**House rent forecasting**

I tried the following approach to solve this problem.

**Approach**: - Using Random forest to predict the rent of the house given the input features like (flat area, flat quality, day, location etc.)

**Other possible approaches**: - Vector Autoregressive model (VAR) also can be used to solve this problem. It’s basically a multivariate linear time-series models, designed to capture the dynamics between multiple time-series.

LSTM’s also can be used to solve multi variate time series problems.

*AWS Deep AR* also can solve this problem. It is a supervised learning model meant to be used for univariate and multi variate data.

In the approach 1, while using random forest I also thought to add some lag variables. Lag variables help us in bringing out patterns from the past to be evaluated at the present.

We can do the similar trick on target variable as well. i.e to predict 6 months into the future, use 6 months of the past data as input.

As an example, to predict the price at 1st December 2019, use data points of 1st June 2019. To predict for 2nd December, use 2nd June. And so on. However, in the dataset given there are multiple data points on a single date. So, I was not completely sure which values to consider while calculating the difference.

Due to deadline requirements and office work I was unable to explore these techniques.

**Approach**

**Aim**: - To develop a model that can forecast the rent price of the house.

**Solution design**: -

Broadly there are four steps in my solution

* Data preprocessing
* EDA and feature selection
* Model building
* Hyperparameter tuning
* Model deployment through docker.

**Data Preprocessing**: - In this step I tried to remove various unnecessary values from the dataset. For instance, feature ‘flat\_thermal\_charactersitic’ have the value almost 32% of the values as null values. I also imputed the null values in the features Weight and publisher with median value. (I selected median because it is not affected by the outliers in the data like the other measure i.e. mean). I have encoded the categorical variables. Some of the columns have two distinct values ‘t’ and ‘f’. I have replaced it with 1 and 0. And I have done the one hot encoding for the other categorical variables.

**EDA and Feature selection***:* - I tried to visualize the distributions of each of the columns and removed few columns which are not required for the analysis (e.g. geo\_city). By looking at the correlation matrix I removed some of the highly correlated features. I didn’t used any feature selection techniques like WOE (Weight of Evidence) or IV (Information value) (for categorical features) as I used Random forest which has the inherent capability of picking best features.

**Model Building***:* - I tried to fit various models on the dataset and evaluated their performance. Finally, I have decided to use Random forest because of following reasons.

1. In this scenario it has higher accuracy than the other models.
2. It can handle the overfitting problem.
3. It can find the important features by itself. (We need not worry about selecting the features)

**Hyperparameter tuning***:* - Tuning the parameters is essential in improving the performance of the model. I have used a grid search technique to tune the parameters. In this case grid search only improved the performance slightly. As time is short, I couldn’t spend much time on this.

**Results**: - Baseline model’s rmse value is 398. Random forest got rmse value of 181 on the test data initially. After performing the random search rmse value got decreased to 178, which is not a significant difference. Performance of this model can be further enhanced by considering more data while training.

“Flat area”, “Flat quality”, “has\_elevator” and “has\_kitchen” are some of the key features which are decisive of the rent price of the house.

**Steps to run the model.**

I have developed the model on windows 10 environment.

Prerequisites: - make sure that following libraries are avaiable to execute the further steps.

sklearn

matplotlib

logging

seaborn

pprint

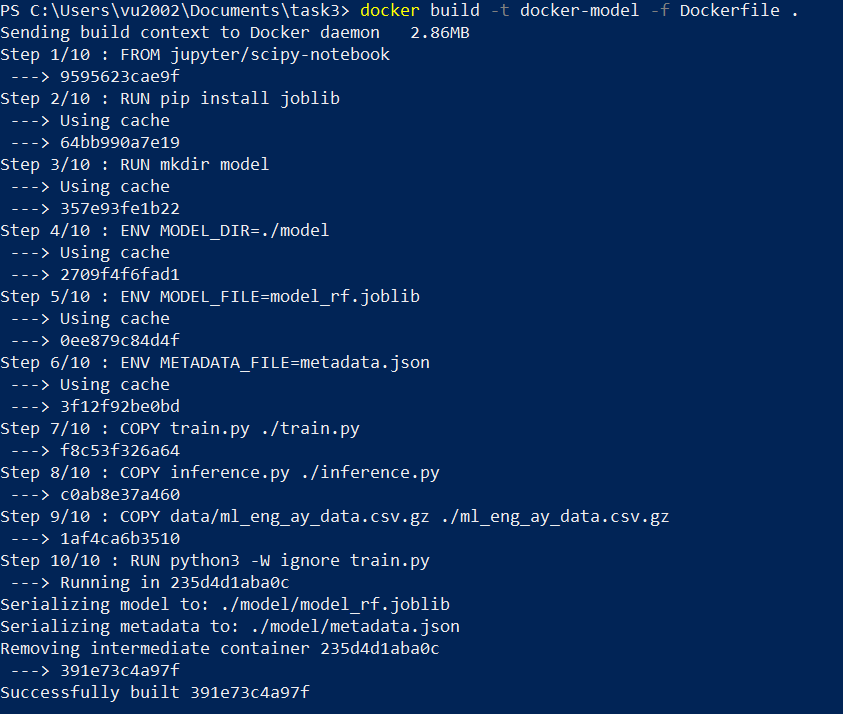
pandas

Step 1: - Build the docker image

**Change the directory to the ‘task3’ folder.**

Run the following command in powershell.

docker build -t docker-model -f Dockerfile .



Step 2:- Now we have built our image, for inspecting the rmse value on the test set run the following command.

docker run docker-model cat ./model/metadata.json



Step 3:- Now we can perform batch predictions. To do that run the following command.

docker run docker-model python3 -W ignore inference.py

