ECE 157A
Homework 1

6.1 Tutorial Quiz

Python Basics

1. Import torch
2. from ucimlrepo import fetch_ucirepo
3. import datetime as dt
4. import os; os.makedirs("./results/model a/run 1", exist_ok=True)
5. def add(a, b):
   ```
   def add(a, b):
       return a + b
   ```

Data Preparation

1. selected_features = ['alcohol', 'density', 'chlorides']
   X = df[selected_features]
2. Building solid machine learning models begins with making sure the distributions of the training and validation data match. Differences between these datasets can produce false performance measurements, and that could jeopardize the model's applicability in real-world situations. By being consistent across datasets, we increase the model's capacity to effectively generalize to possible situations.

Model Building

1. The parameters we need to pass to .fit are X (the training features) and Y (the target labels)
2. Two classifiers—Decision Tree and Logistic Regression—are trained and assessed in the tutorial. For simplicity, the Decision Tree's maximum depth is set to 5, and the accuracy of the training and validation sets is determined. It is possible to see the Decision Tree and save it as "tree.png." For Logistic Regression, default parameters are applied, and both training and validation data accuracy are evaluated. Heatmaps are used to display feature weights. A test dataset is also used to evaluate both models; on this dataset, Logistic Regression scored 54.56% accuracy and Decision Tree achieved 51.65% accuracy.

6.2 Building a Model on a New Dataset

Problem formulation: The UCI Machine Learning Repository's Abalone dataset was compiled with the intention of developing a quicker method of determining an abalone's age than the harder and usual way of piercing the shell and counting rings. The abalones' sex, length, diameter, height, and other weight measurements are all included in this dataset. In this scenario, we use a classification, potentially classifying the abalones into age groups like "young," "middle-aged," or "old." In that case, we used a classification strategy..

Dataset description: The Abalone dataset focuses on the prediction of an abalone's age based on its physical measurements. The traditional method of determining an abalone's age involves counting the number of rings on its shell under a microscope, a method both hard to apply and very time-consuming. This dataset aims to facilitate a more efficient prediction method through machine learning. It contains about 4177 instances which will help us build a more robust model.

Features:

- Sex: data that indicates the gender of the abalone: Male (M), Female (F), and Infant (I).
- Length: data representing the longest shell measurement.
- Diameter: data, perpendicular to length.
- Height: data showing the height with meat in the shell.
- Whole weight: data representing the full weight of the abalone.
- Shucked weight: data indicating the weight of the meat.
- Viscera weight: data showing the gut weight post bleeding.
- Shell weight: data representing the weight after drying.

Label : The label in this dataset is the age of the abalones, aka the number of rings on their shells. This can be categorized for classification tasks, such as 'young', 'middle-aged', or 'old', depending on the number of rings.

Model Selection: The Decision Tree model is really good at capturing the relation between an abalone's physical characteristics and its age, which is why I selected it for the Abalone dataset. Decision Trees are excellent in spotting and segmenting these complex interactions between an abalone's size, weight, and age. The model's hierarchical structure makes sure that key patterns in the data that are important for determining age are not missed. Also, given that the dataset aims to make age prediction simpler, the Decision Tree delivers not just accuracy but also a straightforward, logical method, making it suitable for the task.

Feature Selection: I began by getting the Abalone dataset from UCI's web page and started analyzing the data to get its composition and the inherent statistics. Then, the data was put into histograms, providing a snapshot of the distribution of the dataset's features and making it easier for us to understand the data given. Additionally, a correlation heatmap helped us find relationships and interdependencies among features and the target variable, 'Rings'. Based on all of these, I ended up choosing specific features like 'Length', 'Diameter', 'Height', 'Whole_weight', 'Shucked_weight', 'Viscera_weight', and 'Shell_weight' for modeling because of their correlation with the age of the abalone. With that, I then split the data into training and validation sets, ensuring a balanced representation of the target variable. After that, I started training a Decision Tree Classifier. After the training, I evaluated the model's proficiency on both these datasets, and ended with a visualization of the decision-making structure of the trained model.

Data Splitting: I used a train/validation data split ratio of 80:20. Specifically, 80% of the data was allocated for training purposes, while the remaining 20% was reserved for validation. To ensure that the data distribution remained consistent between the training and validation datasets, I visualized the distribution of the target variable, 'Rings', for both sets using histograms. By comparing the 'Training Label Distribution' histogram with the 'Validation Label Distribution' histogram, I could make sure that both sets had a similar distribution of the target variable, ensuring that the model was not biased during training or evaluation.

Model Evaluation: For model evaluation on the Abalone dataset, I primarily relied on accuracy as a metric. After training the Decision Tree model on the training set, I used it to predict the age ('Rings') of abalones in both the training and validation sets. I then compared these predictions with the actual ages to calculate the proportion of correct predictions, thereby determining the accuracy. This method provided a straightforward number to show us how well the model performed. Additionally, to go even deeper into the model's decision-making process, I visualized the trained Decision Tree using the graphviz library.

Best Model's Performance and Insights: Using the Abalone dataset with a Decision Tree, we found out some interesting things. The model was right about 31.31% of the time when looking at the data it learned from. That might not sound high, but guessing an abalone's age based on its rings is tricky. There are a lot of things that can change how an abalone grows. When we tried the model on new data, it was right about 26.44% of the time. Looking closely, we saw that things like the abalone's 'Length', 'Diameter', and 'Shell_weight' were important for guessing its age. This means these features have a close link to the age of the abalone but guessing an abalone's age isn't easy which is why the numbers given back by our current model might not look very high.