Solution Description:

After spending so much time on analyzing the data on MS Excel, I found that the problem is very interesting. Following are some of the Assumptions I made (which I asked in an e-mail as well.)

I have added the properly packaged source code in python notebook named `n26DSProb`. It should run all of the environments. Although it is well commented, if feel any issue please let me know.

- I needed to predict both income and expense separately. A user can have income or expense at a time. I picked separately.
- For testing set input will be having user history so I trained the model basis on user-specific features(mainly old history) and then predict the user will have income or expense (amount as well).
- From the problem set, what I understood is from 10 k users of training dataset (from February through July)I need to predict income and expenses of all these users for the month of August and then club them into 'in' or 'out' category.
- With these understanding, for both the expense(out[-]) and income(in[+]) feature vector, I created features with users' last credit history such as :
    1. Expense in 90 days
    2. Expense in 30 days
    3. Expense in a week
    4. Expense in 2 days
    5. Expense rate
    6. Income in 90 days
    7. Income in 30 days
    8. Income in a week
    9. Income in 2 days
    10. Income rate
    11. mcc_group
    12. Converted transaction_type(transaction_type_Id)
    13. Converted agent(agent_Id)

Results/Accuracy:

For performance point of view, I have taken R-Squared(R^2) and mean squared error (MSE), although I could not perform the parameter tuning/feature normalization, following are the results of some of the tried algorithms:

R2: A statistical method that explains how much of the variability of a factor can be caused or explained by its relationship to another factor. Coefficient of determination is used in trend analysis. It is computed as a value between 0 (0 percent) and 1 (100 percent). The higher the value, the better the fit.

RMSE: Root Mean Square Error (RMSE) (also known as Root Mean Square Deviation) measures how much error there is between two datasets. In other words, it compares a predicted value and an observed or known value.

	Performance					
	Input Model			Output Model		
Model	MSE	R2	RMSE	MSE	R2	RMSE
LM	3918.431411	0.915408001	62.59738	3017.246	0.717868	54.92946
RandF	3789.717693	0.918186703	61.56068	3001.315	0.719357	54.78426
XgBoost	4037.586677	0.91283565	63.54201	3333.307	0.688314	57.7348
GBM	3902.86625	0.915744026	62.47292	3059.501	0.713916	55.31276

Ps: performance could be improved through some parameter training and feture normalization and ensemble techniques.

Function for testing:
For testing user's credit history and August file data should be process in similar process as in the training.This function does following steps:

- This requires users' old history is required as credit/debit history are used in feature creation.
- Generates transaction agent and mcc group their Id values manually credited
- Goes into the predict function of the finalized model(In/Income or Out/Expense) in my case Random forest regressior i.e pred_out = regr.predict(f_test_x_out)

Step for live production:

Function will accepts the row of a user (/row of users) for particular date and pass in to the function and it would do all the same cleaning and value to index conversion for transaction_type and Agent and then will go the required finalized model (in my case randF) and return the value if the required value is income it will go to Income model otherwise it will go in the expense model.