# *Advanced Topics in Computer Science II (420-G50-HR)*

# *Lab 14 – Linear Regression ML*

Date due: **April 17, 2025**

**Learning Objectives**

Upon successful completion of this lab exercise, the student will be able to:

* Go through the 5 steps of Machine Learning to create a simple model.

Lab Set-Up

1. Download the lab folder from Moodle and rename it to *initials*G50L14 where *initials* are your initials.

To Do

**Part A – Something Fishie**

1. Create a file called fish.ipynb
2. Collect the Data
   1. Use pandas to access the csv file fish.csv from the data\_files folder.
3. Explore the Data
   1. Print the information about the dataset, the description of the dataset and the first five rows of the dataset. The dependent feature is Weight, all other features are independent.
   2. Visualize the data (using matplotlib) to show the relationship between three of the (non-string) independent features and the weight.
   3. Visualize the data to show the distribution of the dependent variable.
4. Prepare the Data
   1. Linear regression cannot have non-numeric data so the Species feature must be removed. Use the drop method on the DataFrame to drop the Species feature.
      1. You will either assign this to a copy of the DataFrame or add the parameter inplace=True to the drop command to have it alter the initial DataFrame.
   2. Set the response variable to the Weight feature and set that feature to the y variable.
   3. Set the predictors to the list of all the (remaining) columns.
   4. Remove the response column from the predictors and assign x to the data for all the independent variables.
5. Modeling
   1. Import train\_test\_split from sklearn model\_selection
   2. Use train\_test\_split to create x\_train, x\_test, y\_train and y\_test from the x and y variables create. Set the random state to 1234 so it is repeatable.
   3. Import LinearRegression from sklearn linear\_model
   4. Create the model using the fit method of the LinearRegression class and the training data.
   5. Display the model intercept, coefficient.
   6. Display the model score using the test data.
   7. Create a y\_pred variable using the x\_test data.
   8. From sklearn metrics import mean\_absolute\_error and display the error based on the tested versus predicted results.
      1. The accuracy should be about **86.9%** and the mean error just over **105**.

**Part B – Life Expectancy (maybe if you eat fish?)**

1. Create a file called life.ipynb
2. Collect the Data
   1. Use pandas to access the csv file lifeExpectancy.csv from the data\_files folder
3. Explore the Data
   1. Print the information about the dataset, the description of the dataset and the first five rows of the dataset. The dependent feature is LifeExpectancy, all other features are independent.
   2. Visualize the data (using matplotlib) to show the relationship between the following independent features and the **LifeExpectancy**:
      * Schooling
      * BMI
      * AdultMortality
      * InfantDeaths
   3. **Add a comment to your code to explain what each plot reveals about the data.**
   4. Visualize the data to show the distribution of the dependent variable.
   5. **Add a comment to your code to explain what this plot reveals about the data.**
4. Prepare the Data
   1. Linear regression cannot have non-numeric data so the two non-numeric features must be dropped.
   2. This data contains a lot of missing rows. Start by dropping all the instances with missing values. This is done using the dropna command as per the class examples. This will reduce the number of rows by more than 40% which, in the end, will not be acceptable, but start there.
   3. Once again, you can assign this to a new variable or use the inplace=True parameter to dropna.
   4. Set the response variable to the LifeExpectancy feature and set that feature to the y variable.
   5. Set the predictors to the list of all the (remaining) columns.
   6. Remove the response column from the predictors and assign x to the data for all the independent variables.
5. Modeling
   1. Import train\_test\_split from sklearn model\_selection
   2. Use train\_test\_split to create x\_train, x\_test, y\_train and y\_test from the x and y variables create. Set the random state to 1234 so it is repeatable.
   3. Import LinearRegression from sklearn linear\_model
   4. Create the model using the fit method of the LinearRegression class and the training data.
   5. Display the model intercept, coefficient.
   6. Display the model score using the test data.
   7. Create a y\_pred variable using the x\_test data.
   8. From sklearn metrics import mean\_absolute\_error and display the error based on the tested versus predicted results.
      * The accuracy should be about **83.5%** and the mean error of about **2.75**.
6. Now go back and try to improve your model to get at least an 85% accuracy. There are three things/combinations to try:
   1. Fill in the missing values for some fields with the median of that missing value.
   2. Use a larger training set (the default is 75%)
   3. Remove features that do not seem to have a strong correlation with the dependent variable (use your plots from “Explore the Data” to do this. You can also leverage <<dataframe>>.corr() which you learnt in previous labs

**To submit**

When you have completed the lab exercise, call the Teacher’s attention and we’ll go over it together. Then, create a single zip file called *initials*G50L14.zip and copy the file to the Moodle page for the course.