Министерство науки и высшего образования Российской Федерации

Федеральное государственное бюджетное образовательное учреждение высшего образования

Ульяновский государственный Технический университет

Кафедра «Вычислительная техника»

Дисциплина «Системы искусственного интеллекта»

**Лабораторная работа №4**

**«Основы нейронных сетей»**

Выполнила:

студентка группы ИВТАСбд-41

Архипова Е.Ю

Проверил работу:

Хайруллин И. Д.

Ульяновск 2025

**Общее задание**

1. Написать программу, которая разделяет исходную выборку на обучающую и тестовую (training set, validation set, test set), если такое разделение не предусмотрено предложенным набором данных.
2. Произвести масштабирование признаков (scaling).
3. С использованием библиотеки scikit-learn обучить 2 модели нейронной сети (Perceptron и MLPClassifier) по обучающей выборке. Перед обучением необходимо осуществить масштабирование признаков. Пример MLPClassifier Пример и описание Perceptron
4. Проверить точность модели по тестовой выборке.
5. Провести эксперименты и определить наилучшие параметры коэффициента обучения, параметра регуляризации, функции оптимизации. Данные экспериментов необходимо представить в отчете (графики, ход проведения эксперимента, выводы).

**Теоретические данные**

**Perceptron** — это простейшая форма искусственной нейронной сети, состоящая из одного или нескольких входных соединений, весов, суммирующих входные сигналы, и выходного сигнала, который обрабатывается через активационную функцию. Основная идея перцептрона заключается в том, что он принимает входные данные, умножает их на соответствующие веса, суммирует полученные значения и применяет активационную функцию к этой сумме для получения выходного сигнала. Перцептроны используются в задачах классификации и регрессии, а также служат основой для более сложных архитектур нейронных сетей.

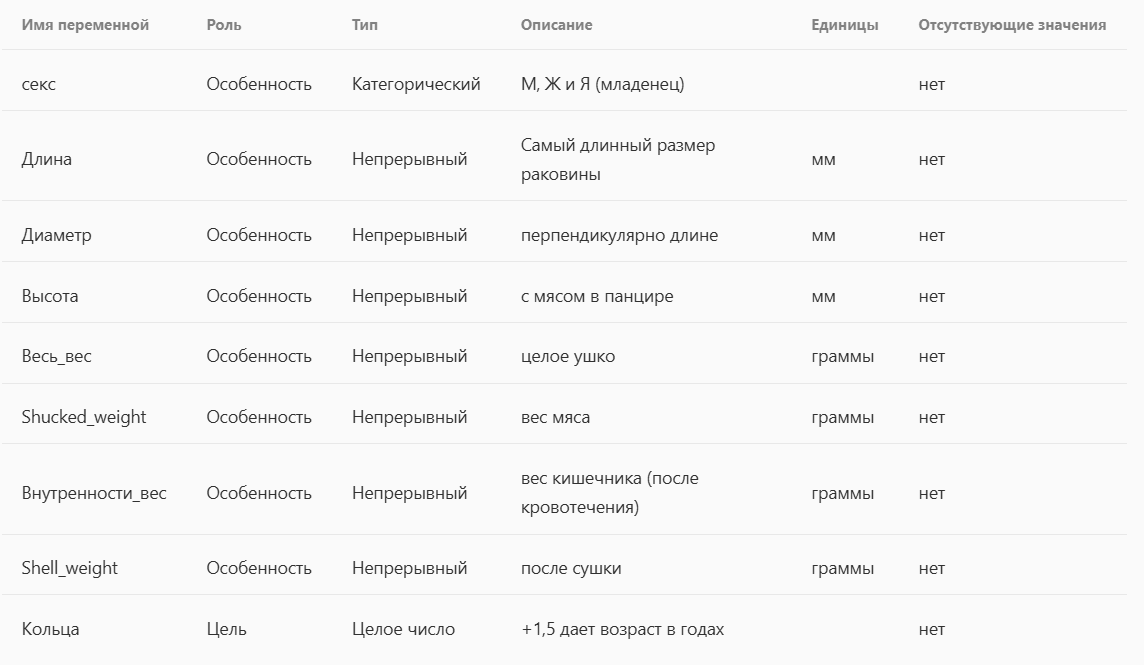
**MLPClassifier** (Multi-Layer Perceptron Classifier) — это классификатор, основанный на многослойной перцептроне, который может использоваться для решения задач классификации. MLP состоит из трех слоев: входного слоя, скрытого слоя и выходного слоя. Каждый слой состоит из нейронов, которые соединены между собой с весами. MLPClassifier использует обратное распространение ошибки для обучения весов с целью минимизации ошибки классификации. Этот метод позволяет решать сложные задачи классификации, особенно когда данные имеют высокую размерность или когда требуется учесть взаимодействия между признаками.

**StandardScaler** — это инструмент из библиотеки scikit-learn, который используется для стандартизации признаков путем вычитания среднего значения каждого признака и деления на его стандартное отклонение. Стандартизация признаков помогает уменьшить масштаб признаков, делая их более совместимыми для обучения моделей машинного обучения, особенно когда признаки имеют разные единицы измерения или разные диапазоны значений. Это улучшает стабильность и скорость обучения моделей, а также может повысить их точность.

**GridSearchCV** — это метод из библиотеки scikit-learn, который позволяет проводить поиск по сетке гиперпараметров для оптимизации производительности модели. GridSearchCV автоматически перебирает комбинации заданных гиперпараметров, обучает модель для каждой комбинации и выбирает ту, которая дает наилучшую производительность на валидационном наборе данных. Это мощный инструмент для настройки гиперпараметров, особенно когда пространство возможных значений гиперпараметров велико.

