Chapter 7
Group Analysis Using Machine Learning Techniques

Ankit Sharma and Jaideep Srivastava

7.1 Machine Learning Techniques and Tools

Our aim in the following text is to provide a hands-on experience for group researchers to use machine learning and data-mining methods. Our main focus is to analyze and understand variables that may affect the group's performance. Keeping that in mind we shall illustrate the use of two machine learning and data-mining methods in a variety of combinations for group performance analysis. We employ an existing implementation of these methods in data-mining GUI based software named Weka (Hall et al., 2009). We shall also illustrate the process of moving from individual level variables to group level metrics in the Data Description Section. In the next subsections we describe the methods (Decision Trees and Feature Selection methods) and introduce the Weka tool.

7.1.1 Decision Trees

In machine learning, decision trees were first introduced by Quinlan (1986) in form of the ID3 algorithm. Later, Quinlan (1993) proposed the C4.5 algorithm to improve upon the limitation of ID3 algorithm. The major improvements upon ID3 are (1) C4.5 can handle both discrete as well as continuous data, (2) it can also handle missing data, and (3) C4.5 also does tree pruning. In the following chapter we shall be using the C4.5 algorithm for building the decision trees because of these reasons.

A. Sharma (✉) • J. Srivastava
University of Minnesota, Minneapolis, MN, USA
e-mail: sharm170@umn.edu; srivasta@cs.umn.edu

© Springer International Publishing AG 2017

A. Pilny, M.S. Poole (eds.), *Group Processes*, Computational Social Sciences,
DOI 10.1007/978-3-319-48941-4_7

Table 7.1 Training samples of 14 days with two features and dependent variable as team played or not that day

#	Outlook	Humidity	Play
1	Sunny	High	No
2	Sunny	High	No
3	Rainy	High	Yes
4	Rainy	High	Yes
5	Rainy	Normal	Yes
6	Rainy	Normal	Yes
7	Sunny	Normal	Yes
8	Sunny	High	No
9	Sunny	Normal	Yes
10	Rainy	Normal	Yes
11	Sunny	Normal	Yes
12	Sunny	High	Yes
13	Rainy	Normal	Yes
14	Rainy	High	No

Decision trees are supervised learning methods that make use of already classified training data to build predictive models. The aim of a decision tree classifier is to divide the training samples into partitions that are homogeneous with respect to the dependent variable (which in our analysis would be the group's performance). The algorithm outputs a model in the form of a tree where the bottom or end nodes (*leaves*) are the final predictions (or the classification class) and all the other nodes (*non-leaves*) represent some independent variables. During the construction of a tree, that independent variable is chosen as the node which splits its set of samples in the most homogeneous fashion i.e. each split is homogeneous with respect to the dependent variable. For this, the C4.5 algorithm employs a normalized information gain (Quinlan, 1993) as the criterion for variable selection and the variable with the highest normalized information gain (i.e., best predictor) is chosen as the node.

As an example we have 14 samples where each sample has a day's humidity and outlook and depending upon these variables if a group plays a cricket game or not, given in the Table 7.1. Using the C4.5 implementation in Weka software we achieve the decision tree shown in the Fig. 7.1b. If we look at the tree, the root is chosen as "humidity" by the algorithm and not the "outlook" variable. To understand this, if we try to split the days if the team will play or not, on the basis of the values of "outlook" and "humidity" variables individually, we get splits as shown in Fig. 7.1a. As we can see that if "humidity" variable is "normal" then we get a split of seven instance days on which the group always plays. In this sense, this split generated by "humidity" variable is pure i.e. all the instances are "yes" only. This purity is what we have been referring to as homogeneous split. Given that "humidity" is able to generate a more homogeneous split we say it is a more informative variable and thus, choose it over the "outlook" variable. Right now for illustration purposes we diagrammatically illustrated the splits and just by eye balling we can understand which split is homo-

