International Research Journal of Engineering and Technology (IRJET) e-ISSN: 2395-0056
Volume: 08 Issue: 09 | Sep 2021 www.irjet.net p-ISSN: 2395-0072

Multi-label Classification Approaches to Assist in Identifying Optimal Agricultural Crop for Cultivation Using the Predicted Crop and Its Yield

Kruthi N Raj

Pre-Final Year(B.E.) Student, Department of Computer Science and Engineering, Bangalore Institute of Technology, Bangalore

---

Abstract - *Artificial Intelligence and Machine Learning are making a major impact across many industries, including agriculture. The agricultural industry is turning to Artificial Intelligence technologies to help enhance a variety of agricultural-related jobs. Identifying an appropriate crop for cultivation that provides a good yield is only one of the many significant agricultural challenges where ML may help. Multi-task learning (MTL), in the classification context, aims to improve the performance of many classification problems by learning them simultaneously. To better characterise a particular data/object instance, multi-label classification tries to associate multiple labels to it. In this paper, I discuss multi-label classification approach, which are used to solve classification problems with at least two targets and a target cardinality two or more. The major goal of this research is to create and evaluate predictive models that can help farmers identify the appropriate crop for cultivation among India's top commercial crops, such as rice, maize, cotton, and jute, based on crop and yield predictions. Predictions are based on key soil nutrients and climatic factors. In a variety of new application domains, including agriculture, data sets having more than one property (target) and more than two classes per property are frequent. This research makes use of a comparable dataset that has two target variables: crop and yield. Rice, maize, cotton, and jute are the four classes in the crop target variable. The yield target variable is divided into three classes: low, moderate, and high. In such datasets, a multi-label classification technique may handle multiple classification tasks at the same time. In this work, three approaches are used to tackle the multi-label classification problem. The techniques are Problem Transformation, Adapted Algorithm, and Ensemble approach. In the problem transformation method, multi-label classification problems are transformed into single-label problem(s). Binary Relevance (BR), Classifier Chains (CC), and Label Powerset (LP) are the problem transformation approaches implemented. Algorithm adaptation method for multi-label classification concentrate on adapting single-label classification algorithms to the multi-label case rather than transforming the problem into different subsets of problems. The adapted algorithm methods implemented are Binary Relevance k-Nearest Neighbours (BRkNN) and Multi-label k-Nearest Neighbours (MLkNN), adapted from the k-Nearest Neighbors method. Ensemble techniques combine and employ many learning algorithms to achieve greater prediction performance than any of the individual learning algorithms could achieve on its own. Ensemble methods in most cases produces better results than individual models. Multi-label classifier is given as base classifier to the ensemble models in this case. The ensemble multi-label classifiers implemented are sistinct RAndom k-labELsets (RAkELd) and overlapping RAndom k-labELsets (RAkELo).*

Key Words: **Multi-label Classification, Crop, Yield, Binary Relevance, Label Powerset, Classifier Chain, Multi-label k-Nearest Neighbours, Binary Relevance k-Nearest Neighbours, Random k-labelsets**

1.INTRODUCTION

In some ways, effective farming comes down to making complicated judgments based on the interconnections of a plethora of variables, such as crop specifications, soil conditions, climate change, and so on. Traditionally, agricultural methods were applied to a full field or, at best, a portion of it. Machine learning in agriculture allows for considerably better accuracy, allowing farmers to treat plants and animals nearly individually, increasing the effectiveness of farmers' decisions dramatically. Crop selection is a time-consuming process of looking for crops that produce good yields and are suited for the water and nutrients available, as well as the climatic conditions such as temperature, rainfall, and humidity. Machine Learning and Deep learning algorithms use decades of field data to assess crop performance in a variety of climatic and soil conditions. Based on this information, they can create a model that predicts crop and its yield. Yield prediction is one of the most significant and often discussed subjects in precision agriculture, since it determines yield mapping and estimation, crop supply and demand matching, and crop management. The methods investigated in this paper aims to assist farmers in determining the best crop for cultivation among India's main commercial crops, such as rice, maize, cotton, and jute, utilising crop and yield predictions based on soil and climatic data.

