Project 4: Offline Model Development
CSC 6605 ML Production Systems

Overview

The goal of this part of the project is to create a Scikit-Learn pipeline that specifies how to extract features from a Pandas DataFrame and train a model. You'll spend most of your time exploring modeling approaches in a local notebook and then once you identify which features you want to use, which type of model, and selecting hyper-parameters, you'll create, train, and serialize a Scikit-Learn pipeline object.

Instructions

Part I: Setup a Notebook and Grab Data

1. SSH into your VM from your local computer and forward port 5432 for PostgreSQL.

2. Start Jupyter notebook locally (saved in your "notebooks" directory in your git repository).

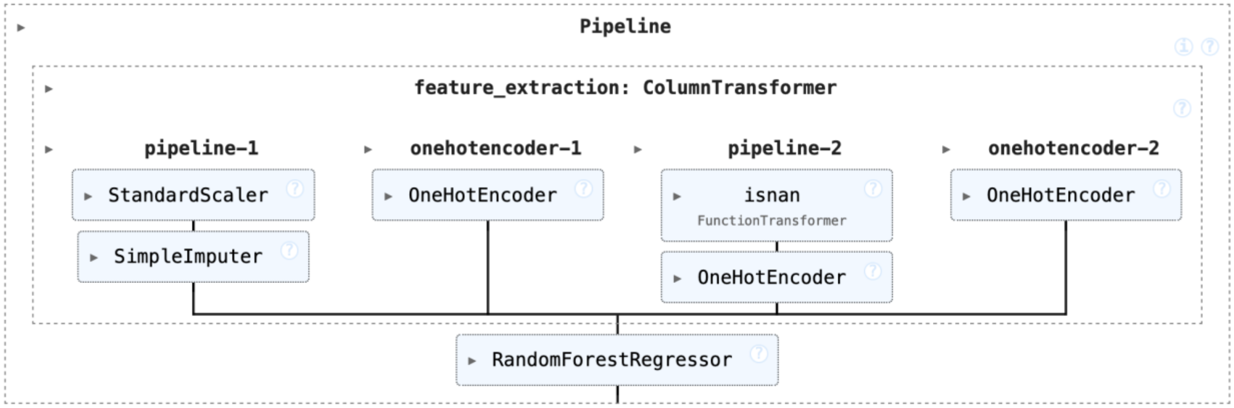
3. Download the cleaned and enriched home sale events from PostgreSQL.

Part II: Create and Evaluate a Model

Create a regression model to predict the price from the other variables. Evaluate the model using the metrics RMSE, MAPE, and R^2. Optimize your model through feature engineering and evaluating different model. For the evaluation, you should use a cross-fold validation strategy that takes time into account as an independent variable such as Scikit-Learn's TimeSeriesSplit.

Part III: Implement Feature Extraction and Model Prediction using Scikit-Learn Pipelines

Use Scikit Learn's Pipeline functionality to create a single pipeline object that will take a Pandas DataFrame as input. It can be helpful to draw out the tree of operations that you want to guide your implementation. You can also display your pipeline in your notebook to double-check it:



Ensure that your pipeline implementation matches the performance of your model from Part II.

Part IV: Train and Save the Production Model

When preparing a model for use in production, you want to use all the data rather than using some for training and the rest for testing.

1. Use the fit() method of your pipeline to train a model on ALL of the data you have.
2. Save the model as a file to disk using the dill library.

Part V: Written Report

Write a 2-3 page report. Describe your final model, its features, and its performance. Describe your pipeline (and include a diagram).

Submission

Submit your report and model development notebook to Canvas as PDF and HTML files, respectively. Your instructor will check that the notebook is committed to your GitHub repository.

Rubric

Description	Percentage
Written Report: • Report is written in a professional manner using proper grammar and spelling. • Report is a useful standalone document that can be shared with a business partner.	15%
Plots: • Appropriate types of plots were chosen for each analysis. • Axes are properly labeled. • Used legends if appropriate. • Chose appropriate axis limits to make plots readable and avoid misleading interpretations. • Font sizes are legible. • Figures are saved at high resolutions.	15%
Offline Model Development • Set up proper experimental setup using the time-series split cross-fold validation and appropriate metric for evaluating predictions. • 2-3 model types were tried. • Features are scaled and transformed correctly (e.g., one-hot encoding). • Different approaches for engineering features were tried. • Reasonable effort to tune hyper-parameters was made. • Model selection was performed to identify best combination of choices.	40%
Scikit-Learn Pipeline • A pipeline was implemented that performs feature engineering, scaling, and transformation. • Performance of pipeline matches the optimal model from Part II.	20%
Trained and Saved Model • A model was trained on all data. • The model was written to a file using Dill.	5%
Development Process • Notebook was committed to the git repository.	5%