DAY 1

Goal: Gain hands-on experience with machine learning tools used in atomistic simulations, as encountered in academic research. You will work with a dataset consisting of allotropes of carbon, both crystalline and amorphous.

To start, please visit and download the GitHub repo:
https://github.com/zakmachachi/Atomistic-ML-Tutorial

Part 1: Understand, explore and visualise atomistic data

1. *Working with atomic structures in Python:*

Task: Load the dataset using load_atoms in python. It is quite large so we will take out x number of structures to work with. Visualise these structures in python by indexing from 0 and viewing it. Cycle through different indices to see the different structure types. View the .xyz file in the text editor to understand the format properly

1. *Generate Descriptors*

Descriptors convert atomic structures into fixed-length numerical features for ML models. A good descriptor is rich in information enabling differentiation between atomic motifs and locality.
Examples include:

- Coordination number
- Pairwise distances
- Angular information

Tasks:

- Generate two or more types of descriptors for your systems.
- Visualise the data. E.g.: coordination number could be shown as a histogram.
- Plot the radial distribution function and angular distribution function for a high density and low density carbon structure.

Part 2: Exploring the structural space of carbon using SOAP

SOAP is many-body descriptor built on spherical harmonics and radial basis functions (which you will have seen in your quantum chemistry courses!). SOAP contains a **lot** of information since large *n* and *l* are used making it difficult to interpret due to the high dimensionality. As a result, dimensionality reduction techniques are used to help understand how we can differentiate between carbon structures and local atomic environments.
Tasks:

- Build SOAP descriptors for a single structure.
- Apply a dimensionality reduction method (PCA) to the SOAP vectors and visualise the two principal components using chemiscope.
- Colour code each environment using coordination number, and then by atomic energy.
- Next, do the same but on 100 structures of varying density. Notice the plot contains a large number of points. Think about how you can make a per structure SOAP vector and plot that.

Part 3: Supervised learning – Predicting local energies

Fit ML models to predict the total energy of structures in the dataset.
Tasks:

- Train regression models using:
  - Linear regression
  - Kernel ridge regression
  - (Optional) Neural networks from scikit-learn
- Compare the performance of models using different descriptors.
- Evaluate model performance with metrics such as MAE and RMSE.

Questions:

- Which descriptor and model combination gives the best results?
- What other structure-property predictions could be explored?

Part 4 (Optional): Predicting NMR Chemical Shifts in SiO_2

Now shift focus to predicting atomic-level properties rather than system-level totals.
Tasks:

- Modify your pipeline to predict chemical shifts on a per-atom basis.
- Update input and output formatting to support atomic-level targets.
- Train and evaluate your model.

Questions:

- What changes were needed to adapt the pipeline for atomic targets?
- How does performance compare to the energy prediction task?
- Would you use a different descriptor for this property?

Summary and Discussion
Group discussion prompts:

- What worked well and what didn't in your ML pipeline?
- Which descriptors were most effective and why?
- How closely did the ML results match physical intuition?
- What aspects of model performance were surprising or insightful?

Optional Bonus Tasks
For those who finish early or want to explore further:

- Use your model to make predictions on unseen or extrapolated structures.
- Generate and interpret learning curves to understand data requirements.
- Train models to predict atomic forces or stress tensors in addition to energies.
- Visualise feature importance or atomic contributions using interpretability methods.