University of Essex
CE807 – Assignment 2 - TASK 3: Report

Registration Number: 1901423

University of Essex – Colchester Campus
Department: Computer Science and Electronic Engineering
CE807 – Assignment 2 - Final Practical Text Analytics and Report
Dr habil Ansgar Scherp(Supervisor)

April 21, 2020

1 Abstract

Extreme Multi-label class (XMC) is a great and interesting device studying mission within the discipline of natural language processing (NLP) that assigns to each instance its most relevant labels from a a huge collection of label where the numbers of labels, features and illustrations can be millions or trillions. he tremendously large label collections produce challenges such as computational complexity and huge processing power which is required to process data. [1]

Many methods have been projected to meet the challenges.Machine learning knowledge of XMC version has been proposed with a Multi label K nearest neighbor (MLkNN) [2] technique,tf idf vectorizer and pipeline approach from sklearn followed by a multi label binarizer has been used from processing part of sklearn to normalize data. [2]

Classifier build in our code is specifically a pipelining classifier to meet the requirements. In the beginning of the approach, some data preprocessing techniques has been applied which are necessary to make our textual data smooth and workable as well as gives the best results.[3]After making data usable we come to feature extraction which is an essential part of a Machine Learning model.

After attempting the task1 we got to know various Extreme Multilabel Classification(MLkNN) Techniques and as a consequence tried different different techniques and methods in our this task.After exploring various areas it has been decided to use Multilabel K-nearest neighbor from scikit-multiaern accompanied by pipelining and tf idf vectorizer followed by multi label binarizer as the best approach to fullfill our requirements.(MLkNN) is said to be the technique derived from KNN algorithm.[2]

For each unknown illustration , there has been devised a solution where K nearest neighbors in the training set has been primarily recognized.By using the statistical information obtained from collection of labels of the neighbouring illustrations i.e., the quantity of illustrations which are neighbouring fits into each class.Maximum a posteriori (MAP) principle has been employed to devise a bunch of labels for the unobserved illustrations. The approach which should be followed is that we have to test on both datasets named as pubmed and econbiz and ultimately manifest that ML-KNN accomplishes great performance while being deep-rooted around multi-label learning algorithms.[4]

After applying MLKNN,it has been computed that it is not giving a reasonable accuracy on both the data sets.The only solution left with us is to go for some other classifier for carrying out the task of extreme multi label classification. To be carried out with this approach we have tested different classifiers on both the datasets in order to come up with high accuracy.After going through lots and lots of classifiers specifically each and every classifier of sklearn and skmultilearn libraries we finally came up with a GaussianNB classifier which proves to be best for both the datasets.

The pipelining approach has been used in our task.Basically,we have pipelined the classifier with tfidf vectorizer and labelPowerset classifier.Although ,we have also tried some other classifiers too from skmultilearn like Binaryrelivence, OVA, and chainclassifer.Having tried all the classifier , at last we have found that labelpowerset will be best approach for both our datasets.As per the research has been done it has been analyzed that when XMC is combined with GaussianNB and tfidf vectorizer it will be the best approach for getting high accuracy for carrying out the process of XMC(Extreme multilabel classification).

Keywords:Multi label, Machine Learning, Extreme Multi label classification, Huge data sets, MLkNN, MAP, Natural language processing, NLP, Text processing

2 Multi label classification

If we are going with the machine learning approach ,there are some variants associated with classification problem which are multi-label classification and strongly related problem of multi-output classification where multiple labels has been assigned to each instance of the class.Multi class classification has been inherited from Multi-label classification which is considered as the single-label problem for categorizing instances precisely into more than two classes.In multi label problem there is no limitation for maximum how many classes the classes can be assigned.

We can say that multi-label classification is the basically a puzzle for finding a model that best fits or map our input x to binary vectors y by assigning a value of 0 or 1 for each label or element in y. Multi-label classification methods demand has been increased in modern applications such as such as protein function classification, music categorization, and semantic scene classification.

