To begin the project I mined data using <https://seart-ghs.si.usi.ch/>. On this site, I created a dataset using only Java repositories with at least 10 commits, no forks, and at least 100 stars. In order to extract the methods from this project, I use [PyDriller.ipynb](https://colab.research.google.com/drive/1DdPGqJhkLpsfk2yiPGz87rKFRhDHIcMO). Luckily, all of the code was already put together, allowing for me to copy and paste this code to extract all of the methods from the following repositories. After using the Pydriller to extract the methods, I used the [Preprocessing\_Code.ipynb](https://colab.research.google.com/drive/1JOkcDrWI6iuMwY3_jGsmuIAFfhLubDWu) code to help tokenize the methods and remove unnecessary methods or comments. After tokenizing the methods, I created a new column in the data filled with the new tokenized methods. Once I had tokenized the methods, I created the Ngram model. The Ngram class has Train, Predict, and Computer\_Perplexity methods. Train reads in the dataset and the column in which the tokenized methods are held. Using the data set it stores each Ngram and their frequencies in the model directory. The predict method starts by checking the status of the method and if it is at <END> returns <NULL>. Then it extracts the context and returns the most common token and, therefore most probable next token. Lastly the Computer Perplexity method calculates the log probability of each token and uses that to calculate perplexity of the data. Next, LoadCorpus method is designed to take in a dataset, in this case the Teacher\_Data and put it into the same format as the original dataset, having the data in a column in a DataFrame with a “Tokenized Methods” column. After this the model takes the data and trains it with different values of N and selects the N values that yield the lowest perplexity. We then do the same thing with the teacher data after Using loadcorpus to format this data correctly. After we have found the best model for each data set, we use the generate predictions method to generate the json file with the 100 Java method predictions. The generatepredictions method first selects 100 random tokens from the dataset. The method then iterates through the tokens and uses predict to get the probability of the next following token. It stores the predictions in puts them into an output file which holds each value.