Protein Sequence Classification using Natural Language Processing

[1]Aditya Shinde, [2]Mitchell D'Silva
[1]Student, [2]Assistant Professor
[1]Department of Information Technology
[1]Dwarkadas J. Sanghvi College of Engineering, Mumbai, India

***Abstract*— Protein is an important component of every cell in the body. It is an important building block of bones, muscles, cartilage, skin, and blood. For developing competitive pharmacological products, classifying a protein sequence precisely from a large biological protein sequences dataset plays a significant role. Comparing the unseen or novel sequence with all the identified protein sequences and accurately predicting the category of protein requires efforts and are usually time consuming. Therefore, in order to improve the efficiency of protein classification, a protein sequence classification system is developed using machine learning and data mining techniques. In protein analysis, sequence alignment, sequence searching and sequence classification can be done using sequence mining techniques. Protein sequence classification has also become a field of interest for many scientists. It has the potential for discovering the recurring structures that exist in the protein sequences and precisely classify those sequences. This paper provides a novel approach for protein sequence classification using Natural Language Processing.**

***Index Terms*— Word Embedding, Embedding Layer, Classification report, Confusion Matrix, Neural Networks.**

I. INTRODUCTION

Bioinformatics is a field which involves various aspects of genetics and molecular biology, mathematics, statistics and computer science. Using the view of computational science, various data intensive and large-scale biological issues are addressed. The most common problems include modeling biological processes at a very detailed level and making conclusions and deriving patterns from collected data. In the last few years, there is an immense growth in the development in the field of genomics. Due to this, a large amount of biological data is generated. Various complex computational methods are used for making inferences from such data. It also comprises the development of efficient algorithms necessary for the analysis and simplification of sequential data. Such techniques can be used for classification and prediction of sequential data. Due to which we are able to get a gist of various experiments of life sciences. Due to frequent increase in rates of data generation, it becomes extremely essential to use data mining and machine learning techniques for such applications [3].

Due to the growth of biological information, the field of bioinformatics has developed to a large extent [3][8]. Data mining can help the researcher in finding out the new information from plethora of biological data. Sequential pattern mining is one of the important fields of data mining, in which small length patterns i.e., (20 for protein sequences and 4 for DNA sequences) and long patterns with a length of few thousands frequently appear. Such kind of data appears frequently in many fields of science, security, medical, business, etc. wherein a sequence is generally an ordered list of objects. Sequence data mining also provides essential techniques for revealing useful knowledge invisible in such colossal amount of data [4].

Protein sequence classification is a task to classify or label proteins based on the protein sequences. The word sequence in proteins denotes the arrangement of sequences of amino acids that forms protein [5]. Databases are created which stores various identified sequences of proteins. Proteins are large molecules in which one or more chains of amino acids are arranged in a specific order. The normal size of a protein molecule may be hundred amino acids, whereas the large protein molecules can have a size of thousand amino acids. 20 amino acids present in proteins are A, C, D, E, F, G, H, I, K, L, M, N, P, Q, R, S, T, V, W, Y, such a variety of proteins are found in living things [3].

An example of protein sequence of Haemoglobin beta chain is as shown below:

VHLTPEEKSAVTALWGKVNVDEVGGEALGRLLVVYPWTQRFFESFGDLSTPDAVMGNPKVKAHGKKVLGAFS
DGLAHLDNLKGTFATLSELHCDKLHVDPENFRLLGNVLVCVLAHHFGKEFTPPVQAAYQKVVAGVANALAHKY
H

This paper discusses the steps such as data pre-processing, visualization, feature engineering, modelling, training and testing the data to accurately classify various proteins into different types. Finally, a classification report is generated which shows the classification results.

