Dependency used:

sklearn.metric
pickle
tensorflow
numpy

For my second scoring function, I choose to use the support vector machine learning method. I used 60000 natural and random protein peptides and generated labels for them. Then based on the index of the amino acid in the list, I was able to convert each peptide into a list of integers. Then I padded them to make sure all the input length is the same which is a requirement for the training process. By using the regression method, I was able to develop a SVR model. By running predictions on the input file with that model, I was able to achieve an AUROC score of around 0.52.

Note that I trained the model for score 2 in Google Colab with the file “score2_model.py”, I saved the trained model and read it in my ass2n.py file to make the predictions.

For each protein sequence in the uniprot/swissprot protein sequence database, I truncated them down to a subsequence with length 20-40 and randomly shuffled each natural peptide to generate the random peptides. Then I initialize two different dictionaries to store the number of occurrences for each possible 3-mer each of which has size 20^3 = 8000. After normalizing values in two dictionaries, I was able to obtain a 3-mer frequency vector for both natural and random peptide population. This information is stored locally in a file named “kmer_freq.txt”, the first 8000 lines representing values in the natural population and the last 8000 representing random.

For the testing data, I chose 20000 proteins from the reviewed uniprot/swissprot protein sequence database and truncated them to length between 20-40 to produce the natural and proceeded to randomly shuffle them. These 40000 peptides are stored locally in a file names “input.txt” with the first 20000 being natural and the last 20000 being random.

“kmer_freq.txt” and “input.txt” is being obtained by running the command line
python3 process.py uniprot_sprot.fasta input.txt
The main program is run with the command line
python3 assn2.py input.txt output.txt