Data Mining Techniques for Analysing Employment Data

Anatoli Nachev

Abstract: ***This paper proposes a methodology that uses a large-scale employment dataset in order to explore which factors affect employment and how. The proposed methodology is a combination of predictive modelling, variable significance analysis, and VEC analysis. Modelling is based on logistic regression, linear discriminant analysis, neural network, classification tree, and support vector machine. Following the CRISP-DM standard process model, we train binary classifiers optimising their hyper-parameters and measure their performance by prediction accuracy, ROC analysis, and AUC. Using sensitivity analysis, we rank the variable significance in order to identify and measure factors of employment. Using VEC analysis, we further explore how values of those factors affect employment. Findings show that best performing models are neural networks and support vector machines with preference to the latter for quality of VEC. Experiments also suggest that education and age are primary contributors for correct classification with specific value distribution, discussed in the paper. All results were validated using a rigorous testing procedure that involves training, validation, and test data partitions and a combination of multiple runs along with three-fold cross-validation. This study addresses some gaps in previous research publications, which lack quantification of the conclusions made.***

Keywords: ***classification, data mining, employment data,machine learning.***

I. INTRODUCTION

In recent years, analysing large or big-data sources has become focus to many studies related to data mining and knowledge discovery. Labour data, in particular, have been used to get insights that can drive policies and active management directed towards dealing with unemployment. Knowledge obtained discloses relationships between factors associated with employment and recognises their role. The tools and methodologies used in that analysis become a valuable mean for empirical validation of hypotheses and theoretical considerations in that domain.

This study aims to analyse data form a large-scale nationwide survey of households in Ireland in order to identify empirically employment factors and to find how their values impact on employment. A major component of this analysis is building machine learning classification models that fit the data. Classification is one of the most prominent and effective supervised learning methods, which allows to explore the role of demographic characteristics, education level, dwelling information, family status, and other characteristics in employment status.

A number of works in the labour and employment domain have been published recently, most of which exploring students' and graduates' employment data [15]-[17], [23], or data about employees at organisational level for the purposes of effective HR management [18], [19]. Data mining techniques used include decision trees and Bayesian methods [15]-[19], [24], ensemble methods, MLP, and SVM [16], [25].

The Irish labour data have been explored by Kelly and al. [20], [21], who use non-linear decomposition models to make their conclusions on the unemployment rate. This approach, however, does not provide sufficient measurement of the factors' role and how their values impact on the employment status.

This study addresses the gap by proposing a method that uses several machine learning algorithms for building predictive models that identify, measure, and rank the employment factors and also provide further variable effect characteristic analysis. Classification algorithms used here include logistic regression, linear discriminant analysis, neural networks, classification trees, and support vector machines.

The remainder of the paper is organized as follows: Section II provides an overview of the dataset, data pre-processing steps, modelling algorithms, and performance evaluation metrics. Section III presents the experiments carried out and comments on the results obtained. Section IV gives the conclusions.

II. MATERIALS AND METHODS

This study follows the Cross-Industry Standard Process for Data Mining (CRISP-DM) [27], which remains the most popular methodology for analytics, data mining, and data science projects. The CRISP-DM stages for this study are presented by the sections below as follows:

- *Data understanding* and *data preparation* stages are presented by Sections II A and II B.
- *Modelling* stage is presented by Sections II C, III A, and III B.
- *Evaluation stage* is presented by Sections III A, III B, and III C.

A. Dataset

This study uses the Quarterly National Household Survey (QNHS) [1], which is a large-scale dataset containing responses from nation-wide survey of households in Ireland. It is designed to produce quarterly labour force estimates that include the official measure of employment and unemployment in the state. Data is collected over a twenty-four-months period broken down into four six-month consecutive terms denoted as T1 to T4. The terms capture a period of recovery after economic downturn, which allows to assess the employment factors in the context of changing economic climate.

Revised Manuscript Received on December 15, 2019.
***** Correspondence Author
Dr. Anatoli Nachev, Lecturer, NUI, Galway, Ireland.

Originally, the dataset contains 115 variables grouped into three categories:

- Core variables, which provide information about respondent's demographics, labour status, employment characteristics, atypical work, hours worked, second job, previous work experience of unemployed, search of employment, education and training, and dwelling unit information.
- Derived variables, which are labour related.
- Family unit related variables.

Further details can be found in [1].

Originally, the four terms were represented by separate datasets of size: T1 - 52,763; T2 - 50,515; T3 - 50,939; and T4 - 45,047 observations.

B. Data Pre-Processing

A large number of the original dataset variables were deemed unrelated to the data-mining task of this study and were subsequently eliminated as part of the pre-processing step. For example, variables containing data about few respondents only were discarded as not representative for the entire population. Also, variables deemed identical or dependent to other were eliminated due to the strong correlation. A new binary variable ILO_BIN was added to the dataset. It was derived from the non-binary ILO and used as target variable representing the employment status. After variable elimination, the original 115 were reduced to 17 in five groups, namely:

- Demographic: SEX (gender); MARSTAT (marital status); NATIONAL_SUMMARY (nationality of the respondent); YEARESID_SUMMARY (years of residence in this country).
- Education: EDUCLEVEL (recent/ongoing education and training level); HATLEVEL (highest level of education successfully completed) HATFIELD (field of highest level of education successfully completed);
- Dwelling unit information: DWELLINGUNIT (type of dwelling the respondent lives in); NUMBEROFROOMS (number of rooms); CONSTRUCTIONDATE (construction date of the dwelling); NATUREOFOCCUPANCY (nature of occupancy of the dwelling);
- Technical items related to interview: REGION (region of household); AGECLASS (age class of the respondent);
- Family status: FAMILYTYPE_SUMMARY (type of family); FAMILYPERSON_SUMMARY (person role within the family); FAMILYSTRUCTURE_SUMMARY (summary of family type)
- Target variable: ILO_BIN.

  As part of the CRISP-DM data preparation stage, we did data cleansing by removing records corresponding to age below 16 or above 75, because they were deemed not relevant to employment. Records with missing values were also removed. The number of remaining records after the pre-processing stage were: T1: 35978; T2: 30409; T3: 34240; and T4: 28978 records.

Another pre-processing step is data partitioning. It is required by the supervised machine learning algorithms in order to train, validate, and test the models. There are two possible approaches to take: to break the dataset into two partitions – one for training and another for both validation and testing; or to break the dataset into three partitions: one for training, another for validation, and the third one for testing. The former approach is straightforward and quicker to use, but not reliable enough in obtaining realistic model performance estimates. The rationale for that is that during training, the model fits its parameters to the training data and its hyper-parameters to the validation data. Thus, measuring the model performance by scoring the validation set is not data-neutral and the figures of merit might be unrealistically optimistic. We took the second approach using three partitions: 20% of the original data were selected randomly and set aside as test partition; the rest of observations were split randomly into training and validation partition in ratio 2:1. By not presenting the test data to the models during their training and validation and using the test data solely for testing, makes the model estimation realistic and data-neutral.

C. Modelling Techniques

With reference to the CRISP-DM modelling stage, this study considers five binary classification algorithms: Logistic regression, linear discriminant analysis, neural networks, classification trees, and support vector machines, each outlined below briefly.

1) Logistic Regression (LR): In summary, LR is a statistical technique, which establishes relationships between independent variables X_1, X_2…, X_n and a dependent binary variable Y [6]. The target variable Y can be either continuous or categorical. If p denotes the probability that Y=1, then:

$$Logit(p) = \ln\left(\frac{p}{1-p}\right) = \beta_0 + \beta_1 X_1 + \cdots + \beta_n X_n \quad (1)$$

where β_i are regression coefficients, usually computed by maximum likelihood estimation [6].The probability p can be calculated by:

$$p = \frac{1}{1 + e^{-(\beta_0+\beta_1 x_1+\beta_2 x_2+\cdots+\beta_n x_n)}} \quad (2)$$

Having the probability p, which is a number between 0 and 1, we can map p to the class labels 0 or 1. For example, using a cutoff of 0.5 means that p(Y=1)>0.5; p(Y=0) otherwise. The cutoff need not be set at 0.5.

The LR is one of the most common tools for applied statistics and discrete data analysis.

