Московский Авиационный Институт

(Национальный Исследовательский Университет)

# Отчет по лабораторной работе 2

## по курсу “Искусственный интеллект”

**На тему: “Алгоритмы классификации”**

Студент: Буркевич В. Д.

Группа: М80-307Б-17

Преподаватель: Самир Халид

Москва, 2020

**Постановка задачи**

Необходимо реализовать алгоритмы машинного обучения. Применить данные алгоритмы на наборы данных, подготовленных в первой лабораторной работе. Провести анализ полученных моделей, вычислить метрики классификатора. Произвести тюнинг параметров в случае необходимости. Сравнить полученные результаты с моделями реализованными в scikit-learn. Аналогично построить метрики классификации. Показать, что полученные модели не переобучились. Также необходимо сделать выводы о применимости данных моделей к вашей задаче.

* Логистическая регрессия
* KNN
* Дерево Решений
* Random Forest

**Реализация**

**Логистическая Регрессия**

**Описание модели**

Апостериорные вероятности задаются преобразованием softmax линейных функций переменных признаков, так что:

В реализации используется метод максимального правдоподобия для отдельного определения условных плотностей классов и априорных значений класса. После их нахождения выполняется поиск соответствующих апостериорных вероятностей с помощью теоремы Байеса (определяются параметры {wk}). Для этого потребуются производные от yk по всем a\_j: , где - элементы единичной матрицы. Затем записывается функция правдоподобия: , где , а T - матрица целевых переменных с элементами .

Применяем логарифм и получаем

Градиент функции ошибки относительно одного из векторов параметров имеет вид: ,

С помощью метода Ньютона-Рафсона, получаю алгоритм IRLS для задачи мультикласса. Это требует оценки матрицы Гессе, которая содержит блоки размера M × M, в которых блок j, k: .

**Реализованная модель**

**Без регуляризации:**  
Ошибки на кросс валидации:

Accuracy = 0.4614

Precision = 0.4614

Roc auc = 0.5

Ошибки на выборках

Train Test

Accuracy = 0.7528571428571429 | 0.752

Precision = 0.7528571428571429 | 0.752

Roc auc = 0.5 | 0.5

**С регуляризацией:**

Ошибки на кросс валидации:

Accuracy = 0.5558

Precision = 0.5558

Roc auc = 0.5

Ошибки на выборках

Train Test

Accuracy = 0.7528571428571429 | 0.752

Precision = 0.7528571428571429 | 0.752

Roc auc = 0.5 | 0.5

**С Sklearn**  
Ошибка на выборках:

Accuracy = 0.9986666666666667

Ошибки на кросс валидации:

Accuracy = 0.9986666666666667

Ошибки на выборках

Train Test

Accuracy = 0.9997142857142857 | 0.9986666666666667

Precision = 0.9997142857142857 | 0.9986666666666667

Roc auc = 0.999421965317919 | 0.9982126515671471

**Вывод:**

Так как метрики классификации на трейновой и тестовой выборках почти не отличаются, можно утверждать, что модель не переобучилась.

Модель логистической регрессии показалась довольно сложной и заняла достаточно времени.

Из полученных оценок можно сделать вывод, что данная модель подходит для решения задачи данной задачи.

**KNN**

**Описание модели**

Для классификации каждого из объектов тестовой выборки необходимо последовательно выполнить следующие операции:

* Вычислить расстояние до каждого из объектов обучающей выборки
* Отобрать k объектов обучающей выборки, расстояние до которых минимально
* Класс классифицируемого объекта — это класс, наиболее часто встречающийся среди k ближайших соседей

Было выбрано двумерное пространство, в котором случайным образом на определенном участке по каждой оси выбирается местоположение мат.ожидания двумерного гауссиана со среднеквадратичным отклонением 0.5 (выбор следует из правила трех сигм). Для определения расстояния между объектами можно использовать евклидово расстояние, манхэттенское расстояние, косинусная мера, критерий корелляции Пирсона и др.

На вход подается матрица расстояний между объектами обучающей и тестовой выборки, метки обучающей выборки и число ближайших соседей. На выходе предсказанные метки для новых объектов и вероятности каждой метки.

