**Самостоятельная работа 4**

Создайте небольшой скрипт для запуска модели машинного обучения

Давайте создадим задание для классификации числовых данных на Python с использованием библиотеки scikit-learn. В этом задании будем использовать готовый датасет, который является классическим датасетом в машинном обучении и содержит данные о характеристиках цветков ириса трёх разных видов.

**Выполняемые шаги:**

1. **Импорт библиотек и данных**
   * Импортировать необходимые библиотеки (scikit-learn, NumPy, pandas).
   * Загрузить датасет с помощью scikit-learn.
2. **Анализ и предобработка данных**
   * Изучить структуру данных с помощью методов head(), describe().
   * Проверить данные на наличие пропусков.
   * Разделить данные на признаки (features) и целевую переменную (target).
3. **Разделение данных**
   * Разделить данные на обучающую и тестовую выборки (использовать функцию train\_test\_split).
4. **Выбор модели**
   * Выбрать и создать модель классификации (например, LogisticRegression, KNeighborsClassifier или DecisionTreeClassifier).
5. **Обучение модели**
   * Обучить выбранную модель на обучающей выборке.
6. **Оценка модели**
   * Оценить точность модели на тестовой выборке с помощью метрик точности (accuracy).
   * Вывести матрицу ошибок и отчёт о классификации для детального анализа производительности модели.

Пример кода на Python

from sklearn.datasets import load\_iris

from sklearn.model\_selection import train\_test\_split

from sklearn.linear\_model import LogisticRegression

from sklearn.metrics import accuracy\_score, confusion\_matrix, classification\_report

# Загрузка данных

iris = load\_iris()

X, y = iris.data, iris.target

# Разделение данных на обучающую и тестовую выборки

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size=0.2, random\_state=42)

# Создание и обучение модели логистической регрессии

model = LogisticRegression(max\_iter=200)

model.fit(X\_train, y\_train)

# Предсказание на тестовых данных

y\_pred = model.predict(X\_test)

# Оценка модели

accuracy = accuracy\_score(y\_test, y\_pred)

conf\_matrix = confusion\_matrix(y\_test, y\_pred)

report = classification\_report(y\_test, y\_pred)

print("Точность модели:", accuracy)

print("Матрица ошибок:\n", conf\_matrix)

print("Отчёт о классификации:\n", report)