Identifying Compulsive Gamblers Using Bayesian Networks

Conference Paper · June 2023

CITATIONS
0

READS
119

2 authors, including:

Mark Anthony Caruana
University of Malta
17 PUBLICATIONS **54** CITATIONS

SEE PROFILE

Identifying Compulsive Gamblers Using Bayesian Networks

Mark A. Caruana and Christian A. Farrugia
Department of Statistics and Operations Research
Faculty of Science
University of Malta,
Msida, Malta.
E-mail: mark.caruana@um.edu.mt

KEYWORDS
Bayesian Networks, Classification, Misclassification, Compulsive Gambling, Statistics, Machine Learning, Graph Theory, Trees.

ABSTRACT

The grave consequences suffered by online problem gamblers has led to a growing interest in responsible gambling measures with the intention of preventing players from reaching such a vulnerable state. The focus of this dissertation is to apply statistical machine learning techniques to predict whether a player is most likely a problem gambler or not, whilst identifying which variables were deemed useful predictors of problem gambling. Bayesian networks are implemented on a data set containing historical data obtained from a local medium-sized Malta Gambling Authority (MGA) gambling operator. The models are tesed based on a number of goodness of fit measures such as the predictive accuracy and Area Under the Curve (AUC).

INTRODUCTION

The term *problem gambling* refers to all betting behaviour patterns which may disrupt or compromise the personal life of a gambler. Such behaviour can lead to grave consequences suffered by the staker, such as, but not limited to, debt, failed relationships, and loss of employment. In particular, (Petry 2006) found that online gambling is associated with inferior mental and physical health. On the other hand, a company having a large amount of problem gamblers in their clientele may enjoy higher revenues but risk incurring hefty fines or the loss of their gambling licenses. These clashes of interests have prompted betting companies to investigate different approaches that seem viable when dealing with problem gamblers.

The ease of access to gambling through the internet has let to a vast body of literature which aims to study the impact online gambling has on individuals when compared to traditional (offline) gambling. A study conducted by (Griffiths et al. 2009) suggests that online gambling is more likely to lead to problem gambling than offline gambling. (McCormack and Griffiths 2012) conducted a survey on a sample of both online and offline gamblers and found that the players perceived online gambling to be more addictive. Furthermore, convenience and anonymity were both identified as major factors contributing to the players' inclination towards online gambling.

Since online gambling has led to an abundance of data on players' transactions on betting sites, statistical and machine learning techniques are frequently used to analyze problem gambling. Statistical and machine learning consist of algorithms which are trained on a data set and are able to detect patterns and / or make predictions. Using such techniques to detect problem gambling can be useful for both the gamblers and the gambling operator. By preempting that a user is likely to become a problem gambler, the betting operator may opt to send responsible gaming messages to the gambler in an attempt to increase their awareness on the amount of time or money they have spent on the site. This might avoid potential socio-economic problems for the gambler while saving the company from incurring a fine due to lack of adherence to responsible gambling regulations. In literature, various techniques have been used to detect problem gambing, which include, and are not limited to, Support Vector Machines (SVM) (Akhter 2018), Artificial Neural Networks (ANNs) and Bayesian Neural Networks (Buttigieg et. al 2022), Recurrent Neural Networks (RNN) (Bermell 2019) Random Forests (Farrugia 2022), Logistic Regression (Farrugia 2022).

Data sets related to problem gambling usually contain individuals that have decided to stop gambling because they realized that they were becoming addicted. In literature this is called self-exculsion. This is used as a proxy for problematic gamblers (Percy et al. 2016) . In this paper, we shall mainly focus on Bayesian networks (BNs) because such networks do not assume independence between the predictor variables in a data set. These networks try to find hidden relationships between the variables and exploit them to enhance the predictive power of the model. Indeed, it has been noticed by a number of authors which include Percy et al. 2016) that BNs have superior classification capability than other statistical or machine learning algorithms. Moreover, to the best of our knowledge, BNs have never been used to detect problem gambling on a local (Maltese) data set.

In the next section we introduce BNs, discuss their construction, and the algorithms used to fit such networks to a data set. Afterwards, we present and discuss the fitted model as well as a number performance measures which are used to test the quality of the fitted model. In the last section we include some concluding remarks and ideas for possible future research in this area.