The basic task performed by the Multi-label classification is that it organizes the sparse related literature into a structured presentation and then performs comparative experimental results on certain multi-label classification methods.It has a conceptual contribution also for quantifying multi-label nature of a data set.

It is not easy to carry out with the approach of Multi label classification.It is very vast as well as deep field majorly focusing upon text mining and data science domains.It allows us to dig into huge datasets from different fields like medical, business industries of different types.Text analysis plays an important role in the field of data science.Whenever there is a thought of multi label classification, we have to consider processing power and required time to process the dataset.

Whenever we are dealing with multi label classification ,we have to keep one thing in mind that it will require heavy computation and large amount of time for the execution.There is a possibility that at the time of getting the results of our classifiers with large amount of data it can require hours also for running the classifier.We know the fact that some classifiers are very slow, some are medium speed in processing data and some are very fast but still we can not deny the fact that majorly the speed of our classifier depends majorly on the structure of the dataset.

2.1 Extreme Multi label Classification

Extreme multi-label text classification (XMTC/XMC) is the process of assignment of most relevant labels from an huge collection of labels for each individual document where there is no limit towards having number of labels it could reach upto hundreds,thousands or millions.

With the advent of having large collection of labels ,there has been an increase in research challenges that can be data sparsity and scalability.It is justified to say that there has been a significant improvement in recent years with the growing usage of machine learning methods such as tree induction with large-margin partitions of the instance spaces and label-vector embedding in the target space. However,We have explored XMTC for our particular piece of work but we can not deny the fact that machine learning are achieving greater success in other realted areas as well.

We have tried to use machine learning specifically for XMTC/XMC by using family Multi label classification model which has been came into picture specifically for multi-label classification concept.With a comparative evaluation of other methods,it has been shown that MLknn approach successfully scaled to the largest datasets and can produce best results consequently on all datasets.

The primary aim of using extreme multi-label classification (XMC) is to train a classifier which will assign relevant labels to an instance from an extremely large group of target labels.

The distribution of training instances among labels depicts that larger the fraction of labels smaller the number of positive training instances.It is one of the important challenge in XMC to detect labels which accounts for the diversity of the label space as well as account for a large fraction of all labels.

The label detection task in XMC is interpreted as robust learning in the presence of worst-case perturbations.We have come up with this view by having an observation that there has been a tremendous change in the distribution of features for instances of these labels from training set to test set.

For the time being considering shallow classifiers, the robust approach to XMC gives us way towards regularized classification as well.It has been shown that by minimizing hamming loss with the appropriate regularization will excel advanced approaches.Furthermore, we additionally highlight the sub-optimality of the coordinate descent-primarily based solver which is interesting in its own way.

Extreme Multi label classification is one of the most widely used method in the field of text mining and machine learning. It's not easy to tackle Extreme multi label classification using machine learning classifiers. The reason behind this discussion is type and structure of datasets used for extreme multi label classification.

In extreme multi label classification multi labels are given in each row, yes not different rows, they exist in rows. A single row can have 5 to 10 or 15 labels. So, when you have a dataset of 1000 or 10000 rows then you have millions of labels, these labels are processed by classifier to predict the right label. This is the reason that classifiers in extreme multi label classification gives lesser amount accuracy then for other text-based datasets.[5]

3 Data Pre-Processing

Data preprocessing have a significant role while carrying out the process of data mining. It basically works on the concept of extracting useful information from the large pile of data and leave the rest which are no use to us.Some of the approaches are unbounded which lead to out of bound values.

Sometimes we can get mislead outcomes by the lack of screening problems for certain issues while carrying out analysis of the data.It has been comprehended that it is very important to handle the way of representation of data as well as quality of data before moving forward towards any analysis.Especially in computational tasks data prepossessing is considered as the most valuable step.

