Journal of Computer Sciences and Applications, 2024, Vol. 12, No. 1, 17-24
Available online at http://pubs.sciepub.com/jcsa/12/1/3
Published by Science and Education Publishing
DOI:10.12691/jcsa-12-1-3

Facing the Clinical Trial Annotation Problem on Breast Cancer: Natural Language Processing & Machine Learning Models Selection

Pablo Eliseo Reynoso-Aguirre[1,*], Pedro Flores-Pérez[2]

[1]Computer Science Department, Universitat Politècnica de Catalunya UPC, Barcelona, Spain
[2]Mathematics Department, University of Sonora, Hermosillo, México
*Corresponding author: pablo.eliseo.reynoso@est.fib.upc.edu, pablo.reynoso9@gmail.com

Received July 15, 2024; Revised August 18, 2024; Accepted August 25, 2024

Abstract: Clinical trial classification problem (CTCP) is one of the cutting-edge real-life applications in biomedical informatics, especially in the domain considered in this paper, namely breast cancer. The task consists in the development of models able to discriminate patient's eligibility profile at breast cancer trials based on performance status (PS) labels. The task has gained relevance at medical research and practice in the framework of decision support systems. Besides, the task has been considered a meaningful instrument for an accurate selection of participants at experimentations resulting in no health-behavioral drug side effects on participants.

Keywords: *ECOG, KPS, performance status, eligibility criteria, clinical trial, classification, multinomial linear regression, multinomial naive bayes, multilayer perceptron, support vector machines*

Cite This Article: Pablo Eliseo Reynoso-Aguirre, and Pedro Flores-Pérez, "Facing the Clinical Trial Annotation Problem on Breast Cancer: Natural Language Processing & Machine Learning Models Selection." *Journal of Computer Sciences and Applications*, vol. 12, no. 1 (2024): 17-24. doi: 10.12691/jcsa-12-1-3.

1. Introduction

Now a day, biomedical informatics has gained a high importance through real life applications and academic research (see [1] for a clear overview of such applications). Focusing on *clinical trial (CT)* of breast cancer, the *National Institute of Health (NIH)* lists biomedical applications related to breast cancer clinical trials [2]. Each study's protocol in *CT [3]* has guidelines for who can or cannot participate in the study. These guide- lines, called *eligibility criteria (EC)*, describe characteristics that must be shared by all participants. They may include age, gender, medical history, and current health status. *EC* for treatment studies often seek a particular type and stage of cancer in patients. Facing the problems involved in these applications consider the following implications:

- The complex and non-unified genres variety in which biomedical information is represented, including: *electronic heath reports (EHR)*, medical publications, medical blogs and social media, drug leaflets, *clinical trial reports (CTR)*, textual in- formation included in ontologies [4,5,6] and knowledge bases. Author's profiles include medical doctors, nurses, radiologists, pharmacologists, scientists, and lay people.
- The difficulty of extracting, normalizing, and classifying medical entities as drugs, diseases, medical findings, anatomical elements, and other medical-related linguistic patterns (doses, formulas, quantities, units), etc.

All these implications are presented in *CT*, documents written for human use. These files frequently contain un- precise information un-useful for *medical decision support (MDS)* [7]. As part of them, *EC* present the same issues.

An important task, related to *MDS CT*, is the automatic computing of score status of a patient given the textual context on *EC* content from CTs. PS, a metric to evaluate prospective patient stage of cancer is contained in most of the EC text in trials. Details related to the *CT Annotation Problem [8]* are described in the following subsections:

1.1. Performance Status Scales & Ranges

According to the literature [9], *PS* has been considered as the standardized metric to Asses *EC* in terms of clinical trial *EC* attribute. *PS* eases tracking of patient's treatment evolution and unifies researching analysis through different institutions and countries. *Different scales can represent PS*: *Eastern Oncology Group (ECOG)* [10], *Karnofsky performance status (KPS)* [11], and *Lansky performance status (LPS)* (a particular case of *KPS* for oncological children's studies). These scales describe the stage of cancer of a patient based on their daily physical-behavioral signs. *PS* scales equivalences [9] and descriptions are found in Table 1.

Table 1. ECOG-KPS scale equivalences and patients profile description

ECOG scoring	KPS/LPS scoring	Patients Profile Description
0	100	Normal, no evidence of disease.
	90	Normal activity, minor symptoms.
1	80	Effort required, some symptoms.
	70	Unable to perform active work.
2	60	Occasional assistance required.
	50	Considerable assistance required.
3	40	Disable, special care required.
	30	Severely disable, hospitalization indicated.
4	20	Very ill, hospitalization required.
	10	Moribund.
5	0	Dead.

1.2. Data Source

Experimental data for model's inference learning/evaluation was obtained from *Clinical Trials Gov. (USA)* [12]. Dataset origin sources, descriptors related to CT XML files are describedby Tables 2 & 3 respectively.

Table 2. CT studies distribution of cancer/breast cancer among U.S. and Non-U.S. countries

Country	Cancer	Breast Cancer
U.S. only	31,268	4,316
Non-U.S. only	29,544	3,791
Total	60,812	8,107

Table 3. KPS & ECOG data samples/features distribution

Performance Status	Samples	Features
KPS	3,767	15,296
ECOG	4,023	15,296

1.3. KPS Class Distribution

For class distribution, KPS scale is featured as main scoring in the problem overviewed based on the scale intrinsic granularity in comparison to ECOG scale. Equivalence among scales was done according to PS scales/ranges (see Table 1). Initial distributional analysis of the labels resulted in a class high imbalance for both KPS min, max values. Based on complexity of data and prediction goals, solving approach splits annotation in 2 learning tasks, predicting CT KPS min, max in a separate way.