BAYESIAN NETWORKS

To properly introduce and define Bayesian networks we first introduce some notation and definitions from Graph theory.

Basic Notation

Let n denote the sample size of our data set and d the number of predictor variables in the data set. The latter random variables are all categorical and will be denoted by $X_1, X_2, \ldots X_d$. We will let $\mathcal{D} =$

$\left(x_{(1)}^T, \dots, x_{(n)}^T\right)$ represent the set of n independent observations of the random vector $X = (X_1, \dots, X_d)$. On the other hand, $\mathcal{Z} = \left(\left(x_{(1)}, c_{(1)}\right)^T, \dots, \left(x_{(n)}, c_{(n)}\right)^T\right)$ will be used to denote a data set containing n independent observations of X and the binary random variable C, taking values in the set $\{0,1\}$, where 0 indicates no compulsive gambling whereas 1 indicates that compulsive gambling was detected.

Concepts from Graph Theory

In this subsection, we will go over a number of definitions and concepts from graph theory which will be used in to properly define a Bayesian network

A *directed graph* $\mathcal{G} = (V, D)$ consists of a finite set of nodes V and an directed edge set D, where each directed edge is contained in the set $V \times V$. This edge set specifies which vertices are directly connected to each other. In the context of BNs, each vertex represents a random variable.

The *parent set* of a vertex X_j in a directed graph is defined mathematically as follows

$$\Pi_j = \{X_k \in V \big| (X_k, X_j) \in D\} \qquad (1)$$

This set contains all the vertices in V that are directly connected to X_j. Hence, we denote $\Pi_i^{\mathcal{G}}$ the parent set of node X_i in graph $\mathcal{G}$.

A *directed acyclic graph* (DAG) is a directed graph in which no cycles are allowed. Hence, there exists no sequence of edges which connect a vertex to itself.
A variable X_k is a *descendant* of X_j if there exists a directed path from X_k to X_j in a DAG.

A DAG is a *tree* if all nodes X_i have only one parent. Only the so-called root (starting) node is exempt from this assumption and therefore is allowed to have no parents.

Defining Bayesian Networks

We now have all the necessary framework to define properly a Bayesian network as follows:
A Bayesian network is a pair $(\mathcal{G}, P)$ where $\mathcal{G} = (V, D)$ is a DAG with node set $V = \{X_1, \dots, X_d\}$ for some $d \in \mathbb{N}$, D is the directed edge set and P is a joint probability density function (PDF) of the discrete random variables $X_1, \dots, X_d$. We denote the parameter space of the PDF as Ψ.
To each node $X_v \in V$ with no parent variables, there is assigned a probability distribution P_{X_v}. To each variable X_v with a non-empty parent set Π_v, we associate the conditional probability distribution $P_{X_v|\Pi_v}$. Moreover, the joint PDF $P_{X_1,\dots,X_d}$ my be expressed as follows:

$$\prod_{v=1}^{d} P_{X_v|\Pi_v}. \qquad (2)$$

An important property of BNs is the so-called Local Markov Property which is defined as follows:
Let $\mathcal{G} = (V, D)$ be a DAG with vertices $V = \{X_1, \dots, X_d\}$. A probability density function P over the random vector $X = (X_1, \dots, X_d)$ satisfies the *Local Markov Property* if for each $j \in \{1, \dots, d\}$, $X_j \perp V \setminus \{V_i \cup \Pi_j\} | \Pi_j$. That is, X_j is conditionally independent of its non-descendants given its parents.

