COMP 468 Machine Learning in Python, Spring 2023
Instructor: Zafer Aydın
Homework 3

Setup

1. Login to your Kaggle account.
2. Click on the Code link.
3. Click on the New Notebook button.
4. Change the title of the page on the upper left corner so that it obeys this format: “ML in Python, Spring 2023, Homework 3, Your Name Surname” (e.g. ML in Python, Spring 2023, Homework 3, Zafer Aydın).
5. Click on the “Add Data” button on the upper right corner.
6. Click on the “Search keyword or URL text box” below Add Data. Search for “Housing Prices Competition for Kaggle Learn Users” by entering this text to the search box.
7. Go to page numbered 2.
8. Click on the + button next to the dataset uploaded by user A.P. to add dataset. When you bring your mouse on the + button you should see Add Dataset not Add Notebook Output.
9. Click on the × button next to Add Data at the upper right corner to close the window for adding data.
10. Click on the +Code button at the lower left side of the code cell to add a new code cell.
11. You can start from the template code called ML in Python, Spring 2023, Homework 3, Template (which is made available as ml-in-python-spring-2023-homework-3-template.ipynb).

Assignment

In this homework, you will work with numeric and categorical features. You will implement the three techniques we learned for handling categorical features and compute performance on validation set as well as the test set (i.e. by submitting your test set predictions to leaderboard of the competition). Different from the exercise notebook of Lesson 3 of Intermediate Machine Learning course, in this homework, we don’t drop features with missing values. Implement each question in a separate code cell in your notebook. Note that the code template reads the original training set as X_train and test set as X_test. After splitting it produces X_train_2, y_train_2 (reduced training set) and X_valid, y_valid.

1. Implement the following steps that drops categorical columns

- Drop columns that have categorical data from reduced training set (X_train_2) and validation set (X_valid). Assume that the data type of categorical features is object. Store the resulting data sets as drop_X_train_2 and drop_X_valid.
- Fill the missing values in the remaining numeric feature columns using SimpleImputer by setting the strategy to median. Convert the output of imputations to pandas data frames. Store the resulting data sets as imputed_X_train_2 and imputed_X_valid.
- Reassign column names from drop_X_train_2 and drop_X_valid to their imputed versions.
- Compute and print mean absolute error on validation set by calling score_dataset function using the imputed versions of the data sets. Include the mean absolute error to your notebook as a Markdown cell.

2. Repeat question 1 this time starting from the original training set (X_train) and test set (X_test). Instead of computing the mean absolute error on validation set, this time you will submit the test set predictions to competition. For this purpose, train an XGBoost regression model (using the same parameter settings

as in score_dataset function) on training set and compute predictions on test set. Save your predictions as a csv file. You can find instructions for submitting a csv file to competition in question 4 of homework 2. The link of the competition is https://www.kaggle.com/competitions/home-data-for-ml-course. Include the leaderboard score to your notebook as a Markdown cell.

3. Implement the following steps that applies ordinal encoding to categorical features

- Start from X_train_2 and X_valid. Remove categorical columns that contain data in validation set but not in training set from X_train_2 and X_valid.
- Apply ordinal encoding to remaining categorical columns. Save the resulting data frames as ordinal_encoded_X_train_2 and ordinal_encoded_X_valid.
- Fill the missing values in ordinal_encoded_X_train_2 and ordinal_encoded_X_valid (which include both numeric and categorical features) using SimpleImputer by setting the strategy to median. Convert the output of imputations to pandas data frames. Store the resulting data sets as imputed_X_train_2 and imputed_X_valid.
- Reassign column names from ordinal_encoded_X_train_2 and ordinal_encoded_X_valid to their imputed versions.
- Compute and print mean absolute error on validation set by calling score_dataset function using the imputed versions of the data sets. Include the mean absolute error to your notebook as a Markdown cell.

4. Repeat question 3 this time starting from the original training set (X_train) and test set (X_test). Instead of computing the mean absolute error on validation set, this time you will submit the test set predictions to competition. Follow the steps similar to question 2. Include the leaderboard score to your notebook as a Markdown cell.

5. Implement the following steps that applies one-hot encoding to categorical features

- Find categorical columns in X_train_2 having low cardinality (i.e. cardinality less than 10).
- Apply one-hot encoding to low cardinality columns of X_train_2 and X_valid and store these as new pandas data frames. Set handle_unknown to ignore and sparse to False when defining OneHotEncoder. Set the index of X_train_2 and X_valid to their one-hot encoded versions.
- Obtain pandas data frames starting from X_train_2 and X_valid that includes numeric features only.
- Combine data frames that contain numeric features and one-hot encoded features separately for training set and validation set. Save the resulting data frames as OH_X_train_2 and OH_X_valid.
- Use the astype method and make sure that the data type of the column names of OH_X_train_2 and OH_X_valid is string. You can use the code templates in tutorial link of Categorical Variables lesson of Intermediate Machine Learning course for this purpose.
- Fill the missing values in OH_X_train_2 and OH_X_valid (which include both numeric and categorical features) using SimpleImputer by setting the strategy to median. Convert the output of imputations to pandas data frames. Store the resulting data sets as imputed_X_train_2 and imputed_X_valid.
- Reassign column names from OH_X_train_2 and OH_X_valid to their imputed versions.

- Compute and print mean absolute error on validation set by calling score_dataset function using the imputed versions of the data sets. Include the mean absolute error to your notebook as a Markdown cell.

6. Repeat question 5 this time starting from the original training set (X_train) and test set (X_test). Instead of computing the mean absolute error on validation set, this time you will submit the test set predictions to competition. Follow the steps similar to question 2. Include the leaderboard score to your notebook as a Markdown cell.

7. Fill the table below that includes your validation set and test set scores.

	Validation Set MAE	Test Set MAE
Drop columns with categorical features		
Apply ordinal encoding to categorical features		
Apply one-hot encoding to categorical features		

8. Which approach gives the best validation set score? Which approach gives the best test set score (i.e. leaderboard score)?

Submission

Once you finish, click File and Download Notebook. Submit your notebook with .ipynb extension to Canvas. Your notebook should include all the codes you developed for each question. Implement each question as a separate and a single code cell and put a comment line that includes the question number at the beginning of the code cell of each question (e.g. #Question 1). Submit your answers to questions 7 and 8 as a text or Word document to Canvas.