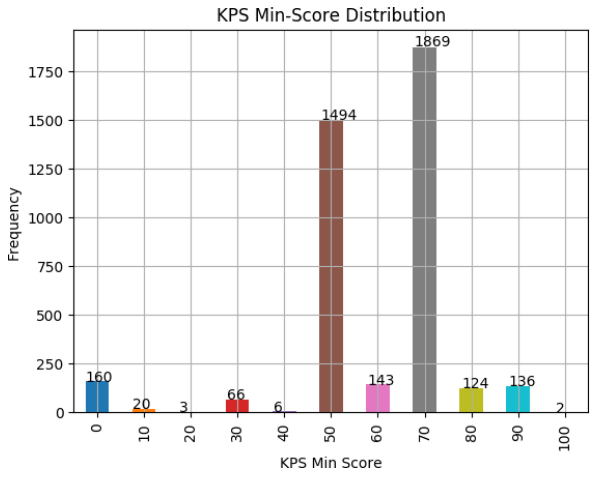

Figure 1. Class Distribution of Min KPS from CTEC KPS Ranges

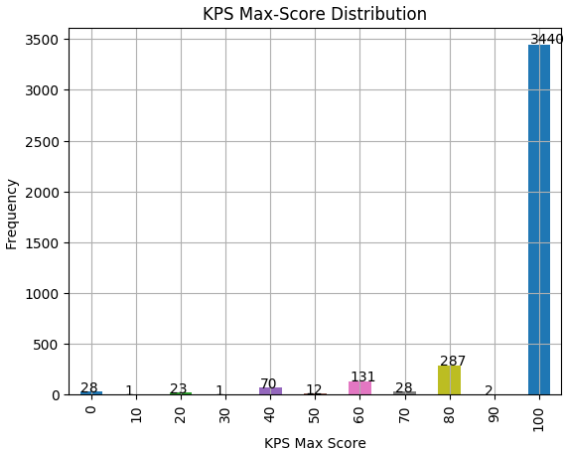

Figure 2. Class Distribution of Max KPS from CTEC KPS Ranges

2. Related Work

The work in [13] considers *KPS* as *EC* classification scale approach at *CT* breast cancer patients profile annotation. Study proposes an algorithm which uses a minimum of two and a maximum of three questions to facilitate an adequate and efficient evaluation of the CT KPS score. According to the authors, the system obtained an average good performance for this type of application. However, their CT classifier suffers from synonymy, polysemy and fuzziness by its framework constrained text nature. Besides, their framework is not capable to classify CT in a range of PS scores as CT real PS scores; it is constrained to single value classification. Finally, their prototype has built in a static learning arquitechture not induced by data. The proposed approach in this work faces most of the drawbacks described in this section.

According to [14], CTR text mining study. This work considers processing CT texts with *NCBO* annotator. In [15] *ExaCT*, researches comprise user assistance locating and extracting key trial characteristics (e.g., *EC*, sample size, drug dosage, primary outcomes) from full-text journal articles reporting on randomized controlled trials are presented. In [16], research faces the problem of extracting *EC*. Since *EC* are represented as free text, their automatic interpretation and the evaluation of patient eligibility is challenging. Processing approach is based on the identification of contextual patterns and semantic concepts that together define the machine-interpretable meaning. In [17], study presents a system working on cancer vaccines *CT*, enabling rapid extraction of information about institutions, diseases, clinical approaches, clinical trials dates to obtain predominant cancer types in the trials, clinical opportunities and pharmaceutical market coverage.

There is a former work on the *Clinical Trial Classification Problem (CTCP)* task overviewed in this article considering the same CT data samples [18]. The initial solving approach considered a multivariate regression modeling to forecast min & max PS scores of a given CT. The aim consists in finding a useful correlation among KPS scores and the CT eligibility criteria clinical terms. The experiment considered the following generic tasks [19,20,21]:

- PS Scoring Extraction (Regular Expressions)
- Data Cleansing (XML Tag Removal, Tokenization)

- Text Normalization (Stopwords Removal, Lemmatization)
- Text Vectorization (Term Frequency - Inverse Document Frequency)
- Features Projections (Single Value Decomposition)
- Extra Considerations (Problematic Samples Removal, Prediction Refining, ECOG PS Predictions),
- Linear / Non-linear Models Tuning (Partial Least Squares, Multilayer PerceptronRegressor)

The final reported results considering 4024 highly imbalanced-labeled CTs from 8107 breast cancer CT (*clinicaltrials.gov - NIH US National Library of Medicine*):

Table 4. Model classification performance comparison among PLS/MLP in terms of 1 R^2 and *MSE* scores using *10-Fold Cross Validation* learning framework

Algorithm	$1-R^2$	$1-R^2$	MSE $_{min}$	MSE $_{max}$
PLS	0.8834	0.9495	140.1356	44.9453
MLP	0.8951	0.9688	141.9824	45.8599

Findings on Table 4, suggested both PLS, MLP models, achieved weak classification performance in terms of their R^2 values [22] for *min* [0.1116, 0.1049] and *max* [0.0312, 0.0505] since typical scores considered for pure science fields required the following condition R^2[0.5, 0.75] for pure science according to [23]. Experimentation results denote learning performance high dependency with data representations e.g., complex clinical terms combinations as bigrams, trigrams and a tendency of better generalization on linear models than non-linear approaches.

3. Solving Approach

The solving approach in this work for the problem proposed at [8] considers a multi classification inference on data with the following confusion matrix and statistical metrics related:

Table 5. Confusion Matrix for Classification Tasks

		Predictions	
		class-A	**class-B**
Truths	**class-A**	True Positive (TP)	False Negative (FN)
	class-B	False Positive (FP)	True Negative (TN)

On classification algorithms considerations, since number of samples in data (*#samples* < 50, 000) the ideal approach to avoid over fitting and computational effort on inference on this work are *classical machine learning* models e.g.: Multinomial Linear Regression (MLR), Multinomial Naive Bayes (MNB), Multilayer Perceptron (MLP) [24], Support Vector Machines (SVM). Deep learning models are not considered in the classification task.