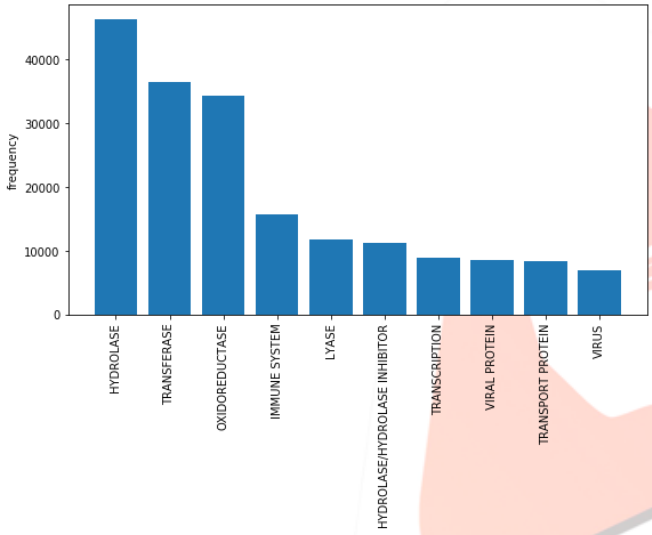
II. DATASET

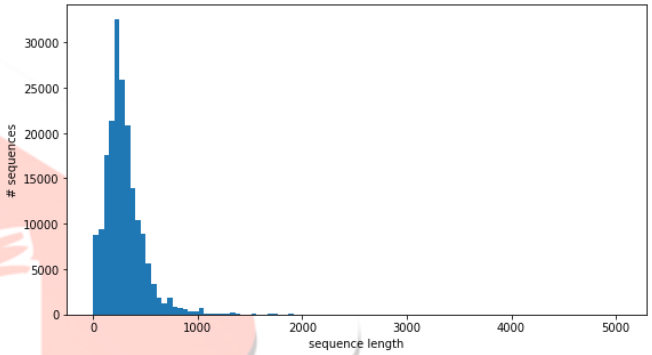
2.1 Dataset Description

Structural Protein Sequences dataset from Kaggle.com is used for this study [1]. The dataset is divided into two subsets - The first part contains protein meta data which includes details on protein classification, extraction methods, etc. and the second part contains the protein structure sequences. Both the datasets are arranged using "structureID" attribute of the protein. The first dataset consists of 1,41,000 rows and 14 columns whereas the second dataset consists of 4,67,000 rows and 5 columns. This protein dataset is obtained from Research Collaboratory for Structural Bioinformatics (RCSB) Protein Data Bank (PDB).

2.2 Data Pre-processing and Visualization

First, the two datasets are merged into a single dataset by merging them on the "structureID" attribute. After merging, the rows that are without labels and sequences are dropped or removed. Next, as the dataset contains various different macro-molecules, only protein are selected for further processing based on "macromoleculeType_x" attribute. The dataset consists of many different types of macromolecules of biological significance. The majority of the data records are of proteins. With DNA being the precursor to RNA, which when translated, proteins are the biomolecules that are directly interacting in biological pathways and cycles. Proteins are usually centered around one or a few functions which is defined by their family type. For example, we can have a protein that is from a Hydrolase group, which focuses on catalyzing hydrolysis (breaking bonds by adding water) in order to help promote destruction of chains of proteins, or other molecules. Another example would be a protein that is a transport protein, which allows other molecules such as sucrose, fructose, or even water come in and outside of the cell. Also, only 10 common protein classes are used, that is the top 10 classes on the basis of row count are selected. Figure 1a shows





the frequency of each of the 10 selected protein classes. Figure 1b shows the length of the protein sequences.

Fig 1(a). Frequency of each protein. **Fig 1(b)**. Length of protein sequences.

It can be observed that Hydrolase protein has the highest frequency whereas Virus has the least frequency among the 10 selected proteins as per Fig. 1(a). Each protein consists of more than 10000 values. As there is a huge amount of data, it helps in more accurate results. From Fig 1(b), it is observed that most of the protein sequences have length between 0 to 1000. Thus, this will help in defining the maximum length of protein sequence with consideration of majority of proteins.

2.3 Feature Engineering

It is observed that all the 10 labels are categorical values. As machine learning models can only interpret numeric values, these categorical values are need to be converted into binary or numeric form. For this, the labels in string are transformed to one hot representation using LabelBinarizer. In one hot representation, the values are assigned 1 if that value is present else the values are assigned 0 [12][13].

For further pre-processing of sequences, Tokenizer method from Keras library is used which translates every character in the sequence into a number. Also, the length of every sequence is made uniform for precise processing. Here, a maximum length of 256 characters is used.