Bayesian Network Classifiers

Having defined a Bayesian network and given examples of its basic structures, we now move on to define Bayesian network classifiers. A Bayesian network classifier is a model used for predicting a discrete class variable C which as stated earlier, can take any value in the set $\mathcal{C} = \{0,1\}$. Given an observation $\boldsymbol{x}$ of a random vector $X = (X_1, \dots, X_d)$ the Bayesian classifier aims to find the most probable class c^* given by:

$$c^* = argmax_{c \in \mathcal{C}} P(c|\boldsymbol{x}) = argmax_{c \in \mathcal{C}} P(\boldsymbol{x}|\boldsymbol{c}) \qquad (3)$$

In equation (3) we used Bayes' formula.
The easiest type of Bayesian network found in literature was proposed by (Minsky 1961) and is the naïve Bayes classifier. This classification technique however assumes that all explanatory variables are conditionally independent. This is a very strong assumption and is clearly very unrealistic especially in our application. We will therefore discuss more elaborate types of Bayesian networks with more realistic assumptions. In this paper we discuss the Chow-Liu (CL) algorithm and its adaptation by (Friedman et al. 1997) as well as the Forward Sequential Selection and Joining (FSSJ) algoritm which was introduced by (Pazzani 1996).

Chow-Liu Algorithm

The CL algorithm assumes that each variable has at most one parent and thus the DAG returned by the CL algorithm is a tree. This assumption is an improvement over the independence assumption imposed by the naïve Bayes classifier. Moreover, this assumption can guarantee the existence of an optimal solution.
Since we have d variables with $d-1$ dependencies and each variable may be conditioned on at most one other variable, then the distribution over the random vector $X = (X_1, \dots, X_d)$ is of the form:

$$P_{X_1,\dots,X_d} = \prod_{i=1}^{d} P_{X_i|\Pi_i^{\mathcal{G}}} \qquad (4)$$

To find the optimal solution we make use of Kruskal's algorithm which finds the so-called maximum weight dependence tree which we denote as σ and is such that for any other tree $\hat{\sigma}$, $\sum_{i,j=1}^{d_\sigma} \hat{I}(X_i, X_j) \geq \sum_{i,j=1}^{d_{\hat{\sigma}}} \hat{I}(X_i, X_j)$, for $i \neq j$. Note that d_σ and $d_{\hat{\sigma}}$ denote the number of nodes in trees σ and $\hat{\sigma}$ respectively. Moreover, $\hat{I}$ is called the mutual information function and is defined as follows:

$$\hat{I}(Y, Z) = \sum_{y,z} \hat{P}_{Y,Z}(y, z) log\left(\frac{\hat{P}_{Y,Z}(y, z)}{\hat{P}_Y(y)\hat{P}_Z(z)}\right) \qquad (5)$$

where $\hat{P}$ is the empirical estimate of the PDF P of two random variables Y and Z. These two random variables represent two nodes in a Bayesian Network.
Kruskal's algorithm operates on a network in such a way that all the mutual information of each edge in a network is computed and arranged in decreasing order. The algorithm starts adding one edge at a time starting with the one with the highest mutual information. Each time an edge is added, the algorithm checks if a cycle is formed. If so, the edge is discarded otherwise it is retained.
It can be proven that this algortihm returns a tree with maximum weight. The proof is not essential in this paper and therefore will not be included however the interested reader may refer to (Loberman and Weinberger 1957).

The CL algorithm is based on Kruskal's algorithm and creates a Bn as follows:
First the CL algorithm computes $\hat{I}(X_i, X_j)$ for each pair of variables where $i \neq j$. After, a graph is built with nodes representing variables $X_1, \dots, X_d$ and a weight is associated to each edge connecting X_i **to** X_j by $\hat{I}(X_i, X_j)$. Kruskal's algorithm is then used to find the maximum weighted spanning tree. The root node is identified and we set the direction of each edge to be outward of the said root node.
A further improvement of the CL algorithm takes into considertation the so-called conditional mutual information between variables X, Y given the class variable C. This is defined as follows:

$$I(\boldsymbol{X}, \boldsymbol{Y}|\boldsymbol{C}) = \sum_{x,y,c} P(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{c}) log\left(\frac{P(\boldsymbol{x}, \boldsymbol{y}|\boldsymbol{c})}{P(\boldsymbol{x}|\boldsymbol{c})P(\boldsymbol{y}|\boldsymbol{c})}\right) \qquad (6)$$

As done in equation (5), in (6) we replace I , by its empirical estimate $\hat{I}$ in which we replace P with its empirical counterpart $\hat{P}$. The adaptation of the CL algorithm which we denote by CL-adapt follows closely the general outline of the CL algorithm. However, since each predictor variable will have the class variable as a parent, instead of considering the mutual information between attributes only, we take into account their conditional information given the class variable C which was defined in (6). This in turn allows better results when fitting a Bn to a given data set. Moreover it can still be proven that CL-adapt still returns a maximum weight tree.