**Описание набора данных**

Прогнозирование возраста морского ушка по физическим измерениям. Возраст морского ушка определяется путем разрезания раковины на конус, ее окрашивания и подсчета количества колец с помощью микроскопа — скучная и трудоемкая задача. Другие измерения, которые легче получить, используются для прогнозирования возраста. Вот краткое описание ключевых столбцов:



**Рис 1.** Таблица переменных

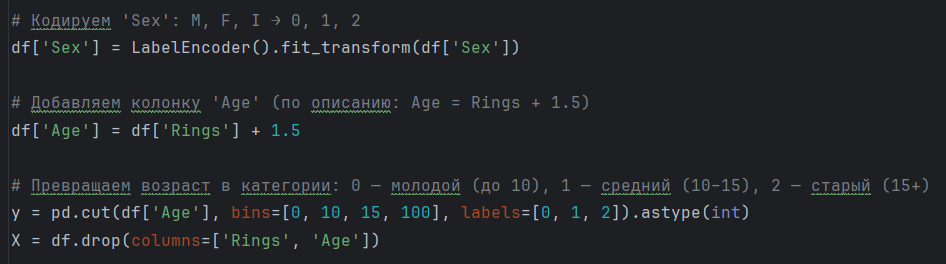
**Описание реализации**

Для экспериментов использовался набор данных Abalone из репозитория UCI. В наборе содержатся физические характеристики моллюсков и возраст, выраженный через количество колец. В работе возраст был переведён в три категории: молодой, средний и старый. Категоризация позволила решить задачу классификации. Категории возраста:

0 — молодой (возраст до 10 лет)

1 — средний (возраст 10–15 лет)

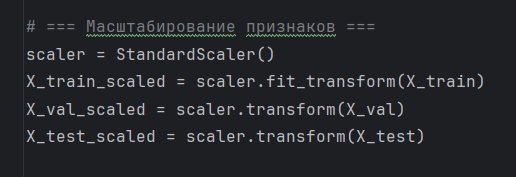
2 — старый (старше 15 лет)



**Рис 2.** Разделение на категории

Данные были загружены и обработаны: категориальный признак пола был закодирован численно, возраст преобразован в категориальный признак, исходные признаки разделены на обучающую, валидационную и тестовую выборки в соотношении 60/20/20.

Для улучшения качества обучения и сходимости алгоритмов применялось масштабирование признаков с помощью StandardScaler.



**Рис 3.** Масштабирование признаков

Обучение моделей:

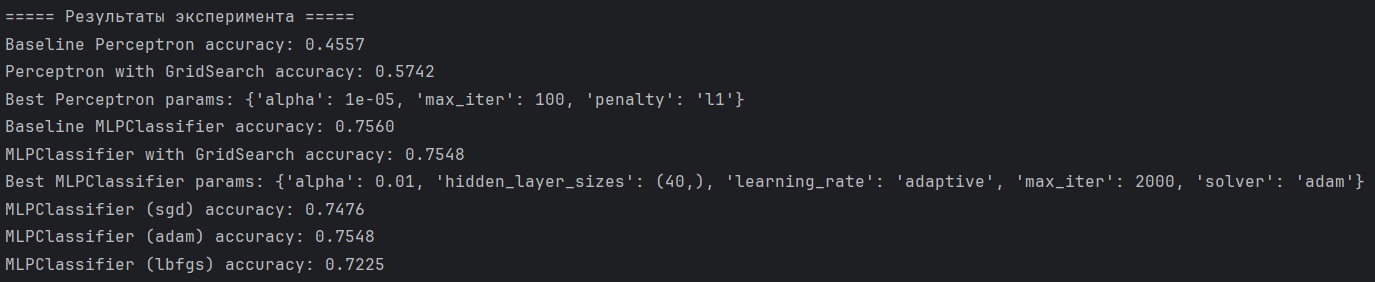
Perceptron: обучается стандартными параметрами, а затем повторно обучается с наилучшими параметрами, найденными через GridSearchCV.

MLPClassifier: также обучается стандартными параметрами, а затем повторно обучается с наилучшими параметрами, найденными через GridSearchCV. Для модели Perceptron сначала была обучена базовая версия, показавшая точность около 45%. Далее был проведён подбор параметров регуляризации, коэффициента регуляризации и максимального числа итераций с помощью GridSearchCV, что повысило точность до 57%. Для модели MLPClassifier была также обучена базовая версия с максимальным числом итераций 5000 и проведён подбор параметров, включая выбор алгоритма оптимизации (solver), число нейронов в скрытом слое, коэффициент регуляризации и скорость обучения. Результаты показали, что MLPClassifier достигает точности порядка 75%, с наилучшими параметрами, подобранными с помощью GridSearch.

Особое внимание было уделено сравнению трёх алгоритмов оптимизации: sgd, adam и lbfgs. При фиксированных параметрах скрытого слоя и числе итераций они были обучены отдельно, что позволило выявить, что adam и sgd показывают лучшие результаты, чем lbfgs.

Стохастический градиентный спуск (SGD). Позволяет постепенно приближаться к оптимальному решению, обновляя параметры модели на основе случайных подвыборок данных.

В целом, использование масштабирования, подбор гиперпараметров и исследование различных методов оптимизации существенно влияют на качество моделей и позволяют достичь высокой точности классификации.



**Рис 4.** Результат