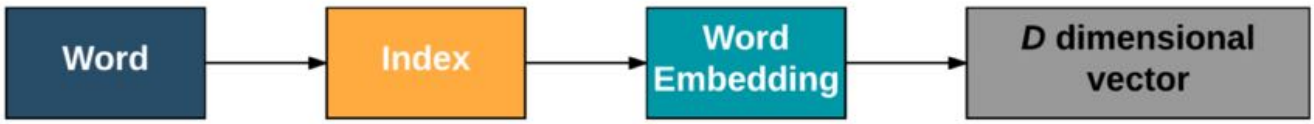
III. ALGORITHMS USED

3.1 Word Embedding

A word embedding is a representation for text where words that have a similar meaning have a corresponding similar representation. This approach of representing words is considered as a major breakthrough of deep learning on challenging Natural Language Processing (NLP) problems. Word embeddings are techniques using which individual words are represented as real-valued vectors in a predefined vector space. Each word is mapped to one vector and the vector values are learned in such a way that resembles a neural network, and thus this technique is generally considered into the field of deep learning.

Each word is represented by a real-valued vector. The vector can be of tens or hundreds of dimensions. On the other hand, for one-hot encoding, a thousands or millions of dimensions are required. Embeddings overcome the two limitations of a common method for representing categorical variables such as one-hot encoding. First, the dimensionality of the transformed vector becomes unmanageable for high-cardinality variables - those with many unique categories. Second, the mapping is completely uninformed i.e. "similar" categories are not placed closer to each other in embedding space.

Based on the usage of words, the distributed representation is learned. This allows words that are used in similar pattern to result in having similar representations, in-turn capturing their exact meaning. For example, cat and dog, they both are different



in their own way but similar in a lot of ways, they both have 2 eyes and 4 legs.

Fig 2. Word Embedding [10]

3.1.1. Word Embedding Algorithms

This section discusses the algorithms used in word embedding.

3.1.1.1. Embedding Layer

An embedding layer, is a word embedding that is learned along with a neural network model on a specific NLP task, such as classification of documents. One-hot encoding of each word is performed so that text in the documents must be cleaned and prepared. The size of the vector space is 50, 100, or 300 dimensions. It is defined as a part of the model. Small valued random numbers are used to initialised the vectors. The embedding layer is used on the front end of a neural network i.e, the first layer of network and is fitted in a supervised manner using an algorithm called backpropagation [9].

Keras library provides an embedding layer that can be used for neural networks on textual data. Its requirement is that the input data should be integer encoded. Therefore, each word is represented using a unique integer. Tokenizer API is used for data pre-processing. It is a flexible layer and can be used in several ways such as it can be used solely to learn word embedding that can be saved and used in the future with different models. It can also be used as a part of a deep learning model where the embedding is learned simultaneously along with the model. Another way to use it is to load a previously trained word embedding model [10].

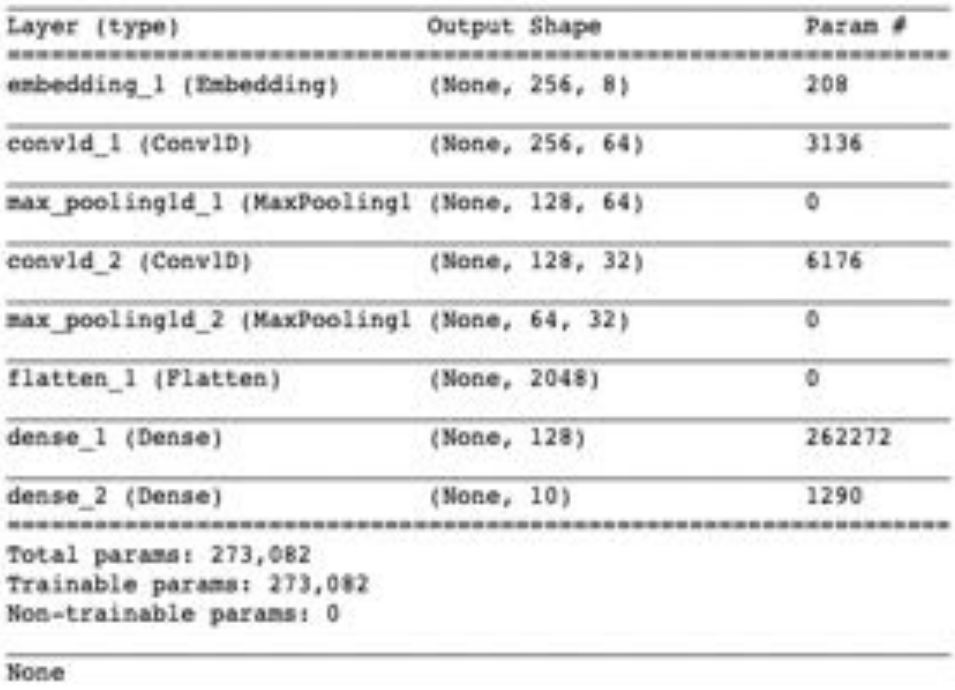
An embedding layer is usually the first hidden layer of a network. It has 3 arguments - **input_dim**: size of the vocabulary in the textual data, **output_dim**: size of the vector space in which words will be embedded, **input_length**: length of input sequences. The output of the embedding layer is a 2D vector where each word in the input sequence of words has one embedding. To connect a Dense layer directly to an Embedding layer, it is first required to flatten the 2D output matrix to a 1D vector using the Flatten layer [15].

