Automatic Mapping of Graduates' Skills to Industry Roles using Machine Learning Techniques: A Case Study of Software Engineering

[1]Fullgence Mwachoo Mwakondo, [2]Lawrence Muchemi, [3]Elijah Isanda Omwenga
[1]Institute of Computing & Informatics, Technical University of Mombasa
[2,3]School of Computing & Informatics, University of Nairobi

Abstract
The main focus is determine a machine learning model for mapping graduates' skills to industry roles using skills profile of employed graduates. A hierarchical classification strategy using a bottom-up approach was designed based on a taxonomy that is bottom-up friendly and was applied to construct the model. Two machine learning techniques, naiveBayes and support vector machines, and software engineering employees' profile dataset with 113 instances and 18 attributes were adopted in the investigation using experimental design. Experiments to evaluate the model were designed using pretest-posttest with control group. While the aim was to assess performance of the model under effect of various machine learning techniques and taxonomic structures, performance reported on carefully selected benchmark on bottom-up multi-classification method was adopted for validation.
Findings indicate model performance is not only considerably fair both under naïve Bayes (57.85%) and SVM (67.15%) but also slightly above the reported benchmark score of 61%. However, difference between the two model designs is significant (t=2.602, p=.029; t= -2.939, p=.017). In conclusion, automatic mapping of graduates' skills to industry roles with the aim to improve employability and productivity prediction of new graduates must involve both a suitable machine learning technique and a bottom-up friendly taxonomic structure.

Keywords
Automatic Mapping of Skills, Long Term Unemployment, Machine Learning, Software Engineering

I. Introduction

Employment Trends [1] indicate rapid growth of long term unemployment (LTU) which is as a result of increased unemployment rate currently standing at 13 per cent, originally at 5.6 and 6.2 per cent in 2007 and 2010 respectively [2]. In Europe, number of unemployed persons went up from 30.6 million in 2007 to 47 million in 2010, while LTU went up from 7.9 million to 14.9 million in the same period [3]. In Kenya, number of unemployed persons increased from 1.8 million in 1998/99 to 1.9 million in 2005/2009 [4]. Empirical studies indicate unemployment problem relates to either workers unable to match their skills to requirements of advertised jobs [4], or employers unable to find workers with important skills, especially both before and after economic recession of 2008 to 2010 [5]. Large companies have the highest trouble (30% before and 25% after recession), than smaller companies (19% before and 17% before recession).
Despite rapid development in information technologies, employers still suffer from challenges not only in recruiting the right talent but also predicting which applicant would have better work performance [6]. These challenges have also been observed in graduates when choosing industry jobs that befit their skills. Yet, using machine learning techniques, these challenges can be reduced or eliminated. Machine Learning (ML) can be used to view challenges facing both employers and graduates as a skill mapping problem and computationally modeled as a pattern recognition problem and solved using appropriate ML techniques.
ML has been applied in many areas including medicine and manufacturing industry [7] and has exhibited outstanding results. However, studies reveal there is very little research in human resource management especially towards improving graduates employability using machine learning techniques [2]; [6]. Yet, with increasing unemployment trend in the world and decreasing capacity of most economies to create employment opportunities, employability of young and productive graduates is at threat of long term unemployment if something is not done to reverse the trend. To increase graduates' chances to obtain jobs that match their training knowledge and skills, universities need to not only equip their graduates with necessary knowledge and skills to enter the labour market but also promote their capacity to achieve enhanced performance in the job by mapping their skills to the right job in the industry [2]. With ever increasing pool of qualification mix for new graduates each year, employers are also at risk of not only taking longer to search the pool but also selecting graduates whose skills do not match their needs [8].
This paper examines potential of ML techniques both in mapping problem solving skills of employed graduates to industry jobs and predicting suitable industry jobs for new graduates. The rest of this paper is organized as follows: section II highlights related work, section III outlines the problem statement, section IV describes ML techniques, section V discusses the applicability of ML in automatic mapping of skills to industry roles, section VI describes methodology of the study, and while section VII presents results and discussions, section VIII presents conclusion and recommendations.

II. Related Work

Literature reveals there are fewer studies towards improving graduates employability especially using machine learning techniques [2]; [6]. Zaharim et al. [9] applied requirements of professional bodies and accrediting bodies to construct an Engineering employability skills Framework for Malaysian graduates. Chien & Chen [6] built a classification model using data mining techniques for prediction of employee retention of new job applicants. Jantawan & Tsai [2] presented a classification model based on decision trees and naiveBayes for predicting graduate employment twelve month after graduation based on attributes that influence graduate employment identified from actual data collected from graduates. Many of these studies approach skill mapping to industry roles using top-down method yet natural mobility of employees in the industry is bottom-up.

