Machine Learning Assignment 1

Prepare and upload one zip file which you will name as <your first name>_<your last name>_ assignment1.zip. This zip file should contain all of the materials used in this assignment except the dataset.
hacertilbec@std.sehir.edu.tr

In this project, you will do exploratory data analysis to understand your dataset and your features, do feature processing, use machine learning methods on real data, analyze your model and generate predictions using those methods. This project is mainly about the examples in the 2. chapter of the course book with some changes. You may benefit from the book. This project is 5% of your overall grade.

Dataset

This dataset is provided for you to test your codes. We will test your functions with different datasets too. Be sure your functions can work in different datasets.
To install the dataset, you need to follow the link below. It will automatically download the .tgz file which contains the training dataset, available in .csv format.

Download link: https://raw.githubusercontent.com/ageron/handson-ml/master/datasets/housing/housing.tgz

Data Analysis & Pre-processing (70 points)

In Data Analysis & Pre-processing part, you are going to implement some functions using the scikit-learn library. You will analyze your dataset and change some values in it in order to prepare your data to be used in machine learning models.

You have been provided with the python script *analysis_and_preprocessing.py* in the assignment folder. Within this script, you have skeleton functions defined for you.

(a) The *create_df* function should take the path of the csv file and turn it into a pandas data frame. **(5 points)**

(b) The *nan_columns* function should take a data frame as an input and return a list of names of columns that contain nan values in it. **(5 points)**

(c) The *categorical_columns* function should take a data frame as an input and return a list of column names that contain categorical values in it. **(5 points)**

> For example, if a data frame has 'gender' column and 'female' and 'male' values in it, then 'gender' column should be in the returned list.

(d) The *replace_missing_features* function should take a data frame and a list of column names that contain nan values as input.The function should replace all nan values in the column with the median of this column's values and return this new data frame as output. **(5 points)**

(e) The *cat_to_num* function should take a data frame and a list of categorical feature column names as input. Perform one-hot-encoding on categorical features. Modify your data frame and replace its nominal features with their one-hot encoding representation. **(15 points)**

(f) The *standardization* function should take a data frame and label column as input and scale all columns except the label column with standardization. Return a new data frame with scaled values. **(10 points)**

> Scikit-Learn provides a transformer called StandardScaler for standardization. The output of the scaler is an array, you need to convert it dataframe after standardization. Don't forget to add indexes and columns of the original dataframe to the new dataframe.

(g) The *my_train_test_split* function should take a data frame, name of the label column and percentage of test size as input. Split dataframe as X and y where y is the label column values and X is the feature values (all column values except the label column). Then, the function should split X and y into test and train sets as

X_train, X_test, y_train and y_test with the given test size. Output datatype should be numpy array. **(10 points)**

> For example, if the test size is 0.3, then the function should divide the data frame into 70% training and 30% test data.

(h) The *main* function will use all functions you created and return a train and test set at the end. The function should take path of a csv file, test data percentage and name of the label column as an input. First, convert the csv data (which you downloaded at the beginning) into a data frame (a). Fill nan columns of this data frame (b). Find in which column there are categorical values (c). Fill up missing features in data frames (d). Convert categorical features into numerical format (e). Scale all feature columns with standardization (f). Split the final data frame into train and test numpy arrays according to given label column and test ratio. Return *X_train, X_test, y_train* and *y_test* numpy arrays (g). **(15 points)**

Model Evaluation (30 points)

In this section, you are going to train a Linear Regression model that learns to predict the label column according to rest of the features in our dataset.

You have been provided with the python script custom_model.py in the assignment folder. Within this script, you have skeleton functions defined for you.

(i) *model_evaluation* function should take training instances (train_x) and labels (train_y) as numpy arrays. Create a LinearRegression model with default parameters. Train the model with the train_x and train_y. Your function should return *coef_* and intercept_ arrays of the model as output. **(10 points)**

(j) *predict* function should take instance array, coef_ array and intercept_ array as input. This function will create a Linear Regression model and set its coef_ and intercept_ parameters to input coef_ and intercept_ values. Using this model, predict the label column values of given instance/s. Return list of predictions as output. **(20 points)**

Note: We will test your program with any number of instances so you should take that into account.