Multi-label classification on EURLEX dataset

Pawan Joshi(223752), Shivani Jadhav(223856), Vritika Kalra(223674)

June 18, 2019

1 Introduction

1.1 Motivation and Problem Statement

A single text document can be tagged into one or more types based on its content. The EUR-Lex text dataset is a collection of documents about European Union law. The documents are classified according to several categorization schemes, most important one being **EURVOC descriptors**. The EUR-Lex dataset constitutes a very challenging Multi-label scenario due to the high number of possible labels (upto 4000).

1.2 Multilabel Vs. Multiclass

In Multiclass Classification there are more than two labels but each instance is associated with just one label. Example: Suppose we want to classify fishes into three categories viz. Salmon,Tuna and Mackerel.Each fish can either be Salmon, Tuna or Mackerel and thus implying this task as Multiclass classification.

In Multilabel classification, there are more than two labels and each instance is associated with one or more than one label. Example: suppose we want to give different genres to a song. Each song can belong to either one genre or multiple genres.

The documents in the Eurlex Dataset are allocated different labels or categories manually. Given a new document, the aim of project is to compare different machine learning approaches to automatically annotate the document with respective labels.

2 Dataset

The dataset is in the form of html files(approx 20k). Each HTML file contains a unique id, which is simply used to identify a file. The file also includes another heading named "Title and references" under which it describes some kind of agreements. Next heading is "text", under which there is some link which consists of information about contents in several languages. Then there is heading name "Dates" which tells about some dates and deadline about the documents. The next heading is "Classification" which contains a subheading name "EUROVOC descriptor" under which all the labels are defined and contains distinct subheading namely "Directory code", and "subject matter". The HTML file contains next heading as "Miscellaneous information" which emphasizes about Author and Form. Finally, some additional headings are "Relationship between documents" and "Text" which describes the Treaties and some new additional documents which are related to it. We are interested in the EUROVOC descriptor section for labels and the title and text section for the content.The procedure used to build the dataset in the form of a data frame from html files is as follows: The html files were parsed for getting the content and the labels (Eurovoc). During the parsing process,files that did not have text or did not contain Eurovoc labels were skipped.

3 Concept

3.1 EDA

This process is important because ,it gives us an intuition about the data. It helps to summarize the main characteristics of data and we can see what the data can tell us beforehand.

Number of Labels	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	17	18	24
Number of Documents	17	397	1427	3048	4950	7871	1042	459	230	91	18	8	8	3	4	2	1	1

Table 1

This table groups the documents with respect to how many label they contain.Inferring from the table,we can see that the mostly the document has 5 to 6 labels and the maximum number of labels a document contains is 24.

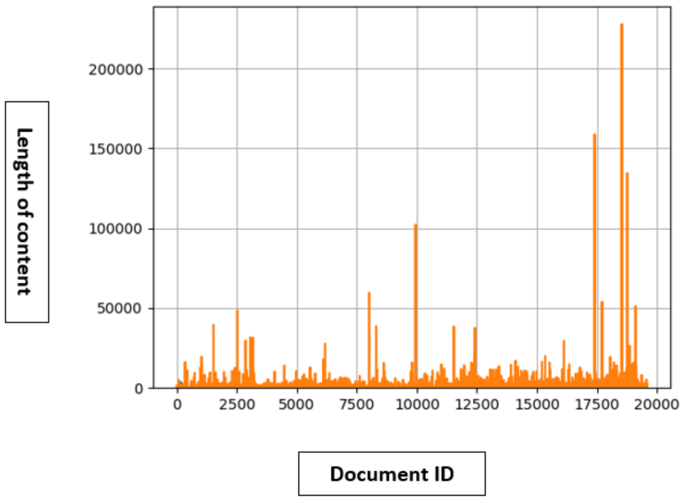


Figure 1

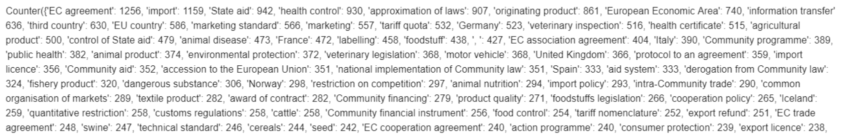
The graph above visualizes how many words the content of each document has. The maximum length of the content is above 2lacs, mostly the length of the document is between 0 to 15000.