If there is much irrelevant and redundant information present or noisy and unreliable data, then knowledge discovery during the training phase is more difficult. Data preparation and filtering steps can take considerable amount of processing time. Data preprocessing includes cleaning, instances, normalization, transformation, and selection. The product of data preprocessing is the final training data set.

Knowledge is one most basic and formost step while devising a machine learning algorithm.It is well known that if our source data is not relevant as well as carries redundant information and noise then while traning the model knowledge discovery will become the difficult step to proceed with.It is fact that preparing and filtering data will take ample time.Some of the basic steps of data preprocessing consists of cleaning ,creating instances,normalizing the data and extract the required data.

Different text preparation techniques are used alongside the machine learning model in the code to make the data normalized and smooth. These techniques are essential part of the text processing to build a machine learning model. [6]

First of all we make title column in both data sets lower case, then we removed the punctuation symbols from both datasets, then we used NLTK library to remove stop words from the both datasets.The words which are occurring very frequently then search engine has been programmed in such a way that it will remove all frequent words while carrying out any search operation.These words are known a as stop words.[7]

Stop word removal is applied on title column of econbiz dataset as well as on the title and labels columns of pubmed dataset.

Then text lemmatization (The process of having returning root words only by removing inflectional endings) is applied on title column of econbiz dataset and title and labels columns of pubmed dataset.[8]

All above mentioned techniques and functions are useful for almost all the datasets and in their classifications but when we have tested these techniques on our datasets and then we applied our classifiers the results were horrible and unbelievable, accuracy percentage of classifiers dropped in a unexpected manner, this was undeliverable because our knowledge told us that these techniques increase the accuracy rate of a classifier. So, due to these reasons we removed these techniques from our project and we decided to complete our project or task without these techniques. So, we did this.

We are not going to ignore these techniques in future because they are not useless after all but for the time being, we are forced to consider them useless techniques as they are not working well for this project.

4 Libraries used

Libraries are basically the short cut approach approach of performing certain action with writing lines of code.To complete assigned task few libraries are used named as Pandas,genism,numpy,sklearn.Pandas is well known library of python used specifically for data analysis such as time series and numeric tables by providing optimized outcomes.[9]Genism is used in unsupervised models and natural language processing majorly. Numpy is used for providing multidimensional arrays and different tools to do processing on these arrays.Sklearn is used while using various machine learning and cross validation algorithms specifically.[6]

All these libraries are not the end of text mining, machine learning or data science but for the time being and for this project it has been decided to conclude on the use of these libraries as we do not have more time to entertain and test some other libraries although many other libraries do exist in the world of data sciences, machine learning and extreme multi label classification but we can not entertain them for the time being and for this project specifically.

It has been tried to work on other libraries as well if we would have got some chance ,projects or tasks to work on them. It is quite possible that we would like to start work on them by our own as text mining and machine learning is very vast and attractive field. so, we will surely work on other libraries in future as per the requirements.

5 Features selection and extraction

In features selection title and labels columns from both the data sets named as pubmed and econbiz respectively are used. [10]

While examining both the datasets it has been witnessed that Title is used as input variable (X) and labels is used as output variable (y).[11]

tf-idf(term frequency-inverse document frequency) is the measure of finding out how relevant a word is in a document.Text mining is the major field where it has been used while scoring a relevance of the document.It is used in such a way that each word in a document has been scored on the basis of its tdf and idf values which in turn tells us which word is more important and which is not.Higher the score ,more important will be the term consequently.tf is calculated by calculating the frequency of each word in a document. For calculating idf first we have to divide total number of documents by no of times word appears in a document and then applying algorithm on the whole fraction.Finally ,by combing both tf and idf we will obtain final scoring of particular term.vectorizer is used for feature extraction and text vectorization. [1]

Multi label binarizer is used to encode the data. It has the capability to encode more than two labels for each object.It can transform the array with the help of dataframe. [12]

Pipeline is normally being in use by association with feature union which adds the final outcome of transformer into a complex feature space.It has the authority to allow us to bind multiple processes into one estimator of scikit-learn.It contains methods just like score,predict,fit similar to any other estimator.There has been one importation called from sklearn which has been used extensively for having more efficiet and accurate predictions at the time doing classification.Train_test_split is basically splitting the array or matrices randomly into training and test set.The splitting of data into training and test set will be carried out using from sklearn to make use of data more efficiently.Testing data is 20% and training data is 80%.random_state parameter is used in train_test_split to avoid random results from model. [13]

6 Model Building and Testing

Finally using all previously mentioned tools model is built for training and testing.

