report

October 20, 2018

1 Data Preprocessing

1.1 Dummy Coding

- categorical variables was stored by datatype `object`, I extracted all the object columns and conducted dummy coding by one-hot encoding.
- To ensure the same columns between train and test, I used `align` from pandas to ensure same features between train-test.(This `align` fucntin is heavily used in this file to ensure same columns left for train-test)

1.2 Feature Selection

Feature selection process was done thorugh gbm method, as tree model has the ability to select features based on importance-- frequency certain features being chosen.

1.3 Misssing Values and other issues

- Imputation on Missing values,replaced by median.
- StandardScale on train,test features.
- Replace method is being used to deal with infinite numbers.

2 Method

Method here being used are Gradient Boosting Machine(gbm), due to the fact that GBM method has lots of hyperparameters to be tuned,so parameters optimization method like `GridSearchCV` and `RandomizedSearchCV` is being used to find optimal hyperparameters.

2.1 RandomizedSearchCV

In this step,loss,n_estimators,max_depth,min_samples_leaf,min_samples_split,max_features are parameters being searched based on cross-validation.

2.2 GridSearchCV

After RandomizedSearchCV, we use GridSearchCV to specific a range of trees([100, 150, 200, 250, 300, 350, 400, 450, 500, 550, 600, 650, 700, 750, 800]) to find best tree numbers for the model.

3 Result

3.1 RMSE

- Model 1: Test:0.110
- Model 2: Test:0.114

3.2 Running Time

- 1345.7330601215363

3.3 Computer System

- MacBook Pro,3.3 GHz Intel Core i7,16 GB 2133 MHz LPDDR3 Memory

4 Project 2

Note: `np.dot` function return different value when tring to get the matrix multiplication between `test` and `beta`, result Notebook(html version) is appended in the folder to present I do get the result around 0.125 however once call `np.dot` again I couldn't get same result.