The task investigated in this paper is classification. An inducer is a classifier-building method that learns a set of labelled instances (training set) whose classification (label value) is already known. The classifier (also known as a classification model) can then be used to classify instances that have not yet been categorised. The examples are usually connected with a single label, which might include two (binary) or more (multi-class) unique values. However, there are numerous situations in which one instance must be connected with several labels. There is a set of labels L= {Li} linked with the

training set in these situations. Each training instance's classification is supplied as values in a subset of labels selected from L. The subsets do not have to be disjoint. Every label is generally considered to be binary for the sake of simplicity. However, each label in the case of this paper is multi-class. The kind of classification is multi-label classification.

Multi-task learning (MTL), or training a single model to perform multiple tasks, is becoming a common technique for current machine learning practitioners. MTL is a branch of machine learning in which several learning tasks are addressed at the same time while making use of commonalities and differences between them. When compared to training the task-specific models separately, it often results in computational gains (one model performing many tasks takes up less memory and storage) as well as performance gains (learning to do well on a related auxiliary task can improve the model's ability on the primary task), resulting in improved learning efficiency and prediction accuracy. There are many classification problems that must be handled concurrently in multi-label classification. Each problem has several classes (multi-class). In this study, there are two multi-class classification problems that must be solved simultaneously. The crop label must be predicted among the four classes of rice, maize, cotton, and jute, which are among India's top ten commercial crops and may bring in lots of income to the farmers, in the first classification task. The yield label must be predicted among the three classes of low, moderate, and high yields in the second classification task. The predictions are based on three major soil nutrients: nitrogen (N), phosphorus (P), and potassium (K), as well as soil type and climatic conditions such as temperature, humidity, pH, and rainfall.

Three techniques to addressing the multi-label classification problem are presented in this paper. Problem Transformation, Adapted Algorithm, and Ensemble Approach are the three methods used. Problem transformation, in which a multi-label problem is transformed into one or more single-label, multi-class problems, is a popular technique to solve multi-label classification. A single-label, multi-class classifier may be used to make single-label classifications, which are then converted back into multi-label representations. In literature, there are numerous families of problem transformation approaches to choose from. These techniques are derived from one or more fundamental problem transformation approaches that serve as the foundation for more sophisticated frameworks or as extensions to existing algorithms. I use three primary approaches in this study. The approaches are Binary Relevance(BR), Classifier Chains (CC), and Label Powerset (LP). A direct adaptation of an existing single-label multi-class algorithm for the purpose of multi-label classification is an alternative to problem transformation. A direct modification of an existing single-label, multi-class algorithm for the purpose of multi-label classification is an alternative to problem transformation. Existing single-label, multi-class algorithms may be adapted, expanded, and customised to handle multi-label learning using algorithm adaptation approaches based on fundamental machine learning algorithms. Binary Relevance k-Nearest Neighbours (BRkNN) and Multi-label k-Nearest Neighbours (MLkNN), adapted from the k-Nearest Neighbors algorithm are the algorithm adaptation approaches used in this study. To enhance the performance of machine learning models, model optimisation approaches are frequently utilised. Ensemble learning is one such approach. The approach intelligently integrates the results of the constituent base models in such a way that the overall accuracy is higher than any single model. For multi-label classification tasks, ensemble techniques have been found to be a useful tool. The ensemble approaches implemented in the paper include Distinct RAndom k-labELsets (RAkELd) and overlapping RAndom k-labELsets (RAkELo). Along with implementing the methods stated above, I have provided a quantitative comparison of the methods utilising evaluation metrics such as accuracy score, hamming score, and F1 score.

2. MULTI-LABEL CLASSIFICATION

In multi-label classification, X represents the domain of cases to be classified, Y represents the set of labels, and H represents the set of classifiers for f: $X \rightarrow Y$, where f is unknown. The aim is to discover the classifier $h \in H$ that gives the maximum probability of $h(x) = y$, where $y \in Y$ is the ground truth label of x. Another approach to display the attribute–value to cope with multi-label problems is as follows: N instances $z_1, z_2,..., z_N$ make up a dataset, each of which has m characteristics $X_1, X_2,..., X_m$ and c labels $Y_1, Y_2,..., Y_c$. If i denotes the i-th instance (i=1, 2,...,N), x_{ij} denotes the value of instance i's j-th attribute (j = 1, 2,...,m), and output y_{ik} denotes the value of instance i's kth label (k = 1, 2,...,c). The instances are the tuples $z_i = (x_{i1}, x_{i2},..., x_{im}, y_{i1}, y_{i2},..., y_{ic}) = (x_i, y_i)$, also referred as $z_i = (x_i, y_i)$, where z_i, x_i, and y_i are all vectors. Each y_i belongs to the set $Y_1 x Y_2 x... x Y_c$; $Y_i = \{0,1\}$.