Classifier is fitted with X_train and y_train parameters.[4] Predict function has been combined with classifier where X_test parameter is provided to test the model.Predicted results have been stored in a variable named as y_predict.[14]

Datasets used in this coursework are huge datasets which requires heavy processing power and a lot of processing time.In order to avoid this hurdle few rows of each dataset are taken to train and test the model.

With all these mentioned details accuracy obtained by model on dataset named as pubmed is not good enough but its accuracy will increase remarkably when the complete dataset will be used for model testing and training as amount of data plays a significant role in obtaining accuracy. [15]

Similar Classifier which has been discussed in previous paragraphs are applied on the second dataset named as econbiz.Accuracy obtained by model on this dataset is little bit lesser then dataset named as pubmed.The reason behind this lesser accuracy is that the dataset named as enconbiz.csv has different type of data in its labels column.It has numerical data with dashes, this data cause lesser accuracy as multi label binarizer is built specially for textual data.

Hamming loss is also calculated for each dataset to judge the classifier in better way. Hamming loss is 0.74% for dataset named as pubmed.csv which negligible. Hamming loss for data set named as econbiz is 0.59% which is also negligible which clearly shows that it is less then dataset named as pubmed.csv. This point can be noted and highlighted here that dataset named as pubmed.csv has high accuracy rate as well as high hamming loss on built classifier. On the other hand ,dataset named as econbiz.csv has lesser accuracy and lesser hamming loss as compared to data set named as pubmed.csv. hamming loss matrix will be discussed later on, for the time being let's talk about f1 score.f1 score is used in the model to calculate the accuracy of the model. F1 score is a part of matrics which is sub part of sklearn library.[16]

6.1 f1_score

While doing binary classification by using specifically statistical analysis F1 score will be used to compute the test's accuracy of the classifier.It is calculated as the weighted harmonic mean of test's precision and recall.The approach uses precision and recall both. Precision is the ratio of the quantity of accurate positive outcomes and total number of positive outcome.While Recall is the total number of correct positive divided by total number of samples which are relevant.F1 score is equal to harmonic mean of precision and recall followed by condition that the best range will be reach out at 1 which is most accurate precision and recall.[17]

$$F_1 = \left(\frac{2}{\mathrm{recall}^{-1} + \mathrm{precision}^{-1}} \right) = 2 \cdot \frac{\mathrm{precision} \cdot \mathrm{recall}}{\mathrm{precision} + \mathrm{recall}} .$$

There is a need of this evaluation parameter at the time of maintaining proper balance between precision and recall parameters by considering the test values of precision and recall respectively.Precision is the ratio of true positive and sum of true positive and false positive while recall is the ratio of true positive and sum of true positive and false negative.F1 score is the average of both precision and recall.It is computed by dividing multiplication of 2, precision and recall with addition of precision and recall.It is of more use as compared to other evaluation parameter called accuracy.The worst value of f score will be when there is low value of precision and recall which is 0 While the best value will be when both precision and recall values are 1.

Here in place of f1-sore we could use accuracy_score from sklearn matrix but we did not because accuracy_score matrix is good for numerical data and other simple text data but it is not enough good for multi label classification. As it was oroginally built for calculating and estimating numerical and mathematical data, it can not be used for extreme multi label classification.