Table 6. Statistical Metrics for Classification Tasks based on Confusion Matrix

Metric	Mathematical Expression
Precision	$PPV = \frac{TP}{TP+FP}$
Recall	$TPR = \frac{TP}{TP+FN}$
Specificity	$TNR = \frac{TN}{TN+FP}$
Accuracy	$ACC = \frac{TP+TN}{TP+FN+TN+FP}$
$F1-Score$	$F1_{score} = 2 * \frac{PPV*TPR}{PPV+TPR}$
$Youden-J-Stat$	$YJS = TPR + TNR - 1$

Algorithm 1 Breast Cancer Trials Annotation Framework

```
procedure FINDBESTANNOTIATIONMODEL (CT XML Data)
    EC Texts, Labels (min/max) ← XML Pre Processing for each CT EC
    if (EC Texts, Labels) are imbalanced then
        return EC Texts, Labels (min/max) ← Random Oversampling(EC Texts, Labels)
    EC Texts ← Tokenization for each EC Texts
    if NoStopWords.performance > StopWords.performance then
        return EC Texts ← NoStopWords(EC Texts)
    if Stemming.performance > Lemmatization.performance then
        return EC Texts ← Stemming(EC Texts)
    else
        return EC Texts ← Lemmatization(EC Texts)
    if CountVectorizer.performance > TFIDFVectorizer.performance then
        return BestFeatureVectorizer ← CountVectorizer
    else
        return BestFeatureVectorizer ← TFIDFVectorizer
    BestVectorizer ← SelectBestNgramVectorizer from BestFeatureVectorizer, monogram, bigram, trigram
    X ← BestVectorizer(EC Texts)
    BestFeaturesSVD ← SelectBestSVDFeatures from TruncatedSVD, X
    if BestFeaturesSVD.performance > BestVectorizer.performance then
        return X ← BestFeaturesSVD(X)
    MLR_min ← RandomizedSearchCV(MLR, Settings, X, Labels_min)
    MNB_min ← RandomizedSearchCV(MNB, Settings, X, Labels_min)
    MLP_min ← RandomizedSearchCV(MLP, Settings, X, Labels_min)
    SVM_min ← RandomizedSearchCV(SVM, Settings, X, Labels_min)
    BestMinModel ← SelectBestTrialMinModel(MLR_min, MNB_min, MLP_min, SVM_min)
    MLR_max ← RandomizedSearchCV(MLR, Settings, X, Labels_max)
    MNB_max ← RandomizedSearchCV(MNB, Settings, X, Labels_max)
    MLP_max ← RandomizedSearchCV(MLP, Settings, X, Labels_max)
    SVM_max ← RandomizedSearchCV(SVM, Settings, X, Labels_max)
    BestMinModel ← SelectBestTrialMaxModel(MLR_max, MNB_max, MLP_max, SVM_max)
    return BestMinModel , BestMinModel
```

4. Experiments

In this section is described the experimentation done at the different stages of the inference analysis. To begin with, a single multi label classification algorithm (MNB) is implemented in the initial experimental stages, to continue evaluating learning correlations among clinical textual features and KPS labels. All experimental stages [25,26] consider a *5-K Fold Cross Validation*. Different multi classification algorithms as MLR, MLP, SVM are implemented in the advanced stages in order to selected the best possible approach in order to maximize the True Positive (TP), True Negative (TN) for every label of KPS range considered in the trials.

4.1. Class Distribution Balancing

As it was seen in Figures 1 & 2, both KPS range limits of CT have a high imbalance distribution, particularly on *max* variable; therefore, we proceed using a sampling approach. Since number of samples in minority classes is very low, oversampling seems to be an appropriate framework to tackle the problem. For this task, implementation consider the most generic technique, RandomOverSampler, oversampling minority classes occurrences up to number of occurrences of majority class (with replacement, without adding noise to the samples

copies) for each classification takes KPS_{min} & KPS_{max}. In this stage only MNB is considered to observe the classification outcome based on statistic descriptors.

Table 7. Imbalance vs. RandomOverSampler KPS_{min} classification comparisons on MNB model, TF-IDF (1,1)

Sampling	*Precision*	*Recall*	*Accuracy*	*F1 − Score*	*Youden − J*
Imbalance	0.440	0.475	0.475	0.324	−0.812
RandomOverSampler	0.294	0.236	0.236	0.239	-0.527

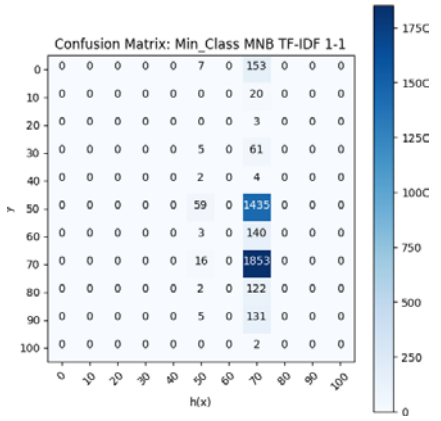

Figure 3. MNB TF-IDF (1,1) on Imbalance KPS_{min}

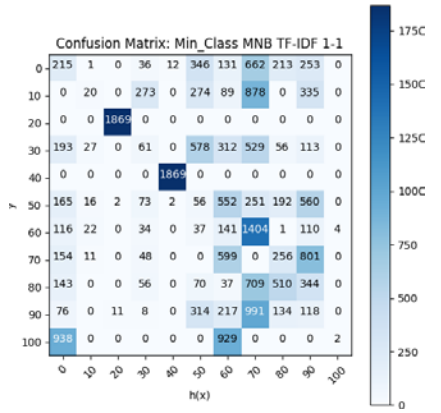

Figure 4. MNB TF-IDF (1,1) on RandomOverSampler KPS_{min}

As it can be seen in Table 7 & Table 8, no sampling on imbalance data may seem to achieve better classification results. However, *Youden J* statistic reflects how generalization of the models differentiate among the different *n classes* predicted for either KPS_{min}, KPS_{max}. The higher the *Youden J* statistic value, the better generalization we obtain in the model to predict all the different classes of KPS range. Therefore, RandomOverSampler simple assumption to sample up the number of samples in majority class seem to heal the imbalance problem, and helps the model inference to escape from over fitting the majority class in variable distribution.

