Bias / Variance Exercises: (Deadline: $MAY,\ 25^{th}$)

- Implementing Bias-Variance Decomposition: Write a function that takes as input a dataset (X, y) and a machine learning model. The function should perform the following steps:

  1. Split the dataset into training and testing sets.
  2. Train the model on the training set.
  3. Make predictions on the testing set.
  4. Calculate the mean squared error (MSE) between the predicted values and the true values.
  5. Calculate the bias and variance using the bias-variance decomposition formula.

- Bias and Variance Visualization: Create a script that generates a synthetic dataset with a single input feature and a target variable. Use a known mathematical function to generate the true relationship between the input feature and the target variable. (For example, if you choose a linear relationship between the input feature and the target variable, you can define a mathematical function like: *target = m * input_feature + c*) Then, implement a machine learning model (e.g., linear regression, polynomial regression) and train it on the dataset. Generate predictions for different subsets of the dataset (e.g., subsets with different sample sizes). Plot the bias and variance as a function of the sample size. Visualize how the bias decreases and variance increases with increasing sample size.

- Regularization and Bias-Variance Trade-off: Implement a Python function that takes as input a dataset (X, y) and performs the following steps:

  1. Split the dataset into training and testing sets.
  2. Implement a regularized regression model (e.g., Ridge regression, Lasso regression).
  3. Train the regularized model on the training set.
  4. Make predictions on the testing set.
  5. Calculate the mean squared error (MSE) between the predicted values and the true values.
  6. Compare the bias and variance of the regularized model with that of a non-regularized model (e.g., linear regression).
  7. Discuss the bias-variance trade-off observed between the regularized and non-regularized models.