2) Linear Discriminant Analysis (LDA): LDA is a statistical classification method, formulated by Fisher [8], which maps linear combination of input variables to two or more class labels. The resulting combination can be used as a linear classifier, or for dimensionality reduction before classification. It first computes mean vectors for the different classes from the dataset, then computes in-between-class and within-class scatter matrices and computes eigenvectors (e_1, e_2, ... ,e_d) and corresponding eigenvalues (λ_1, λ_2, ... ,λ_d) for the scatter matrices. Both eigenvectors and eigenvalues provide information about the distortion of a linear transformation. The eigenvectors represent direction of distortion, and the eigenvalues represent the distortion. LDA is closely related to analysis of variance (ANOVA), however, ANOVA uses categorical independent variables and a continuous dependent variable, whereas LDA has continuous independent variables and a categorical dependent variable.

LDA is also closely related to principal component analysis (PCA), which finds directions (a.k.a. principal components) that maximize the variance in a

dataset. In contrast to PCA, LDA is supervised and computes the directions (a.k.a. linear discriminants) that will represent the axes that that maximize the separation between multiple classes. Usually LDA is superior to PCA. It should be mentioned that LDA assumes normally distributed data, input variables that are statistically independent, and identical covariance matrices for every class, but it works reasonably well without those assumptions.

3) Neural Network (NN): Inspired by biological neural systems, the artificial NN are machine learning modelling techniques and algorithms, which perform well in many tasks that require clustering, classification, or regression. Among various NN architectures, the feed-forward multilayer perceptron (MLP) is the most common one and used in this study. An MLP is made of nodes (a.k.a. artificial neurons) organised in layers (Fig. 1). The input and output layers are mandatory for any MLP, but the hidden layer(s), sandwiched between the input and output, can be zero (missing), one, or more.

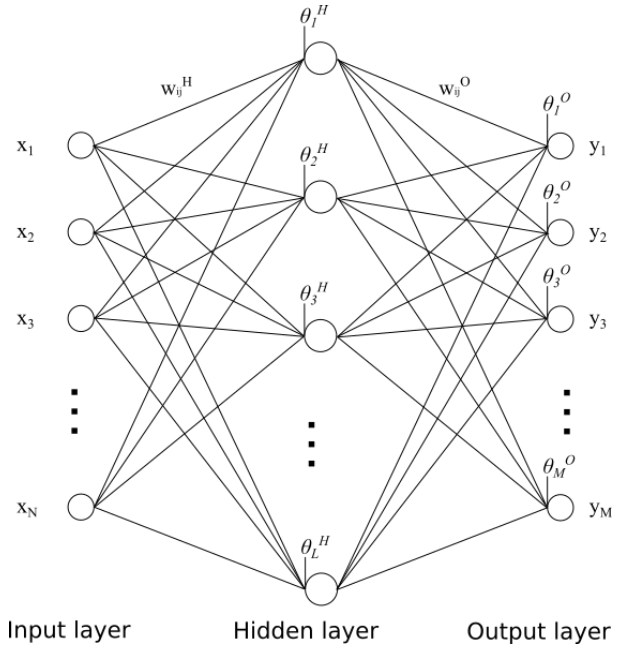


Fig. 1 Architecture of MLP neural network with one hidden layer.

Each layer is fully connected to the neighbouring layers by connections with weights ($w_{i,j}$). Each hidden and output node has an extra bias input with signal value 1 and weight θ_i.

The input layer, being the first taking the NN input data, has size corresponding to the input sample size. The output layer has size determined by theNN purpose (e.g. regression or classification). NN used for binary classifiers have output layer of size 1. The hidden layer(s) number and size may vary, forming different architectures, each of which performing differently. Choosing the correct NN architecture is an application-specific task, which is essential stage of the model building process.

A hidden or output node computes its internal activation signal by

$$s_i = \theta_i + \sum_j w_{i.j} x_i \qquad (3)$$

The signal is then transformed by an activation function, such as the non-linear logistic function:

$$f_i(s_i) = \frac{1}{1 + e^{-\beta s_i}} \qquad (4)$$

where β is slope parameter. Equations (3) and (4) imply that an MPL with one hidden layer outputs

$$y_j = f^O(\theta_j^O + \sum_{k=1}^{L} w_{kj}^O f^H(\theta_k^H + \sum_{i=1}^{N} x_i w_{ij}^H)). \qquad (5)$$

The MLP uses supervised training to fit to the training data in order to be able to make predictions with generalisation. The most common training algorithm is the error backpropagation (BP). R packages used for experiments in this study use BP.

It should be noted that the duration of training requires special attention in order to avoid underfitting or overfitting the models. Monitoring the error levels by scoring the training and validation partitions allows to find the balance between bias and variance, which ensures a good model performance. Among the factors, which control the goodness of the model fit are architecture complexity, size of the training set, training epochs, and other hyper-parameters.

4) Classification Tree (CT): Classification and Regression Trees (CART) is a term referred to the Breiman's work [9] which introduces the C4.5 algorithm for building CT. Fig. 2 shows example of a binary CT with two split variables (height and weight in this example), each having a split value (180 and 80 respectively).

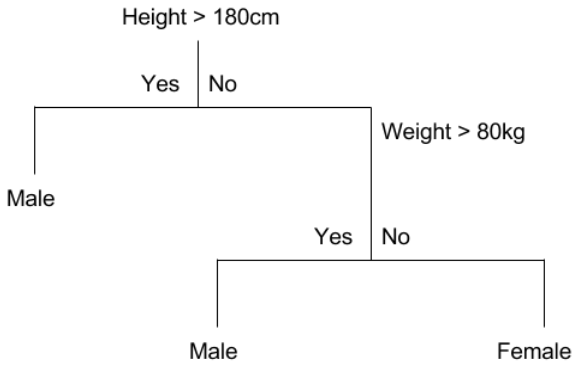


Fig. 2 Example of classification tree.

The tree building algorithm uses two paradigms: recursive partitioning and tree pruning. Recursive partitioning is an iterative process, which splits the training partition into two sub-partitions on the root branches and then continues that splitting recursively [7]. If the training partition has p independent variables x_1, x_2, ..., x_p and one dependent y, the first partitioning estimates each x_i as candidate for being established as tree root along with its split value s_i. The algorithm tries each pair $<x_i,s_i>$ to see how it splits the training partition into sub-partitions, estimating how 'pure' sub-partitions are. Purity is related to the extent of mixing data points from different classes. Pure means all points belonging to one class. The most common metrics for purity used by the CT algorithms are Gini index (6) and information gain (7).

Gini index for a child node partition A is defined as:

$$I(A) = 1 - \sum_{k=1}^{m} p_k^2 = \sum_{i \neq k} p_i p_k \qquad (6)$$

where p_k is the proportion of data point belonging to class k *(k=1..m)*. For binary classifiers (m=2), Gini index values are between 0 (all data points

belong to the same class), and 0.5 (data points are equally distributed between the two classes). The pair $<x_i, s_i>$, which provides the purest split is the successful candidate and established as tree root. The algorithm then applies the same procedure recursively to each partition, splitting it further, and establishing new tree nodes on the tree. The branches grow until the sub-partitions get as homogeneous or 'pure' as possible - that means establishing a terminating leaf node at that branch with the dominating class label on it.

Information gain (IG) is based on the on the concept of entropy from the information theory. Entropy (H) is defined as

$$H(A) = -\sum_{k=1}^{m} p_k log_2(p_k) \quad (7)$$

The IG counts decrease of entropy after splits and the selected $<x_i, s_i>$is the one with maximum decrease.

Tree pruning is a procedure, which reduces a full-grown tree by removing the 'weakest' branches, thus reducing the model overfitting. A good approach to find the best tree pruning is to use the tree complexity parameter (*cp*), discussed later on in Section III.

CT provides some advantages over the other modelling techniques. For example, they don't need variable selection prior to the model building as the selection is happening naturally by the splitter selection mechanism. The closer a variable to the root, the more significant it is. Another advantage of CT is their non-linearity, which makes them great performers for non-linear tasks. This is specifically the cases where discriminating between classes can be described as horizontal and/or vertical splits of the data space. At the same time CT are not that good in capturing diagonal or arbitrary non-linear splits of the data space, where other techniques may perform better. Decision trees are also very robust to outliers.