Figure 2: Word Cloud

A word cloud was created and as per the Zipf's law, the words occurring maximum times were added to the conventional stop word list. After removing the stop words, porter stemming was done.

Label co-occurrence graph is created where two labels are connected with an edge only if both of then occur in the same content. For 3967 labels 65698 edges were created.

A list showing in how many documents a particular label is present is shown as a snippets below:



```
Counter({'EC agreement': 1256, 'import': 1159, 'State aid': 942, 'health control': 930, 'approximation of laws': 907, 'originating product': 861, 'European Economic Area': 740, 'information transfer'
636, 'third country': 630, 'EU country': 586, 'marketing standard': 566, 'marketing': 557, 'tariff quota': 532, 'Germany': 523, 'veterinary inspection': 516, 'health certificate': 515, 'agricultural
product': 500, 'control of State aid': 479, 'animal disease': 473, 'France': 472, 'labelling': 458, 'foodstuff': 438, ', ': 427, 'EC association agreement': 404, 'Italy': 390, 'Community programme': 389,
'public health': 382, 'animal product': 374, 'environmental protection': 372, 'veterinary legislation': 368, 'motor vehicle': 368, 'United Kingdom': 366, 'protocol to an agreement': 359, 'import
licence': 356, 'Community aid': 352, 'accession to the European Union': 351, 'national implementation of Community law': 351, 'Spain': 333, 'aid system': 333, 'derogation from Community law':
324, 'fishery product': 320, 'dangerous substance': 306, 'Norway': 298, 'restriction on competition': 297, 'animal nutrition': 294, 'import policy': 293, 'intra-Community trade': 290, 'common
organisation of markets': 289, 'textile product': 282, 'award of contract': 282, 'Community financing': 279, 'product quality': 271, 'foodstuffs legislation': 266, 'cooperation policy': 265, 'Iceland':
259, 'quantitative restriction': 258, 'customs regulations': 258, 'cattle': 258, 'Community financial instrument': 256, 'food control': 254, 'tariff nomenclature': 252, 'export refund': 251, 'EC trade
agreement': 248, 'swine': 247, 'technical standard': 246, 'cereals': 244, 'seed': 242, 'EC cooperation agreement': 240, 'action programme': 240, 'consumer protection': 239, 'export licence': 238,
```

Figure 3

3.2 Approaches

For solving the multilabel classification problem two types of approaches were used

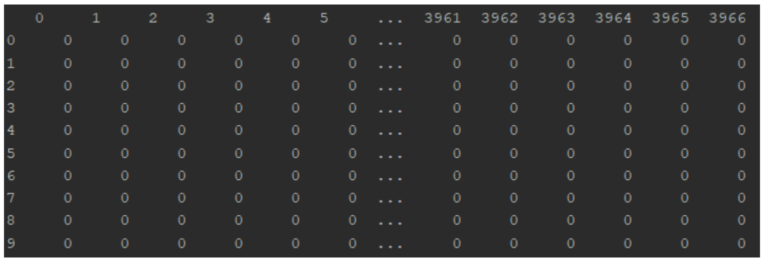
1. Adapted algorithm
2. Problem transformation

3.2.1 Preprocessing

Before building the model the first step was some preprocessing. Firstly, the content and labels were read from the preprocessed files content.csv and label.csv than multilabel binariser was used to transform the labels and a sparse matrix were created.