After leaving these details behind we decided to try this matrix on our dataset at least once to check its behaviour, working style, performance and accuracy.Finally, we applied it on our dataset and results were as expected.Its performance and accuracy was far worst then f1_score .so, it has been decided to continue with f1_score.It has been found that F1_score is the best tool for extreme multi label classification.

Few other scorers are also available for performance calculation or what we call accuracy score but as we were running out of time.So, it has been decided not to waste time on other well-known scorers or accuracy calculators. This decision of us saved our time which we invested in our coursework to learn something more from it and optimize.

6.2 Hamming loss

Hamming loss is other one of the important evaluation parameter for machine learning algorithms which is calculated as the ratio of wrongly predicted labels and total number of labels in the classification approach.

$$\frac{1}{|N| \cdot |L|} \sum_{i=1}^{|N|} \sum_{j=1}^{|L|} \mathrm{xor}(y_{i,j}, z_{i,j}), \text{ where } y_{i,j} \text{ is the target and } z_{i,j} \text{ is the prediction.}$$

When we are talking about carrying out multiclass classification,this evaluation parameter is calculated as the hamming distance between y_true and y_pred which is equal to subset zero_one_loss method at the time of assigning the normalize parameter to true.[6].When we are referring to hamming loss in multilabel classification it is far peculiar from the zero-one loss.Zero one loss is precisely a loss function used at the time of using classification approach.Zero will be used for correct classification and 1 is used for misclassification respectively.[15]

At the time of having normalize parameter to true, hamming loss is occupied by the subdivision zero-one loss. It is always between 0 and 1, lower being better. Hamming loss metrics which has been used to check the loss resulted at the time of built a model.Percentage of this loss depicts+ the estimated number of wrongly predicted labels in the extreme multi label text classification XMLTC/XMC.

6.3 Multi label k Nearest Neighbors (MLknn)

Multi-label is began to came into picture from the examination of content classification issue where each report may have a place with a few predefined points at the same time.[2]

In multi-label learning, the trained data is made out of occasions each related with a lot of labels, and the assignment is to anticipate the mark sets of unseen cases through examining training occurrences with known labels consequently.

In this paper, a multi-label lethargic learning approach named ML-KNN is introduced, which is gotten from the customary K-closest neighbour (KNN) calculation. In detail, for each shrouded occasion, its K closest neighbour's in the trained set are recognized from the outset specifically.[14]

From that point onward, in light of measurable data picked up from the labelled data of these neighbouring objects, for example the quantity of neighbouring examples having a place with every conceivable class. maximum a posteriori (MAP) (MAP) rule is used to decide the label set for the unnoticed occurences.

MLknn is an Adapted calculation, as the name proposes, adjusting the calculation to straightforwardly perform multi-label arrangement as opposed to changing the issue into various subsets.

For instance, multi-label variant of kNN is depicted by MLkNN. Along these lines, let us rapidly execute this. Multi-label arrangement is a unique learning task in which any occurrence is perhaps connected with multiple classes at the same time.

Step by step instructions to plan and execute proficient and powerful multi-label calculations is a difficult issue. The k-closest neighbor (kNN) technique and its weighted structure (MLkNN) are basic yet successful for binary and multi-class characterization. In this bit of work, there has been an arrangement of weighted kNN variant for multi-label characterization (MLkNN) as indicated by Bayesian hypothesis.

Through approximating a question example by the linear weighted whole of k-closest neighbors as far as least squared error, the loads are resolved adaptively by tackling a quadratic programming with a condition of unit simplex. Extraordinarily, our MLkNN is yet a free model and case-based learning strategy which just includes a tunable parameter k. Trial concentrate on two datasets specifically outlines that our MLkNN beats seven existing high-performed multi-label calculations.

The multi-label order is one of the latest and flow examine directions since it tackles numerous genuine issues where each article can have a few semantics. One of the classifications committed for gaining from such information is the adaptation strategies.