IV. EXPERIMENT

The data set consists of total 18 columns or attributes. The target is to classify proteins into labels - Hydrolase, Hydrolase / Hydrolase Inhibitor, Immune system, Lyase, Oxidoreductase, Transcription, Transferase, Transport Protein, Viral Protein and Virus. Deep Learning Model is created to predict the desired outcome. The dataset is divided into two parts - Training data and Testing data. 80% of dataset is used for training purpose that is to train the model and 20% of the dataset is used for testing purpose that is to test the model. The test dataset is used to evaluate the accuracy of the model. The target value in the test set is compared with the predicted value and a classification matrix is generated for each of the protein.

V. IMPLEMENTATION

For this study, Jupyter Notebook with python 3.0 is used for implementing the model. For making predictions with more accuracy, scikit-learn libraries are used. Scikit-learn is a free machine learning library and it is an extension to SciPy. Keras Library is used for the deep learning neural network implemented in the study. The architecture of the neural network developed is as shown in Fig.3.



Layer (type)	Output Shape	Param #
embedding_1 (Embedding)	(None, 256, 8)	208
conv1d_1 (Conv1D)	(None, 256, 64)	3136
max_pooling1d_1 (MaxPooling1	(None, 128, 64)	0
conv1d_2 (Conv1D)	(None, 128, 32)	6176
max_pooling1d_2 (MaxPooling1	(None, 64, 32)	0
flatten_1 (Flatten)	(None, 2048)	0
dense_1 (Dense)	(None, 128)	262272
dense_2 (Dense)	(None, 10)	1290

Total params: 273,082
Trainable params: 273,082
Non-trainable params: 0

None

Fig 3. Architecture of the model

The classification report is generated for evaluating the model. It shows a representation of the main classification metrics on a per-class basis. The metrics are defined in terms of true and false positives, and true and false negatives. Positive and negative in this case are generic names for the classes of a binary classification problem. The measures are:

- Precision
  It is the percentage of instances which are correctly classified as positive are actual positive. It is the measure of exactness.
- Recall
  It is the percentage of instances which the classifier labelled as positive. It is the measure of completeness.
- F1-score
  The f1 score is a harmonic mean of precision and recall. Higher the f1 score, higher is the accuracy.
  $$f = (2 * precision * recall)/(precision + recall)$$
- Support
  Support is the number of instances of true response in that class [11].

VI. RESULTS

Evaluation is done based on accuracy and a confusion matrix which is already implemented in sklearn. The confusion matrix is as shown in Fig.4.

