**Лабораторная работа №1**

1. **Название работы:**

Алгоритм K ближайших соседей

1. **Цель работы:**
2. Понимание принципа работы алгоритма K ближайших соседей (KNN).
3. Умение подбирать оптимальное количество соседей для алгоритма.
4. Применение KNN на практических примерах.
5. **Основные задачи работы:**
6. Научиться предобрабатывать данные используя статистические методы.
7. Научиться разделять выборку на тестовую и обучающую.
8. Научиться оценивать результаты модели.
9. **Теоретическая часть работы:**

**Алгоритм K ближайших соседей (KNN)** - это один из самых простых алгоритмов машинного обучения. Идея KNN основана на предположении о близости объектов: если объекты "похожи", то их целевые переменные также будут "похожи". Похожесть объектов определяется на основе некоторой функции расстояния, например, Евклидова расстояния.

В контексте классификации алгоритм KNN предсказывает класс нового объекта на основе классов его K ближайших соседей в обучающем наборе. Самый распространенный класс среди соседей выбирается в качестве предсказанного класса.

1. **Порядок выполнения работы:**
2. **Создание или загрузка набора данных**: В зависимости от варианта необходимо создать искусственный набор данных или использовать один из предоставленных.
3. **Предварительная обработка данных**: Подготовьте данные для обучения. Данные могут требовать нормализации, кодирования категориальных признаков или заполнения пропущенных значений.
4. **Разделение данных на обучающую и тестовую выборки**: Разделите данные на обучающую и тестовую выборки, чтобы можно было оценить качество модели.
5. **Моделирование алгоритма KNN**: Смоделируйте алгоритм KNN, выберите подходящее количество соседей и обучите модель на обучающей выборке.
6. **Оценка модели**: Оцените качество модели с использованием подходящих метрик, таких как accuracy score.
7. **Теоретические вопросы и ответы:**
8. Что такое алгоритм K ближайших соседей?

*Это алгоритм машинного обучения, который классифицирует новые объекты на основе классов их ближайших соседей в обучающем наборе. "Близость" определяется с помощью функции расстояния, например, евклидова расстояния.*

1. Как выбирается класс объекта в алгоритме KNN?

*Класс объекта определяется на основе большинства классов его K ближайших соседей.*

1. Как влияет количество соседей на работу алгоритма?

*Если выбрано слишком малое количество соседей, модель может быть переобучена и чувствительна к шуму. Если выбрано слишком большое количество соседей, модель может быть недообучена и показывать плохое качество на новых данных.*

1. Какую метрику Вы использовали в алгоритме KNN?

*Евклидова.*

**Вариант 1.**

Имеется N точек в R[0, 10]xR[0, 10]. Эти N точек разделены на k кластеров: красные, синие, зеленые и др. Эти точки составляют собой тренировочную выборку данных. Необходимо подготовить тестовые данные и реализовать $knn$ алгоритм классификации для новых 0.1\*N, начертить вывести матрицу ошибок (confusion matrix) и оценку точности предсказания (accuracy score), показать работу алгоритма графически(вывести раскрашенные точки тренировочной выборки; вывести точки тестовой выборки с, определенным алгоритмом knn, цветом и окружностями, включающими k ближайших соседей).

Этап 0. Подготовьте данные для тестирования алгоритма.

Необходимо подготовить тренировочные данные, чтобы можно было наблюдать видимое различие между разными сегментами.

Например, красные точки лежат в левом нижнем углу квадрата R[0, 10]xR[0, 10], cиние по центру, зеленые в правом верхнем углу.

Этап 1. Создайте тестовую выборку точек с предопределенными значениями цвета. Используя евклидову метрику, найдите k ближайших соседей к каждой точке, определите цвет в зависимости от доминирующего цвета соседних точек, нарисуйте окружность, в которую входят k соседних точек.

Этап 2. Оцените качество модели с использованием метрики accuracy score и выводом матрицы ошибок.

**Вариант 2.**

Имеется датасет с признаками пациентов проходящих проверку на диабет и в последнем столбце указан флаг принадлежности пациента к диабетикам (diabetes.csv). Необходимо произвести предобработку признаков. Произвести оценку качества работы knn алгоритма для этих данных.

-Используйте библиотеку pandas для работы с csv файлом.

-Также для простоты вычислений можно использовать библиотеку scikit-learn.

Этап 0. Избавиться в исходных данных от пропусков данных ( Они имеют значение 0. Заменить на среднее\моду\медиану, для номинальных признаков вероятностное). Разделить исходный набор данных на тренировочный и тестовый. (Попробуйте использовать from sklearn.model\_selection import train\_test\_split).

Этап 1. Нормируйте исходные данные (попробуйте использовать fit\_transform()). Обратите внимание, что признаки необходимо нормировать и в тестовых, и в тренировочных данных.

Этап 2. Выберите k и с использованием евклидовой метрики найдите k ближайших соседей к каждому пациенту тестовой выборки. Предскажите наличие диабета у пациентов тестовой выборки.

Этап 3. Оцените качество модели с использованием метрики accuracy score и выводом матрицы ошибок.

**Вариант 3.**

Имеется датасет с признаками цветов-ирисов. В последнем столбце указан сорт цветка. Необходимо произвести предобработку признаков. Произвести оценку качества работы knn алгоритма для этих данных.

-Используйте библиотеку pandas для работы с csv файлом.

-Также для простоты вычислений можно использовать библиотеку scikit-learn.

Этап 0. Разделить исходный набор данных на тренировочный и тестовый. Данные отсортированы по сорту цветка, поэтому выбор из таблицы данных для обучения и теста должен происходить случайно! (Попробуйте использовать from sklearn.model\_selection import train\_ test\_ split).

Этап 1. Нормируйте исходные данные.(Попробуйте использовать fit\_transform()). Обратите внимание, что признаки необходимо нормировать и в тестовых, и в тренировочных данных.

Этап 2. Выберите k и с использованием евклидовой метрики найдите k ближайших соседей к каждому цветку тестовой выборки. Предскажите принадлежность цветка к сорту тестовой выборки.

Этап 3. Оцените качество модели с использованием метрики accuracy score и выводом матрицы ошибок.