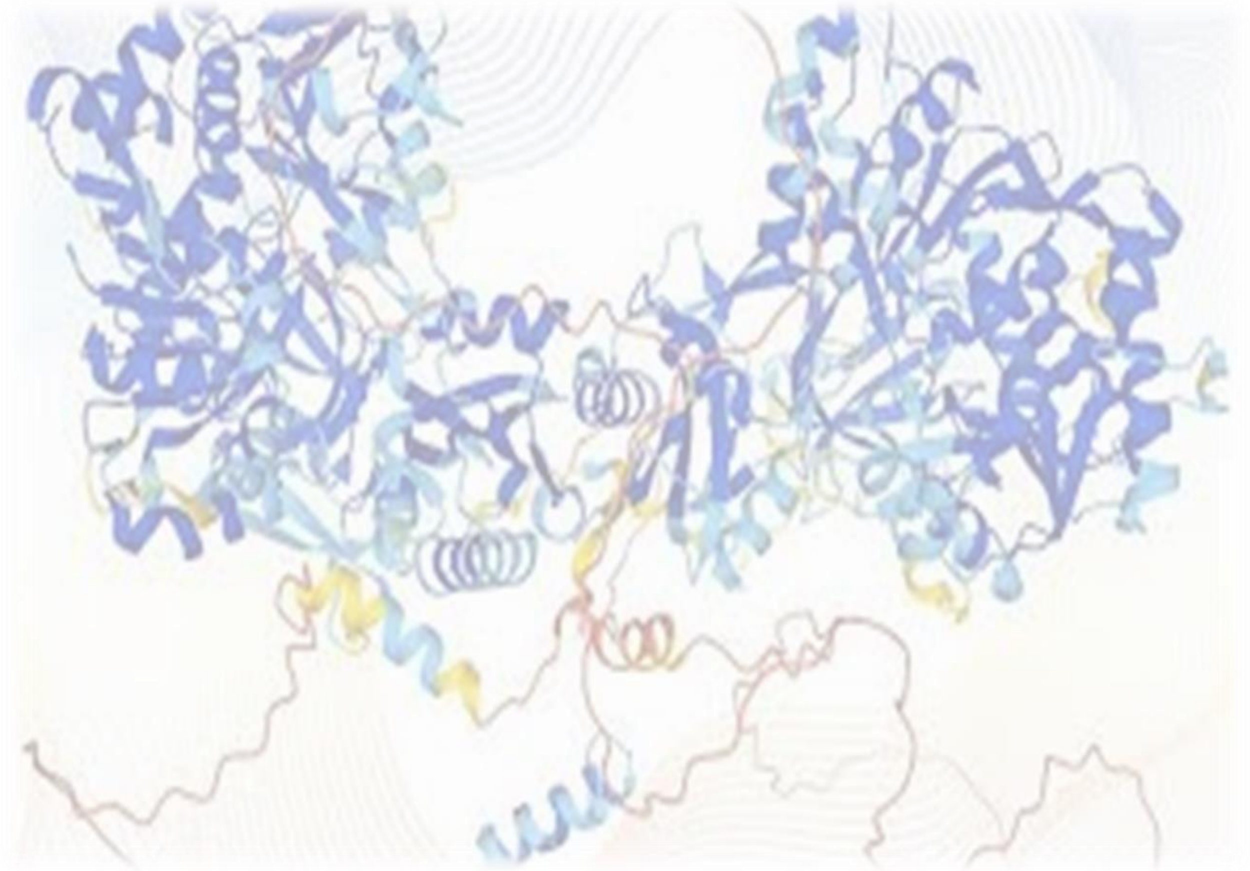
Report of final assignment Programming 3

Predicting of Protein Function based on InterProScan Features

Lecturer: Martijn Herber

Reporter: Azadeh Pirzadeh



Contents

Description

The main aim of this project is to find a model that predicts the largest feature of protein, based on small features of that protein. This model is based on Random Forest which is one of the powerful methods of machine learning to predict the long features based on small features.

In this project:

- ✓ The feature which should be predicted by the model is InterPro with more than 90% length of the protein sequence which is called large features.
- ✓ The features that the model can predict based on them are the InterPro which has less than 90% of the protein sequence which is called small features.