Forward Sequential Selection Algorithm
An improvement over CL algorithm and its adaptation, CL-adapt, is the Forward Sequential Selection (FSSJ) algorithm. This algorithm does not set a restriction on the maximum number of parents a node, which in turn represents variable, can have.
However, as in the case of CL-adapt, the FSSJ algorithm requires that each variable has the class C as one of its parents. The fact that this algorithm may allow each node to have more than one parent means that finding the optimal network is an NP hard problem. The FSSJ algorithm is a heuristic algorithm. In particular, it is an example of a hill climbing algorithm, meaning that it starts off with an initial solution and iteratively aims to improve upon the solution by making a change at each step. It is also a greedy algorithm since once a change is made, it cannot be undone at a later stage of the algorithm.
A common approach when fitting statistical learning models to a data set is to divide it into a training data set and a testing data set. The model is then fitted on the training data set but evaluated on the test data set. The accuracy is then calculated to check how good the classifier is. The accuracy of a model is computed as in equation (7) The accuracy can be computed for both the test set and train sets and ideally the accuracy of both sets should be as close as possible. If there is a large discrepancy then that would be an indication of overfitting.
The FSSJ algorithm starts off with no attributes (the empty set) and then performs several operations until there is no longer any improvement in the accuracy. The algorithm attempts to search for dependencies among pairs of variables by joining them in the DAG. Note that the term "joining" refers to adding a directed edge between two vertices.
Since the FSSJ algorithm iteratively adds variables and dependencies until there is no longer a significant improvement in accuracy, then it can be seen as a feature selection algorithm since it selects a subset of variables (and their corresponding dependencies) whose predictive power is not significantly different from a model containing more features.

FITTING A BAYESIAN MODEL TO A DATA SET

Description Of Data
The data set consists of 130588 observations with 22 predictor variables and 1 response variable. The data set was partitioned into a training set and testing set using a 75:25 ratio. The data includes 65389 users who self-excluded themselves from the gambling operator and 65199 users who never self-excluded. The variables related to transaction data such as deposits and money staked are based on a period of thirty days unless specified otherwise. The R package *bnclassify* which was used to fit Bayesian networks to our data set only handles discrete variables. Therefore, all continuous variables in the raw data set were discretized using the *discretizeDF* function found in the R package *arules*. The frequency method was used which ensures that there are an equal number of observations in each category. More information on this technique can be found in the aforementioned R package.
Note that in the online gambling industry, the GGR (Gross Gaming Revenue) generated by a player is defined as the sum of the player's bets subtracted by the sum of their wins. A list of the variables used throughout this study, together with their description, may be found in table 1 below.

