Bot Conversations

Dataset:

1. Total 11 columns, 1st column (named *'source'*) corresponds to target labels, remaining 10 columns contain one number each as output by a bot (label as in *'source'*) during a conversation.

Task:

1. Given an input of 10 numbers (or saveral inputs of 10 numbers each), predict the bot label that could have given as output those numbers while in a conversation, using ML classification models.

Results:

Model	Random Forest	Logistic Regression	Support Vector Machine	Neural Network
Accuracy	0.9996	0.9996	0.9996	0.9996

Exploratory Data Analysis (EDA):

1. There are no missing numbers in the dataset.
2. There are no negative numbers in the dataset.

3. Extracting the data corresponding to each Bot (target label) in a separate dataframe reveals the following interesting patterns:

(i) Bot 0 always outputs a single digit number only. In a given row of input, no other bot outputs all single digit numbers (except for Bot2 & Bot 4 which output all zeros in exactly 24 rows).

(ii) Except for Bot0, all other bots always output a non-decreasing sequence of numbers in a given row.

(iii) Bot2 & Bot4 are the only bots for which all output numbers of a row are a multiple of the first number of that row.

(iv) Also, for Bot4 exclusively gives rows of 10 numbers such that the succeeding number can be obtained by multiplying 5 to the preceding number, i.e., next number is 5 times the previous number.

(v) Bot0, Bot1 & Bot3 give rows of 10 numbers such that the difference of two consecutive numbers in a row is not more than 10.

(vi) Bot3 exclusively outputs rows of 10 numbers such that the difference between any two consecutive numbers is exactly 5.

Feature Engineering:

Construct features in original dataset that correspond to the observations of *EDA* as these features can be very helpful in identifying the patterns that correspond to a given '*source'* label.
Because no specific pattern could be observed for Bot1 during *EDA*, construct one more feature '*diff_less_than_10_but_not_5*' that can essentialy capture the rows that correspond to Bot1.

** The actual numbers present in an input row do not seem as important as these features. Because the dataset is large and the numbers vary a lot, it does not seem optimal to use the actual numbers as input to the model as they would not help any model learn patterns due to non-repetitive nature. So, dropping the columns corresponding to actual numbers is an optimal strategy at this point.

Modelling:

1. Basic Classification Models:

1. Import the scikit-learn packages and modules for machine learning classification algorithms (namely Random Forest, Logistic Regression, Support Vector models).
2. Prepare the model inputs and target labels by dropping the actual number columns and only keeping the newly prepared features in the input. Then, split the dataset into train & test sets using the *model_selection* package of scikit-learn library.
3. Cross validation could be ignored at this point because it is a seemingly simpler dataset and the patterns are too simple for a model to learn given the prepared features.
4. Fit the input data by training the model.
5. Predict the target labels on test data using the trained model.
6. Evaluate *accuracy* and *confusion_matrix*.

2. Neural Network Model:

1. Prepare target labels in *one-hot-encoding* format.
2. Import the necessary packages from *keras* library.
3. Because the simple logistic regression model has produced very good results, a three layer neural network should suffice (Input layer 7 units, dense hidden layer 5 units and (softmax) output layer units).
4. Fit the model on training data. Predict the labels for test data.
5. Evaluate *accuracy* and *confusion_matrix*.