**Реализованная модель**

CV scores: [0.5985714285714285, 0.5842857142857143, 0.6085714285714285, 0.5892857142857143, 0.6064285714285714]

Train data accuracy: 0.5974285714285714

Ошибки на кросс валидации:

Accuracy = 0.651

Precision = 0.651

Roc auc = 0.49999976393217693

Ошибки на выборках

Train Test

Accuracy = 0.8175714285714286 | 0.651

Precision = 0.8175714285714286 | 0.651

Roc auc = 0.499617188541433 | 0.49999976393217693

**С Sklearn**

Ошибка на выборках:

Accuracy = 0.6556666666666666

Ошибки на выборках

Train Test

Accuracy = 0.8187142857142857 | 0.6556666666666666

Precision = 0.8187142857142857 | 0.6556666666666666

Roc auc = 0.4996235296003279 | 0.4999997439250601

**Вывод:**

Так как метрики классификации на трейновой и тестовой выборках не отличаются в случае RocAuc, а в остальных метрики на тестовой выборке сильно хуже следует, что модель переобучилась, вероятно недообучилась.

Модель KNN считается не сложной, но даже при этом на трейновых и тестовых данных выдает хороший результат: accuracy свыше 0.8 на трейне.

Из полученных оценок можно сделать вывод, что данная модель подходит для решения задачи данной задачи.

**Решающее дерево**

**Описание модели**

Дерево решений представляет собой двоичное дерево:

* Узел — это одна входная переменная (x) и точка ее разделения.
* Конечные узлы (терминальные) дерева содержат выходную переменную (y), которая используется для прогнозирования.

На каждом шаге выбирается признак, при разделении по которому прирост информации оказывается наибольшим:

1. Вычисление показателя Джини:
2. Разделение набора данных:
   1. Проверка нахождения значение атрибута ниже или выше значения разделения.
   2. Присвоение его левой или правой группе.
3. Оценка разделений:

* Проверяются значения каждого атрибута, оценивается стоимость разделения и находится наилучшее возможное разделение.
* Лучшее разделение используется в качестве узла в дереве решений.

Процедура повторяется рекурсивно к каждой из частей, пока энтропия не окажется равной нулю или очень малой величине.

Дополнительные условия останова: достижение нужной глубины дерева и/или достижение минимальной записи узла.

Для представления узла в дереве используется словарь для представления решений, поскольку мы можем хранить данные по имени. При выборе наилучшего разделения и использовании его в качестве нового узла для дерева, хранится индекс выбранного атрибута, его значения, по которому нужно разделяться, и две группы данных, разбитые по выбранной точке разделения. Каждая группа данных представляет собой собственный небольшой набор данных.

**Реализованная модель**

Ошибки на кросс валидации:

Accuracy = 0.9800000000000001

Precision = 0.9800000000000001

Roc auc = 0.9833333333333334

Ошибки на выборках

Train Test

Accuracy = 1.0 | 1.0

Precision = 1.0 | 1.0

Roc auc = 1.0 | 1.0

**С Sklearn**

Ошибки на выборках

Train Test

Accuracy = 1.0 | 1.0

Precision = 1.0 | 1.0

Roc auc = 1.0 | 1.0

**Вывод:**

Так как метрики классификации на трейновой и тестовой выборках не отличаются следует, что модель не переобучилась.

Как видно из полученных результатов дерево решений очень хорошо подходит для решения данной задачи классификации.

**Random Forest**

**Описание модели**

Сначала генерируется подвыборка из обучающей выборки. Далее строится решающее дерево, которое классифицирует образцы данной подвыборки. В ходе создания каждого следующего узла дерева выбирается набор признаков, на основе которых производится разбиение. Используются не все признаки, а некоторые, выбранные случайно. Дерево строиться пока не закончится вся подвыборка.

**Реализованная модель**

Ошибки на валидации

accuracy = 0.61

precision = 0.6873669467787116

roc auc = 0.5906870906870907

Ошибки на выборках

Train Test

Accuracy = 0.6285714285714286 | 0.6666666666666666

Precision = 0.9069767441860465 | 0.9444444444444444

Roc auc = 0.5974499089253188 | 0.701923076923077

**С Sklearn**

test\_accuracy = 1.0

test\_precision\_macro = 1.0

test\_roc\_auc = 1.0

Ошибки на выборках

Train Test

Accuracy = 1.0 | 1.0

Precision = 1.0 | 1.0

Roc auc = 1.0 | 1.0

**Вывод:**

Так как метрики классификации на трейновой и тестовой выборках почти не отличаются следует, что модель не переобучилась.

Как видно из полученных результатов дерево решений очень хорошо подходит для решения данной задачи классификации.