COMPX310 – Machine Learning

Lab 1: k-Nearest Neighbors (kNN) Classification

Important Note on Academic Integrity

You must be able to **explain working of your code you submit**. During or after submission, you may be asked to explain how your code works.

- If you cannot explain your own code, it will be treated as **copied work**.
- **Copied code without understanding will receive ZERO marks.**
- AI tools may only be used to **brainstorm ideas** or to **understand concepts**.
- Do **not** copy-paste AI-generated solutions directly into your lab work.

Always write your own code, discuss ideas with a partner if needed, but make sure you fully understand what you are submitting.

Due Date: Sunday, September 21 at 21:00 PM

Worth: 3% of your total COMPX310 grade

Platform:

- **Offline:** You can use a local Python setup with VS Code to complete this assignment. OR
- **Online:** You can use Kaggle (https://www.kaggle.com/) or Google Colab (https://colab.research.google.com/) to complete this assignment.

Dataset

Explore the field of breast cancer diagnosis with the insightful Wisconsin Breast Cancer dataset (Original). You can get the dataset from the Canvas. This dataset provides detailed attributes representing tumor characteristics observed in breast tissue samples. By analyzing these attributes, researchers and medical professionals can gain insights into tumor behavior and develop predictive models for cancer detection and prognosis.

A detailed description of the dataset features is given below:

1. Sample code number: Unique identifier for each tissue sample.
2. Clump Thickness: Assessment of the thickness of tumor cell clusters (1 - 10).
3. Uniformity of Cell Size: Uniformity in the size of tumor cells (1 - 10).
4. Uniformity of Cell Shape: Uniformity in the shape of tumor cells (1 - 10).
5. Marginal Adhesion: Degree of adhesion of tumor cells to surrounding tissue (1 - 10).
6. Single Epithelial Cell Size: Size of individual tumor cells (1 - 10).
7. Bare Nuclei: Presence of nuclei without surrounding cytoplasm (1 - 10).
8. Bland Chromatin: Assessment of chromatin structure in tumor cells (1 - 10).

9. Normal Nucleoli: Presence of normal-looking nucleoli in tumor cells (1 - 10).
10. Mitoses: Frequency of mitotic cell divisions (1 - 10).
11. Class: Classification of tumor type (2 for benign, 4 for malignant).

What You Need to Do

Create a Python notebook and complete the following steps:

- Write your full NAME and Student ID at the top of the notebook.
- Load the dataset. Use seaborn's pairplot for basic visualization.
- Set input features "X" = all columns except "class". Set target "y" = "class".
- Split data using "train_test_split()" with "test_size=0.2" and "random_state=your_student_id".
- Train a kNN classifier and use it to predict on the test set.
- Print the confusion matrix and classification report.
- Train and test kNN classifiers for k from 1 to 100. Record and plot k vs. accuracy.

Use **Markdown cells** to explain each step and discuss results clearly.

Submission Instructions

After you complete your notebook and run all code cells:

- If you're using Kaggle or Colab, click File → Download → Download .ipynb.
- If you're working locally in VS Code, simply save your notebook as a .ipynb file (e.g., Lab1.ipynb).
- Make sure the file is saved to your computer for conversion or submission.
- Open a terminal or command prompt on your computer.
- Run the following command to convert the notebook to HTML:

```
jupyter nbconvert --to html Lab1.ipynb
```

- This will create a file named "Lab1.html" in the same folder.
- Open the HTML file in a browser and check that all outputs are visible.
- Submit the '.html' file on Canvas before the deadline.

Pair Programming (Optional)

You can do this lab with one partner. Make sure:

- Your notebook includes both names and IDs.
- Both of you submit the same PDF notebook to Moodle.

Important Notes

- Always use your student ID as `random_state` when splitting data.

- If working in a pair, use the **sum** of both student IDs.

Hints & Troubleshooting

If you get an error when training the model:

- Check the data types of your features.
- Some models do not support non-numerical values. Convert them if needed.
- If accuracy is low (60–70%), review the features. Do they make sense? Are they clean?
- Ask your tutor in the lab if you are stuck.