Structure

This project is written by python code and an open-source library of python, Dask which is useful in parallel computing of big data. Another reason for using Dask is that Scikit-Learn can use Dask, so we can use machine learning methods to train and test the data when we use it.

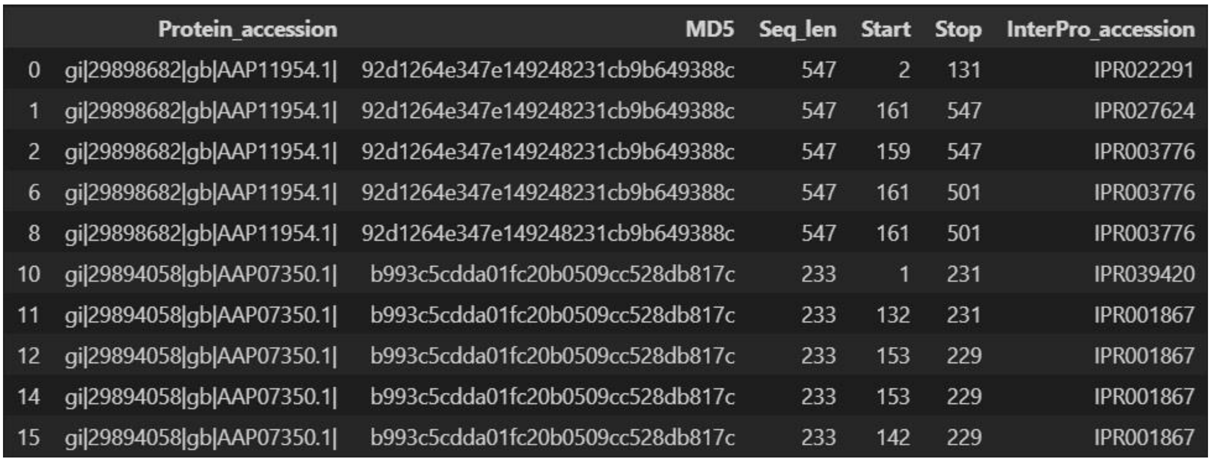
The machine learning method that I used in this assignment was Random Forest. Random Forest is one of the supervised machine learning algorithms that is used in both classification and regression issues.

As we know the issue of this project is a kind of classification because we have specific long features that are known, and I want to predict which one of those long features is related to small features. This model builds many decision trees on various samples and takes their majority vote to predict.

Also, Random Forest is a great method when we have high dimensional data, and it is faster than decision trees in training. This model is based on a bagging algorithm and uses the technique of Ensemble learning. After it builds many trees on the subset of data, it combines the output of all those trees, so it decreases the overfitting and increases the accuracy

Preprocessing

This program analyses “all_bacilli.tsv”. This tsv file contains many proteins with all the information related to those proteins. I just fetch some important ones and create a pandas data frame and then change it to the Dask data frame. Thanks to Dask, I split the data into 20 partitions because the data is so huge. The below picture is an example of this dask data frame.



	Protein_accession	MD5	Seq_len	Start	Stop	InterPro_accession
0	gi|29898682|gb|AAP11954.1|	92d1264e347e149248231cb9b649388c	547	2	131	IPR022291
1	gi|29898682|gb|AAP11954.1|	92d1264e347e149248231cb9b649388c	547	161	547	IPR027624
2	gi|29898682|gb|AAP11954.1|	92d1264e347e149248231cb9b649388c	547	159	547	IPR003776
6	gi|29898682|gb|AAP11954.1|	92d1264e347e149248231cb9b649388c	547	161	501	IPR003776
8	gi|29898682|gb|AAP11954.1|	92d1264e347e149248231cb9b649388c	547	161	501	IPR003776
10	gi|29894058|gb|AAP07350.1|	b993c5cdda01fc20b0509cc528db817c	233	1	231	IPR039420
11	gi|29894058|gb|AAP07350.1|	b993c5cdda01fc20b0509cc528db817c	233	132	231	IPR001867
12	gi|29894058|gb|AAP07350.1|	b993c5cdda01fc20b0509cc528db817c	233	153	229	IPR001867
14	gi|29894058|gb|AAP07350.1|	b993c5cdda01fc20b0509cc528db817c	233	153	229	IPR001867
15	gi|29894058|gb|AAP07350.1|	b993c5cdda01fc20b0509cc528db817c	233	142	229	IPR001867

- ## Finding the large features

  In the first step, I removed the noise and calculated the length of each feature of the proteins, If the length was more than 90% of the protein length I categorized it as a long feature and used the column "class_p" to identify the long features and small features.

