Prince Genel R. Umali

BSCS – 3B

Task Title: "My First End-to-End ML Experience"

The first end-to-end ML experience taught me a lot about making a model. The dataset for building a model is the California Housing Dataset, which trains the model to predict house prices based on the features of the dataset like MedInc, HouseAge, AveRooms, and so on. At first, all I know is to load the data and train it, but it must go step by step, like import the libraries, load the data, check and handle for missing values, and so on, which can help me to improve the model.

I learned that ML is not just coding the model and training it. But it is about understanding the dataset first and how to prepare it. First is import all the libraries that will be needed for training a model and cleaning the dataset. Then load the dataset and see how many rows and columns there are in the dataset. Next is by using the info(), which helps me to see how many rows are missing and their data types for each column. If the dataset has missing values or duplicates, it must use pandas to handle the problems in the dataset. To understand more of the data, it should have data visualization to learn more about the features of the dataset, and to see how it relates to the house prices. After that, I use a library that helps me split the data to split into a training set that has 80% of the dataset and a test set for 20%, so that it can train my models on one part of the data and test my model for its accuracy on unseen data. Then, scaling the dataset so that the data of all made numbers have a similar size because some models are sensitive to differences in scale, which can lead to either overfitting or underfitting.

Next, import the libraries of linear regression and the decision tree regressor. Then write the code to train the models. After training the data from both models, it is time to evaluate their performance using the RMSE (Root Mean Squared Error) and R^2 score. The RMSE tells me the average of how far the errors of my predictions are. In my linear regression, the RMSE of the model is 0.7456, and the Decision tree is 0.6497, which shows that the decision tree is better because it has less error than the linear regression. The R^2 tells how much of the variation in house prices is in the model. In this metric, the decision tree performs better than linear regression. Next is plotting the actual values vs predicted values for both models so that I can see how accurate the models and decision tree plots are have the accurate model to the linear regression in the dataset.

The most challenging part of this is cleaning the dataset, if the dataset has null values or duplicates, it will need compensation because once there is a dataset that has null values or duplicates, it will lose the accuracy of the model. It is important to double-check the dataset to clean all the data so that the model has a better performance.

An example in a real-world scenario is a website of a car dealership that predicts the fair prices for vehicles. By feeding information about the car, it will help to determine

the sale price of the car. So that the dealership prices are fair and not overpriced or underselling.

In conclusion ML pipeline helps us to build a model step-by-step. It uses a dataset for the model to train on and a test set. Then it needs some libraries for the models and the cleaning of the dataset. And see the dataset if it has null values or duplicates, then determine the features that will be needed for the model. Then train the model and use metrics to see the error and the score of the models to see which is better. Then use visualization to see the actual and predicted values of the model.