Table 8. Imbalance vs. RandomOverSampler KPS_{max} classification comparisons on MNB model, TF-IDF(1,1)

Sampling	*Precision*	*Recall*	*Accuracy*	*F1 − Score*	*Youden − J*
Imbalance	0.731	0.855	0.855	0.788	−0.818
RandomOverSampler	0.336	0.266	0.266	0.241	-0.466

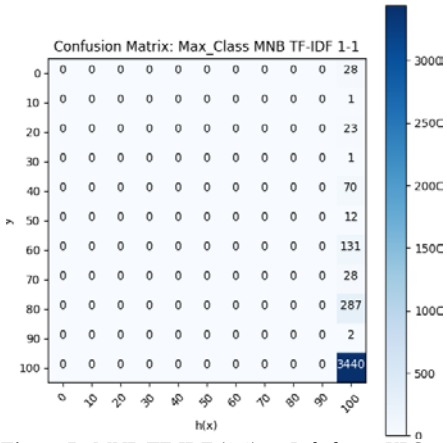

Figure 5. MNB TF-IDF (1,1) on Imbalance KPS_{max}

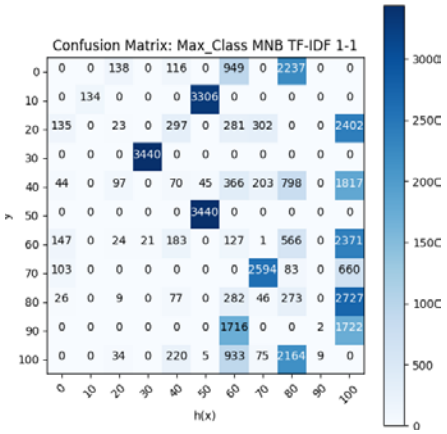

Figure 6. MNB TF-IDF (1,1) on RandomOverSampler KPS_{max}

4.2. Feature Extraction (Weighting)

An important part of model's inference is the extraction of features for training the models. Moreover, tasks that involve natural language text require a text embedding (*Word2Vec*, *Sentence2Vec*, and *Doc2Vec*) as numerical matrixes to be a valid input data for *Classical Machine Learning* algorithms. Deep learning approaches consider methods as Keras Embedding's and Bert that automatically calculate text embedding's using initial weights on the input layer of the networks. In this experimentation phase we consider a *Doc2Vec* type of embedding to find relationships among CT XML documents based on documentwords similarities.

Doc2Vec can be implemented by different ways of weighting: CountVectorizer, TFIDF Vectorizer. CountVectorizer, recalls for frequency of word in a given document from CORPUS, while TfidfVectorizer, considers a special weight combining frequency of word in a given document times a normalizing factor of how common the term is for all documents in CORPUS overall. *Doc2Vec* columns, Bag of Words (BOW) are represented by universe of words in all documents of CORPUS. Bag of Words (BOW) may contain mono-grams, bi-grams, tri-grams, n-grams or combination of them if needed. *Doc2Vec* rows represent the documents in the CORPUS. After *Doc2Vec* textual model is implemented on CT documents, the numerical representation of text is known as *document term matrix (DTM)* (see Figure 7). In this

stage only MNB is considered to observe the classification outcome based on statistic descriptors.

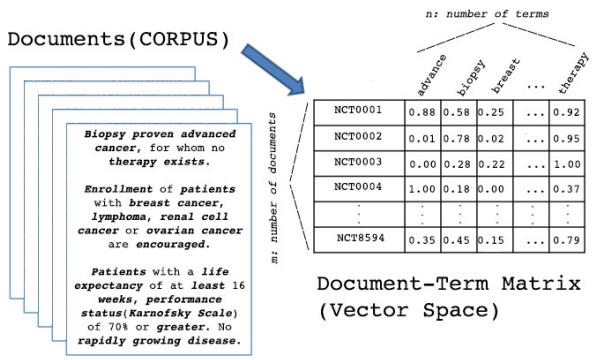

Figure 7. CT CORPUS representation as Document Term Matrix using TfidfVectorizer

Table 9. TF-IDF vs. Count rep. (1,1) for *KPS* $_{min}$ Oversampling classification comparisons on MNB model

Weighting	*Precision*	*Recall*	*Accuracy*	*F1 − Score*	*Youden − J*
TF − IDF	0.294	0.236	0.236	0.239	−0.527
Count	0.550	0.402	0.402	0.370	−0.194

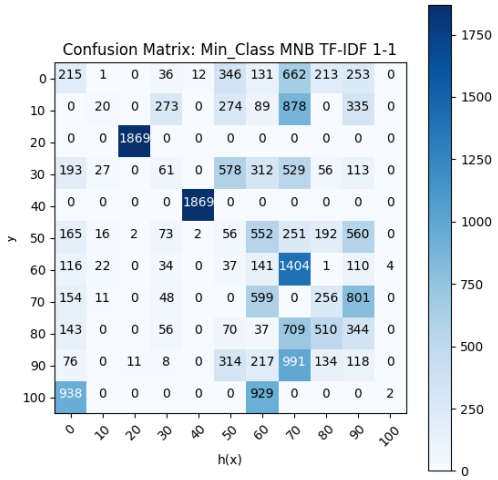

Figure 8. MNB TF-IDF (1,1) on RandomOverSampler-KPS_{min}

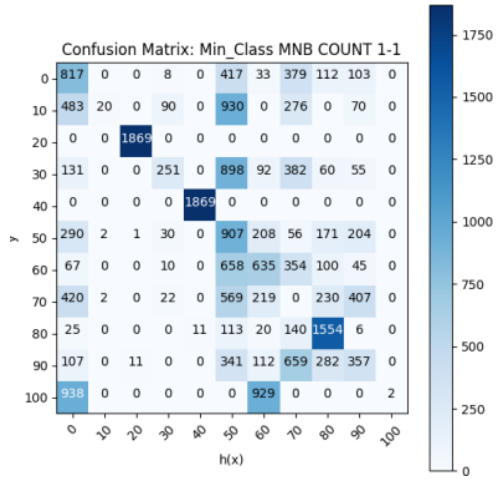

Figure 9. MNB Count (1,1) on RandomOverSampler- KPS_{max}

Table 10. TF-IDF vs. Count rep. (1,1) for KPS_{max} Oversampling classification comparisons on MNB model