On the other hand, CT have a requirement to dispose with large dataset for training in order to build a good model. They don't perform well with small datasets. CT are also computationally expensive with large number of variables, in contrast to other classification techniques.

5) Support Vector Machine (SVM): SVM, introduced by Vapnik et al. [10], became very popular supervised machine learning classification technique due to the satisfactory results over a wide range of application domains. It provides a well-balanced trade-off between complexity and learning ability in order to achieve a strong generalization and accuracy [11].

SVM binary classifiers can learn from a two-class training dataset how to construct the best linear separator between the classes among many possible. Intuitively, an SVM selects one, which is maximally far away from any data point of each class. The distance from the linear separator to the closest data point of each class determines the margin of the classifier. The nearest data points to the separator from each class are referred to as support vectors. This method of construction means that the decision function for an SVM is fully specified by the support vectors. The other data points play no role in determining the class separator. Fig. 3 illustrates support vectors, maximum margin, and class separating hyperplane for a sample problem.

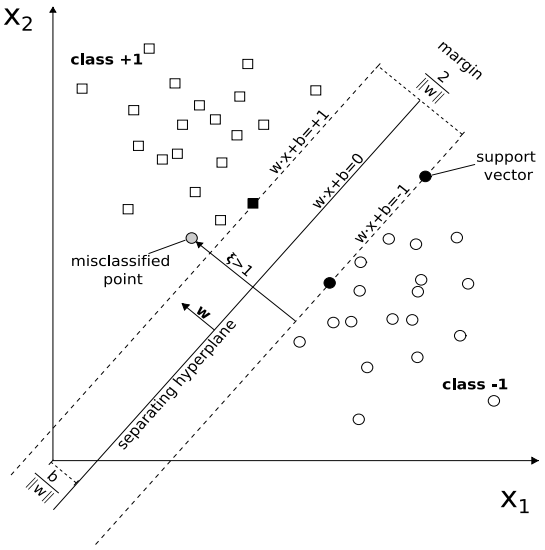


Fig. 3 Support vector machine geometry for a two-dimensional dataset.

Formally, SVM can be defined as follows:

Let the training dataset *D* contains *n* data points $\boldsymbol{x_i}(i=1..n)$, each of which is a *p*-dimensional vector of numeric values, and a class labels y_i with values either -1 or +1.

$$D = \{(\boldsymbol{x_i}, y_i)|\boldsymbol{x_i} \in \Re^p, y_i \in \{-1, +1\}\}_{i=1}^{n} \quad (8)$$

The SVM constructs a hyperplane in $\Re^{p-1}$, (a line in 2-D), that separates between the two classes. The hyperplane can be defined by $\boldsymbol{w} \cdot \boldsymbol{x} + b = 0$, where *w* is the normal vector perpendicular to the hyperplane (a.k.a. weight vector) and *b* is term that specifies the choice of hyperplane among all perpendicular to the normal vector. The hyperplane can also be denoted as $(\boldsymbol{w}, b)$. Any data point x_i would fall into one or another side of the hyperplane, turning (8) into inequality.

The SVM can be defined as a function, which maps data points to a class label +1 or -1.

$$f(\boldsymbol{x_i}) = sign(\boldsymbol{w} \cdot \boldsymbol{x_i} + b) \quad (9)$$

The geometric margin of an individual data point x_i is

$$\delta = \frac{y_i(\boldsymbol{w} \cdot \boldsymbol{x_i} + b)}{\|\boldsymbol{w}\|}, \quad (10)$$

which corresponds to the perpendicular distance to the point. As $y_i$is either +1 or -1, its purpose in the equation is to assure that the distance is a non-negative number.

It can be shown that maximizing the margin of a classifier is the minimisation problem to find $\boldsymbol{w}$and *b*, which minimize (11) for all $\boldsymbol{x_i}, y_i$

$$\Phi(\boldsymbol{w}, b) = \frac{1}{2}\boldsymbol{w} \cdot \boldsymbol{w} \quad (11)$$

and

$$y_i(\boldsymbol{w} \cdot \boldsymbol{x_i} + b) \geq 1 \quad \forall i \quad (12)$$

This is optimisation of a quadratic function subject to linear constraints. The solution involves constructing a dual form of the optimisation problem where a Lagrange multiplier α_i is associated with each constraint (12).

Most Lagrange multipliers found by the optimization problem are zero. Each non-zero α_i^* indicates that it corresponds to a support vector. The optimization also finds the optimal bias b^*. The classification function (9), or the SVM, for a new observation $\boldsymbol{x}$,can be presented in the form:

$$f(\boldsymbol{x}) = sign\left(\sum_i \alpha_i^* y_i \boldsymbol{x_i} \cdot \boldsymbol{x} - b^*\right) \quad (13)$$

Equation (13) implies that SVM is linear, but the dot product between the input vectors opens a door for constructing non-linear SVMs by a technique known as the kernel trick: the input vectors can be mapped from their original input space into a high-dimensional feature space through some non-linear mapping function chosen a priori. The linear decision surface is then constructed in this higher-dimensional feature space. The SVM with kernel function can be defined as

$$\hat{f}(\boldsymbol{x}) = sign\left(\sum_i \alpha_i^* y_i K(\boldsymbol{x_i}, \boldsymbol{x}) - b^*\right) \quad (14)$$

There are several well-known kernel functions that are commonly used for a wide variety of applications with SVM. These are:

- Linear: $K(\boldsymbol{x}, \boldsymbol{z}) = \boldsymbol{x} \cdot \boldsymbol{z}$ (15)
- Polynomial: $K(\boldsymbol{x}, \boldsymbol{z}) = (\gamma \boldsymbol{x} \cdot \boldsymbol{z} + r)^d$ (16)
- Gaussian RBF: $K(\boldsymbol{x}, \boldsymbol{z}) = exp\left(-\frac{\|x-z\|^2}{2\sigma^2}\right)$ (17)
- Sigmoid (*tanh*): $K(\boldsymbol{x}, \boldsymbol{z}) = tanh(\gamma \boldsymbol{x} \cdot \boldsymbol{z} + r)$ (18)
- Laplacian: $K(\boldsymbol{x}, \boldsymbol{z}) = exp(-\frac{\|x-z\|}{\sigma})$ (19)

An SVM assumes that the training dataset is linearly separable into two non-overlapping groups either directly in the input space or in the feature space. However, perfect separation may not be possible, or it may result in a model in so high-dimensional space that the model does not generalize well. To allow some flexibility in separating the classes, Cortes and Vapnik [10] propose soft-margin SVM, which permits some misclassifications. In order to achieve that, the method modifies (14) by introducing slack variables ξ_i and parameter*C* that represents the cost of misclassification. *C* controls the trade-off between allowing training errors and forcing rigid margins.

D. Performance Estimation

- The most common approach for estimating performance of binary classifiers is to use confusion matrix containing four categories of responses: true positives (TP), true negatives (TN), false positives (FP), and false negatives (FN). Performance analysis also refers to Building optimal binary classifiers for each technique for each term T1 - T4 and estimate their performance. Selection of correct hyper-parameters is important part of these experiments.
- Further improvement of models by applying variable selection. This requires estimation of variable significance and reduction of data dimensionality. Methodology used is a combination of sensitivity analysis and backward selection strategy.
- Exploring values of the most significant employment factors by the means of VEC analysis.

terms derived from the matrix, such as True Positive Rate TPR=TP/(TP+FN), referred to as the sensitivity, True Negative Rate TNR=TN/(TN+FP), referred to as the specificity, and False Positive Rate FPR=FP/(FP+TN), referred to as 1-specificity, or anti-specificity.

Accuracy (ACC) of classification is the primary figure of merit of a model. For a given operating point of a classifier, ACC is the total number of correctly classified instances divided by the total number of all available instances ACC=(TP+TN)/(TP+TN+FP+FN). ACC is a good metric that suggests the level of confidence in future predictions, but at the same time it can be misleading, because it varies dramatically if the classes representation in the dataset becomes unbalanced, or if there are different misclassification costs. For those cases sensitivity and specificity can be more relevant performance metrics.