The multi-label classification problem in this study is solved using three approaches:

1)Problem Transformation

2)Adapted Algorithm

3)Ensemble methods

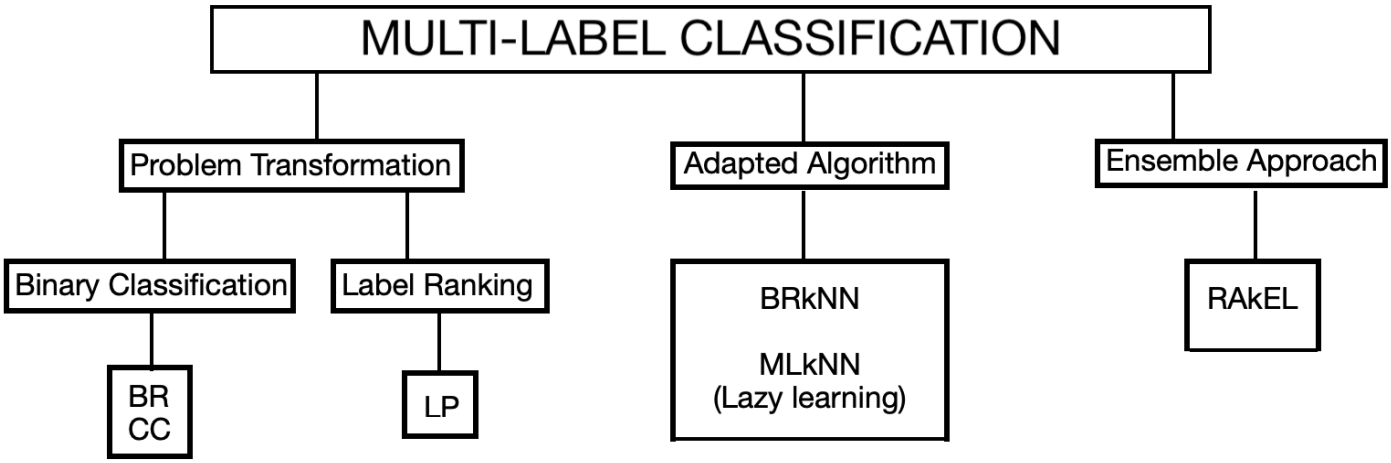

Figure 1 : MLC problem solving approaches discussed in this paper

2.1. MULTI-LABEL CLASSIFICATION APPROACHES IMPLEMENTED

2.1.1. PROBLEM TRANSFORMATION

The problem transformation(PT) methods transform the original problem into one or more single-label, multi-class classification problems. Classification is performed using singe-label classifiers. Problem transformation methods are flexible, and the data is adapted to the algorithm. This work was done utilising PT approaches including Binary Relevance(BR), Label Powerset (LP), and Binary Relavance's enhancement Classifier Chains (CC).

Binary Relevance(BR)- The main idea behind binary relevance [13,16] is to break down a multi-label classification issue into several independent binary classification problems, each of which corresponds to a different potential label in the label space. The binary relevance approach creates a binary training set for class j using the measure shown in Figure. Then, using current binary classification methods, a binary classifier for class j is created to justify whether the instance belongs to class j. Finally, total Q binary classifiers will be created, with Q being the entire number of potential classes. As a result, Q binary classifiers will classify the instance in multi-label classification learning. The sum of all binary classifiers yields the final classification result for an instance.

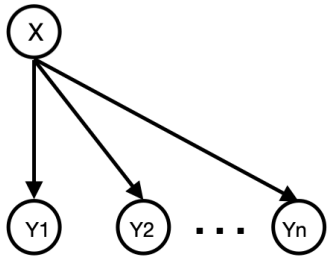

Figure 2 : For a multi-label dataset with n labels, BR divides the dataset into n subsets, each of which contains x and y_i, where $1 \le i \le n$.

$$\mathcal{D}_j = \left\{ \left(x_i, \phi\left(Y_i, y_j\right)\right) \mid 1 \le i \le m \right\}$$

$$\text{where } \phi\left(Y_i, y_j\right) = \begin{cases} +1, & \text{if } y_j \in Y_i \\ -1, & \text{otherwise} \end{cases}.$$

Figure 3 : The metric used by BR technique to create a binary training set for class j

