Лабораторная работа 3. ОБУЧЕНИЕ С УЧИТЕЛЕМ. ЗАДАЧА КЛАССИФИКАЦИИ

1. Изучение примеров. Изучите примеры: Lab3_Ex1_Basic_Classifiers.ipynb, Lab3_Ex2_LogisticRegr_HotCode.ipynb, Lab3_Ex3_Naive_Bayes.ipynb, Lab3_Ex4 Imbalance.pdf, Lab3_Ex5 ClassificationModel.ipynb

2. Загрузка и подготовка данных.

● В соответствии с индивидуальным вариантом загрузите предобработанный датасет в формате CSV для решения задачи классификации.

● Выделите целевой признак и предикторы. Определить тип задачи классификации (бинарная или мультиклассовая).

● Проверьте баланс классов. В случае дисбаланса классов проведите их балансировку (см. Lab3_Ex4 Imbalance.pdf).

● При разделении данных на обучающую и тестовую выборки используйте два метода:

– **Метод отложенной выборки (hold-out)** с учётом *стратификации*[1], чтобы выборки сохраняли относительную пропорцию классов.

Изучите самостоятельно: онлайн-документация, раздел [train_test_split в Sklearn] https://scikit-learn.org/stable/modules/generated/sklearn.model_selection.train_test_split.html

– **Метод k-fold[2] (кросс-валидация)**.

Изучите самостоятельно:

онлайн-документация, [Кросс-валидация] https://scikit-learn.org/stable/modules/cross_validation.html

онлайн-документация, [k-fold] https://scikit-learn.org/stable/modules/generated/sklearn.model_selection.KFold.html

онлайн-документация, [функция cross_val_score] https://scikit-learn.org/stable/modules/generated/sklearn.model_selection.cross_val_score.html

3. Решение задачи классификации.

● Решите задачу классификации на ваших данных с использованием следующих классификаторов из Scikit-learn:

– Logistic Regression.

– kNN. Исследуйте различные методы расстояния и их влияние на классификацию.

– Naive Bayes.

– SVM, в том числе с ядерными функциями (Linear Kernel, Polynomial Kernel, RBF Kernel, Sigmoid Kernel, Exponential Kernel).

● При необходимости подобрать гиперпараметры для соответствующего классификатора тремя способами (GridSearchCV; RandomizedSearchCV; фреймворк Optuna).

● Постройте итоговую модель классификаторов (см. *Lab3_Ex5 ClassificationModel.ipynb*).

---

[1] **Стратифицированная выборка (stratified sampling)**, это делает функция **test_train_split**, когда устанавливаете параметр **stratify** для меток.

[2] **Основной класс для использования** — **KFold**, также есть функция **cross_val_score**, которая позволяет проще применить кросс-валидацию при моделировании алгоритмов ML

4. Визуализация. Постройте графики для визуализации результатов классификации, чтобы показать, как работают классификаторы. Например, представьте график, демонстрирующий влияние изменения k на точность модели kNN. Для бинарной классификации постройте ROC curve.

5. Оценка качества моделей. Вычислите значения метрик оценки качества для обученных моделей классификации: F1, Accuracy, Precision, Recall, матрица ошибок (confusion matrix), ROC AUC score. Поясните ошибки первого и второго рода.

6. Реализация пользовательских функций.

● Самостоятельно реализуйте вычисление следующих метрик оценки качества модели классификации: Accuracy, Precision, Recall, F1, матрица ошибок.

● Поместите пользовательские функции в отдельный файл[3] и подключите его к основной программе как библиотеку функций.

7. Реализация алгоритма классификатора kNN.

● Самостоятельно разработайте и реализуйте классификатор на основе алгоритма kNN в контексте реализации Scikit-learn.

● Поместите разработанный классификатор kNN в отдельный файл и подключите его к основной программе как библиотеку алгоритмов ML.

● Создайте таблицу (объект DataFrame) и выведите в них наименования используемых классификаторов и значения вычисленных метрик оценки качества как с использованием Scikit-learn (Образец 1).

Образец 1

	Train Data				Test Data			
Классификатор	**F1**	**Accuracy**	**Precision**	**Recall**	**F1**	**Accuracy**	**Precision**	**Recall**
kNN из Sklearn	0.XX	0.XXXX	0.XXXX	0.XXXX	0.XX	0.XXXX	0.XXXX	0.XXXX
программный kNN								

Указания к выполнению:

1). Исследование алгоритма kNN.

Изучите теоретические основы алгоритма kNN, включая:

– принцип работы алгоритма;

– методы вычисления расстояния между точками (например, евклидово, манхэттенское);

– влияние гиперпараметра k на производительность модели kNN.

2). Реализация алгоритма.

Напишите класс для классификатора kNN. Основные функции должны включать:

– метод для обучения модели;

– метод для предсказания класса для новых данных;

– расчет расстояния между точками;

– возможность выбора количества ближайших соседей k.

3). Сравните производительность разработанного классификатора с эталонным классификатором kNN из Scikit-learn.

[3]Этот файл разработан при выполнении **Лабораторной работы №2**.

8. Создание таблицы результатов.

● Создайте две таблицы (объект DataFrame) и выведите в них наименования используемых классификаторов и значения вычисленных метрик оценки качества как с использованием Scikit-learn, так и пользовательских функций (Образец 2 и Образец 3).

● Убедитесь, что при выводе значений предусмотрено необходимое количество знаков после запятой (см. Образец 2 и Образец 3).

Образец 2

	Train Data				Test Data			
Классификатор	**F1**	**Accuracy**	**Precision**	**Recall**	**F1**	**Accuracy**	**Precision**	**Recall**
Logistic Regression	0.XX	0.XXXX	0.XXXX	0.XXXX	0.XX	0.XXXX	0.XXXX	0.XXXX
…								

Образец 3

	Метод hold-out (отложенная выборка)				Метод k-fold (кросс-валидация)			
Классификатор	**F1**	**Accuracy**	**Precision**	**Recall**	**F1**	**Accuracy**	**Precision**	**Recall**
Logistic Regression	0.XX	0.XXXX	0.XXXX	0.XXXX	0.XX	0.XXXX	0.XXXX	0.XXXX
…								

9. Вывод. Напишите вывод о выполненной Лабораторной работе №3, в котором выберите лучшую модель классификации и обоснуйте свое решение.