The Receiver Operating Characteristic (ROC) analysis is a more sophisticated approach to estimate binary classifiers [12], [26]. It addresses the ACC deficiencies by plotting a curve with sensitivity on the y-axis against 1-specificity on the x-axis across all possible classifier's operating threshold values. A classifier, which discriminate perfectly between the two classes has a ROC plot that passes through the upper left corner (100% sensitivity, 100% specificity). Therefore, the closer a ROC plot is to the upper left corner, the better performer it is. Fig. 4 illustrates ROC curves of three binary classifiers A, B, and C , which suggest that A outperforms B, which outperforms C.

The area under the ROC curve (AUC) is a scalar metric that represents the classifier performance over all possible threshold values, therefore it is threshold independent. The higher the AUC, the better the model is. In Fig 4 A has largest AUC, followed by B and C.

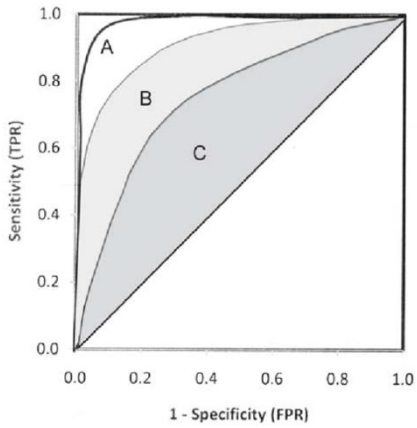


Fig. 4 ROC curves of three classifiers A, B, and C. A has largest area under the curve (AUC).

III. RESULTS AND DISCUSSION

Experiments were carried out in R environment [2]-[5], addressing three major issues:

A. Model Optimisation

In accordance with the assumptions made in Sections II A) and B), we built statistical binary classifiers based on LR and LDA for each period T1-T4. For given training, validation, and test partitions, these models are deterministic as they have no hyperparameter to tune. The variance of results may occur with the random selection of the t-v-t partitions.

In order to validate results and avoid the lucky-set composition effect, models were tested 10 times using different randomly selected t-v-t and for each fit, we applied internally 3-fold CV. Performance was measured by ACC and AUC and results were averaged. Experiments show that ACC for LR varies between 72.1% and 74.2% with average 73%; AUC ranges between

0.788 and 0.807 with average 0.796. Similarly, LDA shows ACC between 71.7% and 73.8% with average 72.9%; AUC varies between 0.786 and 0.807 with average 0.797. Summary of result is presented in Table V.

Searching optimal MLP models, we explored how different architectures affect their performance. As the input and output layer size is pre-determined by the task (16 and 1 respectively), the architectures vary as the number of the hidden layers varies along with their size (H). Best ACC and AUC results were obtained using one hidden layer of size 13. Summary of performance results on the size is presented in Table I. ACC and AUC represent average, while ACC_{max} and AUC_{max} show the maximal values obtained over experiments.

Table-I: Performance of MLP Architecture 16-H-1.

H	ACC	ACC_{max}	AUC	AUC_{max}
0	72.123	72.123	0.788	0.788
1	72.079	72.096	0.788	0.788
2	75.567	75.834	0.823	0.823
3	75.634	75.973	0.828	0.831
4	75.862	76.098	0.832	0.835
5	75.781	76.265	0.832	0.835
6	76.001	76.334	0.834	0.836
7	76.323	76.695	0.837	0.839
8	76.188	76.543	0.838	0.840
9	76.387	76.876	0.840	0.842
10	76.371	76.529	0.840	0.842
11	76.385	76.529	0.840	0.842
12	76.406	76.723	0.842	0.843
13	**76.499**	76.793	**0.842**	0.843
14	76.488	76.779	0.841	0.843
15	76.455	76.790	0.841	0.842
16	76.320	76.668	0.841	0.844
17	76.417	76.668	0.842	0.844
18	76.358	76.487	0.841	0.844
19	76.406	76.612	0.840	0.842
20	76.408	76.570	0.841	0.843

Models based on CT were built using the *rpat*package of R for recursive partitioning for classification [5]. The primary model hyper-parameter is the complexity parameter (*cp*), which is a penalty term that controls the tree size. The *cp* is always monotonic, so that the smaller the *cp*, the more complex the tree. Overgrown trees should be avoided as they are considered as overfitted models. The optimal *cp* value can be found using the following performance metrics:

- Relative error (*rerror*), equivalent to the root-mean-squared-error (RMSE)
- Cross-validation relative error (*xerror*) - the error produced by the built-in 10-fold CV when scoring the training partition.
- Cross-validated standard deviation (*xstd*), referred to as the standard error (*SE*).

Table II shows the region of metric values, where optimal *cp* is expected. Columns represent the seq. number of *cp* (#), the *cp* value, number of splits (ns) in the tree for that *cp*, *rerror*, *xerror*, and *xstd*. The optimal *cp* minimises the *xerror*, which is *cp*=0.000266 in row 29.

Table-II: Tree Pruning Using Complexity Parameter (cp).

#	cp	ns	rerror	xerror	xstd
...	...	...	...	...	...
14	7.81E-04	24	0.514	0.526447	0.005613
15	7.42E-04	28	0.511	0.524572	0.005606
16	**6.25E-04**	**31**	**0.509**	**0.520353**	**0.005590**
17	5.86E-04	32	0.508	0.519963	0.005589
18	5.66E-04	34	0.507	0.519884	0.005588
19	5.47E-04	39	0.504	0.518947	0.005585
20	5.08E-04	44	0.501	0.518947	0.005585
21	4.69E-04	48	0.499	0.517540	0.005579
22	4.30E-04	53	0.497	0.518166	0.005582
23	3.91E-04	56	0.495	0.516056	0.005574
24	3.65E-04	66	0.491	0.516369	0.005575
25	3.52E-04	69	0.490	0.516134	0.005574
26	3.44E-04	71	0.489	0.516134	0.005574
27	3.13E-04	78	0.486	0.515040	0.005570
28	2.73E-04	87	0.484	0.514728	0.005569
29	**2.66E-04**	**94**	**0.482**	**0.514650**	**0.005568**
30	2.60E-04	99	0.480	0.514650	0.005568
31	2.34E-04	108	0.478	0.516056	0.005574
...	...	...	...	...	...

Breiman et al. [9] recommend using the "1SE rule" for selecting optimal *cp*. They suggest adding one *SE* to the minimal *xerror*, where the tree is less complex. This is 0.514650 + 1*0.005568 = 0.520218, corresponding to row 16 in Table II, where *cp*=0.000625. Fig. 5 plots the “1SE rule” tree with *cp*=0.000625 giving ACC=76.5% and AUC=0.813. A CT naturally measures variable significance for the classification task. For each tree node, the algorithm selects for split variable the one, which is most significant among all candidates. That means the closer a variable to the root, the more significant it is. Apart from building CT, the *rpart* package weights variables with respect to their significance. Results are summarised in Table III. It is evident, that the CT model recognises education, training, and age as primary factors that determine employment status.

SVM optimisation requires careful tuning of model hyper-parameters. As discussed in Section C, SVM performance largely depends on the choice of kernel and its specific parameter values, as well as the value of the cost parameter C.The optimisation task is primarily experimental, because the SVM behaviour is driven by the data specifics and depends on the nature of task. Although there are some common recommendations about SVM settings and default parameter values, there is no solid theory on that or universal rules to follow.