Label Powerset(LP)- It's a straightforward and infrequent problem transformation technique. The idea behind LP [8,12] is to treat each different label-set in a multi-labeled dataset as a single label, converting the original dataset into a single-label dataset that can be used with any single-label classifier. The single-label classifier of LP returns the most likely class label, which is really a set of labels, given a new instance. While the classifier can generate a probability distribution for every class labels, LP can rank them all together. To use the label ranking, it adds up the probabilities of all class labels that contain the label. It coverts the multi-label problem into a single multi-class problem of 2^L class values where L is the number of labels. As a result, LP is capable of doing MLC tasks. Despite the fact that it considers label correlations, it suffers from rising complexity due to the high number of different label sets. Even while the number of different label-sets is generally less, it is still more than the total number of labels, posing a major complexity challenge, especially for large numbers of instances and labels. It also suffered from imbalance as there are not many examples per class label and overfitting.

Classifier Chains(CC)- CC [6] is an improvement to the BR approach. As in a BR approach, it uses n-binary classifiers. The total number of classifiers required for this method is n, which is the same as the number of classes. By taking into consideration the label correlation task, it overcomes the BR constraints. The classifiers are connected in a chain, with each classifier dealing with the label's BR issue. The 0/1 label associations of all preceding links are used to express each link in the chain. When a chain of binary classifiers C_0, C_1,..., C_n is built, a classifier C_i utilises all of the classifier C_j's predictions, where j < i. This allows the technique to account for label correlations. The related label set for test cases is predicted by repeatedly traversing the classifier chain. The ordering of classes during the construction of classifier chains will obviously have a significant impact on the efficacy of the classifier chain technique. Furthermore, due to the chaining characteristic, classifier chains have the benefit of leveraging label correlations while losing the ability to implement in parallel.

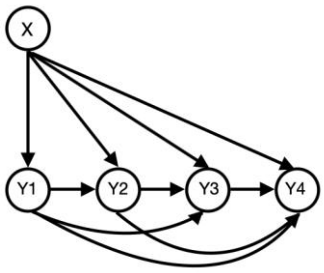

Figure 4 : CC divides a dataset into four subsets (in this case), with previous subsets included as feature attributes in each subset.

2.1.2. ADAPTED ALGORITHM

Adapted method, as the name implies, adapts the algorithm to conduct multi-label classification directly rather than converting the problem into various subsets. It modifies and extends the current specialised learning method to tackle the multi-label problem directly. It's an approach that's based on an algorithm. The algorithm is adapted to fit the data. It adapts a single-label classification algorithm to produce multi-label outputs. Two variants of Binary Relevance k-Nearest Neighbours (BRkNN) and Multi-label k-Nearest Neighbours (MLkNN), modified from the k-Nearest Neighbors technique are the adapted algorithm methods implemented.

Multi-label k-Nearest Neighbours (MLkNN)- It's a Bayesian-based version of the popular k Nearest Neighbors (kNN) lazy learning method [9,17]. The majority class of the k nearest neighbours is assigned to x̃ by kNN. The most common labels of the k nearest neighbours are assigned to x̃ by MLkNN. It specifies the relevant label-set for the new supplied instance using the Maximum A Posteriori (MAP) principle(see Figure 5 , where c_j , x = number of examples in neighbourhood of x with y_j = 1; probabilities estimated from training data), which is based on prior and posterior probabilities for the frequency of each label within the kNN. This probability estimation will be utilised to make a final prediction of a test instance's label set [13]. Distances between instances are measured using the Euclidean metric.

$$y_j = \begin{cases} 1, & \text{if } P(c_{j,\mathbf{x}}|y_j = 1)P(y_j = 1) \geq P(c_{j,\mathbf{x}}|y_j = 0)P(y_j = 0) \\ 0, & \text{otherwise} \end{cases}$$

Figure 5 : Maximum A Posteriori (MAP) principle

Binary Relevance k-Nearest Neighbours (BRkNN)- BRkNN is a multi-label classification method based on the lazy k-Nearest Neighbor (kNN) algorithm introduced in [9]. Despite the similarity in the algorithms, BRkNN is significantly quicker than the kNN method applied using the BR method, since BRkNN only does one search for the k nearest neighbours. The extensions BRkNN-a and BRkNN-b were proposed in [9] to enhance prediction performance and directly address the multi-label problem. Both extensions are based on a label confidence score, which is calculated for each label based on the proportion of the k nearest neighbours that possess that label. BRkNN-a classifies an unknown example E using labels with a confidence score larger than 0.5, i. e., labels that appear in at least half of E's k nearest neighbours. If no label meets this criterion, the label with the highest confidence score is chosen. BRkNN-b, on the other hand, categorises E using the [s] (nearest integer of s) labels with the highest confidence score, where s is the average size of the label sets of E's k nearest neighbours. The BRkNN-a model is implemented in this study.

