Approach for Analytics Vidhya Jobathon June-2021

Requirement:

Create fact table based on user_dimension and visitlog dimension tables.

Approach:

Created three transformer classes for transforming User dimensions and visitlog dimensions and one for creating the features for the user_visitlog features.

Class user_transformer:

#description: Used to create features on user dimension and to clean the user data as well

Classs_Attributes:

Today date: though set as 28-05-2018 can be altered easily to date of now to get vintage days of user.

Methods:

Cleandata():

Takes in the user_df and provides with cleaned data by performing following actions

1.Drops duplicates on UserID inorder to avoid duplicates.
2.Drops rows where the UserID is null.
3.converts the Sign up date column to datetime column
4.fills the null values in the user segment column to unknown.

Vintage_days():

Takes in the user_df and returns the df with vintage days column.

Class Visitlog_transformer:

Methods:

Cleandata():
Takes in the visitlog_df and provides cleaned data by performing following actions

1.Drops duplicates on the whole dataframe in order to avoid duplicates
2.Removes all the rows where userid is not available
3.cleans the string columns by calling in string_clean method
4.cleans the datetime column by calling in the datetime method
5.imputes the missing value in Activity and ProductID columns by calling in the impute_activity method.

String_clean():
Takes in the visitlog_df and provides cleaned string columns by performing following actions

1. Sets the string columns in same cases

Datetime_clean():
Takes in the visitlog_df and provides cleaned datetime columns by performing following actions
1.Find the unix column and convert to standard datetime format
2.Converts the string column to Datetime column

Impute_activity():
Takes in the visitlog_df and provides imputed values for activity and product by performing following actions

1.Creates Session Id by following actions
(i).removing VisitDateTime null rows and sorting the df on user_id and Visitdatetime columns
(ii).remove the rows where both the productid and activity columns are null.
2.Create a column to represent the imputed values
3.imputes the values as click if the users previous product in the same session is same as current product.

4.imputes the values as click if the users previous product in the same session is not same as current product.
5.calls the imputeproduct function to impute products
6.concates the rows that are dropped before imputation

Impute_product():

Takes in the visitlog_df and provides imputed values for product by performing following actions

1.If the current Activity in the current session is click then imputes values of product as the previous product.

Class user_visitlog_transformer:

Methods:

user_visitlog_feats():
Takes in the user_df, visitlog_df and returns the user_df with the set of features by performing the following actions
1.calling in user.Vintage_day() method
2.calling most_active_os method and other self.methods

Most_active_os():
Takes in the user_df, visitlog_df and returns the user_df with the most active os by performing the following actions

1.groups by UserID and gets the mode for each group and joins it with the User_df

Recent_viewed_products():
Takes in the user_df, visitlog_df and returns the user_df with the recent viewed products by performing the following actions

1.removes the rows of users who are having no visittime entries and store it in a seprate df since those users products cannot be sorted to find the recent one.

2.Considers the products viewed as recent if only one product is viewed by the user even though visittime is unknown
3.finds the reacent products of user with visitdate time entries with help of visitdatetime column
4.joins both the recent products with and w/o visitdate time and merges with the user_df

Seven_days_feats():

Takes in the user_df, visitlog_df and returns the user_df with the features based on last seven days history of user by performing the following actions

start date and end date are set as local variables which can be modified easily to current seven days.
1.creates the no of days visited feature by calling unique function over the date column
2.calculates the no of clicks and pageload
3.merges the features with the user_df

Fifteen_day_feats():

Takes in the user_df, visitlog_df and returns the user_df with the features based on last fifteen days history of user by performing the following actions

start date and end date are set as local variables which can be modified easily to current fifteen days.
1.Creates the no of viewed products
2.merges the feature with the user_df

Most_viewed_feats():

Takes in the user_df, visitlog_df and returns the user_df with the most viewed products feature on last fifteen days history of user by performing the following actions

1.Creates the most viewed products by filtering the rows that belong to pageloads and has some entry in the product column.
2.Finding number of times each product has been viewed and filter the products which has maximum views
3.finding the product which has been viewed recents with visitdatetime
4.remove if there are more than one products and merge with the user_df

RUNTIME INSTRUCTIONS:

1.Read in the data

2.CALL THE **user_visitlog_feats()** method on the User_visitlog_transformer to get the set of transformed table with features.

Technologies Used:

Python, Pandas.

For further queries contact me at Arulprabhakaran.a@gmail.com