**Введение**

Машинное обучение - один из наиболее обширных и значимых разделов искусственного интеллекта, основным назначением которого является построение обучаемых моделей и оценка их качества. Среди алгоритмов машинного обучения обычно различают обучение с учителем (supervised learning) и обучение без учителя  (unsupervised learning). Классический пример обучения с учителем - обучение многослойного перцептрона методами Back propagation и Resilient propagation. В этом же ключе можно упомянуть и метод опорных векторов (support vector machine), а также деревья принятия решений (decision trees). Среди алгоритмов обучения без учителя наиболее известным и, зачастую, упоминаемым в первую очередь является алгоритм K-средних (K-means).

В обучении с учителем алгоритм получает несколько примеров пар «входные значения – выходные значения» и обучает функцию, которя ставит выходы в соответствие входам. С другой стороны, в обучении без учителя испытуемая система должна обнаружить скрытые взаимосвязи между объектами без вмешательства со стороны экспериментатора [1].

Наиболее базовыми задачами машинного обучения являются задачи классификации и кластеризации данных. Задача классификации состоит в том, чтобы определить класс объектов из некоторой тестовой выборки на основе имеющихся данных о классах объектов из обучающей выборки. Задача же кластеризации состоит в организации объектов в однородные группы на основе совокупности признаков. При этом задача классификации относится к классу задач обучения с учителем, в то время как задача кластеризации относится к классу обучения без учителя.

Одним из наиболее интересных инструментов машинного обучения являются нейронные сети, которые могут решать широкий круг задач обработки и анализа данных: распознавание и классификация образов, прогнозирование, управление, автоматическая торговля, анализ естественного языка, медицинских данных и т.д. Конкурентами нейронных сетей являются классические методы анализа данных, однако нейронные сети имеют над ними ряд преимуществ. Используя способность обучаться на множестве примеров, нейронная сеть способна решать задачи, в которых неизвестны закономерности развития ситуации и зависимости между входными и выходными данными. Нейронные сети устойчивы к зашумленности входных данных и способны адаптироваться к изменениям окружающей среды. Также нейронные сети обладают потенциальным сверхвысоким быстродействием и значительной отказоустойчивостью за счет массового параллелизма обработки информации.

Практически в каждой предметной области при ближайшем рассмотрении можно найти постановки нейросетевых задач [2]:

* Экономика и бизнес: предсказание рынков, автоматический дилинг, оценка риска невозврата кредитов, предсказание банкротств, оценка стоимости недвижимости, выявление пере- и недооцененных компаний, автоматическое рейтингование, оптимизация портфелей, оптимизация товарных и денежных потоков, автоматическое считывание чеков и форм, безовасность транзакций по пластиковым карточкам.
* Медицина: обработка медицинских изображений, мониторинг состояния пациентов, диагностика, факторный анализ эффективности лечения, очистка показаний приборов от шумов.
* Авионика: обучаемые автопилоты, распознавание сигналов радаров, адаптивное пилотирование сильно поврежденного самолета.
* Связь: сжатие видео-информации, быстрое кодирование-декодирование, оптимизация сотовых сетей и схем маршрутизации пакетов.
* Интернет: ассоциативный поиск информации, электронные секретари и агенты пользователя в сети, адресная реклама, коллаборативная фильтрация.
* Политические технологии: анализ и обобщение социологических опросов, предсказание динамики рейтингов, выявление значимых факторов, объективная кластеризация электората, визуализация социальной динамики населения.
* Безопасность и охранные системы: системы идентификации личности, распознавание голоса, распознавание лиц, автомобильных номеров, мониторинг информационных потоков.

Однако есть у нейронных сетей и недостатки. Как следствие отсутствия формализованных алгоритмов настройки сети и ее высокой сложности, возникает необходимость в привлечении высококлассных специалистов, обладающих необходимыми знаниями. Отсутствует строгая теория по выбору архитектуры нейронной сети под специфические задачи, хотя некоторая работа в этом направлении ведется. Из обученной сети практически невозможно извлечь приобретенные знания, то есть, нейронная сеть является своего рода черным ящиком.

Для реализации нейросетевых алгоритмов используются различные языки программирования, в том числе и специализированные диалекты LISP. Для написания реально используемых систем часто используют С++ благодаря его чрезвычайно мощным возможностям параллельного программирования, в том числе и на GPU, а также высокой скорости выполнения. В то же время, для прототипирования приложений, связанных с машинным обучением, обычно используются более специализированные языки: Matlab, R, Python.

В то время, как для задачи классификации используются многослойный перцептрон, метод опорных векторов и деревья принятия решений, для задачи кластеризации используют нейронные сети Кохонена, алгоритмы K-means, Affinity propagation, Spectral clustering, Hierarchical clustering, DBScan и Gaussian mixtures.

Наиболее серьезным препятствием перед использованием алгоритмов машинного обучения в реальных условиях является выбор оптимального алгоритма для определенной задачи. Помочь в этом могут алгоритмы оценки качества обучения. Однако и они тоже подвержены проблеме оптимального выбора. Разные метрики, предназначенные для разных алгоритмов машинного обучения, оценивают алгоритмы с разных позиций. Нельзя с уверенностью сказать, какая метрика будет более точно отображать реальную эффективность алгоритма на определенном наборе данных. Поэтому целью данной работы будет являться создание инструмента, позволяющего исследовать поведение различных алгоритмов классификации и кластеризации на различных наборах данных с помощью различных метрик.

Основным языком для приложения является Python – популярный язык программирования, используемый для разработки самостоятельных программ и различного рода прикладных сценариев в самых разных областях применения [3]. Связано это, прежде всего, с возможностью использования библиотек машинного обучения PyBrain и Scikit-learn о которых подробнее будет рассказано в постановке задачи.