2.1.3. ENSEMBLE APPROACH

In the context of a classification task, various models have varied strengths depending on whether they are trained using the same or different techniques and capture distinct characteristics and connections hidden in a data set. Even if the most accurate model is chosen, there is no assurance that it will generalise well when exposed to unknown data in the production environment. As a result, relying on a single model typically results in varying accuracy depending on whether the model encounters the same sort of unknown data that it is adept at predicting. Ensemble learning is a viable solution to this problem. In this case, ensemble approaches are built on top of conventional problem transformation and algorithm adaptation techniques. They build a classifier system and then use a weighted vote to categorise incoming data points. They're utilised to improve the accuracy of prediction performance. They are attempting to combine the predictions of various base estimators created using a specific learning method. RAndom k-labEL sets (RAkEL), Ensemble Pruned Set (EPS), and Ensembles of Classifier Chains (ECC) are the ensemble approaches used in this work.

RAndom k-labEL sets (RAkELd & RAkELo)- It is an ensemble based on LP. The goal of RAkEL is to assist LP in dealing with a high number of label sets [14,15]. To address the LP's mentioned problems, Tsoumakas et al. [14,15] proposed randomly breaking the initial set of labels into a number of small-sized label-sets, resulting in a number of small-sized, computationally simpler, and less skewed multi-label classification tasks, and then applying LP to these tasks. The size of the label-sets and the difficulty of the multi-label classification tasks are determined by the k parameter. RAkEL is available in two variants. RAkELd randomly divides the whole label set L of size M into N = [M/k] given a specified value of k. RAkELd trains m multi-label classifiers h_j using LP utilising disjoint label-sets R_j, j = 1,...,N, and splits the training set into subsets Dj accordingly. Let Lk indicate the set of all distinct k-label sets of L in RAkELo. The binomial coefficient: Lk = (M! / k!(M-k)!) determines the size of L_k. RAkELo picks m k-label sets R_i, I = 1,..., m from the set L_k through random sampling without replacement, given a size of label-sets k k and a number of required classifiers m |Lk|. It's worth noting that the label sets in this situation may overlap. RAkEL's computational complexity is proportional to the number of LP classifiers used and their complexity. Furthermore, in the case of disjoint label-sets, the number of LP classifiers is linear with respect to M, but in the case of overlapping label-sets, a value of m that is linear with respect to M can also produce satisfactory empirical results. The predictions hi(x,j) are gathered for a fresh instance x in order to generate the final multi-label classification outputs. In the instance of RAkELo, each model hi generates binary predictions $h_i(x, j)$ for each label j in the associated k-label set R_i, and then uses the average votes rule to combine the LP classifier outputs.

3. EXPERIMENTAL WORK

This section is divided into two subsections: Section 3.1 discusses the dataset and data pre-processing techniques used, Section 3.2 provides the list of parameters given to each MLC algorithm implemented.

3.1. DATASET AND DATA-PREPROCESSING

The dataset utilised in this study comprises real-world data for nitrogen (N), phosphorus (P), and potassium (K), as well as soil type, temperature(in Celsius) , rainfall (in Centimetres), humidity (in Grams of water per cubic metre of air), pH, crop and yield gathered across India between 1998 and 2014. The soil data is represented by digits as follows: alluvial-0, clay loam-1, clay loam-2, volcanic-3, sandy loam-4, and black-5. All of the feature variables are numerical data. Crop and yield, the dependent variables, are categorical data.

One of the pre-processing techniques applied to the dataset is one-hot encoding the categorical data of the dependant variables(crop and yield). Initially, the dataset comprises two multi-class labels, as shown in Figure. The first label, crop, is divided into four classes: rice, maize, jute, and cotton. The second label is divided into three classes: low, moderate, and high. When one-hot encoding is applied to each of these two columns, the first label column (crop) is divided into four

columns, one for each crop class, and the second label column (yield) is divided into three columns, one for each class (low, moderate, and high), resulting in a total of seven binary class labels (0 or 1).