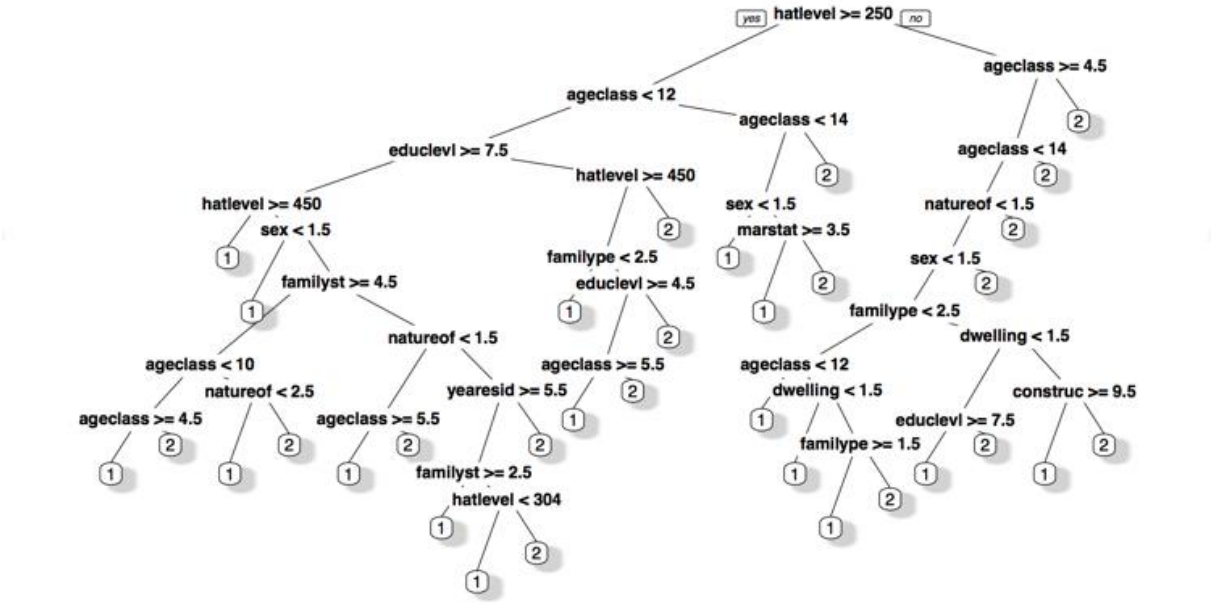


Fig. 5 Optimal classification tree with 31 nodes (cp=0.000625). Leaf nodes represent class labels: 1- employed; 2 – unemployed.

Table-III: Weighted Variable Significance for the Optimal Classification Tree with Complexity cp=0.000625.

#	Variable Name	Weight
1	AGECLASS	25
2	HATLEVEL	25
3	HATFIELD	23
4	EDUCLEVEL	15
5	FAMILYPERSON_SUMMARY	2
6	CONSTRUCTIONDATE	1
7	FAMILYSTRUCTURE_SUMMARY	1
8	NATUREOFOCCUPANCY	2
9	SEX	3
10	NUMBEROFROOMS	0
11	MARSTAT	2
12	FAMILYTYPE_SUMMARY	0
13	DWELLINGUNIT	1
14	YEARESID_SUMMARY	0
15	NATIONAL_SUMMARY	0
16	REGION	0

Considering the cost parameter C for soft-margin SVMs, it should be noted that it controls the trade-off between the training error and model complexity, expressed by the number of support vectors. With too large C, we have high penalty for non-separable data points and many support vectors, which in fact turns the soft-margin SVM into hard-margin SVM. This leads to overfitting. On the other extreme when C is zero, we have no penalty for misclassifications, few support vectors, and model underfitting. For each group of experiments in this study, C was tested in [0,10].

In order to cast a broad catchment area in search of optimal kernel type and parameter values, we tested five kernels with parameter intervals as follows: linear kernel, no parameters; polynomial kernel of degree d=2, $\gamma \in [0,5]$, $r \in [0,5]$; sigmoid (tanh) kernel, $\gamma \in [0,5]$, $r \in [0,5]$; Gaussian RBF kernel, $\sigma \in [0,5]$; and Laplacian kernel, $\sigma = [0,5]$. Table IV summarises results. Best performer is SVM with Gaussian (RBF) kernel and parameters C=3.5, and σ=0.051. It provides ACC=76.9% and AUC=0.830. The Laplacian and polynomial kernels are also good, showing similar results.

Table-IV:SVM Performance: Kernels and Parameters.

Kernel	C	σ	γ	r	ACC	AUC
Linear	3	—	—	—	69.8%	0.775
RBF	3.5	0.051	—	—	**76.9%**	**0.830**
Polynomial	3	—	1.5	2.5	76.4%	0.830
Sigmoid	3	—	1.5	3	58.5%	0.528
Laplacian	3.25	0.052	—	—	76.8%	0.830

Summarising performance and optimisation, we can note that best accuracy is shown by the SVM with RBF kernel (ACC=76.9%), followed by MLP (ACC=76.5%), CT (ACC=76.4%), LR (ACC=73%), and LDA (ACC=72.9%).

The following section discusses experiments for further improvement of the models by variable selection, because building models on reduced number of variables has the potential to make them to generalise better and predict more accurately.

B. Feature Selection

Building binary classifiers allows to estimate the significance of the predictor variables in their ability to discriminate between the classes. Using a subset of the most significant variables for training, validation, and testing would eventually lead to a better fit and improved classification performance.In order to rank variable significance for each model,

we used the sensitivity analysis (SA) method proposed by Kewley et al. [14]. It varies each input variable x_a through its range with L levels from the minimum to the maximum value. Given x_{a_j} denotes the j-th level of input x_a and $\hat{y}$denotes the value predicted, significance can be measured by the gradient measure:

$$S_g = \frac{\sum_{j=2}^{L} |\hat{y}_{a_j} - \hat{y}_{a_{j-1}}|}{(L-1)} \qquad (20)$$

This method was initially proposed for neural networks, but then applied to any supervised learning technique, inclusive those discussed here. Fig. 6-10 show ranking of relative variable significance for each modelling technique, measured by S_g. Whiskers represent confidence intervals.

Fig. 6 and 7 show that LR and LDA provide similar ranking that recognises education/training-related variables and the age variable as the top three most significant with relative importance between 0.15 and 0.20. It is also evident that the least significant variables are region, nationality, and family_structure with relative importance close to 0. A rationale for this observation is that the Irish labour data show homogeneity with respect to those features. These variables appear as primary candidates for elimination, because they do not contribute much to the classifier's ability to discriminate.

Fig. 8 show that the neural network recognises the top three most significant variables in the same way as the statistical LR and LDA above. The low-significant variables, such as region, constructiondate, dwellingunit, and other are also the same, but not ranked in the same way. It can be noted that MLP does not make that sharp distinctions between significant and insignificant as the statistical techniques above do. Indeed, the MLP's highest significance values are between 0.12-0.14; the lowest between 0.02-0.03.

Fig. 9 shows SA made by a best prune CT with complexity factor cp= 0.000625. The CT has 31 nodes but holds a relatively small number of split variables on the nodes. That explains why Fig. 9 lists fewer variables - the others are eliminated by the CT algorithm as insignificant during the tree building. Here we have again the education and age variables recognised as most significant with higher relative importance values ranging between 0.2 and 0.27. The least significant are not listed in Fig. 9.

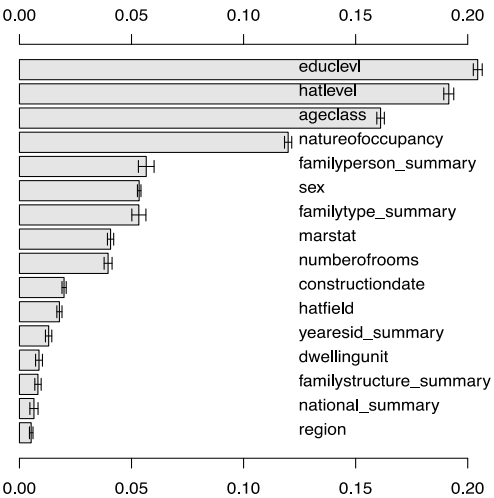


Fig. 6 Variable significance measured by sensitivity analysis of the LR model.

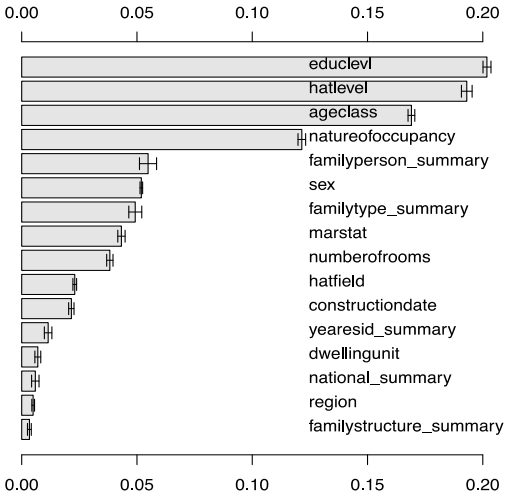


Fig. 7 Variable significance measured by sensitivity analysis of the LDA model.