Weighting	*Precision*	*Recall*	*Accuracy*	*F1 − Score*	*Youden − J*
TF − IDF	0.336	0.266	0.266	0.241	−0.466
Count	0.519	0.435	0.435	0.425	−0.129

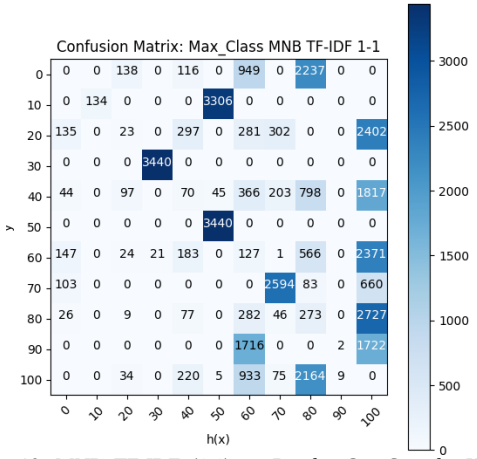

Figure 10. MNB TF-IDF (1,1) on RandomOverSampler KPS_{max}

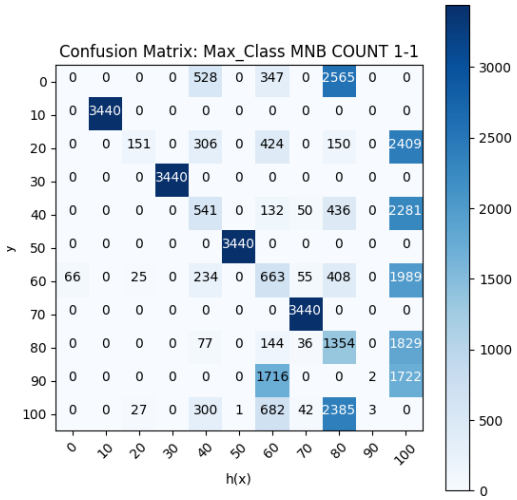

Figure 11. MNB Count (1,1) on RandomOverSampler KPS_{max}

4.3. Trial Text Preprocessing

After experimenting on feature extraction weighting to finding a useful numerical representation of features, a text preprocessing [27] stage is considered to boost the feature extraction approaching different techniques of text normalization. All approaches involve Tokenization, and subsequent NLP preprocessing methods as: Stopwords Removal, Stemming Algorithm, and English Lemmatization Processes. In this stage only MNB is considered to observe the classification outcome based on statistic descriptors on Count Feature Extraction for both *KPS* $_{min}$, and *KPS* $_{max}$.

Table 11. Text Pre Processing Comparisons for Count ngram (1,1) representation on MNB *KPS* $_{min}$ model

Preprocessing	*Precision*	*Recall*	*Accuracy*	*F1 − Score*	*Youden − J*
None	0.5506	0.4027	0.4027	0.3702	−0.1944
Stemming	0.5387	0.3983	0.3983	0.3654	−0.2032
Lemmatization	0.5506	0.4027	0.4027	0.3702	−0.1944
Stop Words	0.5515	0.4042	0.4042	0.3731	−0.1914
Stop Words/ Stemming	0.5265	0.3953	0.3953	0.3628	−0.2092
Stop Words/Lemmatization	0.5515	0.4042	0.4042	0.3731	−0.1914

Table 12. Text Pre Processing Comparisons for Count ngram (1,1) representation on MNB *KPS* $_{max}$ model

Preprocessing	*Precision*	*Recall*	*Accuracy*	*F1 − Score*	*Youden −J*
None	0.5193	0.4352	0.4352	0.4259	−0.1294
Stemming	0.5322	0.4392	0.4392	0.4329	−0.1215
Lemmatization	0.5193	0.4352	0.4352	0.4259	−0.1294
Stop Words	0.5225	0.4369	0.4369	0.4265	−0.1261
Stop Words/ Stemming	0.5358	0.4394	0.4394	0.4334	−0.1211
Stop Words/Lemmatization	0.5225	0.4369	0.4369	0.4265	−0.1261

Different types of preprocessing approaches for normalizing text before extracting numerical features on Multinomial Naive Bayes (MNB) suggested that for that specific algorithm text normalizations result in better classification results by performing Stopwords Removal and Stemming Chunking to posterior extract features by a Count Weighting. The experimentation was extended to implementing all different text normalization approaches, both Count & *TF IDF* feature weighting, and different ngram combinations for other ML supervised learning algorithms (default parameters) as: Multinomial Logistic Regression (MLR), Support Vector Machines (SVM) and Multilayer Perceptron Neural Network (MLP). The extended experimental results are shown in Tables 13, 14.