	N	P	K	temperature	humidity	ph	rainfall	Soil	crop	Yield
0	73	43	42	26.583610	78.007748	6.310700	93.234670	0	jute	low
1	91	29	40	19.357000	81.417538	5.386168	264.614870	0	rice	moderate
2	102	46	19	22.770764	82.599331	6.631005	81.495434	5	cotton	high
3	78	58	44	26.800796	80.886848	5.108682	150.355600	0	rice	high
4	73	45	21	24.605322	73.588685	6.636803	96.591953	1	maize	high
...	...	...	...	...	...	...	...	...	...	...

Figure 6 : A snippet of the dataset

label1	label2	label3	label4	label5	label6	label7
0.0	1.0	0.0	0.0	0.0	1.0	0.0
0.0	0.0	0.0	1.0	0.0	0.0	1.0
1.0	0.0	0.0	0.0	1.0	0.0	0.0
0.0	0.0	0.0	1.0	1.0	0.0	0.0
0.0	0.0	1.0	0.0	1.0	0.0	0.0
...	...	...	...	...	...	...

Figure 7 : Binary class labels generated after one-hot encoding the dataset's labels: crop and yield (see Fig 6)

3.2. HYPERPARAMETER TUNING

We always describe two things while creating a Machine learning model: model parameters and model hyper-parameters of a prediction algorithm. Model parameters are intrinsic components of the model, and their values are derived automatically by the model using data. However, hyper parameters are those that may be changed by the programmer to increase the model's performance. Hyper-parameter tuning is the process of tuning such parameters while building machine learning models. It aims to find such parameters where the performance of the model is highest or where the error rate is least. These parameters are defined by us, can be manipulated and the ML algorithms never learn these parameters.In this technique, several parameters and multiple values for each parameter are assessed using cross-validation technique, and the best parameters are retrieved to use in a prediction model. Every model in this study is subjected to hyper-parameter tuning. The models were developed and trained on the dataset, employing the optimal hyper-parameters predicted in the search, resulting in high-accuracy predictions.

The parameters for each of the selected MLC algorithms, given prior to their training are listed below.

Problem Transformation-

The base estimator for each of the three methods implemented in this approach is a decision tree classifier whose hyper parameters are adjusted using cross-validation. '*gini*' for parameter *criterion*, *20* for *max-depth* parameter, and '*best*' for *splitter* parameter are the optimal parameters as observed.

Binary Relavance(BR) is given 1 other parameter:

1. Dense representation for input features and and classes/labels matrices in fit/predict : [False, True]

Label Powerset(LP) is given 1 other parameter:

1. Dense representation for input features and and classes/labels matrices in fit/predict : [True, False]

Classifier Chains(CC) is given 2 other parameter:

1. Dense representation for input features and and classes/labels matrices in fit/predict : [False, True]

2. The order in which the chain should go through labels : range(n_labels) i.e range(7)

Adapted Algorithm-

Multi-label k-Nearest Neighbours (MLkNN) is given 2 parameters:

1. Number of neighbours : 5

2. The smoothing parameter : 1.0

Binary Relevance k-Nearest Neighbours (BRkNN)-

BRkNN-a is given 1 parameter:

1. Number of neighbours : 5

Ensemble Approach-

The base classifier for each of the three methods implemented in this approach is a decision tree classifier whose hyper parameters are adjusted using cross-validation. '*gini*' for parameter *criterion*, *20* for *max-depth* parameter, and '*best*' for *splitter* parameter are the optimal parameters as observed.

RAndom k-labEL sets (RAkEL)-

RAkELd is given 2 other parameters:

1. Dense representation for input and output matrices : [True , False]

2. Size of each of the partitions : 3

RAkELo is given 3 other parameters:

1. Dense representation for input and output matrices : [True , False]

2. Size of each of the partitions : 3

3. Number of classifiers : y_train.shape[1] = 7

4. EVALUATION AND RESULTS

The comparison of the MLC algorithms used in this study will be assessed from two perspectives:

1. Predictive performance assessed using evaluation metrics

2. Time-complexity

4.1. PREDICTIVE PERFORMANCE

The evaluation of multi-label algorithms requires different measures than those used in single- label classification. Different evaluation metrics are used to assess multi-label classification algorithms.

Before we can establish those evaluation metrics, we need to specify a few things:

1. The instances of multi-label dataset (x_i , Y_i), I = 1……m, where $Y_i \subseteq L$ is the set of true labels and L = (j : j = 1…q) is the set of all labels.

2. Given a new instance x_i, the set of labels that are predicted by an MLC algorithm is denoted as Z_i.

3. $r_i(l)$ is denoted as the LR method for the label *l.*

The metrics used to evaluate the MLC algorithms employed are listed below.