A gist of how the unique 3966 labels look after transformation is given in the figure below:

After that the data was split into train and test. To summarise the content in the form of numerical vectors Tf-Idf vectorizer was applied on the content.



```
   0  1  2  3  4  5  ...  3961  3962  3963  3964  3965  3966
0  0  0  0  0  0  0  ...     0     0     0     0     0     0
1  0  0  0  0  0  0  ...     0     0     0     0     0     0
2  0  0  0  0  0  0  ...     0     0     0     0     0     0
3  0  0  0  0  0  0  ...     0     0     0     0     0     0
4  0  0  0  0  0  0  ...     0     0     0     0     0     0
5  0  0  0  0  0  0  ...     0     0     0     0     0     0
6  0  0  0  0  0  0  ...     0     0     0     0     0     0
7  0  0  0  0  0  0  ...     0     0     0     0     0     0
8  0  0  0  0  0  0  ...     0     0     0     0     0     0
9  0  0  0  0  0  0  ...     0     0     0     0     0     0
```

Figure 4: mlb Output

In adapted algorithm approach the single label classification algorithms are used on Multilabel classification task without transforming the problem into different subsets of problems. Scikit-learn provided Mlknn which is based on the above mentioned explanation.

3.2.2 MLkNN

This is an approach which is derived from the traditional K-Nearest neighbor(KNN) algorithm. So, in this algorithm, for each unseen example, firstly its K nearest neighbors in training set are identified. After that, based on some statistical information obtained from the labels of these neighboring examples a maximum posterior principle(MAP) is employed to determine the label set for unseen examples.

Advantages: Simple Implementation.

Disadvantages: It does not exploit associations between the set of potential labels.

In problem transformation, the problem is first transformed into single label problem and then the classification algorithms are applied. This method is carried out in three different ways :

3.2.3 Binary relevance

It is an adaption of OVA(one-vs-all) to the multilabel scenario and transforms the multilabel dataset into several binary datasets. It independently trains one classifier per label. The trained classifier predicts either membership or non-membership of one class, then all the predicted class are associated and taken as the multilabel output. Suppose we have n target variable(Y1,Y2.....Yn), In binary relevance these target variables are treated independently and the problem is reduced to n classification problem.

Advantages:Simple,computationally efficient. It is slow if the label space is large.

Disadvantages: It ignores the possible correlation between the class labels.

3.2.4 Classifier chains

In this method, a chain of binary classifiers is constructed as C0, C1, . . . , Cn, where a classifier Ci uses the predictions of all the classifier Cj where $j < i$. Here, the first classifier is trained on the input data and then the next classifier is trained on input space. In this approaches the total number of classifiers needed is equal to the number of classes. Consider the target variable (Y1,Y2,.....,Yn) and features (F1,F2, ...Fm) in this approach first, the features F1,F2,......Fm are used to predict Y1. Next step (F1,F2......Fm,Y1) is used to predict Y2. To sum up at nth step (F1,F2......Fm,Y1,Y2,........,Yn-1) predicts Yn.

Advantages: This approach takes into account label correlation, and Generalizes to label combination not present in the training data unlike label powerset.

Disadvantages:One of the disadvantage of classifier chain is the error propagation down the chain, error propagation occurs when one (or more) of the first classifiers down the chain predicts poorly. In simple word, the quality of classifier is heavily dependent on ordering(Note there are n factorial possible ordering).

3.2.5 Label Powerset

In this approach, the problem is transformed from one multilabel dataset into the multiclass dataset by using the label set of each instance as a class identifier.

Advantages: This approach takes possible correlations between class labels into account.
It reduces the problem into one classifier unlike binary classifier.

Disadvantages:The drawback of this approach is the Computational infeasibility, i.e due to the increase in the number of classes the number of distinct label combination can grow exponentially.
All Possible label need to present in training set , so if an instance with completely new label is to be predicted this approach will give incorrect prediction.

Then random forest is used with all these three methods to carry out the classification.

3.2.6 Random Forest

Random forest is a kind of ensemble classifier which uses decision trees in a randomized fashion. This algorithm creates a forest with n number of trees.