In this undertaking it has been proposed to improve the presentation of the Multi-label K Nearest Neighbours (MLknn) where it adjusts K Nearest Neighbours calculation to Multi-label information. The exploratory outcomes on five little to bigger multi-label datasets from various domains shows the viability of the ensemble techniques to improve the root algorithm.

Sluggish multi-label learning calculations have become a significant subject of research inside the multi-label network. These calculations for the majority cases considers the arrangement of standard k-Nearest Neighbors for another object to foresee its labels (multi-label).

The forecast is made by following a democratic standard inside the multi-labels of the k-Nearest Neighbors set of the new example. The Mutual and Not Mutual Nearest Neighbors rules which have just been utilized by languid single-learning calculations. These systems have been utilized to expand the apathetic multi-label calculation knowna as MLkNN.

A trial assessment completed to contrast both mutual procedures and the root MLkNN calculation just as notable aas MLkNN lazy calculation on 15 benchmark datasets which eventually shows that MLkNN introduced the best prescient presentation for the Hamming-Loss assessment measure. In spite of the fact that it essentially beat by the commonality systems at the time of talking about F-Measure . The best after effects of the lazy calculations were additionally contrasted and the outcomes acquired by the methodology utilizing three diverse bases in Multi label binarizer learning model.[14]

After testing on both data sets, we found MLKNN classifier is not a good approach towards extreme multilabel classification, we tried different other classifiers from sklearn and skmultilearn libraries and we found that GaussianNb classifier combined with labelpowerset classifier from skmultilearn supported by tfidf vectorizer is best approach towards extreme multi label classification and to get highest accuracy percentage in the filed of XMC.

6.4 LabelPowerset

Extreme multilabel classification has been categorized into two approaches called adaptation and transformation. With the help of skmultilearn library,transformation technique called Labelpowerset has been carried out.

Three most important techniques are considered for transformation (Binary Relevance, Label Power-Set and Classifier Chain). We found labelpowerset is the best fit for the datasets. Label Powerset is a simple transformation method to predict multi-label data. This is based on the multi-class approach to build a model where the classes are each labelset.

Significant techniques taken into account are Binary Relevance, Label Power-Set and Classifier Chain.It has been analysed that labelpowerset is best option which will fit the classifiers designed for the datasets.For predicting multi-label data this simple transformation method has been used.Multi class approach has been followed to build a model where classes are labelset specifically.

Labelpowerset is capable enough to do the transformation of Extreme multi label by doing the convergence of it into one or more single label classification.There are three significant limitations for carrying out the transformation process:

- the dependency between labels,
- the complexity of the algorithms
- the choice of single-label classifier.

Labelpowerset is an transformation technique which is specifically designed to build a multi-label classification which will convert multi label problem into multi class problem being associated with one multi class classifier which will consequently train all unique label combinations available in training set.The function is doing mapping for each combination into a unique combination of id number and multi class classification where multi class classifier and combination id's are used as classifier and classes repectively.

Parameters:

Classifier: (Base Estimator) – scikit-learn compatible base classifier

require_dense:It will check whether dense representation is required by base classifier for input the features and labels matrices are fitting the model or if there is no value given then sparse representation will came into picture where base classifier is an object of skmultilearn.base.MLClassifierBase else it will be dense only.

6.5 Gaussian Naïve Bayes classifier (GaussianNB)

A Gaussian Naive Bayes calculation is an extraordinary kind of naive Bayes calculation. It is explicitly utilized when the features have persistent qualities. It is additionally expected that all the features are following a gaussian distribution i.e., normal distribution. A Gaussian classifier is a generative methodology as in it endeavours to display class posterior and class-conditional distribution.

Along these lines, we can create new examples in input space with a Gaussian classifier. AI Classifiers can be utilized to predict. Given model information (estimations), the calculation can foresee the class the information is associated with. Training data inputed to classification algorithm . In the wake of preparing the classification algorithm (the fitting capacity) we can make predictions.

In Gaussian Naive Bayes continuous values has been related with each component where they are thought to be disseminated by a Gaussian distribution. A Gaussian distribution is additionally called Normal distribution.

