YAP470/BIL570 Artificial Learning
Term Project Implementation Guide

Group ID : 28

Students :
Adem ŞAHİN 221211022 ademsahin@etu.edu.tr

Instructor : Batuhan Bardak

1. Files

There are 3 code files that need to be used to implement this project:

- missing_values.py
- encoding.py
- deploy_local.ipynb

Test dataset must be saved to working directory as 'test_file.pkl' alongside python modules and other pickle files.

2. Required Libraries

Required libraries are given in "requirements.txt".

3. Code Explanation

1. missing_values.py

This module includes functions that I wrote to implement imputation on the training and test datasets. There are also some auxiliary functions to make the flow clearer.

Functions

a. get_relevant_feats(data, relation_threshold)

This function calculates correlation matrix of the features and returns a list of features for each feature.

b. get_RF_dict(data, relevant_feats)

This function creates a Random Forest Regressor for each feature in the dataframe, trains them. It returns a python dictionary where keys are the feature names and values are the corresponding regressors.

c. *get_LR_dict(data, relevant_feats)*

This function is the same as “get_RF_dict” except that this returns linear regressors for features.

d. *get_mean_mode(data)*

This function returns the means or modes of the dataframe according to data types of the features.

e. *Simple_imputer(data, mean_mode)*

This function makes simple imputation. It fills the missing values with mean or mode of the other training dataset, which should be given as an argument to the function.

f. *my_train_imputer(data, relation_threshold, regressor)*

This function imputes the training dataset in my way. It returns the imputed dataframe, regressors for the features that are created during the imputation, and means and modes of the features.

g. *my_test_imputer(data, regressor_dict, mean_mode)*

This function imputes the test dataset in my way. It needs the regressors of the features for the imputations of features. It also needs the means and modes of the training dataset in order to use as a preliminary step in imputation.

2. encoding.py

This module includes my encoding scheme, which is a part of preprocessing and comes before imputation.

Functions

a. *my_encoder(data, encoder = None)*

This function make encoding, which is a little specialized encoding for the housing datasets. It converts literal grades into numerical grades, then makes ordinal encoding. It returns the encoded dataframe.

This function takes an optional argument, encoder. If a training dataset will be encoded, this argument is left as None. However, if a test dataset will be encoded, the encoder, which was trained with training dataset, should be given as argument to the function.

3. deploy_local.ipynb

This notebook is the main code that creates the predictions in csv format. Test file should be saved in the local directory as pickle file.

4. streamlit_deployment

This python module is required for deployment on the cloud.

4. Prediction

Once the ‘deploy.ipynb’ notebook is run, the predictions will be saved to current directory with the name ‘predictions.csv’