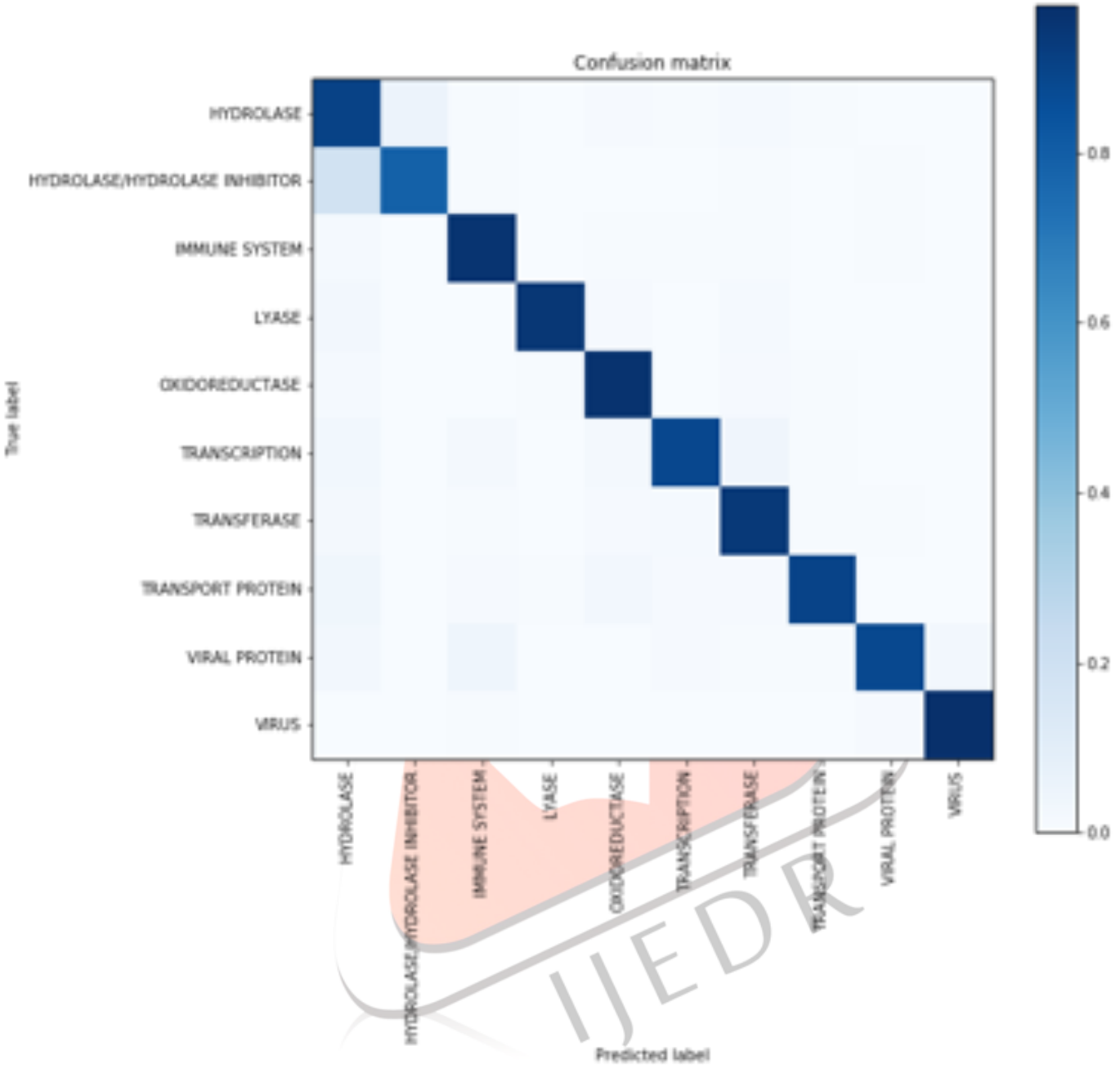

Confusion matrix
HYDROLASE
HYDROLASE/HYDROLASE INHIBITOR
IMMUNE SYSTEM
LYASE
OXIDOREDUCTASE
TRANSCRIPTION
TRANSFERASE
TRANSPORT PROTEIN
VIRAL PROTEIN
VIRUS
True label
Predicted label
0.8
0.6
0.4
0.2
0.0

Table 1. Classification Report

Protein	Precision	Re-call	F1-score	Support
Hydrolase	0.90	0.90	0.90	9146
Hydrolase / Hydrolase Inhibitor	0.77	0.79	0.78	2214
Immune System	0.92	0.96	0.94	3165
Lyase	0.96	0.95	0.95	2335
Oxidoreductase	0.96	0.96	0.96	7013
Transcription	0.88	0.89	0.88	1792
Transferase	0.95	0.94	0.94	7241
Transport Protein	0.91	0.90	0.90	1613
Viral Protein	0.93	0.88	0.91	1739
Virus	0.95	0.97	0.96	1413
Avg / total	0.92	0.92	0.92	37671

This result suggest the applicability of the NLP-theory to protein sequence classification.