It is a classification system dependent on Bayes' Theorem with a condition of having independence among indicators. Basically, a Naive Bayes classifier accept that the presence of a specific element in a class is disconnected to the presence of some other element.

Naive Bayes strategies are group of supervised learning algorithms which depends on applying Bayes' hypothesis with the " Naive " presumption of conditional independence between each two of kind features given the estimation of the class variable.

we can utilize Maximum A Posteriori (MAP) estimation to appraise P(y) and P(xiy); the previous is then the general occurrences of class-y in the train set. The naive Bayes classifiers contrast for the most part by the presupposition they make with respect to the appropriate P(xi—y) value.

Notwithstanding their clearly over-disentangled suppositions, naive Bayes classifiers showed their significance in some practical applications in world, for example, classifying bunch of documents and spam filtration. They require a modest quantity of data information used for training to do the estimation of the important parameters.

Naive Bayes students and classifiers can be incredibly quick contrasted with increasingly modern techniques. The decoupling of the class restrictive component conveyances implies that every distribution can be freely evaluated as a one-dimensional distribution.Thus assists with easing issues of originating from the scourge of dimensionality.

On the other side, although naive Bayes is known as an average classifier, it is known to be an awful estimator. The likelihood yields from indicators are not to be paid attention so well.

$$P(x_i \mid y) = \frac{1}{\sqrt{2\pi\sigma_y^2}} \exp\left(-\frac{(x_i - \mu_y)^2}{2\sigma_y^2}\right)$$

The Gaussian Naive Bayes belongs to the category of classification in machine learning. Close to the Gaussian Naive Bayes other models are also available, for example, Multinomial naive Bayes and the Bernoulli naive Bayes. I picked the Gaussian Naive Bayes since it is the least complex and the most famous one.

It has been understood that GaussianNb classifier when joined with labelpowerset classifier from skmultilearn by tfidf vectorizer is considered as best methodology for XMC as well as for getting most elevated precision rate in the area of classification which is multi label specifically.

7 Comparison and contrast

Task 1 urged us to build a classifier different then itself. So, multi label classification has been explored specificaally extreme multilabel classification. After having careful consideration of outcomes of task 1 ,the classifier has been developed which has been tested on 2 datasets named as pubmed.csv and econbiz.csv. Classifier gave best performance on both datasets individuals classifiers.

Different methods were mentioned in the task 1. Named as Extreme multi label classification XMC, one-vs-all (OVA) approaches, Partitioning methods, and Deep learning-based methods.The idea was to adapt something to advance approach rather then going on to these basic approaches.

Pipelined classifier combined with tf-idf vectorizer and (MLknn) classifier has been built. The final Classifier is entirely based on Extreme multi label classification (XMC)[4]

After testing different approaches and classifiers from sklearn and skmultilearn libraries it is proved that gaussiannb is a best classifier for extreme multi label classification. It gives higher accuracy from other classifiers. And labelpowerset which is a transform approach for multi label classification from skmultilearn stands the best approach from other approaches as described in the task1.

So, Gaussian classifier has been implemented with labelpowerset approach to meet requirements of the task. The task majorly focussing on our abilities regarding extreme multi label classification, text mining, data sciences and machine learning.

8 Conclusion

Working with Extreme multi label classification is not a piece of cake. In Extreme multi label classification Huge data is processed, cleaned and analyzed.

We find that the accuracy rate obtained by is average accuracy percentage, it is not best. But when we talk about the field of extreme multilabel classification, this accuracy achieved by us on extreme multi label classification is good enough to meet the requirements of task. The reason behind this less accuracy is already addressed and explained in the previous section of this coursework.The best of efforts are put for delivering a best extreme multilabel classifier for both the datasets and follow the proper text mining approach to prepare the data for the different models.

To Recapitulate from the above explanation and discussion we can say that Our proposed classifier can be improved by applying parameter tuning but still it is useful and meaningful classifier.

