COMP 468 Machine Learning in Python, Spring 2023
Instructor: Zafer Aydın
Final Exam

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 Final Exam Your Name Surname" (e.g. ML in Python Final Exam 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, Final Exam Template, (which is made available as ml-in-python-spring-2023-final-exam-template.ipynb).

Assignment

The template code applies SimpleImputer (strategy set to median for numeric columns and most_frequent for categorical columns) followed by one-hot encoder for categorical columns. It then trains an XGBRegressor on training set and computes predictions on test set. This is the baseline model for this exam. The leaderboard score of submission.csv generated by the template code is 14690.14211.

Implement each question in a separate code cell (i.e. question 1 will be implemented in a code cell, question 2 will be implemented in another code cell, etc.).

1. Call make_mi_score function by sending X_train and y_train as input and storing the resulting pandas series as mi_scores. Then call plot_mi_scores function for the first 20 features that have the highest mutual information scores. You can access these functions and related commands for calling them from the Mutual Information lesson of the Feature Engineering course. Which five features have the highest mutual information score with the target?

2. Similar to the first step of the exercise page of Creating Features lesson of Feature Engineering course, generate a new feature called LivLotRatio by following the steps below.

- Generate empty data frames called X_1_train and X_1_test.
- Compute X_1_train["LivLotRatio"] as the ratio of the GrLivArea feature of X_train and LotArea feature of X_train.
- Do a similar computation of LivLotRatio feature for X_1_test by computing the ratio of GrLivArea feature and LotArea feature of X_test.
- Concatenate X_1_train to X_train using the join method and store the resulting data frame as X_train_new.
- Concatenate X_1_test to X_test using the join method and store the resulting data frame as X_test_new.
- Perform one-hot-encoding on X_train_new and X_test_new using get_dummies method of pandas. Store the resulting data frames as X_train_new and X_test_new. Call the align method through X_train_new and by sending X_test_new as input and store the resulting data frames as X_train_new and X_test_new.

- Define an XGBoost regressor model by setting n_estimators to 1000, learning_rate to 0.05, random_state to 0 and n_jobs to -1.
- Train the XGBoost regressor model on training set (X_train_new and y_train) and compute predictions on test set (X_test_new).
- Prepare a pandas data frame that contains the Id column and SalePrice column, in which the SalePrice column contains your predictions on test set.
- Convert the pandas data frame to a csv file called submission_new_features.csv. Submit that csv file to competition by visiting https://www.kaggle.com/competitions/home-data-for-ml-course and enter your leaderboard score as a Markdown cell below your code cell for question 1.
- Did you get improvement over the baseline model? Enter your answer as a Markdown cell.

3. Perform k-means clustering and generate a new feature based on cluster labels by following the steps below.

- Import StandardScaler and KMeans classes of scikit-learn.
- Store 'LotArea', 'TotalBsmtSF', '1stFlrSF', '2ndFlrSF', 'GrLivArea' as a Python list named features.
- Select those features from X_train and X_test and store the new data frames as X_train_scaled and X_test_scaled.
- Define a StandardScaler object with default parameter settings.
- Apply standard scaler to X_train_scaled by calling the fit_transform method and convert the result to a pandas data frame. Store the output data frame as X_train_scaled. Apply standard scaler to X_test_scaled by calling the transform method and convert the result to a pandas data frame. Store the output data frame as X_test_scaled.
- Make a copy of X_train as X_train_new_2 and a copy of X_test as X_test_new_2.
- Define a KMeans object by setting n_clusters to 10, n_init to 10, and random_state to 0.
- Apply k-means clustering by sending X_train_scaled as input to fit_predict method of the KMeans object and store the cluster labels as a new feature called Cluster in X_train_new_2. Obtain the cluster label feature for test set similarly by sending X_test_scaled as input to predict method of KMeans object and store the cluster labels as a new feature called Cluster in X_test_new_2.
- Apply one-hot-encoding to X_train_new_2 and X_test_new_2 using the get_dummies and align methods as in question 2. Store the resulting data frames as X_train_new_2 and X_test_new_2.
- Train an XGBoost regressor model using X_train_new_2 and y_train. Compute predictions on X_test_new_2. Save your predictions as submission_new_features_2.csv.
- Submit your predictions to competition as in question 2. Include the leaderboard score as a Markdown cell.
- Did you get improvement over the leaderboard score of the baseline model by adding cluster labels as a feature? Include your answer as a Markdown cell.

Submission

Once you finish, click Share at the upper right corner and share your notebook with user Zafer Aydin. My user name is Zafer Aydin with space between Zafer and Aydin. I am not mahmut zafer aydin and not ZaferAydin (madmachine). There should be space between Zafer and Aydin. If you cannot share your notebook with me, you can submit it with .ipynb extension to Canvas. Please try to share your notebook first. 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 or #Q1, etc).