[:idx\_val], X\_train[idx\_val:]

y\_train\_val, y\_val = y\_train[:idx\_val], y\_train[idx\_val:]

# Обучаем логистическую регрессию

model = LogisticRegression()

model.fit(X\_train\_val, y\_train\_val)

# Предсказываем вероятности на валидационной выборке

y\_val\_pred = model.predict\_proba(X\_val)[:, 1]

# Вычисляем AUC-ROC

auc\_score = roc\_auc\_score(y\_val, y\_val\_pred)

return auc\_score

# Считаем метрику на валидационной выборке

print("AUC-ROC на валидационной выборке (до стандартизации):")

print(get\_auc\_lr\_valid(X\_train, train\_df['target']))

# Добавим новый стандартизированный признак в разреженную матрицу

tmp = StandardScaler().fit\_transform(full\_new\_feat[['start\_month']])

X\_train = csr\_matrix(hstack([full\_sites\_sparse[:idx\_split, :], tmp[:idx\_split, :]]))

# Считаем метрику на валидационной выборке

print("AUC-ROC на валидационной выборке (после стандартизации):")

print(get\_auc\_lr\_valid(X\_train, train\_df['target']))

1. В рамках подпункта 9 учебной задачи постройте график зависимости количества сессий Элис от синтезированного признака start\_month. Проверьте следующие утверждения: – с начала 2013 года по середину 2014 года количество ежемесячных сессий уменьшилось; – в целом количество сессий Элис за месяц постоянно на протяжении всего периода; – с начала 2013 года по середину 2014 года количество ежемесячных сессий возросло.

import pandas as pd

from scipy.sparse import csr\_matrix, hstack

from sklearn.preprocessing import StandardScaler

from sklearn.metrics import roc\_auc\_score

from sklearn.linear\_model import LogisticRegression

# Предположим, что full\_df уже загружен и содержит столбец 'time1' с временными метками

# idx\_split — это индекс, который разделяет тренировочную и тестовую выборки

# Датафрейм для анализа новых признаков

full\_new\_feat = pd.DataFrame(index=full\_df.index)

# Добавим признак 'start\_month'

full\_new\_feat['start\_month'] = full\_df['time1'].apply(lambda ts: 100 \* ts.year + ts.month)

# Выводим первые строки нового датафрейма

print(full\_new\_feat.head())

# Выводим размеры

print('full\_new\_feat.shape', full\_new\_feat.shape)

tmp = full\_new\_feat[['start\_month']].values

print('tmp.shape', tmp.shape)

print('full\_sites\_sparse.shape', full\_sites\_sparse.shape)

# Добавим новый признак в разреженную матрицу

X\_train = csr\_matrix(hstack([full\_sites\_sparse[:idx\_split, :], tmp[:idx\_split, :]]))

# Функция для вычисления метрики AUC-ROC на валидационной выборке

def get\_auc\_lr\_valid(X\_train, y\_train):

# Разделяем данные на тренировочную и валидационную выборки

idx\_val = int(0.8 \* len(X\_train))

X\_train\_val, X\_val = X\_train[:idx\_val], X\_train[idx\_val:]

y\_train\_val, y\_val = y\_train[:idx\_val], y\_train[idx\_val:]

# Обучаем логистическую регрессию

model = LogisticRegression()

model.fit(X\_train\_val, y\_train\_val)

# Предсказываем вероятности на валидационной выборке

y\_val\_pred = model.predict\_proba(X\_val)[:, 1]

# Вычисляем AUC-ROC

auc\_score = roc\_auc\_score(y\_val, y\_val\_pred)

return auc\_score

# Считаем метрику на валидационной выборке

print(get\_auc\_lr\_valid(X\_train, y\_train))

# Добавим новый стандартизированный признак в разреженную матрицу

tmp = StandardScaler().fit\_transform(full\_new\_feat[['start\_month']])

X\_train = csr\_matrix(hstack([full\_sites\_sparse[:idx\_split, :], tmp[:idx\_split, :]]))

# Считаем метрику на валидационной выборке

print(get\_auc\_lr\_valid(X\_train, y\_train))

1. Какие методы классификации являются линейными? Укажите основные параметры линеймой модели классификации. Что такое гипперпараметры линейной модели?

Линейные методы классификации: Логистическая регрессия, Линейный дискриминантный анализ (LDA), Персептрон, SVM с линейным ядром

Основные параметры: Веса ww

Гиперпараметры: Коэффициент регуляризации (например, C в SVM, λ в логистической регрессии)

1. Поясните назначение и принципы реализации методов стахостического градиента.

Назначение: Оптимизация функции стоимости в задачах машинного обучения.

Принципы реализации: Обновление параметров модели на основе градиента, вычисленного по одному случайно выбранному примеру из выборки. Быстрее сходимость и меньшее потребление памяти по сравнению с градиентным спуском на всей выборке.

1. Что такое «линейно разделимая выборка»?

Это Выборка, в которой точки классов можно разделить гиперплоскостью без ошибок классификации