# МИНОБРНАУКИ РОССИИ

**САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ ЭЛЕКТРОТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ**

# «ЛЭТИ» ИМ. В.И. УЛЬЯНОВА (ЛЕНИНА)

**Кафедра ИС**

# ОТЧЕТ

**по лабораторной работе №3**

# по дисциплине «Машинное обучение» Тема: Исследование алгоритмов классификации

|  |  |
| --- | --- |
| Студент гр. 2373 | Гермаш А. В. |
| Преподаватель | Татчина Я.А. |

Санкт-Петербург 2024

Цель работы: познакомиться с алгоритмами классификации kNN и Decision Tree, а также с различными метриками качества: Accuracy, Presicion, Recall, F-measure, ROC

# Краткое описание набора данных

Для работы был взят отличный от предыдущего набор данных. Датасет является характеристикой использования смартфонов пользователями. Датасет создан искусственно, представлены следующие атрибуты:

* + User ID - идентификатор пользователя
  + Operating System - Android или iOS
  + App Usage Time (min/day) - использование приложений (мин в день)
  + Screen On Time (hours/day) - время включенного экрана смартфона (часов в день)
  + Battery Drain (mAh/day) - разряд батареи в сутки
  + Number of Apps Installed - количество установленных приложений
  + Data Usage (MB/day) - использование сети в мб в день
  + Age - возраст
  + Gender - Пол

User Behavior Class - классификация поведения пользователя

В качестве целевого класса выступит атрибут “User Behavior Class” (от 1 до 5).

# Первичная обработка данных

* + Дубликатов не оказалось
  + Пропущенных значений не оказалось
  + Выбросов не оказалось
  + Удалены атрибуты User ID, Operating System, Gender, Device Model
  + App Usage Time (min/day) переименовано в Usage Time и приведено к часам в день
  + Упрощены другие названия атрибутов

# Разделение данных на тестовую и обучающую выборки

Разделение на тестовую и обучающую выборки выполнено при помощи метода train\_test\_split из sklearn.model\_selection (см. рис. 1).

Изображение выглядит как текст, снимок экрана, Шрифт, линия

Автоматически созданное описание

Рисунок 1 – Разделение данных

1. **Применение метода kNN**

Применим метод KNeighborsClassifier из библиотеки sklearn (см. рис 2). Вначале происходит обучение алгоритма на обучающей выборке, затем тестирование на тестовой выборке.

Изображение выглядит как текст, Шрифт, снимок экрана, линия

Автоматически созданное описание

Рисунок 2 – Использование kNN

Результат использования kNN (см. рис. 3):

Изображение выглядит как снимок экрана, текст

Автоматически созданное описание

Рисунок 3 – Результат использования kNN

По графику видна лишь одна ошибка в результате работы алгоритма.

1. **Применение метода дерева решений**

Применим метод DecisionTreeClassifier из библиотеки sklearn (см. рис 4). Вначале происходит обучение алгоритма на обучающей выборке, затем тестирование на тестовой выборке.

**Изображение выглядит как текст, Шрифт, снимок экрана, линия

Автоматически созданное описание**

Рисунок 4 – Применение Decision Tree

Выведем дерево решений (см. рис. 5).

Изображение выглядит как текст, диаграмма, линия, снимок экрана

Автоматически созданное описание

Рисунок 5 – Дерево решений

Выведем результат работы алгоритма (см. рис. 6).

Изображение выглядит как снимок экрана

Автоматически созданное описание

Рисунок 6 – Результат работы Decision Tree

По графику видно 2 ошибки, алгоритм неплохо справился с поставленной задачей.

1. **Применение метрик оценки качества**

Применены следующие метрики качества: Accuracy, Presicion, Recall, F-measure, ROC

Применение Accuracy:

Изображение выглядит как текст, снимок экрана, Шрифт, линия

Автоматически созданное описание

Рисунок 7 – Accuracy

Применение Precision:

Изображение выглядит как текст, снимок экрана, Шрифт, линия

Автоматически созданное описание

Рисунок 8 – Precision

Применение Recall:

Изображение выглядит как текст, снимок экрана, Шрифт, линия

Автоматически созданное описание

Рисунок 9 – Recall

Применение F-Measure:

Изображение выглядит как текст, снимок экрана, Шрифт, линия

Автоматически созданное описание

Рисунок 10 – F-Measure

Применение ROC:

Изображение выглядит как текст, Шрифт, линия, снимок экрана

Автоматически созданное описание

Рисунок 11 – ROC

Таким образом, почти все метрики качества показали эффективность kNN на уровне примерно 99.5%, а Decision Tree – на уровне 99%.

# Выводы

В ходе работы были изучены методы классификации kNN и Decision Tree, проведено обучение и тестирование алгоритмов на обучающей и тестовой выборке, использованы метрики качества для определения точности работы алгоритмов.