Table 13. Text Pre Processing Comparisons for Count ngram representations on ML *KPS* $_{min}$ models

Algorithm	*PP*	*Ngram*	*Precision*	*Recall*	*Accuracy*	*F1 − Score*	*Youden − J*
MNB	*SW*	(1,3)	0.6006	0.4116	0.4116	0.3843	−0.1767
MLR	*SW/Stem*	(1,3)	0.8174	0.8877	0.8877	0.8493	0.7755
SVM	*SW*	(1,3)	0.8240	0.8923	0.8923	0.8547	0.7846
MLP	*Lemma*	(1,3)	0.8553	0.9048	0.9048	0.8736	0.8097

Table 14. Text Pre Processing Comparisons for Count ngram representations on ML *KPS* $_{max}$ models

Algorithm	*PP*	*Ngram*	*Precision*	*Recall*	*Accuracy*	*F1 − Score*	*Youden − J*
MNB	*SW/Stem*	(1,3)	0.5586	0.4408	0.4408	0.4400	−0.1183
MLR	*SW/Stem*	(1,3)	0.7163	0.8181	0.8181	0.7541	0.6363
SVM	*SW/Stem*	(1,2)	0.7180	0.8181	0.8181	0.7546	0.6363
MLP	*SW*	(1,3)	0.7480	0.8181	0.8181	0.7685	0.6363

4.4. Feature Selection (Mapping)

In terms of data projections, SVD algorithm [28] was considered to project existing numerical features to a more separable space (latent variables) [29,30]. SVD have been proved as good for experiments as this one with sparse data (has non-numerical nature - text vectorization) and curse of dimensionality issues e.g. (*# f features > #samples*). In this experiment feature mapping consider different numbers of meta-features for data projections [100, 150, 200, 250, 300] on the results obtained for every text preprocessing, feature weighting, ngram settings and algorithms on previous stage. After analyzing classification results in which ngram features are projected into a more compact dimensional space, we obtained the following results for the best SVD configuration on every algorithm prediction for *KPS* $_{min}$, *KPS* $_{max}$:

Table 15. SVD Feature Selection Comparisons for Count ngram (1,3) representations on ML *KPS* $_{min}$ models

Algorithm	*SVDFeats*	*Precision*	*Recall*	*Accuracy*	*F1 − Score*	*Youden − J*
MNB	*N/A*	0.6006	0.4116	0.4116	0.3843	−0.1767
MLR	300	0.7609	0.7996	0.7996	0.7784	0.5992
SVM	300	0.7156	0.6726	0.6726	0.6773	0.3453
MLP	250	0.8251	0.8910	0.8910	0.8546	0.7821

Table 16. SVD Feature Selection Comparisons for Count ngram (1,3) representations on ML *KPS* $_{max}$ models

Algorithm	*SVDFeats*	*Precision*	*Recall*	*Accuracy*	*F1 − Score*	*Youden − J*
MNB	*N/A*	0.5586	0.4408	0.4408	0.4400	−0.1183
MLR	300	0.6944	0.7331	0.7331	0.7086	0.4662
SVM	300	0.6536	0.5833	0.5833	0.6035	0.1666
MLP	200	0.7318	0.8181	0.8181	0.7610	0.6363

After comparing the results of *Trial Text Preprocessing* from Tables 13 & 14 with the results of *Feature Selection (Mapping)* of Tables 15 & 16 respectively we observed that statistical metrics did not improve, therefore the SVD feature projections does not seem to be an efficient approach to boost classification performance metrics (Accuracy, F1- Score, Youden J Statistic) scoring.

4.5. Models Tuning

In the following section experimentation related to model hyper parameters tuning we consider a RandomizedSearchCV approach from a model selection framework to explore different combinations of parameters values in order to find settings that optimize classification performance metrics from former stages.

- On Multinomial Naive Bayes, hyper-parameters and settings considered for testing are: alpha *in* [0.005, 5.000], class prior = none, fit prior = True.
- On Multinomial Logistic Regression, hyper-parameters and settings considered for testing are: penalty = L2, C *in* [0.005, 6.000], tol *in* [0.0001, 0.2000], dual = False, solver *in* [lbfgs, sag, saga], multi class *in* [ovr, multinomial, auto].
- On Support Vector Machines, hyper-parameters and settings considered for testing are: penalty = L2, C *in* [0.005, 10.000], tol *in* [0.0001, 0.2000], dual *in* [True, False], max iter *in* [1, 10], multi class = ovr, random state = 0, loss = squared hinge
- On Multilayer Perceptron, hyper-parameters and settings considered for testing are: activation *in* [identity, logistic, tanh, relu], hidden layer sizes *in* [(5, 1), (10, 1), (15, 1), (20, 1), (25, 1), (50, 1), (100, 1), (200, 1)], solver = lbfgs, max iter *in* [10, 25, 50, 100].

After trying different combinations of model hyper-parameters along generic (up-to-majority) class oversampling, different text preprocessing, different feature extraction, different n-gram representations and

different feature selection (mapping), the following results were obtained:

Table 17. Hyper-parameters Algorithms Tuning Comparisons for Countngram (1,3) representations on ML *KPS$_{min}$* models

Algorithm	PP	Ngram	Precision	Recall	Accuracy	F1 – Score	Youden – J
MNB	SW	(1, 3)	0.8572	0.9070	0.9070	0.8754	0.8141
MLR	SW/Stem	(1, 3)	0.8179	0.8889	0.8889	0.8501	0.7778
SVM	SW	(1, 3)	0.8242	0.8930	0.8930	0.8552	0.7861
MLP	Lemma	(1, 3)	0.8553	0.9048	0.9048	0.8736	0.8097

Table 18. Hyper-parameters Algorithms Tuning Comparisons for Count ngram (1,3) representations on ML *KPS$_{max}$* models

Algorithm	*PP*	*Ngram*	*Precision*	*Recall*	*Accuracy*	*F1 – Score*	*Youden – J*
MNB	*SW/Stem*	(1,3)	0.7463	0.8181	0.8181	0.7678	0.6363
MLR	*SW/Stem*	(1,3)	0.7174	0.8181	0.8181	0.7546	0.6363
SVM	*SW/Stem*	(1,2)	0.7180	0.8181	0.8181	0.7546	0.6363
MLP	*SW*	(1,3)	0.7480	0.8181	0.8181	0.7685	0.6363

