Lab Assignment 2 - Linear Regression - Part 1

Build a **Linear Regression Model** for predicting Car Sales Prediction.

Steps

1. **Dataset:** Download the dataset from the link https://github.com/chandanverma07/DataSets/blob/master/Car_sales.csv. The target Y of the dataset is the column “Sales in thousands”, where other columns are features X.
2. **Pre-processing**
    - **Encoding:** Some of the categorical values in the dataset can be converted into numerical using one-hot encoder. If there are too much unique values in a categorical column then you can think of whether to encode them or drop them.
    - **Normalization:** Since the features are in different ranges, each column can be normalized into 0 to 1 using different methods such as scaling, standardizing etc. This part you have to explore. Note: Normalization should not be done for the target feature.
3. **Data Splitting:** After the range normalization, its time to split the data into training and testing. Dataset should be randomly shuffled. Split the dataset into 60% for training and rest 40% for testing. You can utilize builtin functions (like train_test_split) in sklearn for this task.
4. **Linear Regression Training:** Now we have to train a linear regression model using model written from scratch or using buildin functions from libraries. It is preferred you try both and come up with some analysis.
5. **Testing:** Test the model with the test data and compute the mean squared error (MSE) for test data. Explore more error/similarity measures for regression.
6. **Model Analysis:** After training print the model parameters i.e weights and bias learned. Print the important features from the dataset based on their absolute weight value learned in the model.
7. **Model Experimentation:** You can try different strategies to see whether testing error comes down or not. Strategies can be different
    - Testing between whether label encoder vs one hot encoder for categorical features gives better results.
    - Removal of unimportant features and training with smaller set of features.
    - Running model with different feature scaling methods (i.e. scaling, normalization, standardization etc using sklearn)
    - Training model with different sizes of dataset splitting such as 60-40, 50-50, 70-30, 80-20, 90-10, 95-5 etc
    - Shuffling of training samples with different random seed values. Check the model error for the testing data for each setup.

Reference: Algorithm can be studied from https://machinelearningmastery.com/linear-regression-tutorial-using-gradient-descent-for-machine-learning/. One hot encoding https://scikit-learn.org/stable/modules/generated/sklearn.preprocessing.OneHotEncoder.html. Feature scaling https://scikit-learn.org/stable/modules/preprocessing.html.