III. Statement of the Problem

Industry is facing a problem of finding skilled graduates who fit to their roles while academia is facing a problem of matching graduates' skills with industry roles, due to effects of industry

academia gap. If nothing is going to be done to reverse this trend, employers are at risk of not only employing graduates who do not match their needs but also take longer to search ever growing pool of new graduates with qualification mix, leading to long term unemployment. Hence, employability of graduates will continue to be at risk of long term unemployment. Predictive mapping of skills to industry roles can improve both matching of skills to industry roles and searching periods by both graduates and employers.

A. General Objective

To investigate whether mapping graduate's skills to industry roles using machine learning techniques improves prediction accuracy for both employability and productivity.

1. Specific Objectives

- To develop an effective machine learning model that maps graduates' skills to industry roles
- To evaluate validity of the model.

B. Research Questions and Hypotheses

RQ1: What is the prediction performance of an effective model design for mapping graduates skills to industry roles in the same occupation?

RQ2: what is the validity of the model?

IV. Machine Learning Techniques

Machine Learning (ML), a branch of Artificial Intelligence, is concerned with designing programs that make computers behave intelligently by being able to sense, remember, learn, and recognize patterns [10]. ML problem can be defined as problem of improving some measure of performance when executing some task, through some kind of training experience [11]. Traditionally, ML task can be modeled as a function (f), where learning problem is to improve accuracy of f and training experience consists of a sample data of known input-output pairs (x,y) of the function. Depending on the kind of output (discrete or continuous) f is called a classifier or regression function respectively. In many ML setups the goal is to learn f such that:

$$f: \mathbf{X} \longrightarrow \mathbf{Y} \qquad (1)$$

Where $x \in X$ are inputs while $y \in Y$ are outputs. The goal is to improve performance accuracy of f through optimization procedures and is achieved using various ML algorithms. Conceptually, ML algorithms are viewed to be searching through a large space of candidate functions that optimize performance metric, guided by training experience [11]. A number of ML algorithms have been developed to cover a variety of data and problems exhibited across ML, such as decision trees, Neural Networks, K-Nearest Neighbor, etc.

A number of methods have been devised to solve ML problems and are broadly classified into supervised and unsupervised. Classification is one of the ML methods used to predict group membership for data instances and is of two types: supervised and unsupervised. Objective of supervised classification is to establish a rule/model from a given correctly labeled training data set so as to be able to classify new objects with unknown labels. Two types of supervised classification problems are binary and multiclass problems. Surveys on multiclass classification methods reveal approaches used to solve multiclass problem such as extensible, decomposition and hierarchical methods [12]; [7]. Hierarchical methods involve arranging classes into a tree structure and using a classifier at each node. The goal is to use as few classifiers as possible. Major types of hierarchical classifiers are flat, big bang (also global), and local classifiers [13]. However, multiclass classification is still facing problems, such as large number of classifiers [7]. So far K-1 binary classifiers to classify K-classes problem have been proposed [14].

Studies reveal that most multi-class problems are hierarchical in nature and therefore demand hierarchical classification methods. Three criteria that distinguish these methods are: 1) hierarchical structure (tree or Direct Acyclic Graph), 2) depth of classification hierarchy (mandatory or non mandatory leaf node prediction at any level of hierarchy), 3) hierarchical structure transverse (bottom-up, or top-down). Merschmann & Freitas [13] have distinguish these three criteria very well as follows: 1) in a tree, classes are associated with only one parent while in Direct Acyclic Graph (DAG) classes may have multiple parents, 2) in mandatory leaf node prediction each instance must be assigned classes at the leaf node in the hierarchy while in non-mandatory leaf node prediction instances can be assigned not only leaf nodes but also internal nodes in the hierarchy, 3) in top-down prediction, for each level in the hierarchy (except the top level) each instance classification in the current level is based on the classification in the previous level (parent) while in bottom-up (also known as flat) each instance classification in any level ignores class hierarchy and performs prediction based on leaf node classes only. Tree and DAG hierarchical structures are common with hierarchical classification methods using top-down approach and have yielded very good results. However, applying hierarchical classification methods using bottom-up approach on the conventional tree and DAG hierarchical structure leads to prediction results with multiple class labels, as revealed in [15].

V. Application of ML Techniques in Automatic Mapping of Graduates' Skills to Industry Roles

The concept of industry roles is linked to the concept of occupation which is a collection of jobs, sufficiently similar in work performed and grouped under a common label known as occupational title [16]. Some occupation titles are broad while others are specializations within occupational area. Therefore, occupational titles, also industry roles, are predefined, are structured hierarchically, are associated with a certain skill level and occupational mobility of employees is vertical and upward. Computationally, skill mapping problem can be viewed as a pattern recognition problem and modeled as a ML task by mapping skills to predefined roles in the hierarchical structure and learn a model to classify graduate skills from bottom (entry-level) to top level positions. For this solution to work effectively, a suitable taxonomic structure must be defined that promotes bottom-up transverse and results to a single class label prediction. Consequently, part of the contribution of this study is to propose a bottom-up friendly taxonomic structure.