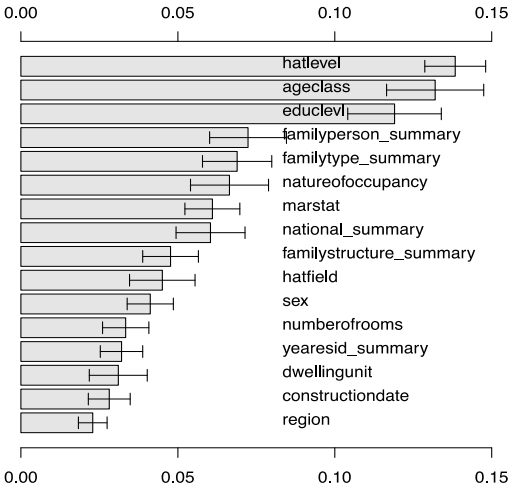


Fig. 8 Variable significance measured by sensitivity analysis of the MLP model.

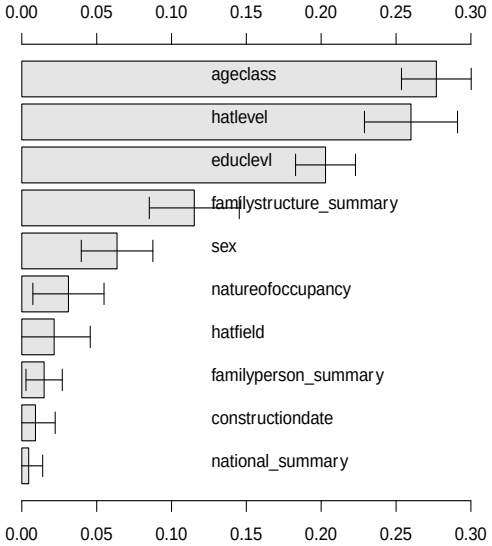


Fig. 9 Variable significance measured by sensitivity analysis of the CT model.

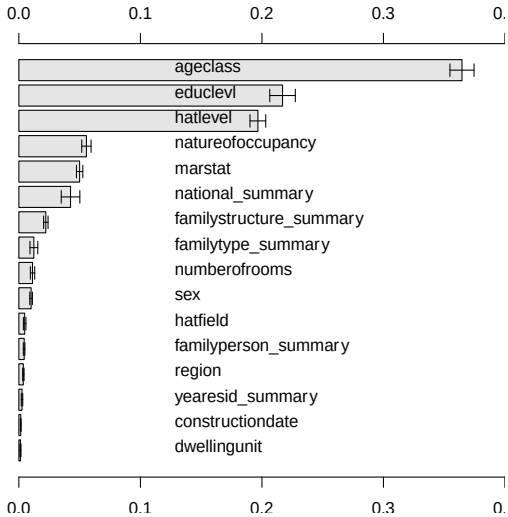


Fig. 10 Variable significance measured by sensitivity analysis of the SVM model.

Fig. 10 shows the SA results for SVM with Gaussian kernel. Again, ranking recognises education and age variables as most significant; those related to region, dwelling, construction date, and other are found least significant. It should be noted, that compared to the other models, SVM makes sharp distinction between significant and insignificant in terms of relative importance values. While the top three have values between 0.2 and 0.35, the lowest go very close to 0.

In summary, Fig. 6-10 show that all models recognise consistently that the top three most significant factors of employment are related to the age and education. The relative significance figures suggest that these factors make collectively contribution to the correct classification about two thirds of all. At the other end, there are many factors identified as insignificant, such as related to region, dwellingunit, constructiondate, etc.

A number of experiments were carried out in order to rebuild models after variable selection using the SA relative significance and backward elimination strategy. Results show that models improve slightly their performance after eliminating the variables dwellingunit, constructiondate, region, and familyperson_summary. Table V summarises results. Performance is measured by ACC and AUC before (old) and after (new) variable selection.

Table-V: Model Performance after Feature Selection.

Model	ACC_{old}	ACC_{new}	AUC_{old}	AUC_{new}
LR	73.0%	73.4%	0.796	0.796
LDA	72.9%	73.1%	0.797	0.797
MLP	76.5%	**77.4%**	0.847	**0.847**
CT	76.5%	76.8%	0.813	0.813
SVM	76.9%	77.1%	0.830	0.830

Results show that the three machine learning algorithms MLP, CT, and SVM outperform the statistical LR and LDA in terms of both ACC and AUC. The feature selection and reducing the data dimensionality slightly improves all model ACC, but not the AUC. That means the variable selection retains of the overall model performance but provides opportunity to improve it by selecting an appropriate classifier operating point.

C. Variable Analysis

Having the variable significance identified by the SA we did experiments for further exploring how values of the most significant variables EDUCLEVEL, HATLEVEL, and AGECLASS contribute to the classifier ability to discriminate between classes. We built variable effect characteristic (VEC) curves of those variables over the four consecutive terms T1–T4. As those terms represent time of economic recovery after recession, VEC curves may provide a valuable insight how employment factors change in that context.

Following the notation of (20), within the range of values of an input x_a with L levels from the minimum to the maximum, the VEC plots x_{a_j} on the x-axis versus responses $\hat{y}_{a_j}$ on the y-axis [22]. The graph plots lines as interpolation between two consecutives values of the x-axis for continuous values and a horizontal segment for categorical variables. In our case the closer the values to 0, the higher contribution to unemployment; the closer the values to 1, the higher contribution to employment.

VEC applied to the statistical LR and LDA produces very coarse almost linear plots, which don't provide the insight sought. Because of that, LR and LDA were excluded from further consideration.

1) Factor Age:

Table VI shows VEC curves of AGECLASS using MLP, CT, and SVM over T1-T4. All models show similar non-linear convex shape curve, which starts to grow from value 4 (corresponding to youngsters of age 15-19), then reaching peak of employment at value 8 (age 35-39), then going down towards values corresponding to retirement age. That distribution of employment by age is relatively consistent over T1-T4 and also captured by all models. As it stands, youngsters and those beyond mid-age show lower employment than people in their 30^{th} to 40^{th}. Along with aging employment goes down steadily, hitting its minimum at value 14 (age 65-69).The AGECLASS graphs have minor deviation in results, but the modelling algorithms show different ability to capture relationships between age and employment. The confidence intervals represented by vertical whiskers are larger for MLP and CT and negligibly small for SVM. Compared to CT curve, the SVM's one finer and smoother. Apparently, SVM provides best quality VEC analysis, followed by MLP, and CT.

2) Factor Education:

Tables VII and VIII show VEC curves of EDUCLEVEL and HATLEVEL by using MLP, CT, and SVM over T1-T4. EDUCLEVEL graphs in Table VII plot values between 1 and 9 representing the formal education level or equivalent training of the respondents over the last 4 weeks, inclusive ongoing.

The link between education and employment is captured similarly by the three modelling algorithms. Graphs show lowest employment at values 1 to 4 (primary education – 1; lower-secondary – 2; upper-secondary – 3; post-secondary non-tertiary - 4). Employment slightly increases at value 5 (short-cycle third-level, corresponding to diplomas) and more substantially at 6 (bachelor degree) and above, hitting the top at value 8 (PhD degree).