Advantages:The overfitting problem will never come when we use the random forest algorithm in any classification problem.
It is Fast to train with test data.
It can handle large dataset with high dimensionality.

Disadvantages Slow in creating predictions once model is made.

3.3 Evaluation

Conventionally, In single label classification we use simple metric, such as accuracy, precision, recall,etc. Here,

$$Accuracy = \frac{number\ of\ correctly\ predicted\ instances}{total\ number\ of\ instances}$$

In multilabel classification,however a misclassification can not be considered entirely wrong. In other words, a prediction containing a subset of actual classes is more correct than the prediction that contains none of them. example: Consider a text document with labels a,b,c, suppose prediction 1 gives the labels text document as a,b and prediction 2 labels the text document as e. So here Prediction 1 is better than prediction 2. Measures for multilabel classification can be categorized in two types: First is the Label based measure where separate evaluation for each label is done during the evaluation process and second the example based measure is evaluated by taking the average differences of the actual and the predicted sets of labels over all the examples of the given dataset. In this project we are trying to explore both of them.

3.3.1 Hamming Loss: Example based measure

It is the fraction of the labels that are incorrectly predicted. It penalizes the individual labels which are wrongly predicted.It is loss function so lesser the value the better it is with the value ranging from 0-1.

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

Figure 5: Hamming Loss

3.3.2 Label Based Measures: Macro and Micro Averaging

In micro averaging metrics are calculated globally by summing up TPs, TNs, FPs and FNs for each class and then the average is taken.

- Microaveraging Precision $Prc^{micro}(D) = \dfrac{\sum_{c_i \in \mathcal{C}} TPs(c_i)}{\sum_{c_i \in \mathcal{C}} TPs(c_i) + FPs(c_i)}$

- Microaveraging Recall $Rcl^{micro}(D) = \dfrac{\sum_{c_i \in \mathcal{C}} TPs(c_i)}{\sum_{c_i \in \mathcal{C}} TPs(c_i) + FNs(c_i)}$

Figure 6: Micro Averaging

In macro averaging , the average of the precision and recall of the system on different sets is taken.

- Macroaveraging Precision $Prc^{macro}(D) = \dfrac{\sum_{c_i \in \mathcal{C}} Prc(D, c_i)}{|\mathcal{C}|}$

- Macroaveraging Recall $Rec^{macro}(D) = \dfrac{\sum_{c_i \in \mathcal{C}} Rcl(D, c_i)}{|\mathcal{C}|}$

Figure 7: Macro Averaging

The F1-Score is the weighted average of precision and recall.

$$F1 = \frac{2 * (Precision * Recall)}{(Precision + Recall)}$$

Jaccard similarity is the size of the intersection divided by the size of the union of two label sets. It is used to compare set of predicted labels for a sample to its corresponding set of true labels. In terms of TPs and FPs Jaccard is given by,

$$Jaccard(A, B) = \frac{TP}{TP + FP + FN}$$

In accordance with the above formula, if there are no true or predicted labels Jaccard is undefined.

3.3.3 Exact Matching score: Subset Accuracy

This score which is calculated using accuracy_score function in scikit gives the fraction of of samples that have all their labels classified correctly. This measure completely ignores the partially correct predictions so cannot be the most accurate measure for evaluating multilabel classification models.

$$ExactMatchRatio, MR = \frac{1}{n}\sum_{i=1}^{n} I(Y_i = Z_i)$$

Figure 8: Exact Matching Score

4 Implementation

4.1 Data Preprocessing

4.1.1 Libraries used

BeautifulSoup
nltk.corpus-stopwords
nltk.stem-PorterStemmer

4.1.2 Process

The HTML document is parsed using BeautifulSoup considering the title,content and the classifier(EUROVOC Descriptor). The most frequent words were added to the nltk-stopword list and the newly created stopword list was applied on both content and label, after that Porter stemming was done on both the content and labels. The idea here is to create two CSV files one with ID and content(content.csv) and other with Id and labels(label.csv).This was achieved by adding the data to both the files only if English text was present and the labels EUROVOC descriptor were present.