A. Proposed Taxonomy

Hierarchical classification, a special type of structured classification, is a problem where there is some structure (hierarchical or not) among the classes and the output of classification algorithm is defined over a class taxonomy. Wu et al. [17] defined a class taxonomy as a tree structured regular concept hierarchy defined over a partially order set (C, R), where C is a finite set that lists all classes in the application domain and the relation, R, represents the "IS-A" relationship. According to Silla & Freitas [18], most hierarchical classification problems are based on: (1) trees or DAG structure whose "IS-A" relationship is asymmetric, anti-reflexive,

and transitive, (2) flat or multi-class classifiers that are multi-label. Fig. 5.1(a) presents the tree and DAG structures commonly used in most hierarchical classification problems. Silla & Freitas [18] listed the properties of the "IS-A" relationship as follows:

1. The only one greatest element R is the root of the tree.
2. For every class c_i ; $c_j \epsilon C$; if c_i is related to c_j then c_j is not related to c_i.
3. For every class $c_i \epsilon C$; c_i is not related to c_i.
4. For every class c_i; c_j ; $c_k \epsilon C$; c_i is related to c_j and c_j is related to c_k imply c_i is related to c_k.

The structures in fig. 5.1(a) have challenges with consistency of class membership in the hierarchy and are only suitable for top-down approach, where classification is approached from general to specifics. However, underlying structure of some problems may not align well to top-down approach but bottom-approach. For example occupational titles, also industry roles, are structured hierarchically according to the organizational structure. This is evident from the hierarchical nature of most organizational structures including line, functional, line & staff organizational structures. Each occupation is associated with a certain skill level. Skill level is defined as the amount and type of education and training required to enter and perform the duties of an occupation [16] and varies increasingly upward in the hierarchy, from lower skilled occupations to higher skilled occupations. Further, occupational mobility of employees is vertical and upward i.e. employees start with occupations at entry level positions and progress to increasingly higher skilled occupations. Occupations at higher levels are characterized by higher levels of responsibility, accountability, and subject matter expertise gained through formal education or extensive experience in lower skilled occupations [16]. Occupational mobility may be through promotion or appointment. Unlike promotion where an existing employee progresses upward the job ladders based on observed job performance and experience, in appointment an employee (new or existing) does not necessarily start at the lower levels occupation but can be appointed to any occupation at any level based on performance predicted from their academic qualifications. Therefore, skill mapping involves classifying a set of skills into one of predefined industry roles in the hierarchy. Since natural occupational mobility of employees is upward then classification strategy that aligns well with this phenomenon is bottom-up approach and therefore the above two structures may not work well with bottom-up approach.

Fig. 5.1(b) presents a proposed bottom-up friendly taxonomic structure (BFTS) that represents the hypothetical structural organization of classes as per the structured classification problem and classification assumptions proposed in this method.

Fig. 5.1(b) illustrates hierarchical structure with two branches (may be more), each branch with three levels, a total of twelve leaf node classes (C1.5, C1.6, C1.1.3, C1.2.4, C1.2.1, C1.2.2, C2.5, C2.6, C2.1.3, C2.1.4, C2.2.1, and C2.2.2), and a total of six parent nodes (1, 1.1, 1.2, 2, 2.1, and 2.2), and root node (R). Leaf nodes represent specialized individual roles while the upward arrow indicates the direction of employees' occupational mobility.

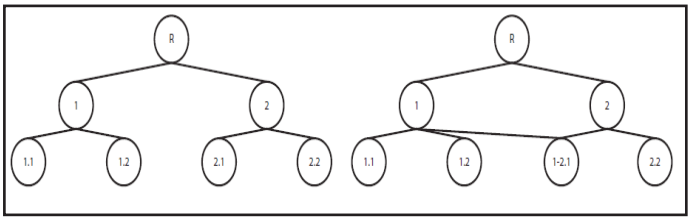

Fig. 5.1(a): Tree structure (left-side diagram) and DAG structure (right-side diagram)

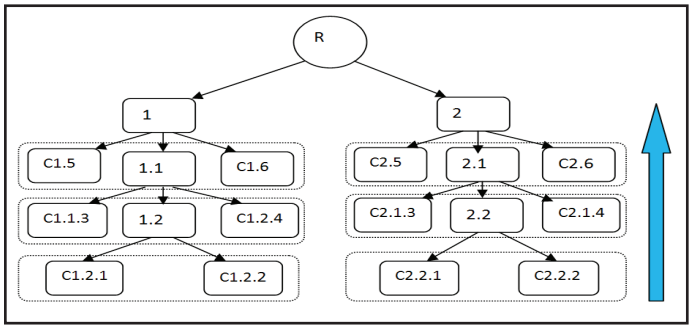

Fig. 5.1(b): Bottom-up Friendly Taxonomic Structure

However, although the proposed taxonomic structure's "IS-A" relationship is asymmetric and ant-reflexive as in Sillas & Freitas (2011) definition of "IS-A" relationship, it departs away from this definition by being anti-transitive with the following properties:

1. The only one greatest element R is the root of the tree.
2. For every class c_i ; $c_j \epsilon C$; if c_i is related to c_j then c_j is not related to c_i.
3. For every class $c_i \epsilon C$; c_i is not related to c_i.
4. For every class c_i; c_j ; $c_k \epsilon C$; c_i is related to c_j and c_j is related to c_k does not imply c_i is related to c_k.

B. Classification Strategy

Based on the problem structure definition, the appropriate classification methodology will be:1) Multiclass, where many classes are involved 2) Hierarchical, where classes are arranged into several levels 3) Supervised, where the model is trained with predefined classes 3) Bottom-up, where the transverse of the structure is performed vertically upward the levels and at each level there are flat classifiers that ignore class hierarchy and perform prediction based only on information gathered from specific leaf node classes. This is as opposed to top-down method where the transverse of the structure is performed vertically downward the hierarchy and at each level there are local classifiers that perform prediction based on information gathered from parent-child relationship [13]. Although the hierarchical structure shows the physical arrangement of classes, the hierarchical classification methodology (strategy) defines the logical arrangement and configuration of classifiers. From Figure 5.1b, the classifier will consists of a collection of flat classifiers organized methodically into layers that will be activated from bottom to top as follows:1) Branch classifiers at the bottom layer 2) Parent classifiers in each branch at the middle layer 3) Child classifiers in each parent node at the top layer.

VI. Methodology

To optimize accuracy of our strategy and also promote validity of the findings, industry roles must be mapped into the proposed taxonomy. Since definition of industry roles is subjective from firm to firm, while some roles have elements from more than one role, mapping of industry roles to the proposed taxonomy involved a preprocessing procedure to the employee profile dataset to discover the underlying functional roles where the goal was to maximize intra-class similarity and minimize inter-class similarity [6]. Our previous work provided the underlying conceptual framework for designing the mapping model [21-22].

A. Experiment Design

In computing two common experimental designs for laboratory experiment well known for their good control of both internal and external validity are pretest-posttest with control group and

Solomon four-group designs. In this case, the pretest-posttest with control group seemed to be the most appropriate, which simply means randomize participants into two groups (experimental and control), take measurement to both before subjecting only experimental group to treatment, then taking measurement in both groups after treatment and conduct a comparison.

Initially, the dataset was randomly split into two groups, experimental and control groups, in the ratio of 2:1 respectively. Experimental group was used for training and control group was used for validation. Class values of both groups were noted before experimental group was applied to selected algorithms to learn the model, after which the model was used to predict the classes of control group. Recorded accuracy was then compared between treatments for any significant differences. However, no model will be able to make good predictions without informative and discriminative features [19]. Thus, to ensure the model was not too complex for underlying data, features that are relevant to the model were selected through sequential backward selection approach using logistic regression algorithm and extracted using Linear Discriminant Analysis (LDA). Besides, to ensure more reliable results with less bias estimate of model's ability to generalize on unseen data [19], an ensemble of hold-out and k-fold cross-validation methods for model evaluation were adopted where dataset was separated into three parts: training, validation and test data. Initially, hold-out was used to partition dataset into training and testing set in the ratio of 2:1 respectively, after which the training set was subjected to k- fold cross-validation where the best fold in each iteration was validated using the hold-out test set. Because of limited size of the dataset and especially fewer instances per class, repeated 5-fold cross-validation was adopted while stratified sampling was applied to ensure class proportions were maintained.

In order to answer the research questions several hypotheses were defined and investigated using designed experiments as follows:

RQ1: What is the prediction performance of an effective model design for mapping graduates skills to industry roles in the same occupation?

To approach the question, two machine learning techniques were adopted in the design of the model and hence the question was rephrased as follows: are there significant performance differences between models built using different machine learning techniques? A research hypothesis was defined to be investigated using an experiment to answer the question:

Hypothesis 1(H_{01}):

H_0: There is no significant performance difference between different model design implementations

H_a: There is significant performance difference between different model design implementations

For this hypothesis, machine learning techniques (naïve Bayes and SVM) were used as test variables for design implementation where the model performance accuracy under each was measured and the difference between the two used as a test statistic. We reject the null hypothesis when the test statistic value (P) is less than significance value (.05), otherwise we accept the null hypothesis.

RQ2: what is the validity of the model?