Table-VI:VEC Plots for AGECLASS

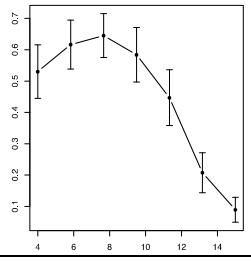
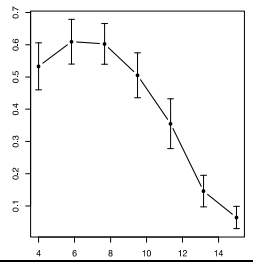
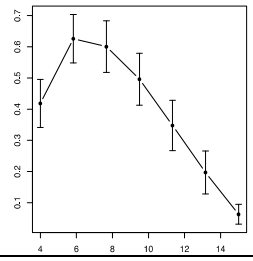
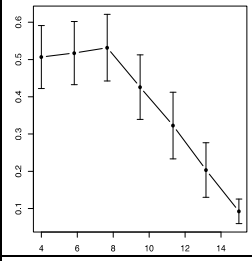
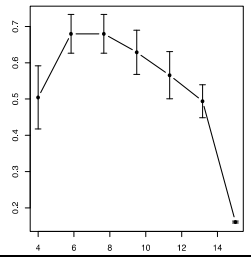
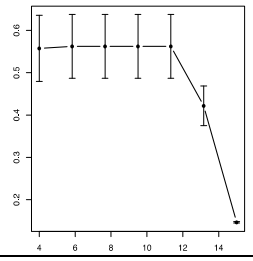
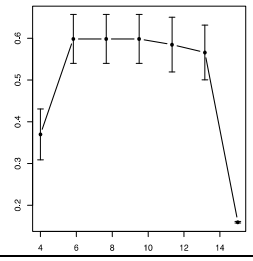
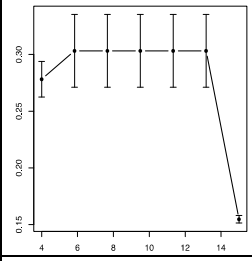
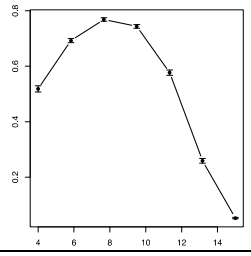
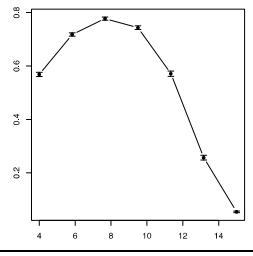
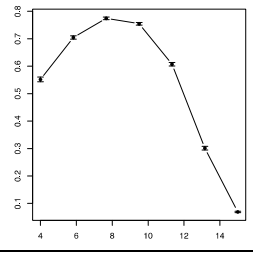
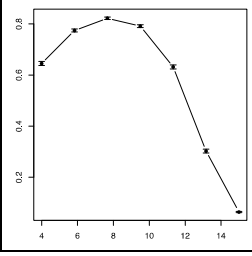
Model	T1	T2	T3	T4
MLP				
CT				
SVM				

Table- VII:VEC Plotsfor EDUCLEVEL

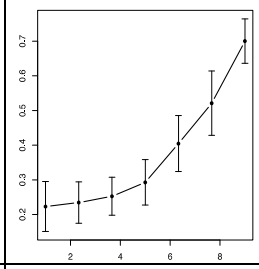
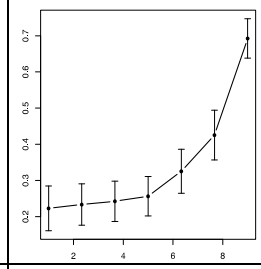
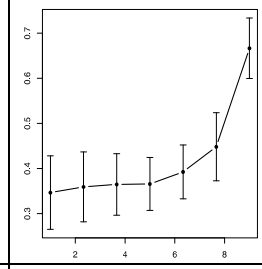
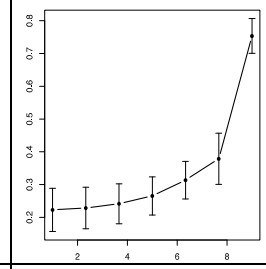
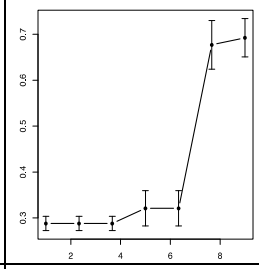
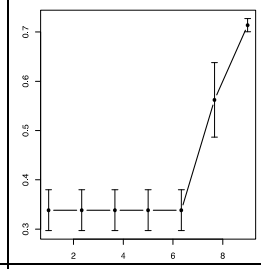
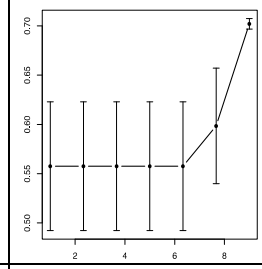
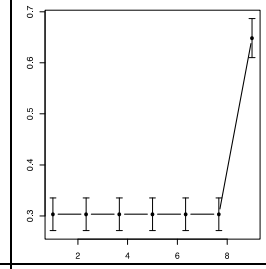
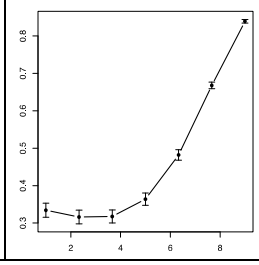
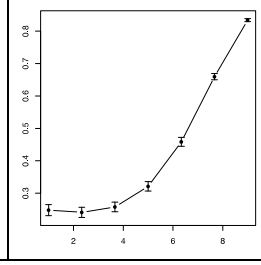
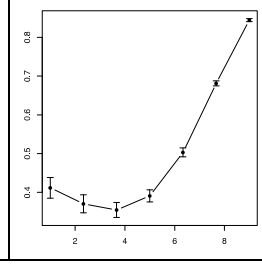
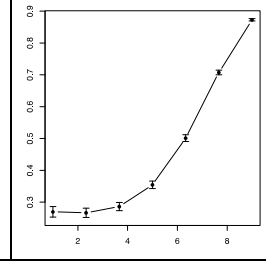
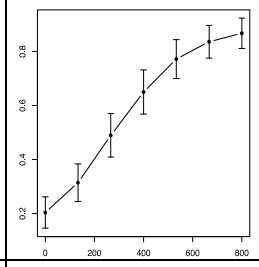
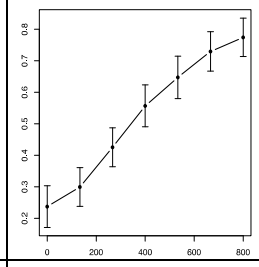
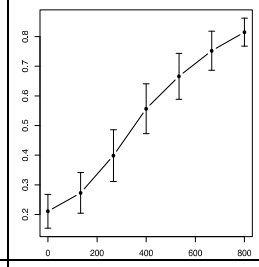
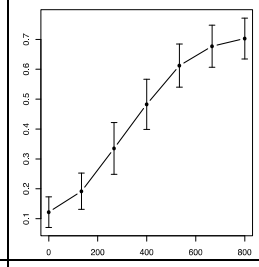
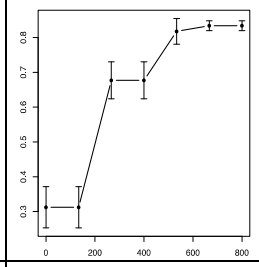
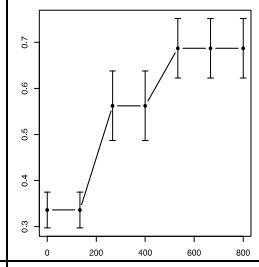
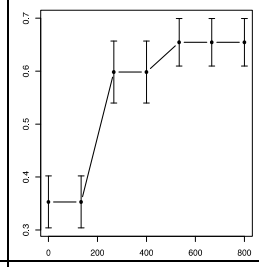
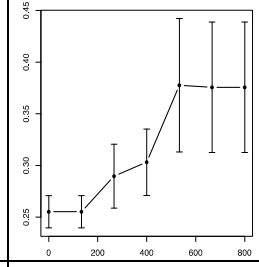
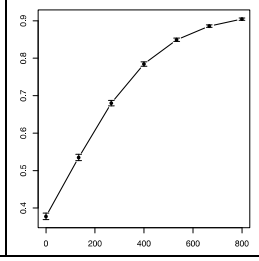
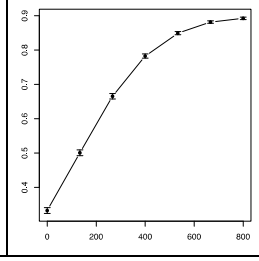
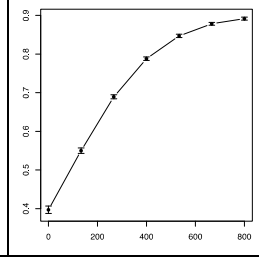
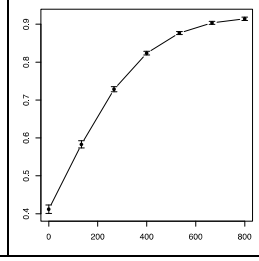
Model	T1	T2	T3	T4
MLP				
CT				
SVM				

Table- VIII: VEC Plotsfor HATLEVEL

Model	T1	T2	T3	T4
MLP				
CT				
SVM				

CT produces coarse and approximated graph with large confidence intervals; MLP graphs are smoother but with large confidence intervals; and SVM graphs are fine and smooth with small confidence intervals. Table VII shows again that SVM produces the best quality VEC curves, followed by MLP and CTs. An interesting observation is the variance of the SVM curves over the periods T1-T4. The left-hand tail of the T3 SVM graph is lifted above the usual, showing that primary level educated have higher employment than secondary level people in that period. A plausible explanation of this observation is that the rapidly developing sectors, such as the building and construction industry needed many workers in the T3 period no matter of their education.

