My proposed methodology for recommending at least 5 pratilipis based on pratilipi info involves 5 major steps

1. Exploratory Data Analysis
2. Data pre-processing (Feature Engineering)
3. Model training and testing
4. Evaluation
5. Recommendation
6. Exploratory data analysis

In this step, the dataset provided was subjected to a preliminary analysis. The pratilipi dataset's shape, data types, and statistical information were all discovered. Time values were converted to float datatype for sorting the complete dataset, while date values were converted from object to datetime format.

To examine the dependencies between the input features, a correlation matrix was used. The correlation between "updated at time" and "published at time" was found to be more than 0.5. Additional analysis included determining unique class values, checking for checking for missing values, and plotting distribution for ‘reading\_time’ for skewness.

1. Data pre-processing

An initial change was made using excel formula for separating out the values of time and date for better analysis using datetime python format. Dropna() was used to impute missing values. SMOTE (Synthetic Minority Over-sampling Technique) implementation is used to eliminate data imbalance in our anticipated classes. This method was later eliminated because of how badly it affected the recommendation system. Other steps included outlier detection and removal using boxplot and log transformation respectively, and label encoding for pratilipi categories.

1. Model training and testing

Several models were implemented for the proposed methodology. Dataset was split intro training and testing having 75% and 25% respectively. In the end random forest classifier gave us approximately a better recommendation prediction. Random forest is an assembling technique that combines multiple decision trees to find optimal prediction. This algorithm uses bootstrap aggregation technique. Reason behind choosing this as the optimal model:

1. Had the least overfitting scenario
2. Handles imbalanced dataset well
3. Works well even with outliers
4. Does not require normalization of data

Support Vector Machine, K Nearest Neighbor, Decision Tree, and other models have also been implemented. These models were excluded because, despite providing a better relative accuracy, they had trouble providing clear likelihood scores for recommendations. Hyperparameter tuning with Grid SearchCV was carried out for comparing accuracies with different model and parameters (snippets of code saved as comments at the end of the notebook).

1. Evaluation

Accuracy, precision, recall, and f1 score matrices were utilised to measure the model's performance. For training and testing data, the model's accuracy was 56% and 6%, respectively. In terms of reliably predicting a single pratilipi category, the model provides a very poor score. The low number of features that are associated to the target variable (the highest one only having a score of -0.03), the extreme data imbalance, and the prediction class's 45 distinct categories are the reason behind this result. The top 5 likely pratilipi categories are nevertheless better estimated despite this decreased accuracy. This classification model can predict the classes more accurately with additional features having more correlation.

1. Recommendation

The methodology of the recommendation system is based on the expected class's probability score. As the machine learning predicts the expected predicted class with the highest probability, we can take out the top 5 scores for recommending the best possible pratilipi category based on provided input. For better visualization, a bar chart has been plotted for displaying the probability comparison among different categories.

For example:

**For input variables in this case,**

Author\_id: 2270000000000000

Pratilip\_id: 1380000000000000

Reading\_time: 810

Updated\_time: 12:09 AM

published\_at\_time: 12:09 PM

**The top recommendations include:**

Pravasi-sahitya, Crime Lekhan, Webseries, Horror-marathon, and pratilip-kids.