Develop scikit-learn machine learning models to predict the function of the proteins on InterPROscan dataset

1. Introduction

In this project, various machine learning models are developed to predict the function of proteins using the InterPROscan dataset. It is implemented using the Spark framework, an open-source distributed processing system for big data.

Generally, three main steps are performed:

1. Data Preparation
2. Machin Learning (ML) Implementation:
3. Evaluation

2. Analysis and Results

This chapter elaborates on the outlined steps including data preparation, ML implementation, and evaluation.

2.1. Data preparation:

This step is the most important step which exploring and identifying the data should be done very strictly. The main dataframe consisted of 1921817 records with information on different annotations for each protein, and 332775 distinct proteins. The Following measures are taken to prepare a subset ready for ML implementation:

- Dropping out the columns that are not required in the next steps.
- Filtering out the rows with no protein annotation. (i.e InterPro_annotations is “-’)
- Checking for null records in the dataframe.
- Adding a new column "size" which is the percentage of protein annotation size to protein size: df = df.withColumn('size', (((df['Stop'] - df['Start']) / df.Seq_len) * 100))
- Creating a separate dataframe “large_df” for large protein annotations (size greater than 90%). Only the largest annotation was selected for classification, as there may be more than one large annotation for a protein.
- Making a list of distinct Proteins with large features for finding small annotations, named all_protein (208741 Protein).
- Creating a new dataframe “small_df” from the main dataframe for small features including only proteins that have the large feature (from the previous step). The size of the feature is also restricted to less than 90%. The number of distinct proteins in this dataframe is less than all_protein as some proteins only had large annotations (5202 small feature)
- Transposing small_df to pivotDF. A dataframe is needed with a row for every protein, and columns for small annotations, filled with the count of small annotations for each protein. It is made using Pivot() function on small_df. Pivot() is an aggregation that one of the grouping column values is transposed into individual columns with distinct data.
- Filling null values with zero. The dataframe may contain a null value If there were no small features in the previous step.

- Changing the type of all numerical columns from long to integer to dedicate less memory.
- Joining pivotDF and large_df to add the corresponding large annotation to each protein.
- Combining multiple columns into a single feature vector to train ML models. all numeric columns are merged into a vector column using VectorAssembler.
- Encoding label column (InterPro_annotations) which is a string to a column of label indices using StringIndexer.
- Splitting the prepared data to train and test the dataset.

I should mention that Spark properties are configured to run with 16 worker threads and both "spark.executor.memory" and "spark.driver.memory" are set to 128g to avoid "OutOfMemoryError" error.

2.2. Machin Learning (ML) Implementation:

In this stage, several ML methods are applied to identify patterns. LogisticRegression, Naive Bayes, and RandomForestClassifier are implemented.

LogisticRegression is a popular method to predict categorical labels.

Naive Bayes is a family of multiclass classifiers based on applying Bayes' theorem.

RandomForestClassifier is an ensemble machine learning method with high predictive performance.

2.3. Evaluation

In this step, the accuracy of ML models is assessed. MulticlassClassificationEvaluator is used for evaluation of the Multi-class(large annotations). The accuracy of different classifiers is listed below:

- NaiveBayes; 78%
- RandomForestClassifier: 16%

In addition, 78% accuracy is achieved using LogisticRegression on the first 100,000 records. But it is not feasible to implement this model to the entire dataset because it is computationally intensive and requires significant computational resources.

It is not expected to achieve high accuracy due to several factors, including:

- The fact that only the largest features are selected
- Total size of proteins are not taken into account, probably two protein have the same small features, but they still have different size and there may be different nucleotides between small annotations.
- Other attributes, such as protein family, sequence length, and size of small annotation are not considered.

3. Conclusion

Overall, this project demonstrated the potential of using machine learning methods to predict protein function on the InterPROscan dataset. Further work is needed to improve the accuracy of the models, including considering additional factors. The Naive Bayes classifier performed better on this dataset because it is a simple model that assumes independence among the features, which is more suitable for the characteristics of the data. The Random Forest classifier is a more complex model that builds multiple decision trees and aggregates their predictions, which causes overfitting and is less accurate in some cases.