**Approach: Jobathon (Sep-2021)**

Name: G.Dinesh kumar

Ph.no: 8919599274

Email:constantine6174@gmail.com

First I have removed “#order” feature and the data which has “sales” == 0 (there were only 19 records).

Handling “Date” feature.

* I have used preprocessing techniques from fastai tabular module called “add\_datepart(df, “Date”)” which is a helper function ,it takes Dataframe and Date feature as input and creates features from it like [“'Year', 'Month', 'Week', 'Day','Dayofweek', 'Dayofyear', 'Is\_month\_end', 'Is\_month\_start','Is\_quarter\_end', 'Is\_quarter\_start', 'Is\_year\_end', 'Is\_year\_start', 'Elapsed'”]
* all the features created from “Date” are categorical in nature except “Elapsed” which is continuous

Next I used “TabularPandas” from fastai to handle “categorical feature”,”missing values” and create training and validation sets

TabularPandas(df , procs , cat , cont ,y\_names = dep\_var , splits )

- df, is dataframe

- procs = [Categorify , FillMissing]

-procs is list of preparatory processing techniques needed to transform the data

- Categorify : replaces categorical columns with a numeric categorical column

- FillMissing : replaces missing values with median of the column and creates additional column which boolean in nature , where True is missing value and False is the opposite

- Cat is a list of feature names which are categorical in nature which is almost all of them in this case execpt “Elapsed”

- Cont is a list of continuous feature in this case “Elapsed”

- y\_names is “dep\_var” or dependent variable of the data in this case we define dep\_var = “Sales”

- splits : splits the data into training and validation , it should contain two lists with indexes of the data in them

- since this is a time series data we need to split train and validation bases on the timeframe chronologically.

- I have split the data such that the validation data resembles the test data meaning the have taken last two months of the training data to be validation data.

TabluarPandas behaves like dataset objects which provides trian and valid attributes

To = TabularPandas(df , procs , cat , cont ,y\_names = dep\_var , splits )

to.train ,to.valid

next we define our independent and dependent variables for train and valid data

xs,y = to.train.xs,to.train.y

valid\_xs,valid\_y = to.valid.xs,to.valid.y

next we use random forest

* I used randomforest from sklearn to create the model with default parameter to create a baseline
* I used MSLE from sklearn to calculate error between “target” and “preds”
* Once the model is trained I select important features from “feature\_importances\_” attribute from sklearn’s randomforest algorithm
* I filter out features with importance less than 0.005
* Remove redundant features
  + Cluster\_columns(xs\_imp) gives a chart which shows which pairs of features are most similar which are the ones that merged together far from the “root” of the tree at the left .
  + We drop one feature from the pair
* Finding Out of domain data: we don’t know whether test set is distributed the same way as training data or not and also which features reflect the difference
  + inorder to do so we combine only independent variables from train and validation datasets and create a new dependent variable with 0 and 1 , where if the data is from training data and 1 if the data us from validation data
  + Next use random forest train and calculate feature importance and select feature which have high values , which means the features with high values indicate they differ significantly with training and validation data ,in this case “Elapsed” ,“Week” and “Dayofyear”.
  + Now we compare the MSLE score by dropping one feature at a time among the 3 with the original score
  + When “Elapsed” was removed the MSLE score was 82.03 , similarly for “Week” was 137.16 and “Dayofyear” was 139.12. so it is ideal to remove two features among them to get better MSLE score , I have chosen “Elapsed” as it gives lowest score and “Dayofyear” becoz there isn’t much difference between “Dayofyear” and “Week” (look at the cluster chart)and also I felt the pattern with “Week” feature would be much better.
  + Next I droped the time variables “Elapsed” and “Dayofyear” from traiing and validation sets
  + Random forest parameter values
    - One of the important properties of random forest is that they aren’t very sensitive to the hyperparameters choices such as max\_features.
    - So what I did was create a range of values for each hyperparameter and fir the model with one parameter at a time and observe its impact on the train and validation msle values.
    - And then i selected hypermeter values which has low validation msle score.
    - Although it necessarily doesn’t mean that by selecting all the parameter values with low msle score gives the best MSLE score with put together but since we are using Randomforest , as said earlier, randomforest isn’t that sensitive to parameter tuning but if had I used any other algorithm I would use GridSeachCV from sklearn to find the optimal values for the hyperparameters.
* Once the parameters are selected and train the model
* Next we transform the test data similar to that of the training data by following the same steps and remove the features which are not required.
* Next we use the trained model to predict the “Sales” for the test data.

I know it seems a bit confusing and also the notebook I submitted wasn’t cleaned very well but , now I have cleaned the notebook , haven’t changed any original code and haven’t messed with any parameter values good or bad.. thank you