Table 1: List of Variables Used

Variable Name	Description	Categories
category_segment	the type of games the user mostly plays	Slots,Slots\&Table,Table, Table\&Slots, Other
segment_value	the customer segment of the user based on segmentation performed by the company	SVP (Super Value Player), HVP (High Value Player), LVP (Low Value Player), MVP (Medium Value Player), SVP MAX, NV (No Value)
season	the season in which the user is active	autumn,spring,summer,winter
age_group	the age group of the user	18-20, 21-30, 31-40, 41-50, 51-60, 61+
brand	the brand of the company which the user engages with	Brand 1, Brand 2,Brand 3, Brand 4, Brand 5
country_code	the country code of the user's nationality	DE (Germany), FI (Finland), NL (Netherlands), ROW (Rest of World)
first_accepted_deposit_amount	the amount in euro of the first succesful deposit made by the user	[1,25), [25,50), [50,999]
log_total_deposit_amount	the logarithm of the total deposit amount made by the user in thirty days	[0,2.18), [2.18,2.85), [2.85,5.41]
log_total_money_bet_amount	the logarithm of the total amount of money staked	[0,2.87), [2.87,3.71), [3.71,7.65]
ggr_to_deposits_ratio	the ratio of deposits to ggr	[-726,0.228), [0.228,1), [1,251]
avg_stake	the average stake of a player	[0,0.48), [0.48,1.41), [1.41,998]
log_total_deposit_count	the logarithm of the total amount of times the user made a deposit	[0,0.415), [0.415,1.28), [1.28,998]
active_days	the amount of days the player was active	[0,2), [2,8), [8,31]
depositing_days	the amount of days the player placed at least one deposit	[0,2), [2,6), [6,31]
depositing_days_last_7_days	the amount of days within the last week of the time frame that the user deposited at least once	[0,1), [1,3), [3,8]
log_avg_deposit_amount_per_active_day	the logarithm of the average deposit amount per active day	[0,1.61), [1.61,2.09), [2.09,4.78]
log_avg_stake_per_active_day	the logarithm of the average amount the user staked per active day	[0,2.32), [2.32,2.92), [2.92,6.4]
log_avg_deposit_count_per_active_day	the logarithm of the average amount of deposits the user made per active day	[0,0.301), [0.301,0.477), [0.477,1.84]
winsorized_game_win_margin	the ratio of the amount of money won to the amount of money staked by the user	[-1,0.0306), [0.0306,0.164), [0.164,1]
login_duration_hours_month	the amount of hours the user was active within the time frame	[0,4.4), [4.4,19.8), [19.8,476]
login_duration_hours_week	the amount of hours the user was active within the last seven days of the time frame	[0,2.17), [2.17,8.28), [8.28,189]
problem_gambler	indicates whether a user self-excluded or not	TRUE,FALSE

Goodness of Fit Measures
In our application, we are keen to predict whether a gambler may be classified as a problem gambler or not. The following basic definitions are fundamental to the majority of the goodness of fit measures we will be using.

- True Positives (TP): The number of players who were classified as problem gamblers and are in reality problem gamblers.
- True Negatives (TN): The number of players who are not problem gamblers and were correctly classified as such.
- False Positives (FP): The number of players that were classified as problem gamblers but are in fact not problem gamblers.

- False Negatives (FN): The number of players that were not classified as problem gamblers but are in fact problem gamblers.

These four categories may be placed in the so-called confusion matrix and is illustrated in table 2:

Table 2: Confusion Matrix

		Actual Classification	
		Not Problem Gambler	Problem Gambler
Predicted Classification	Not Problem Gambler	TN	FN
	Problem Gambler	FP	TP

The performance measures that we shall use to analyse the results are: Accuracy (A), Sensitivity (Se), Specificity (Sp), Precision (P), F1-score (F1) and Area under the Curve (AUC).

$$A = \frac{TN + TP}{TN + TP + FP + FN} \quad (7)$$

$$Se = \frac{TP}{TP + FN} \quad (8)$$

$$Sp = \frac{TN}{TP + FP} \quad (9)$$

$$P = \frac{TP}{TP + FP} \quad (10)$$

$$F1 = 2 \times \frac{P \times Se}{P + Se} \quad (11)$$

Note that some of the performance measures stated in equations (7) to (10) can be incorporated inside the AUC. The AUC will give us a number which idealy should be as close as possible to 1. The closer to 1 the better the accuracy. We will not go into the detail of how the AUC curve is constructed as this is beyond the scope of this paper.

Analysis Of Results

When applying the CL-adapt alogrithm to the above mentioned training and test sets respectively we obtained the following results:
The confusion matrices for the training and test data appear in tables 3 and 4 illustrated below:

Table 3: Confusion Matrix For Training Data by CL-Adapt

		Actual	
		Not Problem Gambler	Problem Gambler
Predicted	Not Problem Gambler	36373	14807
	Problem Gambler	12492	34269

Table 4: Confusion Matrix for Test Data by CL-Adapt

		Actual	
		Not Problem Gambler	Problem Gambler
Predicted	Not Problem Gambler	12192	4998
	Problem Gambler	4142	11315

The performance measures for both tables were computed and feature in table 5:

Table 5: Performance measures for CL-Adapt algorithm

	A	**Se**	**Sp**	**P**	**F1**	**AUC**
Train	0.721	0.698	0.744	0.733	0.715	0.780
Test	0.720	0.677	0.746	0.732	0.703	0.757

All performance measures on the training set and test sets are quite high indicating a good model fit. Also the values in the train set are quite close to the corresponding values in the test set. This indicates that the model is not overfitted. Moreover, the AUC obtained using this model is 0.78 for training ser and 0.76 for the test set which suggests that the classifier is far superior to random guessing.
When implementing the CL-adapt algorithm in R, the Bayesian network seen in figure 1 was returned. The variables present in the DAG are those which the algorithm deemed relevant in order to predict problem gambling.