This question was approached by hypothesizing that bottom-up multi-class classification method applied on bottom-up structured classes problem produces superior performance than when applied on top-down structured classes problem. As a result, our research question was restated as follows: are there significant performance difference between bottom-up multiclass classification using top-down and bottom-up friendly taxonomic structures. A research hypothesis was defined to be investigated using an experiment to answer the question:

Hypothesis 1(H_{01}):

H_0: There is no significant performance difference between bottom-up multiclass classification using top-down and bottom-up friendly taxonomic structures.

H_a: There is significant performance difference between bottom-up multiclass classification using top-down and bottom-up friendly taxonomic structures.

For this hypothesis, taxonomic structures were used as test variables for multiclass classification method where two datasets (one with top-down structured data and the other with bottom-up structured data) were used to experiment the model. In order to investigate validity of the results, performance reported on carefully selected benchmark on bottom-up multi-classification method was also adopted. We reject the null hypothesis when the test statistic value (P) is less than significance value (.05), otherwise we accept the null hypothesis. We also report performance of our method relative to the benchmark method.

B. Benchmarks for Computational Experiments

Since this approach is proposing to use a new dataset that has not been used by similar past approaches, it is not possible to compare directly with those approaches. However, indirect comparison can be done using either performance reported on standard ML datasets whose classes portray hierarchical properties or reported results of similar approaches as used with their respective datasets. In this case, both approaches were adopted. Two well known hierarchical methods found in literature, one using top-down and the other using bottom-up approach, were used for comparative analyses and validation.

After carefully searching for a hierarchical dataset that will suit the purpose of this method, CellCycle, one of the gene function datasets listed by Merschamann & Freitas [13] and previously used by Care & King [20] was selected as baseline to validate our hierarchical approach. The dataset has 78 attributes, 2,486 instances, 5 levels, 97 classes that are hierarchically structured using the conventional top-down tree. Care & King [20] reported an average accuracy of 54% on this dataset using their method. For experiments in this study, the dataset instances were grouped into five bands based on their classes' properties where the longest band with 18 classes and 407 instances was extracted. Initially, preprocessing was done on the classes which resulted into a four layered hierarchical structure and this was used as dataset1. In addition, the work by Barbedo & Lopes [15] that uses bottom-up multi-classification approach applied on the conventional top-down tree was used as the baseline to validate performance of our method. Barbedo & Lopes [15] reported performance result of 61% on genre music classification in their method.

C. Statistical Methods

Both graphical and descriptive analysis procedures were used for analyses, while for significance tests, sample T test methods were used and .05 was used as the test limit for significance.

Accuracy was used as the measure of predictive performance in the method.

VII. Results and Discussions

A. Taxonomic Description of Software Engineers' Industry Roles

Fig. 7.1 illustrates the mapping of 12 industry roles for software engineers into the proposed taxonomic structure using our method. The 12 distinct industry roles were then coded as follows:

1. Mobile system manager
2. Mobile project manager
3. Mobile architect designer
4. Mobile web designer
5. Mobile analyst programmer
6. Mobile test programmer
7. Desktop system manager
8. Desktop project manager
9. Desktop architect designer
10. Desktop web designer
11. Desktop analyst programmer
12. Desktop test programmer

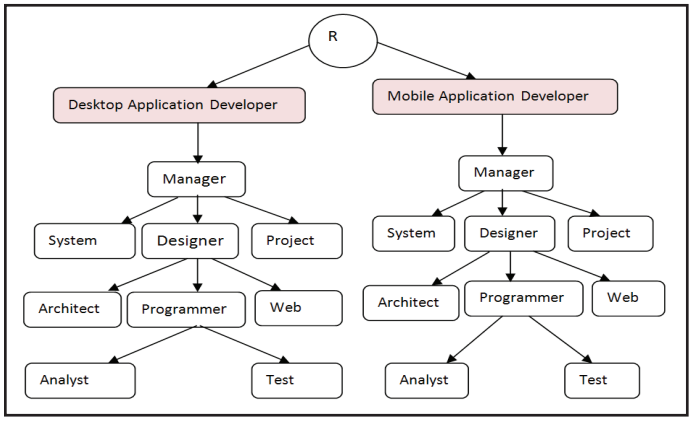

Fig. 7.1: Taxonomy for Software Engineers' Industry Roles

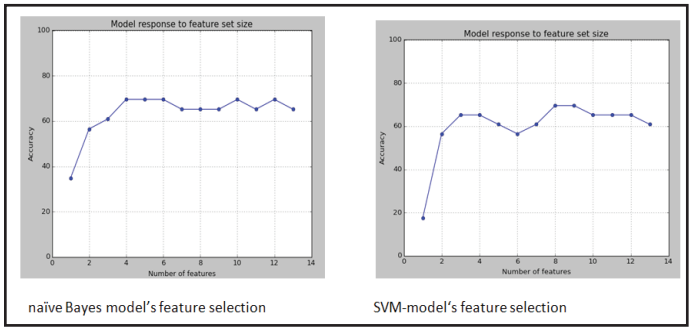

Fig. 7.2: Sequential Backward Selection of Features

B. Experimental Datasets Description

Table 7.2 describes the demographic characteristics of the two datasets used for experimental purpose. Dataset2 represents software engineering employees' profile data while dataset1 represents the CellCycle data that was used as a benchmark.