4.2 EDA

4.2.1 Libraries Used

Matplotlib
wordcloud
PIL
Collections
Pandas

4.3 Approaches

4.3.1 Libraries used

sklearn.modelselection
sklearn.preprocessing
sklearn.feature extraction
sklearn.matrix
skmultilearn.adapt
sklearn.ensemble
skmulitilearn.problemtransform.

4.3.2 Adapted Algorithm-MLkNN

To find the optimal number of k in MLkNN initially a grid search CV was done, however,it was not feasible. Therefore, MLkNN with different values of k was built the evaluation matrix is shown below:

K-Value	**F1 Score**	**Hamming Loss**	**Accuracy**	**Precision**	**Recall**	**Jaccard**
3	0.45(45.8%)	0.001(0.1%)	0.039(3.9%)	0.20(20%)	0.35(35%)	0.099(9.9%)
5	0.436(43.6%)	0.001(0.1%)	0.023(2.3%)	0.185(18.5%)	0.329(32.9%)	0.08(8%)
20	0.284(28.4%)	0.001(0.1%)	0.0035(0.35%)	0.091(9.1%)	0.17(17%)	0.028(2.8%)
134	0.191(19.1%)	0.001(0.1%)	0.0008(0.08%)	0.05(5%)	0.10(10%)	0.012(1.2%)

Table 2: Result for different case

Implying from the above stat we choose MLkNN with k=3, as our Adapted algorithm model.

4.3.3 Problem Transformation Method

A classifier with label powerset as transformation method and base classifier as random forest is built(LP-RF). Another classifier with binary relevance as transformation method and base classifier as random forest classifier is built(BR-RF) and the third model with Classifier chain as transformation method and random forest as base classifier is built(CC-RF).The classifier is made to know that we are dealing with a sparse matrix by giving the parameter require dense.

5 Evaluation

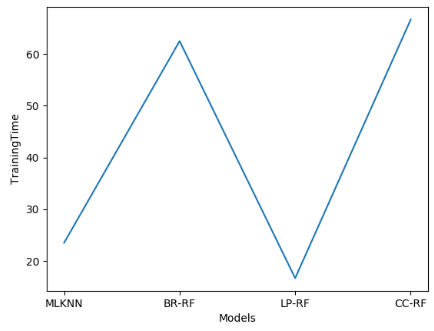


Figure 9: Comparison factor: Time in minutes

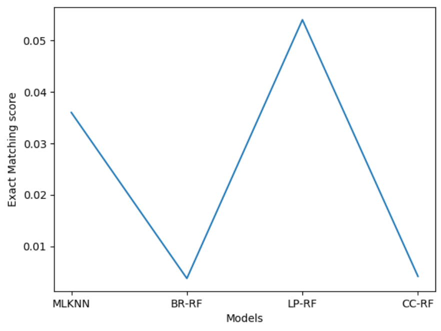


Figure 10: Comparison factor:Exact Matching Score

The models were compared with respect to time taken for them to be trained. It can be seen that LP-RF takes minimum time, followed by MLkNN so, it can be said that LP-RF and MLkNN are faster compared to other two models.

The exact matching score of LP-RF is 0.054 and MLkNN is 0.036. BR-RF model gives the lowest exact matching score(0.0038).