1. *Accuracy Score (Exact Match Ratio)*- Multi-label prediction have the possibility of being partially right. One simple solution is to discard partially right labels (consider them erroneous) and extend the accuracy used for single label prediction to multi-label prediction. The equation depicts the accuracy score measurement, with I denoting the indicator function. Clearly, this metric has a drawback in that it does not differentiate between completely erroneous and somewhat accurate answers, which might be deemed harsh.

$$\text{Exact Match Ratio, MR} = \frac{1}{n}\sum_{i=1}^{n} I(y_i = \hat{y}_i)$$

2. *0/1 Loss*- We determine proportions of instances whose actual value is not identical to projected value using this measure, which is known as 1-Exact Match Ratio.

$$0/1\ \text{Loss} = \frac{1}{n}\sum_{i=1}^{n} I\left(y_i \neq \hat{y}_i\right)$$

3. *Hamming score*- The percentage of predicted accurate labels to the total number of labels (predicted and real) for each occurrence is defined as accuracy. The average of all cases is the overall accuracy. The Hamming score is a less confusing term for it.

$$\text{Accuracy} = \frac{1}{n}\sum_{i=1}^{n} \frac{|y_i \cap \hat{y}_i|}{|y_i \cup \hat{y}_i|}$$

4. *Hamming Loss*- It shows how many times an example's relation to a class label is predicted erroneously on average. As a result, hamming loss considers the prediction error (when an inaccurate label is predicted) as well as the missing error (when a relevant label is not predicted), standardised over the entire number of classes and instances. Figure shows the Hamming score measurement, where I is the indicator function. In an ideal scenario, the hamming loss would be 0, implying no mistake; in practise, the lower the hamming loss, the better the learning algorithm's performance.

$$\text{Hamming Loss} = \frac{1}{nL}\sum_{i=1}^{n}\sum_{j=1}^{L} I\left(y_i^j \neq \hat{y}_i^j\right)$$

5. *Precision*- It's the ratio of correctly predicted labels to the total number of actual labels, averaged over all cases.

$$\text{Precision} = \frac{1}{n}\sum_{i=1}^{n} \frac{|y_i \cap \hat{y}_i|}{|y_i|}$$

6. *Recall*- It's the ratio of expected accurate labels to total predicted labels, averaged over all occurrences.

$$\text{Recall} = \frac{1}{n}\sum_{i=1}^{n} \frac{|y_i \cap \hat{y}_i|}{|\hat{y}_i|}$$

7. *F1-Measure*- Definition of precision and recall naturally leads to the definition for F1-measure (harmonic mean of precision and recall) as shown in the equation.

$$F_1 = \frac{1}{n}\sum_{i=1}^{n}\frac{2|y_i \cap \hat{y}_i|}{|y_i| + |\hat{y}_i|}$$

Metrics \ MLC Algorithm	BR	LP	CC	MLkNN	BRkNN-a	RAkELd	RAkELo
Accuracy Score	0.5625	0.5125	0.7500	0.6500	0.6500	0.6000	0.5625
0/1 Loss	0.4375	0.4875	0.2500	0.3500	0.3500	0.400	0.4375
Hamming Score	0.7531	0.6250	0.8135	0.7625	0.7625	0.7458	0.6729
Hamming Loss	0.0911	0.1821	0.0804	0.1036	0.1036	0.1018	0.1536
Precision	0.8294	0.6878	0.8438	0.8277	0.8294	0.8006	0.7371
Recall	0.8667	0.6849	0.8479	0.8383	0.8335	0.8493	0.7490
F1 Score	0.8473	0.6829	0.8446	0.8269	0.8269	0.8222	0.7393

Table 1 : Predictive performance of MLC classifiers assessed using evaluation metrics

4.2. TIME-COMPLEXITY

In terms of time, I discovered that LP is the least time-consuming algorithm, followed by CC and BRkNN-a, with MLkNN being the most time-consuming. The classification time in seconds consumed during the process for each algorithm is shown in Table 2.

MLC Algorithm	Time (in Sec)
BR	0.0283
LP	0.0164
CC	0.0191
MLkNN	0.3707
BRkNN-a	0.0206
RAkELd	0.0686
RAkELo	0.1279

Table 2 : Time-complexity of MLC classifiers implemented

Overall, based on the results obtained, it can be concluded that CC is the best algorithm among those implemented, since it has the best predictive performance and is also one of the fastest algorithms among those implemented. Furthermore, BRkNN-a is the superior choice between MLkNN and BRkNN-a because, while both have extremely comparable predictive performance, BRkNN-a takes substantially less time than MLkNN.