Table VIII shows VEC curves for the highest level of education, successfully completed. Graphs in Table VII capture nearly directly proportional relationship between completed education and employment. It is evident that the highest employment rate is related to third level education; the lower the education, the lower the employment. Table VIII shows again that SVM produces the best quality VEC curves.

VEC analysis can be conducted with less significant variables in order to disclose other interesting relationships, but that is out of the scope of this study.

All experiments discussed in this section show that well-tuned classifiers and carefully selected model hyper-parameters along with subsequent analysis of the predictor variables and their values provide a great potential to turn employments data into actionable insight.

IV. CONCLUSION

Objective of this study is to present a methodology that explores large-scale nation-wide survey data in order to identify and rank employment factors and to measure their role. We address some gaps in previous research, which lacks quantification of conclusions made.

The proposed methodology is threefold: first, we build optimised predictive models based on logistic regression, linear discriminant analysis, neural network, classification tree, and support vector machine. Secondly, we estimate variable significance based on sensitivity analysis in order to recognise and rank the factors that affect employment. Finally, applying VEC analysis we identify the role of the factor values in employment.

The experiments for building models and estimating their performance found that neural networks with 16-13-1 architecture and support vector machines with RBF kernel are best performers in terms of prediction accuracy and AUC as part of the ROC analysis. Results were validated by 3-fold cross-validation combined with multiple tests per model.

Variable significance experiments found that all models rank education and age as most significant factors for employment with collective contribution to the correct classification about two thirds of all.

Further exploring the most significant variables by VEC analysis identified how age and education values are distributed towards employment. Best quality VEC is provided by the support vector machines.

We believe, that the proposed methodology provides means for facilitating active employment management and also can be used as a tool for empirical validation of hypotheses and theories in the field.

REFERENCES

1. The CSO website. [Online]. Available: http://www.cso.ie/en/qnhs/, 2017.
2. R Development Core Team. (2009) R: A language and environment for statistical computing. R Foundation for Statistical Computing. [Online]. Available: http://www.R-project.org, 2009.
3. T. Sing, O. Sander, N. Beerenwinkel, and T. Lengauer, "ROCR: visualizing classifier performance in R," Bioinformatics, vol. 21(20), 2005, pp. 3940-3941,
4. P. Cortez, Package 'rminer' [Online]. Available: https://cran.r-project.org/web/packages/rminer/rminer.pdf, 2016.
5. T. Therneau, B. Atkinson, and B. Ripley, Package 'rpart' [Online]. Available: https://cran.r-project.org/web/packages/rpart/rpart.pdf , 2019.
6. G. Shmueli, P. Bruce, and N. R. Patel Data Mining for Business Analytics: Concepts, Techniques, and Applications, 3rd ed., Wiley Publishing, 2016.
7. The mylearnmachinelearning website. [Online]. Available: https://mylearnmachinelearning.com/decision-treecart/, 2017
8. R. Fisher, "The Use of Multiple Measurements in Taxonomic Problems.", Annals of Eugenics, vol. 7 (7), 1936, pp. 179-188.
9. L. Breiman, J. Friedman, R. A. Olshen, and C. Stone, Classification and Regression Trees, Wadsworth, Belmont, CA, 1984
10. C. Cortes and V. Vapnik, "Support-vector networks.", Machine Learning, vol. 20(3), 1995, pp. 273-297.
11. D. Bzdok, M. Krzywinski, and N. Altman, "Machine learning: supervised methods", Nature Methods, vol. 15, 2018, pp. 5–6.
12. K. Zou, A. Liu, A. Bandos, L. Ohno-Machado, and H. Rockette Statistical Evaluation of Diagnostic Performance: Topics in ROC Analysis, 1st ed., New York: CRC Press, 2016.
13. P. Kumar and A. Indrayan. "Receiver operating characteristic (ROC) curve for medical researchers," Indian Pediatrics, vol. 48, , 2011, pp. 277-287.
14. R. Kewley, M. Embrechts, C. Breneman "Data strip mining for the virtual design of pharmaceuticals with neural networks," IEEE Transactions on Neural Networks, vol. 11 (3), 2000, pp. 668–679.
15. B. Jantavan, C. Tsai, "The Application of Data Mining to Build Classification Model for Predicting Graduate Employment", International Journal of Computer Science and Information Security, vol. 11 (10), 2013, pp.144-151.
16. T. Mishra, D. Kumar, "Students' Employability Prediction Model through Data Mining", International Journal of Applied Engineering Research, vol. 11. No. 4, 2016, pp. 2275-2282.
17. M. Sapaat, A. Mustapha, J. Ahmad, K. Chamili, R. Muhamad, "A Classification-based Graduates Employability Model for Tracer Study by MOHE", Digital Information Processing and Communications, Springer Berlin Heidelberg, 2011, pp. 277-287.
18. J. Kirimi, C. Moturi, "Application of Data Mining Classification in Employee Performance Prediction", International Journal of Computer Applications, vol. 146, No 7, 2016, pp. 28-35.
19. Y. Alsultanny, "Labor Market Forecasting by Using Data Mining," Procedia Computer Science vol. 18, Elsevier, 2013, pp.1700-1709.
20. Kelly, E., McGuinness, S. [online], Impact of the Great Recession on Unemployed and NEET Individuals' Labour Market Transitions in Ireland, Economic Systems, Available: http://dx.doi.org/10.1016/j.ecosys.2014.06.0042014.
21. Kelly, E., McGuinness, S., O'Connell, P., Haugh, D., Pandiella, A. Transitions In and Out of Unemployment among Young People in the Irish Recession, Comparative Economic Studies, vol. 56, 2014, pp. 616-634.
22. P. Cortez, A. Cerdeira, F. Almeida, T. Matos, J. Reis, "Modeling wine preferences by data mining from physicochemical properties", Decision Support Systems, vol. 47(4), 2009, pp. 547–553.
23. J. T. Avella, M. Kebritchi, S. G. Nunn, and T. Kanai, "Learning Analytics Methods, Benefits, and Challenges in Higher Education: A Systematic Literature Review," Online Learning, vol. 20 (2), 2016, pp.13-29, 2016.
24. E. Sugiharti, S. Firmansyah, and F. R. Devi, "Predictive Evaluation of Performance of Computer Science Students of Unnes Using Data Mining Based on Naïve Bayes Classifier (NBC) Algorithm," Journal of Theoretical and Applied Information Technology, vol. 95 (4), 2017, pp. 902-911.
25. T. M. C. Gatbonton, B. E. Aguinaldo, "Employability Predictive Model Evaluator Using PART and JRip Classifier," n Proc. of the 6th International Conference on Information Technology: IoT and Smart City – ICIT'18, 2018, pp.307-310.
26. J. V. Carter, J. Pan, S. N. Rai, and S. Galandiuk, "ROC-ing Along: Evaluation and Interpretation of Receiver Operating Characteristic Curves," Surgery, vol. 159 (6), 2016, pp. 1638-1645.
27. P. Chapman, J. Clinton, R. Kerber, T. Khabaza, T. Reinartz, C. Shearer, and R. Wirth, "CRISP-DM 1.0 - Step-by-step data mining guide," *CRISP-DM Consortium*, 2000

AUTHORS PROFILE

Dr. Anatoli Nachev received his PhD degree in BAS, Institute of Math and Informatics, section AI. He received his MSc and BSc degrees in Sofia University, FMI. He is currently a lecturer at NUI, Galway, Ireland. Research interests include business intelligence, machine learning, data mining, AI, modelling, etc. He has numerous publications in books, international journalsand conferences in the fields of interest.