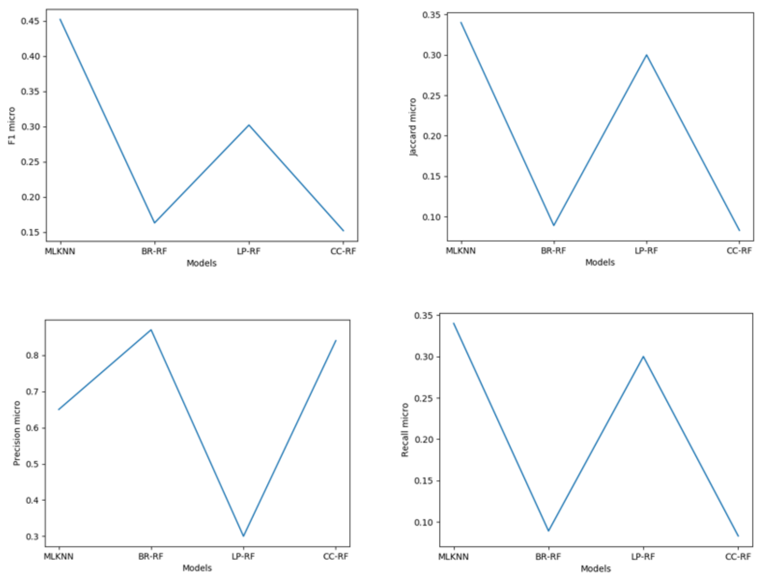


Figure 11: Comparison factor: Micro

The Micro averaging comparison factor shows that even though the precision Micro for LP-RF is minimum but the recall micro compensates this and F1 micro score for LP-RF is better than the F1 micro score BR-RF and CC-RF. On similar grounds MLkNN's F1 micro score is the highest. Jaccard micro is highest for MLkNN followed by LP-RF.

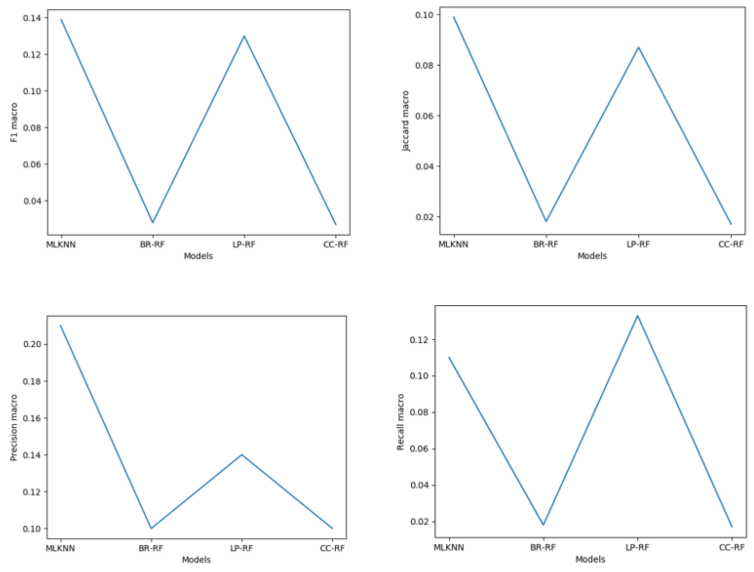


Figure 12: Comparison factor: Macro

The Jaccard and F1-Macro score is highest for MLkNN followed by LP-RF. The Precision macro for LP-RF is less than the precision macro of MLkNN but the recall macro of LP-Rf is gretaer than the recall macro of MLkNN, which gives a very little difference between their F1 micro score.

6 Conclusion

Although the F1 score, Jaccard score for MLkNN is greater than LP-RF, the time consumed by LP-RF is less than mlknn. Also the exact matching ratio for LP-RF is the highest.So, LP-RF and MLkNN both prove to be good models for Eurlex multilabel classification.

BR-RF does not take into consideration the correlation between the labels which can be one of the reason for its poor performance.The future scope would be to prove this point that the labels are correlated by some word association.

CC-RF highly depends on the ordering of the labels. In this project CC-RF was built using the default ordering and no classifier chains by taking random columns from the label matrix. The future scope would be to built different classifier chains and take the best one so as to improve its performance.

It would be interesting to compare other adapted algorithms like MLTSVM to the implemented MLkNN and also to overcome the disadvantages of MLkNN using stack MLkNN which exploits the label associations.