4.6. Additional Considerations: Sampling Tuning

After all the experimentation performed in previous stages to find relevant results, we performed additional considerations to maximize the accuracy results and learning generalization by adjusting sampling framework on both *KPS $_{min}$* & *KPS $_{max}$*. The class distribution balancing considered oversampling of minority classes on difference percentages ranges [5%-25%] in relation with majority class [31]. This implementation only considered monograms (1,1) features in order to avoid *The Curse of Dimensionality Problem*, since sampling tuning considered between 4-6 times less samples than sampling strategy on *Section 4.1:*

Table 19. Sampling Tuning Comparisons for Count ngram (1,1) representations on ML *KPS $_{min}$* models

Algorithm	*PP*	*Ngram*	*Precision*	*Recall*	*Accuracy*	*F1 – Score*	*Youden – J*
MNB	*SW*	(1,1)	0.7421	0.7461	0.7461	0.7436	0.7550
MLR	*SW/Stem*	(1,1)	0.6945	0.7064	0.7064	0.6946	0.7261
SVM	*SW*	(1,1)	0.7236	0.7389	0.7389	0.7259	0.7648
MLP	*Lemma*	(1,1)	0.7481	0.7610	0.7610	0.7501	0.8027

Table 20. Sampling Tuning Comparisons for Count ngram (1,1) representations on ML *KPS $_{max}$* models

Algorithm	*PP*	*Ngram*	*Precision*	*Recall*	*Accuracy*	*F1 – Score*	*Youden – J*
MNB	*SW/Stem*	(1,1)	0.9299	0.9215	0.9215	0.9232	0.9114
MLR	*SW/Stem*	(1,1)	0.9172	0.8930	0.8930	0.8950	0.9312
SVM	*SW/Stem*	(1,1)	0.9260	0.8966	0.8966	0.8949	0.9341
MLP	*SW*	(1,1)	0.9478	0.9374	0.9374	0.9387	0.9525

In the results found we can observe an improvement on *KPS $_{max}$* classification performance metrics. However, *KPS $_{min}$* seem to generalize better on n-gram [(1,2), (1,3)] features data representations than monogram representations.

5. Results

After all the experimentation performed in previous stages to find relevant results, the best generalization found for the models (MNB, MLR, SVM, MLP) on *KPS $_{min}$* & *KPS $_{max}$* annotation tasks are:

Table 21. Final Performance Comparisons for Count ngram (1,3) representations on ML *KPS $_{min}$* models

Algorithm	*PP*	*Ngram*	*Precision*	*Recall*	*Accuracy*	*F1 – Score*	*Youden – J*
MNB	*SW*	(1,3)	0.8572	0.9070	0.9070	0.8754	0.8141
MLR	*SW/Stem*	(1,3)	0.8179	0.8889	0.8889	0.8501	0.7778
SVM	*SW*	(1,3)	0.8242	0.8930	0.8930	0.8552	0.7861
MLP	*Lemma*	(1,3)	0.8553	0.9048	0.9048	0.8736	0.8097

Table 22. Final Performance Comparisons for Count ngram (1,1) representations on ML *KPS $_{max}$* models

Algorithm	*PP*	*Ngram*	*Precision*	*Recall*	*Accuracy*	*F1 – Score*	*Youden – J*
MNB	*SW/Stem*	(1,1)	0.9299	0.9215	0.9215	0.9232	0.9114
MLR	*SW/Stem*	(1,1)	0.9172	0.8930	0.8930	0.8950	0.9312
SVM	*SW/Stem*	(1,1)	0.9260	0.8966	0.8966	0.8949	0.9341
MLP	*SW*	(1,1)	0.9478	0.9374	0.9374	0.9387	0.9525

6. Conclusions

After analyzing final learning performance final results, we can observe the following keypoints:

- Both *KPS $_{min}$* & *KPS $_{max}$* generalization perform better on Count Vectorization (frequency weighting) is considered to build Document Term Matrixes.
- Both *KPS $_{min}$* & *KPS $_{max}$* generalization perform better on Stopwords Removal TextPre Processing.
- *KPS $_{min}$* Annotation Task has a better generalization performance when minority classes oversampled up to 100% majority class framework is considered to heal data imbalance, and combinations of n-grams (single, two, three) features frequencies are considered as feature extraction.
- *KPS $_{max}$* Annotation Task has a better generalization performance when minority classes oversampled up to [5% - 25%] majority class framework is considered to heal data imbalance, and monograms (single word) feature frequencies are considered as feature extraction.
- On *KPS $_{max}$* Annotation Task, best learning performance found is:
  1. Class Imbalance: minority classes Oversampling to 15% of majority class.
  2. Text Pre Processing: Stopwords Removal.
  3. Feature Extraction: Count Vectorizer (1,1) mono-grams.
  4. Model: Multilayer Perceptron.
  5. Settings: MLPClassifier (activation='relu', hidden layer sizes = (100,1), alpha=0.0001, tol = 0.0001, learning rate ='constant', solver ='adam', max iter = 200).
  6. *Accuracy: 0.9374, F1-Score: 0.9387, Youden-J(Informedness): 0.9525.*
- On *KPS $_{min}$* Annotation Task, best learning performance found is:
  1. Class Imbalance: minority classes Oversampling to 100% of majority class.
  2. Text Pre Processing: Stopwords Removal.
  3. Feature Extraction: Count Vectorizer (1,3) mono-grams, bi-grams, tri-grams.
  4. Model: Multinomial Naive Bayes.

5. Settings: Multinomial NB (alpha = 0.0000000001, class prior = None, fit prior = True).
6. *Accuracy: 0.9070, F1-Score: 0.8754, Youden-J(Informedness): 0.8141*

- The best decision support models for annotation of trials found after all of the experimentation done in different stages seem to be: *MLPClassifier* for KPS_{max}, *MultinomialNB* for KPS_{min} achieving multi-class accuracy scores of 0.9374 & 0.9070 respectively.

