**Класифікація**

**Мета роботи**

Ознайомитись з основами машинного навчання та аналізу даних для розв’язання задач класифікації, використовуючи методи К найближчих сусідів та за допомогою багатошарового персептрону (Multi-layer Perceptron / MLP).

**Рекомендована література**

**Про роботу з класифікацією в Python**:

<https://scikit-learn.org/stable/modules/neighbors.html>

<https://scikit-learn.org/stable/modules/neural_networks_supervised.html#classification>

<https://www.tensorflow.org/tutorials/keras/classification>

<https://pytorch.org/tutorials/beginner/blitz/cifar10_tutorial.html>

**PyTorch/Tensorflow/Keras/SciKit Learn**:

<https://drive.google.com/file/d/1faM-aL5Bsea5DcXpFVr6ApOL5qWNWved/view?usp=sharing>

<https://drive.google.com/file/d/1tAD_-4hZ1nqe5Fqfyic92hmyrQlZnzPe/view?usp=sharing>

<https://drive.google.com/file/d/1D5m901531N9-v-UZQC77YhtRCTdbhAG4/view?usp=sharing>

<https://drive.google.com/file/d/10-cRqjUrSp_qOquyjKLuzqO8KPA9yA3r/view?usp=sharing>

**Незбалансована класифікація:**

<https://machinelearningmastery.com/smote-oversampling-for-imbalanced-classification/>

<https://towardsdatascience.com/smote-fdce2f605729>

<https://neptune.ai/blog/how-to-deal-with-imbalanced-classification-and-regression-data>

<https://www.tensorflow.org/tutorials/structured_data/imbalanced_data>

**Хід роботи**

**Дані**

В роботі використовується публічний набор даних з https://www.kaggle.com/datasets (для тих, хто претендує на максимальний бал) або інших відкритих джерел.

**Підготовчий етап**

Провести аналіз вибраного набору даних, визначити вхідні та вихідні параметри, візуалізувати залежності входів на виходу, спробувати виявити основні залежності, детектувати аномалії, неповні зразки тощо у даних. Зокрема виконати кореляційний аналіз входів та виходів набору даних, виявити взаємозалежні фактори. Провести підготовку даних до подальшого використання. Обраний та відфільтрований набір даних розбити на навчальну та тестову частину (70% на навчанання, 30% на тест).

**Реалізація моделі**

Написати код у Python / R, який реалізований з використанням двох підходів на вибір:

1. SciKit learn (KNN, MLP)
2. TensorFlow / PyTorch (MLP)

**Аналіз результатів**

1. Оцінка помилок на начальній та тестовій вибірках

2. Порівняння результатів різних підходів (співпали чи ні, причини чому могли не співпасти, runtime тощо)

3. Вивести та обгрунтувати різні метрики оцінки якості моделі. Знати їх значення, математичне формулювання, відмінності від інших наявних метрик.

**Студенти, що розраховують на високий бал мають:**1. Відкоментувати код (що кожна строка робить, окрім import / library)

1. Якісно візуалізувати результати та проміжні етапи (за необхідністю)
2. Обрати порівняно складний датасет, який потребує додаткового очищення
3. Використати TensorFlow / PyTorch.
4. Виконати небінарну класифікацію.

**Контрольні питання**

1. Як звести задачу небінарної класифікації до бінарної?
2. Навіщо нам потрібні функції активації?
3. Які міри відстані бувають в KNN? Їхні особливості та обмеження?
4. Незбалансована класифікація (imbalanced classification) - як формулюється проблема, які основні підходи до вирішення цієї проблеми?
5. Чим відрізняється класифікація від кластеризації?
6. Чи існують неповнозв’язні нейронні мережі? Якщо так, то в чому їх особливість: плюси / мінуси.
7. Як отримати перенавчання в нейронній мережі?

***NB:   
1) Датасет з рукописними цифрами буде оцінюватися особливо прискіпливо!***

***2) Використовувати датасети з навчальних глав у Scikit Learn / PyTorch / Keras / Tensorflow використовувати заборонено!***