Table 7.2: Demographic characteristics of datasets

Dataset	attributes	instances	classes	levels
Dataset1	78	407	18	4
Dataset2	18	113	12	4

Table 7.3: Distribution of class instances in the datasets

Classes	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	Total
Instances (dataset1)	18	10	10	11	38	15	17	15	75	13	17	12	10	28	23	24	43	28	407
Instances (dataset2)	1	1	5	15	14	8	12	6	14	16	15	6	-	-	-	-	-	-	113

Initially dataset2 had a total of 17 features excluding the class feature after which feature selection was applied and reduced the features to 13. Figure 7.2 presents line graph showing feature selection results. Although training and testing accuracy achieved with all features included was .924 and .529, the same level of accuracy could be achieved with fewer numbers of features such as 4, 5, 6, 7, 8, 9, 10, 11, 12, and 13. However, when the datasets with thesefeature subsets were fitted into our model, only the 13 feature subset produced optimal results.

C. Class Sizes in the Experimental Datasets

Table 7.3 presents table results showing distribution of class instances in the two datasets as revealed by the experiment. While in dataset1 class number 9 has the highest number of instances of 75, 10 is the lowest number of instances in a class. In dataset2 class number 10 has the largest number of instances of 16 and the lowest number of instances in a class is 1.

D. Models Performance Evaluation

Table 7.4a and Table 7.4b present table results showing performance of our method after experimenting with dataset1 and datset2 respectively. The results reveal in either dataset, SVM model seems to be better than naïve Bayes model (8.4% versus 26.45% in dataset1 and 57.86% versus 67.15% in dataset2). Besides, both models seem to be performing more poorly in dataset1 than in dataset2 (8.40% versus 57.86% for naïve Bayes and 26.45% versus 67.15% for SVM).

E. Hypothesis Testing

Table 7.5 presents results of paired samples T tests that were conducted to test model performance differences under various specifications such, machine learning techniques and dataset structures. The results are based on the 10 iterations conducted in Table 7.4(a) and Table 7.4(b) results.

RQ1: What is the prediction performance of an effective model design for mapping graduates skills to industry roles in the same occupation?

Table 7.5 presents test statistic results (t= -2.939, p=.017) and therefore we reject null hypothesis and conclude indeed there is significant performance difference between different model design implementations. Prediction performance of the model under naïve Bayes is 57.85% while under SVM is 67.15%. This also indicates that SVM-based model is the effective model design for mapping graduates' skills to industry roles and its optimal prediction performance is 67.15%. Therefore, our model performs better when implemented using SVM machine learning technique than naïve Bayes.

RQ2: what is the Validity of the Model?

Table 7.5 presents test statistic results (t= -7.272, p=.000) and therefore we reject null hypothesis and conclude indeed there is significant performance difference of model design under the two taxonomic structures. Prediction performance of SVM model under dataset1(top-down structured classes) is 26.45% while under dataset2(bottom-up structured classes) is 67.15%. Therefore, bottom-up multiclass classification is only valid if it is performed using data that has been structured using a bottom-up friendly taxonomic structure.

VIII. Conclusion and Recommendations

A. Conclusion

The key objective of this paper was to investigate whether mapping graduate's skills to industry roles using machine learning techniques improves prediction accuracy for both employability and productivity.