References

[1] *Demner-Fushman D., Chapman WW., McDonald CJ*. What can natural language processing do for clinical decision support? Journal of Biomedical Informatics, Number 42, Vol. 5 (2009).

[2] *National Institute of Health*. Breast Cancer Clinical Trials. (2017)

[3] Clinical Trials Governmental Organization. Protocol Registration Data Element Definitions for Interventional and Observational Studies. http://prsinfo.clinicaltrials.gov/definitions.html, (2017).

[4] *Melnikov M., Vorobkalov P*. Metrics in Ontologies in the Medical Domain. (2014).

[5] *Jain J., Kumari A., Somvanshi P., Grover A., Pai S., Sunil S*. In silico analysis of natural compounds targeting structural and nonstructural proteins of chikungunya virus. F1000Research, Number 1, Vol. 1, (2017).

[6] *National Institutes of Health*. BioPortal Ontology. https://bioportal.bioontology.org/ontologies, (2011).

[7] *Goodwin TR., Harabagiu SM*. Medical Question Answering for Clinical Decision Support. Processing ACM Interantional Conference Information Knowledge Management, Number 1, Vol. 1, Pages = 297- 306, (2016).

[8] *Medbravo Barcelona*. MedBravo Programming Interview Task. https://stackoverflow.com/jobs, (2015).

[9] Ecog-Acrin Organization. ECOG Performance Status Specifications. http://ecog- acrin.org/resources/ecog-performance-status, (2017).

[10] *Zubrod, Charles G. et al*. Appraisal of methods for the study of chemotherapy of cancer in man: Comparative therapeutic trial of nitrogen mustard and triethylene thiophosphoramide. Journal of Clinical Epidemiology, Number 1, Vol. 11, Pages = 7-33, (1960).

[11] *Karnofsky D., Burchenal J*. Evaluation of chemotherapeutic agents: The clinical evaluation of chemotherapeutic agents in cancer. Evaluation of Chemotherapeutic Agents, Number 1, Vol. 11, Pages = 191-205, (1949).

[12] National Institute of Health, ClincalTrial.org. Clinical Trials XML Data Finder. https://clinicaltrials.gov, (2018).

[13] *Peus D., Newcomb N., Hofer S.* Appraisal of the Karnofsky Performance Status and proposal of a simple algorithmic system for its evaluation. BMC Medical Informatics and Decision Making, Number 1, Vol. 13, Pages = 1-7, (2013).

[14] *P. M. Rodda* Text Mining: Automatic Retrieval, Annotation and Visualisation of Clinical Trials Text using Ontology. Master thesis. University of Manchester (2010).

[15] *Kiritchenko, S., de Bruijn, B., Carini, S., Martin, J., Sim, I*. ExaCT: automatic extraction of clinical trial characteristics from journal publications. BMC Medical Informatics and Decision Making, Number 10, Vol. 56, (2010).

[16] *Millian et al*. Eligibility Criteria Text Extraction. (2013).

[17] *Cao X., Maloney K., Brusic V*. Data mining of cancer vaccine trials, a bird's eye view. Immunome Research 2008, Number 4, Vol. 7, (2008).

[18] *Reynoso-Aguirre P., Rodriguez-Hontoria H., Belanche Mun˜oz Ll.* (2018). Natural Language Processing and Machine Learning Techniques to Solve a Breast Can- cer Clinical Trial ECOG-Classification Problem (Master's Thesis). Retrieved from https:// upcommons.upc.edu/bitstream/handle/2117/118759/131668.pdf.

[19] *Anderson P., Thor A., Benik J., Raschid L., Vidal. ME*. PAnG: finding patterns in annotation graphs. SIGMOD Conference, (2012).

[20] *Cotik V., Rodriguez H., Vivaldi J*. Semantic tagging of French medical entities using distant learning. (2015).

[21] *Vivaldi J., Rodrguez H.* Using Wikipedia for term extraction in the biomedical domain: first experience. In Procesamiento del Lenguaje Natural 45, Number 1, Vol. 1, Pages = 251-254, (2011).

[22] *OConnor B*. R2 is rescaled mean squared error. (2009).

[23] *Hiar J., Ringle C., Sarstedt M*. Partial Least Squares Structural Equation Modeling: Rigorous Applica- tions, Better Results and Higher Acceptance. Long Range Planning, Number 1-2, Vol. 46 (2013).

[24] *Ruineihart D., Hint. G., Williams R*. Learning Internal Representations by Error Propagation. Parallel Distributed Processing: Explorations in the Micro structure of Cognition, Number 1, Vol. 1, Pages = 1-33, (1985).

[25] *Raschka, S*. Python Machine Learning. Packt Publishing, ISBN: 9781783555130, (2015).

[26] *Pedregosa F. et al*. Scikit-learn: Machine Learning in Python. Journal of Machine Learning Research, Number 1, Vol. 12, Pages = 2825–2830, (2011).

[27] *Yetisgen M., Gunn M., Xia F., Payne T*. A text processing pipeline to extract recommendations from radiology reports. Journal of Biomedical Informatics, Number 2, Vol. 46, Pages = 354-362, (2013).

[28] *Jia Y*. Singular Value Decomposition. (2017).

[29] *Wold H*. Path models with latent variables: The NIPALS approach. Quantitative sociology: International perspectives on mathematical and statistical modeling, Number 1, Vol. 1, Pages = 307-357, (1975).

[30] *Landauer T., Foltz P., Laham D.* An Introduction to Latent Semantic Analysis. (1998).

[31] *Albisua I., Arbelaitz O., Gurrutxaga I., Lasargueren A., Muguerza J., M. Perez J*. The quest for the op- timal class distribution: an approach for enhancing the effectiveness of learning via resampling methods for imbalanced data sets 2008, Number 2, Vol. 45, (2013).