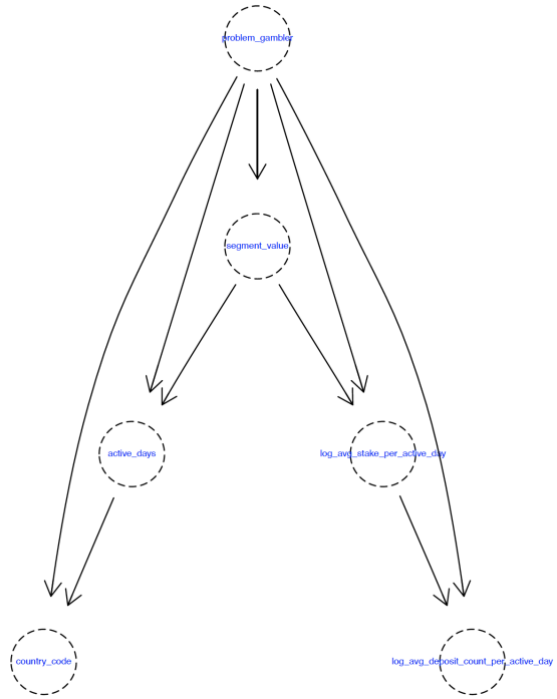

Figure 1: Bayesian Network Obtained From CL-Adapt

We next proceeded to apply the FSSJ algorithm on the same training and test sets. As before we show the confusion matrices for the training and test sets in tables 6 and 7 respectively.

Table 6: Confusion Matrix for Traning Data by FSSJ

		Actual	
		Not Problem Gambler	Problem Gambler
Predicted	Not Problem Gambler	38902	17119
	Problem Gambler	9963	31957

Table 7: Confusion Matrix for Test Data by FSSJ

		Actual	
		Not Problem Gambler	Problem Gambler
Predicted	Not Problem Gambler	13087	5767
	Problem Gambler	3247	10546

Table 8 below gives us all the performance measures for the traning and test sets. It is clear from the figures in the table

that once again the quality of the model is a good one and also it can be said that model is is not suffering from over fitting as the values in the traning set are very similar to the ones in the test set.

Table 8: Performance measures for FSSJ algorithm

	A	Se	Sp	P	F1	AUC
Train	0.724	0.651	0.796	0.759	0.702	0.832
Test	0.724	0.647	0.801	0.765	0.701	0.802

The DAG of the Bayesian Network returned by the FSSJ is illustrated in Figure 2 below:

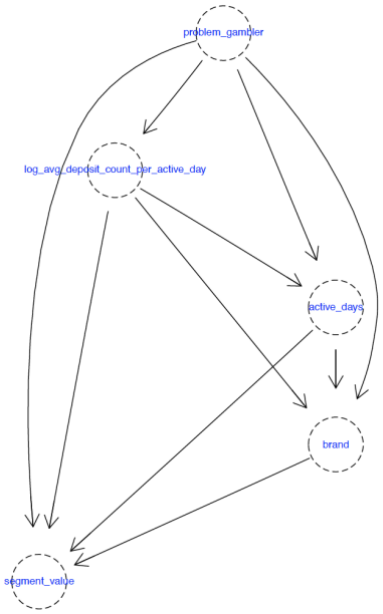

Figure 2: Bayesian Network Obtained From FSSJ

When comparing the variables that feature in figures 1 and 2, it is clear that most of the variables appear in both Bayesian Networks. However, when one compares the performance measures of the test set obtained from CL-adapt algorithm with the same measures obtained by the FSSJ algorithm it is evident that the latter is better for almost all performance measures. Finally, a binary logistic regression model was fit on the same train and test sets. Binary logistic regression is a well-known classification technique in statistics and it is intesting to compare its performance with the two Bayesian networks which were introduced, discussed and fitted to the data set at hand. The results of the Binary logistic regression model feature in the last column of table 9. It is clear that best results were provided by the FSSJ algorithm, followed by the CL-adapt algorithm.