Table 7.4a: Model performance under dataset1

Technique	Iterations	F1 (%)	F2 (%)	F3 (%)	F4 (%)	F5 (%)	Mean (%)	Hold-Out (%)
NAIVE-BAYES	**1**	10.00	10.00	7.57	8.47	5.45	**8.298**	**0.09**
	2	11.43	12.86	13.63	15.25	9.09	**12.452**	**12.64**
	3	10.00	10.00	6.06	15.25	10.91	**10.444**	**14.94**
	4	14.29	12.86	3.03	10.17	9.09	**9.888**	**11.49**
	5	5.71	8.57	16.67	5.08	20.00	**11.206**	**6.90**
	6	18.57	5.71	7.58	6.78	10.91	**9.91**	**10.34**
	7	10.00	5.71	6.06	11.86	14.55	**9.636**	**0.03**
	8	11.43	7.14	13.64	10.17	9.09	**10.294**	**8.05**
	9	8.57	5.71	10.61	8.47	10.91	**8.854**	**6.90**
	10	11.43	10.00	13.64	15.25	12.73	**12.61**	**12.64**
	Average	**11.14**	**8.86**	**9.85**	**10.68**	**11.27**	**10.36**	**8.40**
SVM	**1**	10.00	12.86	9.09	8.47	14.55	**10.994**	**0.09**
	2	5.72	15.72	6.06	6.78	12.73	**9.402**	**0.09**
	3	5.71	4.29	9.09	15.25	7.25	**8.318**	**41.38**
	4	7.14	5.71	10.61	10.17	9.09	**8.544**	**42.52**
	5	12.86	7.14	6.06	10.17	16.36	**10.518**	**49.43**
	6	8.57	8.57	1.52	11.86	14.55	**9.014**	**51.72**
	7	4.29	10.00	9.09	8.47	12.72	**8.914**	**42.53**
	8	8.57	10.00	6.06	18.64	14.56	**11.566**	**10.34**
	9	7.14	8.57	9.09	11.86	12.73	**9.878**	**12.64**
	10	14.29	10.00	16.67	5.08	10.91	**11.39**	**13.79**
	Average	**8.43**	**9.29**	**8.33**	**10.68**	**12.55**	**9.85**	**26.45**

Table 7.4b: Model performance under dataset2

Technique	Iterations	F1 (%)	F2 (%)	F3 (%)	F4 (%)	F5 (%)	Mean (%)	Hold-Out (%)
NAIVE-BAYES	**1**	45.45	57.89	50.00	50.00	66.67	**54.00**	**50.00**
	2	72.73	63.16	56.25	62.50	83.33	**67.59**	**64.29**
	3	68.18	63.16	81.25	50.00	58.33	**64.18**	**64.29**
	4	72.73	68.42	56.25	56.25	66.67	**64.06**	**64.29**
	5	63.64	68.42	56.25	43.75	75.00	**61.41**	**60.71**
	6	63.64	68.42	75.00	56.25	66.67	**65.99**	**67.86**
	7	45.45	47.37	62.50	75.00	66.67	**59.39**	**67.86**
	8	54.54	57.89	68.75	56.25	83.33	**64.15**	**46.42**
	9	63.63	73.68	68.75	62.50	91.67	**72.04**	**42.86**
	10	72.73	63.16	62.5	75.00	50.00	**64.67**	**50.00**
	Average	**62.27**	**63.16**	**63.75**	**58.75**	**70.83**	**63.75**	**57.86**
SVM	**1**	68.18	47.37	56.25	68.75	75.00	**63.11**	**50.00**
	2	59.09	63.16	75.00	50.00	75.00	**64.45**	**64.29**
	3	68.18	52.63	62.50	43.75	58.33	**57.08**	**71.43**
	4	54.54	47.37	68.75	43.75	66.67	**56.22**	**67.86**
	5	59.09	63.16	56.25	75.00	58.33	**62.37**	**71.43**
	6	59.09	47.37	68.75	56.25	50.00	**56.29**	**71.43**
	7	54.54	42.12	68.75	68.75	50.00	**56.83**	**67.86**
	8	45.45	57.89	56.25	50.00	33.33	**48.58**	**71.43**
	9	54.55	52.63	62.50	68.75	58.33	**59.35**	**67.86**
	10	63.64	36.84	62.50	50.00	50.00	**52.59**	**67.86**
	Average	**58.64**	**51.05**	**63.75**	**57.50**	**57.50**	**57.69**	**67.15**

Table 7.5: Paired sample T tests results

Pair	Paired differences					t	df	Sig(2-tailed)
	Mean	Std. dev.	Std. error mean	95% confidence interval for the difference				
				lower	upper			
naiveBayes-svm	-9.287	9.991	3.1594	-16.434	-2.1398	-2.939	9	.017
*TFTS vs BFTS	-4.069E1	17.696	5.596	-53.351	-28.033	-7.272	9	.000

To achieve this objective, a case study of software engineering was adopted where employees' dataset with 113 instances and 18 attributes was investigated using experimental design method. The findings revealed: 1) Figure 7.2 indicates, although training and testing accuracy achieved with all features included was 92.4% and 52.9%, the same level of accuracy could be achieved with fewer numbers of features where the 13 feature subset produced optimal results for our mapping model. Table 7.4a and 7.4b indicate there is significant performance difference between model design implementation where SVM model (67.15%) performs better than naïve Bayes model (57.85%) and this is confirmed in Table 7.5. Table 7.4a and 7.4b indicates that the model performs much better in datasets whose classes are structured using bottom-up friendly taxonomic structure (67.15%) as opposed to datasets whose classes are structured using the conventional top-down taxonomic structure (26.45%) again confirmed in Table 7.5. Also, there is a general observation that our method performed better than the benchmark method for bottom-up multi classification which was 61%. In conclusion, performance and validity of the skill mapping model depends on not only the machine learning

technique implementation but also the taxonomic structure, and especially for bottom-up multi-classification problem a bottom-up friendly taxonomic structure is indeed the best.

