**Chapter 4 – Methodology**

This chapter outlines the machine learning techniques used for predicting the energy density of battery materials using the given dataset. We explore various supervised regression algorithms, drawing on guidance from the textbook \*Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow\* by Aurélien Géron [13], and insights from recent literature in battery informatics and materials science.

**4.1 Linear Regression**

Linear Regression is a foundational algorithm in machine learning, commonly used for predicting a continuous target variable based on one or more input features. In this model, a linear relationship is assumed between the predictors (features like molecular weight, capacity, voltage, etc.) and the response variable (energy density). The prediction function is defined as a weighted sum of input features:

where wᵢ are the weights and b is the bias term. Linear Regression serves as a useful baseline due to its simplicity and interpretability. However, its performance may be limited in the presence of multicollinearity or non-linear relationships.

**4.2 Random Forest Regressor**

Random Forest is a tree-based ensemble learning method that builds multiple decision trees and aggregates their outputs. Each tree is trained on a bootstrapped subset of the training data, and only a random subset of features is considered at each split. This reduces overfitting and variance, making it ideal for tabular data with mixed types.

where hₜ(x) is the prediction of the tth tree. In previous materials science applications (e.g., Jain et al., 2016), Random Forests have demonstrated excellent predictive performance for complex physical properties due to their robustness to noise and non-linear relationships.

**4.3 Support Vector Regressor (SVR)**

Support Vector Regression (SVR) extends the ideas of Support Vector Machines to regression tasks. The goal is to find a function that deviates from actual values by at most a predefined margin ε, while remaining as flat as possible.

where K is the kernel function. While powerful, SVR can be sensitive to feature scaling and parameter tuning.

**4.4 XGBoost Regressor**

XGBoost (Extreme Gradient Boosting) is an optimized implementation of gradient boosting algorithms, designed for speed and performance. It builds models sequentially, with each model attempting to correct the residuals of the previous one.

where fₜ is the new model added at iteration t, and η is the learning rate. XGBoost has shown state-of-the-art results in many machine learning competitions and has recently been applied to materials property prediction due to its ability to capture complex, non-linear interactions among features (Chen et al., 2019).

**4.5 Chosen Methods**

Based on the nature of the dataset and previous literature, the following four regression algorithms were selected for predicting the energy density of battery materials:

1. Linear Regression: Baseline model, high interpretability.
2. Random Forest Regressor: Robust to overfitting, handles non-linearity well.
3. Support Vector Regressor: Capable of modeling complex relationships.
4. XGBoost Regressor: High performance, ensemble model with regularization.

Each model will be evaluated using cross-validation and metrics such as Root Mean Squared Error (RMSE), R² Score, and Mean Absolute Error (MAE). The models will also be compared based on their generalization ability to unseen data.

1. Géron, A. (2019) Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow: Concepts, Tools, and Techniques to Build Intelligent Systems. 2nd edn. Sebastopol, CA: O’Reilly Media.
2. Jain, A., Ong, S.P., Hautier, G., Chen, W., Richards, W.D., Dacek, S., Cholia, S., Gunter, D., Skinner, D., Ceder, G. and Persson, K.A. (2013) 'The Materials Project: A materials genome approach to accelerating materials innovation', APL Materials, 1(1), pp. 011002. doi:10.1063/1.4812323.
3. Chen, T. and Guestrin, C. (2016) 'XGBoost: A scalable tree boosting system', in Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD ’16), San Francisco, CA, 13–17 August. New York: ACM, pp. 785–794. doi:10.1145/2939672.2939785.