Table 9: Comparison of three techniques

	A	Se	Sp	P	F1	AUC
CL-adapt	0.720	0.677	0.746	0.732	0.703	0.780
FSSJ	0.724	0.647	0.801	0.765	0.701	0.802
B-Log	0.686	0.701	0.672	0.681	0.691	0.757

CONCLUSION

This study explored the efficacy of using Bayesian networks to predict problem gambling by using self-exclusion as a proxy. We discussed how the structure of a BN classifier can be determined from a given data set using either the CL-adapt or the FSSJ algorithm. The methods mentioned above, as well as Logistic regression, were applied on a data set provided by a local gaming company. Taking all goodness of fit diagnostics into account it was evident that the FSSJ algorithm had the best overall performance. It would be useful for further studies to apply Random Forests as well as other classification techniques such as Support Vector Machines (SVM) to this data and compare the performace of these models with the results obtained in this paper.

Although past studies, some of which mentioned in this paper advocate for the use of self-exclusion as a proxy for problem gambling, this approach does have some inherent limitations as outlined by (Auer and Griffiths 2022). The authors raise the point that there are various reasons for self-exclusion. Consequently, the assistance of psychologists, psychiatrists and other relevant professionals could be utilised in order to determine whether each individual should be labelled as a problem gambler or not. This procedure is both time consuming and expensive and therefore was not considered in this paper.

REFERENCES

Auer M. and Griffiths, M.D. 2022. “Predicting Limit-Setting Behavior of Gamblers Using Machine Learning Algorithms: A Real-World Study of Norwegian Gamblers Using Account Data”. International Journal of Mental Health and Addiction. 20,771-788.

Akhter, S. 2018. “Using machine learning to predict potential online gambling addicts”. Master's thesis, Aalto University. School of Science.

Bermell, M. 2019. “Identification of Problem Gambling via Recurrent Neural Networks: Predicting self-exclusion due to problem gambling within the remote gambling sector by means of recurrent neural networks”.

Buttigieg, K. D.; Caruana, M. A.; Suda, D. 2022. “Identifying Problematic Gamblers using Multiclass and Two-stage Binary Neural Network Approaches”, International Conference on Agents and Artificial Intelligence (ICAART 2022), Vol. 3, 336-342.

Friedman, N.; Geiger, D. and Goldszmidt, M. 1997. “Bayesian Network Classifiers”. In Machine Learning, 29, 131-163.

Griffiths, M.; Wardle, H.; Orford, J.; Sproston K.; and Erens, B. 2009. “ociodemographic Correlates of Internet Gambling:Findings from the 2007 British Gambling Prevalence Survey”. CyberPsychology & Behaviour, 12, 2, 199-202.

Inglehart, D.L. and G.S. Shedler. 1983. "Simulation Output Analysis for Local Areas Computer Networks." Research Report RJ 4020 (45068), Research Division, IBM, San Jose, CA (Sep).

Loberman, H.; Weinberger, A. (October 1957). "Formal Procedures for connecting terminals with a minimum total wire length". Journal of the ACM. 4, 428–437.

McCormack, A.; Griffiths, M.; 2012. “Motivating and Inhibiting Factors in Online Gambling Behaviour: A Grounded Theory Study”. International Journal of Mental Health Addiction, 10, 39-53.

Minsky, M. 1961. "Steps toward artificial intelligence", Transactions on Institute of Radio Engineers. 49, 8-30.

Pazzani M. 1996. “Constructive induction of Cartesian product attributes”. Proceedings of the Information, Statistics and Induction in Science Conference. 66-77.

Percy, C.; Dragičević, S.; Franca, M.; D’Avila Gracez, A. 2016. “Predicting online gambling self-exclusion: An analysis of the performance of supervised machine learning models”. International Gambling Studies,16, 2, 163-210.

Petry, N. M. 2006. “Internet gambling: An emerging concern in family practice medicine?” Family Practice,23, 421-426