B. Recommendations

To conduct automatic mapping of graduates' skills to industry roles 1) select a suitable machine learning technique that improves performance accuracy, 2) bottom-up multiclass classification has a potential of good results if applied with bottom-up friendly taxonomic structure.

References

[1] International Labour Office (ILO),"Global Employment Trends for Youth 2015: Scaling up Investments in Decent Jobs for Youth", Report (Geneva). 2015.

[2] Jantawan, B. & Tsai, C. "Application of Data Mining to Build Classification Model for Predicting Graduate Employment". International Journal of Computer Science and Information Security, Vol. 11, No. 4, 2013.

[3] Junankar PN. The Global Economic Crisis: Long-Term Unemployment in the OECD. 2011.

[4] Kaminchia S.,"Unemployment in Kenya: Some Economic Factors Affecting Wage Employment". African Review of Economics and Finance Vol. 6, No. 1, June 2014.

[5] Perron R. "Employer Expectations and Experiences. Findings, Training and Keeping Qualified Workers". 2011.

[6] Chien C., Chen L.,"Data mining to improve personnel selection and enhance human capital: A case study in high-technology industry". Expert Systems with Applications Vol. 34, 280–290, 2008.

[7] Mehra, N & Gupta, S. "Survey on Multiclass Classification Methods". International Journal of Computer Science and Information Technologies, Vol. 4 (4) , 572 – 576, 2013.

[8] Quintin G.,"Right for the job: Overqualified or Underqualified". 2011.

[9] Zaharim, A., Omar, M.Z., Yusoff, Y.M., Muhamad, N., Mohamed, A., and Mustapha, R. "Practical framework of employability skills for engineering graduate in Malaysia," In: IEEE EDUCON Education Engineering 2010: The Future of Global Learning Engineering Education, pp. 921–927, 2010.

[10] Leeuwen, J.V.,"Chpater1: Approaches to Machine Learning. Algorithms in Ambient Intelligence". Kluwer Academic Publishers, (2004) ed.

[11] Jordan, M. I. & Mitchell, T. M.,"Machine learning: Trends, perspectives, and prospects", Science Vol. 349, Issue 6245, 2015

[12] Aly, M.,"Survey on Multiclass Classification Methods", 2005.

[13] Merschmann, L.H. C. & Freitas, A.A.,"An Extended Local Hierarchical Classifier for Prediction of Protein and Gene Functions", 2013.

[14] Vural, V., Dy, J.G.,"A hierarchical method for multi-class support vector machines". In Proceedings of the Twenty-First International Conference on Machine Learning, pp. 105-112, 2004

[15] Barbedo J.G.A, & Lopes A."Automatic Genre Classification of Musical Signals". Journal on Advances in Signal Processing Volume 2007, Article ID 64960, 12 pages doi:10.1155/2007/64960

[16] NOC. "Human Resource & Skills Development Canada". National Occupational Classification 2011 Report

[17] Wu F, Zhang J, & Honavar V. "Learning classifiers using hierarchically structured class taxonomies". In: Proc. of the Symp. on Abstraction, Reformulation, and Approximation, Springer, pp. 313-320, Vol. 3607.

[18] Silla, C.N & Freita, A.A.,"A Survey of Hierarchical Classification across different Application Domains". Data Mining And Knowledge Discovery, January 2011.

[19] Raschka, S.,"Python Machine Learning". Packt Publishing, Copyright 2015

[20] Clare A., King R. D.,"Predicting gene function in Saccharomyces cerevisiae". Bioinformatics Vol. 19 Suppl. 2, pp. ii42–ii49, 2003

[21] Mwakondo FM, Muchemi L & Omwenga EI. "Proposed Model for Predictive Mapping of Graduate's Skills to Industry Roles Using Machine Learning Techniques". The International Journal Of Engineering And Science (IJES) Vol. 5, Issue 4, pp. -15-24, 2016.

[22] Mwakondo FM, Muchemi L & Omwenga EI."Trends towards Predictive Mapping of Graduate's Skills to Industry Roles: A case Study of Software Engineering". British Journal of Education, Society & Behavioural Science, Vol. 18, Issue 1, pp. 1-17, 2016.

Mr. Fullgence M. Mwakondo is an assistant lecturer at Technical University of Mombasa, in the Institute of Computing and Informatics. He has over ten years of teaching experience in Computer Science and Information Technology.

He holds a Bachelor of Science degree (Mathematics and Computer Science) from Jomo Kenyatta University of Agriculture and Technology and a Master of Science degree (Information Technology) from Masinde Muliro University of Science and Technology.

He is currently a PhD student (Computer Science) at the University of Nairobi in the School of Computing and Informatics.