Practice: Using Python Functions and Loops for Efficient Coding

1. Create a vector of the three tree-based models that were trained previously. Create another vector of the three-part function names (conn.actionset.action) that are used to score each model.
2. Create a function that returns a list of the CAS action parameters given the model as a parameter.
3. Using the newly created function, use a loop to score each model and print the misclassification rate for each model.
4. Create a function that wraps up the assess action and assesses the model given as the function parameter.
5. Use a loop to assess each model and print the saved tables from the assess action.
6. Create a function to download in-memory tables to the client with the model as the function parameter and add the model name to the local data frame.
7. Use a loop to bring the assess results from each model to the client and stack each resulting table into one data frame.
8. Print the confusion matrix for each model.
9. Print the misclassification rates for each model.
10. Using the addCaslib action from the table action set, create a new caslib named **mycl** with the location as the course data folder, and use the activeOnAdd argument to keep the library local.
11. Use the save action and the attribute action from the table action set to save both the best model and its attributes in the **mycl** caslib.
12. Use the promote action from the table action set to promote the **bank** data to global scope. Use the tableInfo action to view all the tables in the current CAS session and ensure that the table was promoted.
13. End the CAS session.