- ## Deleting useless data

  In the next step, I removed proteins that have no small and no longer features, which means keeping the proteins that have at least one long and one small feature.

- ## Dividing the small and large features into two different Dask data frames

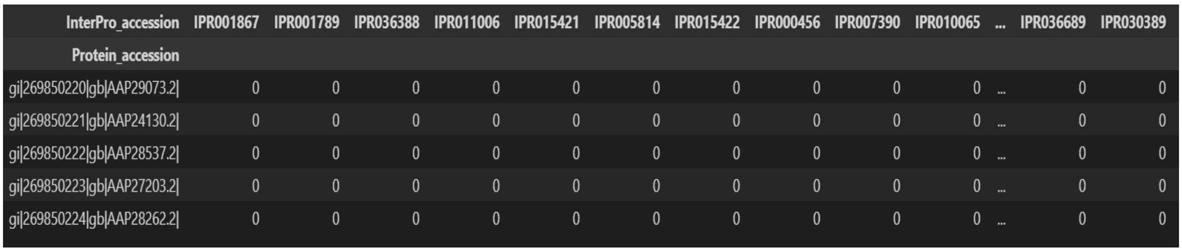
  Then I divided the Dask data frame into two separate Dask data frames: one with small features of each protein and another with large features of each protein.

- ## Finding the largest of long features

  Because some proteins have more than one large feature so I kept just the largest of the long features.

- ## Creating pivot table

  After that, I made a table that contains protein_accessions as rows and interpro_accessions as columns and the value of each cell included the counts of that feature in that special protein. You can see the pivot table below.



<table>
<tr><th>InterPro_accession</th><th>IPR001867</th><th>IPR001789</th><th>IPR036388</th><th>IPR011006</th><th>IPR015421</th><th>IPR005814</th><th>IPR015422</th><th>IPR000456</th><th>IPR007390</th><th>IPR010065</th><th>...</th><th>IPR036689</th><th>IPR030389</th></tr>
<tr><th>Protein_accession</th><th></th><th></th><th></th><th></th><th></th><th></th><th></th><th></th><th></th><th></th><th></th><th></th><th></th></tr>
<tr><td>gi|269850220|gb|AAP29073.2|</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>...</td><td>0</td><td>0</td></tr>
<tr><td>gi|269850221|gb|AAP24130.2|</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>...</td><td>0</td><td>0</td></tr>
<tr><td>gi|269850222|gb|AAP28537.2|</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>...</td><td>0</td><td>0</td></tr>
<tr><td>gi|269850223|gb|AAP27203.2|</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>...</td><td>0</td><td>0</td></tr>
<tr><td>gi|269850224|gb|AAP28262.2|</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>...</td><td>0</td><td>0</td></tr>
</table>

- ### Merge the small and large Dask data frames

  Finally, I merged the pivot table and the data frame of the largest feature base on protein_accession.

Model

I divided the merged table into train and test sets as a portion of 0.7 and 0.3 respectively and used them in the random forest model with 200 trees.

Results

Due to a large amount of data in the first step, I used 100,000 lines of the tsv file and got 33% accuracy then tried to use 1,000,000 lines of the same tsv file and fortunately got 44.3% accuracy, and finally, I got 46.21% accuracy on whole “all_bacilli.tsv”. You can see the third result in the “result.txt”.

Feature approach

The increase in accuracy shows that with more trained data, we will achieve better accuracy. It shows that this model would be perfect if we train more.

I believe that it is not a very good result but it can be a base for working more on this model. In the next step by using some Ensembling methods and changing the parameters, we can improve the accuracy. In addition, if we use the structure of the server and clients, we will improve the speed of the run.