VII. Conclusion

This study performs classification of protein sequence on the Structural Protein Sequences dataset from Kaggle with 18 features. The results obtained from the study suggest that the NLP-theory can be applied for protein sequence classification. From this study, it has been found that every protein is classified by more than 90% precision. Thus, this approach can be used for protein sequence classification which is a significant task for developing various pharmacological products and detect various diseases or disorders. Developing such model will be a step forward in the use of computational science in the field of bioinformatics.

VI. Future Scope

Further improvements in the model can be done using a deeper model, small sequences and using n-gram as words for word embedding to accelerate learning. The model can also be validated using cross-validation. Another major concern of over-fitting data can be overcome by employing techniques such as Dropout or regularizing by adding penalties for large weights (kernel_regularizer) or large activations (activation_regularizer).

References

[1] Structural Protein Sequences Kaggle Inc. [Online] Available: https://www.kaggle.com/shahir/protein-data-set
[2] https://pdfs.semanticscholar.org/3a8f/bd6ed188021aee2f2392b551ba357aa81c21.pdf
[3] Dr. S.Vijayarani, Ms. S.Deepa "Protein Sequence Classification in Data Mining - A Study", International Journal of Information Technology, Modeling and Computing (IJITMC) Vol. 2, No.2, May 2014
[4] Anita Zala, Mehul Barot," Mining Sequential Pattern with Time-Constraint", International Journal of Engineering and Innovative Technology (IJEIT) Volume 2, Issue 7, January 2013, ISSN: 2277-3754
[5] Arabinda Panda, "Bio-Data Mining:Concepts & Applications", International Journal of Computer Science and Management Research Conference Issue DPDM 2012, ISSN 2278-733x
[6] Cornelia Caragea, Adrian Silvescu, Prasenjit Mitra, "Protein Sequence Classification using Feature Hashing".
[7] Jason T. L. Wang, Qic Heng Ma, Dennis Shasha, Cathy H Wu, "Application of Neural Networks to Biological Data Mining: A case study in Protein Sequence Classification", pp: 305-309, KDD, Boston, MA, USA (2000).
[8] P.K.Vaishali, Dr.A.Vinayababu," Application of Data mining and Soft Computing in Bioinformatics", International Journal of Engineering Research and Applications (IJERA) ISSN: 2248-9622
[9] Word embedding- https://machinelearningmastery.com/what-are-word-embeddings/
[10] https://techblog.gumgum.com/articles/deep-learning-for-natural-language-processing-part-1-word-embeddings
[11] Classification report - http://www.scikit-yb.org/en/latest/api/classifier/classification_report.html
[12] Deep Sanghavi, Jay Parekh, Shaunak Sompura, Pratik Kanani, "Data Visualisation and Improving Accuracy of Attrition Using Stacked Classifier", International Journal of Engineering Development and Research (IJEDR), ISSN:2321-9939, Vol.6, Issue 4, pp.284-293, November 2018, Available :http://www.ijedr.org/papers/IJEDR1804054.pdf
[13] Dhvani Kansara, Rashika Singh, Deep Sanghavi, Pratik Kanani, "Improving Accuracy Of Real Estate Valuation Using Stacked Regression", International Journal of Engineering Development and Research (IJEDR), ISSN:2321-9939, Vol.6, Issue 3, pp.571-577, September 2018, Available :http://www.ijedr.org/papers/IJEDR1803097.pdf

[14] https://www.hindawi.com/journals/bmri/2014/103054/
[15] https://machinelearningmastery.com/use-word-embedding-layers-deep-learning-keras/
[16] Suprativ Saha and Rituparna Chaki, "A Brief Review of Data Mining Application Involving Protein Sequence Classification"