References

[1] I. Chalkidis, M. Fergadiotis, P. Malakasiotis, N. Aletras, and I. Androutsopoulos, "Extreme multi-label legal text classification: A case study in eu legislation," *arXiv preprint arXiv:1905.10892*, 2019.

[2] K. Bhatia, H. Jain, P. Kar, M. Varma, and P. Jain, "Sparse local embeddings for extreme multi-label classification," in *Advances in neural information processing systems*, pp. 730–738, 2015.

[3] G. Kurata, B. Xiang, and B. Zhou, "Improved neural network-based multi-label classification with better initialization leveraging label co-occurrence," in *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 521–526, 2016.

[4] F. Gargiulo, S. Silvestri, and M. Ciampi, "Deep convolution neural network for extreme multi-label text classification.," in *HEALTHINF*, pp. 641–650, 2018.

[5] R. Venkatesan and M. J. Er, "Multi-label classification method based on extreme learning machines," in *2014 13th International Conference on Control Automation Robotics & Vision (ICARCV)*, pp. 619–624, IEEE, 2014.

[6] A. Esuli, T. Fagni, and F. Sebastiani, "Boosting multi-label hierarchical text categorization," *Information Retrieval*, vol. 11, no. 4, pp. 287–313, 2008.

[7] J. Lin, Q. Su, P. Yang, S. Ma, and X. Sun, "Semantic-unit-based dilated convolution for multi-label text classification," *arXiv preprint arXiv:1808.08561*, 2018.

[8] I. Chalkidis, M. Fergadiotis, P. Malakasiotis, and I. Androutsopoulos, "Large-scale multi-label text classification on eu legislation," *arXiv preprint arXiv:1906.02192*, 2019.

[9] S. Banerjee, C. Akkaya, F. Perez-Sorrosal, and K. Tsioutsiouliklis, "Hierarchical transfer learning for multi-label text classification," in *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, (Florence, Italy), pp. 6295–6300, Association for Computational Linguistics, July 2019.

[10] F. Figueiredo, L. Rocha, T. Couto, T. Salles, M. A. Gonçalves, and W. Meira Jr, "Word co-occurrence features for text classification," *Information Systems*, vol. 36, no. 5, pp. 843–858, 2011.

[11] J.-H. Oh, K. Torisawa, C. Kruengkrai, R. Iida, and J. Kloetzer, "Multi-column convolutional neural networks with causality-attention for why-question answering," in *Proceedings of the Tenth ACM International Conference on Web Search and Data Mining*, pp. 415–424, 2017.

[12] Z. Ahmadi and S. Kramer, "A label compression method for online multi-label classification," *Pattern Recognition Letters*, vol. 111, pp. 64–71, 2018.

[13] A. Esuli and F. Sebastiani, "Active learning strategies for multi-label text classification," in *European Conference on Information Retrieval*, pp. 102–113, Springer, 2009.

[14] Z. Ren, M.-H. Peetz, S. Liang, W. Van Dolen, and M. De Rijke, "Hierarchical multi-label classification of social text streams," in *Proceedings of the 37th international ACM SIGIR conference on Research & development in information retrieval*, pp. 213–222, 2014.

[15] J. Lee and D.-W. Kim, "Feature selection for multi-label classification using multivariate mutual information," *Pattern Recognition Letters*, vol. 34, no. 3, pp. 349–357, 2013.

[16] R. B. Pereira, A. Plastino, B. Zadrozny, and L. H. Merschmann, "Categorizing feature selection methods for multi-label classification," *Artificial Intelligence Review*, vol. 49, no. 1, pp. 57–78, 2018.

[17] V. Kumar, A. K. Pujari, V. Padmanabhan, S. K. Sahu, and V. R. Kagita, "Multi-label classification using hierarchical embedding," *Expert Systems with Applications*, vol. 91, pp. 263–269, 2018.