Problem:-

To predict the category to which the transaction falls into given transaction_id, amount and a small text description of the nature of transaction.

Note:- The category i.e true label is available in the training data. So it will be possible to model it as a supervised learning problem

Related work in real world:-
There is a more detailed explanation from Stripe as to what is transaction categorization and who it benefits:- More details can be found here.
This is very similar to what someone might see in their banking apps in the form of analytics. Could be like where most of the money is spent etc., Exists in well known banking apps like CIBC and RBC.

EDA:-
Training data:-

1) 238947 records, with 4 cols including one true label
2) All transaction ids are unique in nature.
3) None of the columns have any missing values i.e. null
4) Also none of the columns have any empty strings i.e. so far no data quality issues found
5) In terms of transactions (as in the actual descriptions), each of them are unique too. Which is a bit strange because one can possibly expect that multiple transactions can originate from one store, but maybe the sample is drawn in that way
6) In terms of categories - 16 in total, the dataset is quite imbalanced as expected, with close to 80% of training data accumulated in the first 5 categories as seen below

category	proportion
Food and Drink	0.365022
Shopping	0.229013
Transportation	0.084337
Groceries	0.072250
Transfers and Withdrawals	0.056649
Health	0.047881
Recreation	0.047341
Financial Services	0.028642
Housing and Utilities	0.021147
Income	0.018326
Travel and Leisure	0.015510
Education	0.004566
Obligations	0.003235
Donations	0.002741
Miscellaneous	0.002624
Benefits	0.000716

7)

In conclusion, no data quality issues were noticed. Training data is quite imbalanced as expected. Data also includes amount - which contains both positive and negative values showing that the transaction could either be a debit or a credit transaction.

Data chosen for training:-
1) For brevity will use cols transaction and category alone.
2) transaction_id has no for training atleast.
3) amount column is debatable. At least by intuition is probably not a required predictor but can be used for future experimentation.
4) Chosen cols - transaction and category

Modelling/design choices
Approach 1:- Use traditional supervised ML

1) Assumption that the number of categories won't change.
2) Will use explicit encoding for category columns instead of using something like LabelEncoder, this is due to reproducibility purposes. Plus it's easier to identify 'label drift' with this method than LabelEncoder
3) Usage of TF-IDF - mainly because just by eyeballing one can see words like POS repeated everywhere - doesn't say much about the transaction itself. TF-IDF penalizes/ignores such words.
4) Usage of lightgbm, due to fast training and less assumptions about the underlying training data.

5) Usage of TPE sampler for intelligent hp tuning.
6) Minimization of multi_logloss due to > 2 categories
7) Usage of weighted-f1 score. This seems to be the best choice instead of accuracy -even if weighted.

Other ideas/future enhancements:-

1) Using more sophisticated encoding with rich representations like BERT.
2) More aggressive hyperparameter tuning - learning rate, regularization_params and have been set to relatively large numbers and num_boost_rounds a smaller number for training purposes.
3) Trying other modelling methods, which possibly more probabilistic in nature

Approach 2:-

1) Use a LLM to categorize all transactions into pre-defined categories
    a) Pros
        i) Highly likely much better performance
        ii) No need for training a model and maintaining it for foreseeable future
        iii) Companies like Stripe are moving towards foundation models for transactions.
        iv) Can use an open source model. Needs inference pipeline.
    b) Cons
        i) Expensive if models like gpt or claude or used