5. CONCLUSION

In this paper, I implemented and compared several MLC predictive models with an aim to create a system that may assist farmers in identifying the appropriate crop for cultivation among India's main commercial crops, including rice, maize,

cotton, and jute, based on crop and yield predictions. I examined seven effective MLC approaches in order to determine which classification algorithm is best suited for the dataset and for predicting crop and yield. The feature variables in the dataset are data related to a variety of climatic and soil characteristics. There are two labels: crop and yield, each having four classes (crop) and three classes (yield). The CC algorithm was found to be extremely successful, and it is recommended as the best classification technique for this particular task. Also, the findings show that BRkNN is the superior choice between MLkNN and BRkNN since both models have comparable predictive performance but BRkNN is much quicker.

REFERENCES

1. Batista, V., Pintado, F. P., Gil, A. B., Rodríguez, S., & Moreno, M. (2011). A System for Multi-label Classification of Learning Objects. Springer, Soft Computing Models in Industrial and Environmental Applications, 6th International Conference SOCO 2011 Advances in Intelligent and Soft Computing, 87, 523-531.

2. Comp R. Cerri, R. R. Silva, and A. C. Carvalho,. "Comparing Methods for Multilabel Classification of Proteins Using Machine Learning Techniques", BSB 2009, LNCS 5676, 109-120, 2009.

3. El Kafrawy, P., Mausad, A., & Esmail, H. (2015). Experimental Comparison of Methods for Multi- Label Classification in Different Application Domains. International Journal of Computer Applications, 114(19), 1-9.

4. Krzysztof D., W. Cheng & Eyke H. (2010). Bayes Optimal Multilabel Classification via Probabilistic Classifier Chains. ICML 2010.

5. Madjarov, G., Kocev, D., Gjorgjevikj, D., and Deroski, S. An extensive experimental comparison of methods for multi-label learning. Pattern Recognition, 45 (9):3084 – 3104, 2012.

6. Read, J., Pfahringer, B., Holmes, G., & Frank, E. (2009). Classifier chains for multi-label classification. Springer,Machine Learning, 5782, 254-269.

7. Santos, A. M., P Canuto, A. M., & Neto, A. F. (2011). A Comparative Analysis of Classification Methods to Multi-label Tasks in Different Application Domains.International Journal of Computer Information Systems and Industrial Management Applications, 3, 218-227.

8. Sorower,M.S.(2010).A Literature Survey on Algorithms for Multi-labelLearning, 25.

9. Spyromitros, E., G. Tsoumakas and I. Vlahavas, An empirical study of lazy multilabel classification algorithms, in: Hellenic conference on Artificial Intelligence (2008), pp. 401–406.

10. Tawiah, C. A., & Sheng, V. S. (2013). Empirical Comparison of Multi-Label Classification Algorithms. Proceedings of the Twenty-Seventh AAAI Conference on Artificial Intelligence, 1645- 1646.

11. Trohidis, K., G. Tsoumakas, G. Kalliris and I. Vlahavas, Multi-label classification of music into emotions, in: International Conference on Music Information Retrieval, 2008, pp. 1–6.

12. Tsoumakas, G., and Katakis, I. (2007). Multi-label classification an overview. International Journal of Data Warehousing and Mining. 3(3), 1-13

13. Tsoumakas,G., Katakis,I., & Vlahavas,I.P.(2010).Mining Multi-label Data,1-20.

14. Tsoumakas, G., Katakis, I., & Vlahavas, I. (2010). Random k-Labelsets for Multi-Label Classification. Knowledge and Data Engineering, IEEE Transactions, 23(7), 1079 - 1089.

15. Tsoumakas, G. & Vlahavas, I. (2007). Random k-Labelsets: An Ensemble Method for Multilabel Classification. Machine Learning: ECML 2007, 2007, Volume 4701.

16. Zhang, M.-L., Li, Y.-K., Liu, X.-Y., and Geng, X. (2018). Binary Relavance for multi-label learning: Overview. Frontiers of Computer Science, 12(2):191-202.

17. Zhang, M., & Zhou, Z. (2007). ML-KNN: A lazy learning approach to multi-label learning. ScienceDirect, Pattern Recognition, 40(7), 2038-2048.