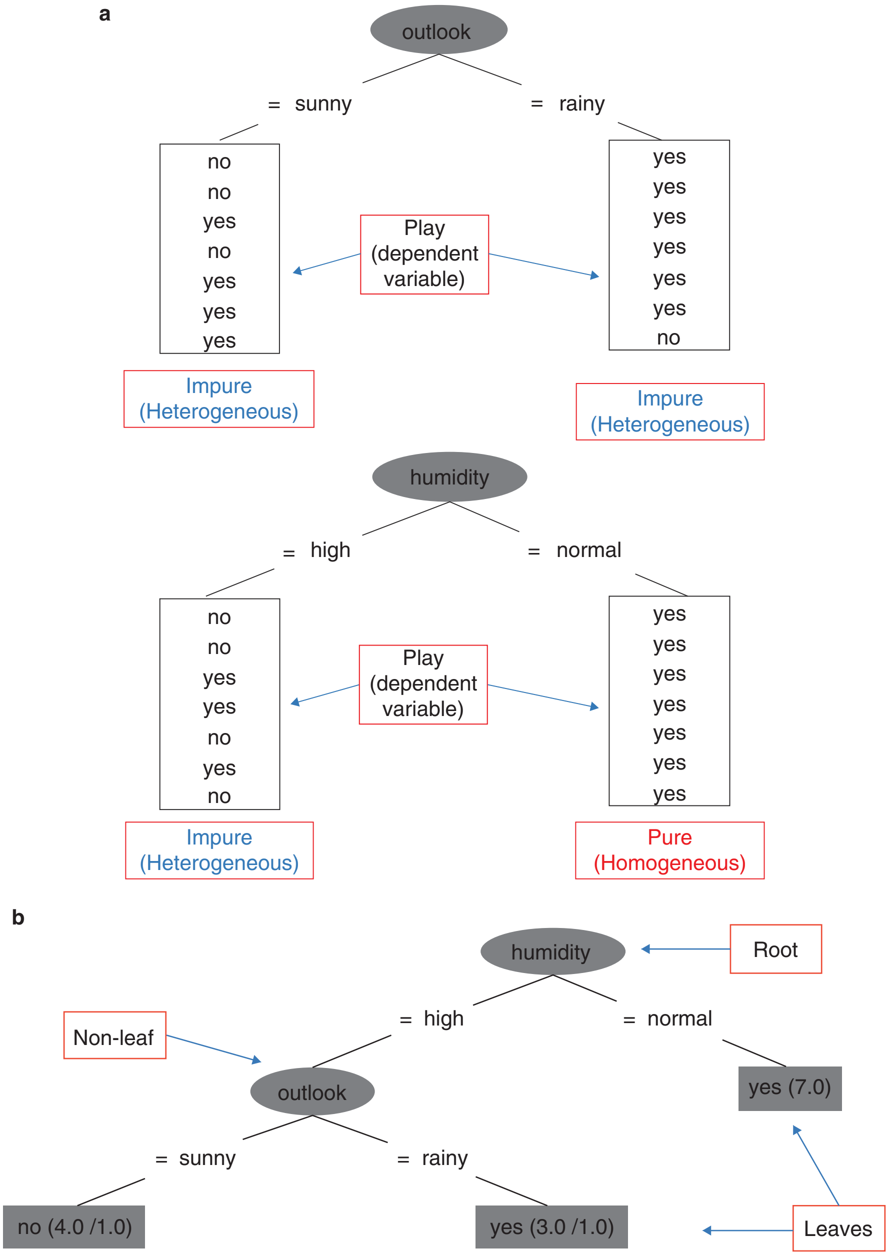

Fig. 7.1 (**a**) Splits generated by individual features. (**b**) Decision Tree classifying the samples from Table 7.1

geneous or more informative or not. However, this is impractical in practice and C4.5 employs an information theoretic measure of normalized information gain (Quinlan, 1993) as the criterion for variable selection. For further details of this measure we encourage readers to visit the Quinlan's text (Quinlan, 1993).

The biggest advantage of decision trees is that a single tree has the ability to describe the whole feature space. This ease of interpretability makes them quite popular among practitioners and therefore, we propose them for social scientists as a tool to understand the feature space pertaining to groups. We make use of an open source implementation of this algorithm available in the Weka software we use.

7.1.2 Feature Selection

Given the training samples, the aim of *feature selection* is to select a compact subset of independent variables that can predict the dependent variable without much loss of information. In other words, the purpose is the trim the dataset into a manageable one by focusing on independent variables that have high predictive power. Feature selection mines the most informative features and gets rid of the redundant or strongly correlated features. This process helps achieve a compact smaller set of features (i.e., parsimony) and therefore, improves model interpretability as well as training time and generalization by less over fitting (modal selection) (Guyon, Saffari, Dror, & Cawley, 2010). For a general overview of feature selection in machine learning we refer to (Guyon & Elisseeff, 2003) and the survey (Chandrashekar & Sahin, 2014).

Feature selection methods are mainly categorized into three types: (1) Filter, (2) Wrapper and (3) Embedded (Guyon et al., 2010). A subset of features can be judged as informative or not irrespective of how well they are able to predict the target or dependent variable. Algorithms that perform feature selection in this manner are called Filtering methods but as the selection is independent of the prediction accuracy, they usually may not perform optimally. Wrapper methods evaluate the model accuracy using a learning method for different subset of features and return the best performing feature subset. But the evaluation and search are done separately, making wrapper methods often computationally expensive. Embedded methods, on the contrary try to merge the subset search and evaluation phase, by incorporating the search within the machine learning model itself. Therefore, the information obtained while training the model are used to eliminate or retain features, all this done while model training itself.

In this paper we describe the application of a popular embedded method called SVM-RFE (Support Vector Machine based on recursive feature elimination) (Guyon Weston, Barnhill, & Vapnik, 2002). This algorithm reclusively learns SVM based model and eliminates independent variables or features with low weights. For further details of the algorithm we refer the reader to the original paper in (Guyon et al., 2002). We make use of the open-source implementation of SVM-RFE in Weka, which is called "SVMAttributeEval".

7.1.3 Introducing WEKA: GUI Based Machine Learning Tool

We conduct analysis using the tool called Waikato Environment for Knowledge Analysis (Weka), written in Java and developed at University of Waikato, New Zealand. This is a free software available for Windows, Linux as well as Macintosh environments at (Hall et al., 2009). The tool's website has link to numerous tutorials and they also have video based courses at YouTube. The best part of tool is the easy Graphical User Interface (GUI) which makes it very popular among data-mining and machine learning practitioners.

7.2 Dataset and Metrics

7.2.1 Dataset Collection and Description

The dataset was collected using a game based test-bed: SABRE - Situation Authorable Behavior Research Environment, developed by BBN Technologies, using the Bioware's Neverwinter Nights game and its provided toolset (Leung, Diller, & Ferguson, 2004). In this research we employ a NATO dataset collected using the game-based test-bed (SABRE) (Fig. 7.2). During the experiment 56 teams, of four members

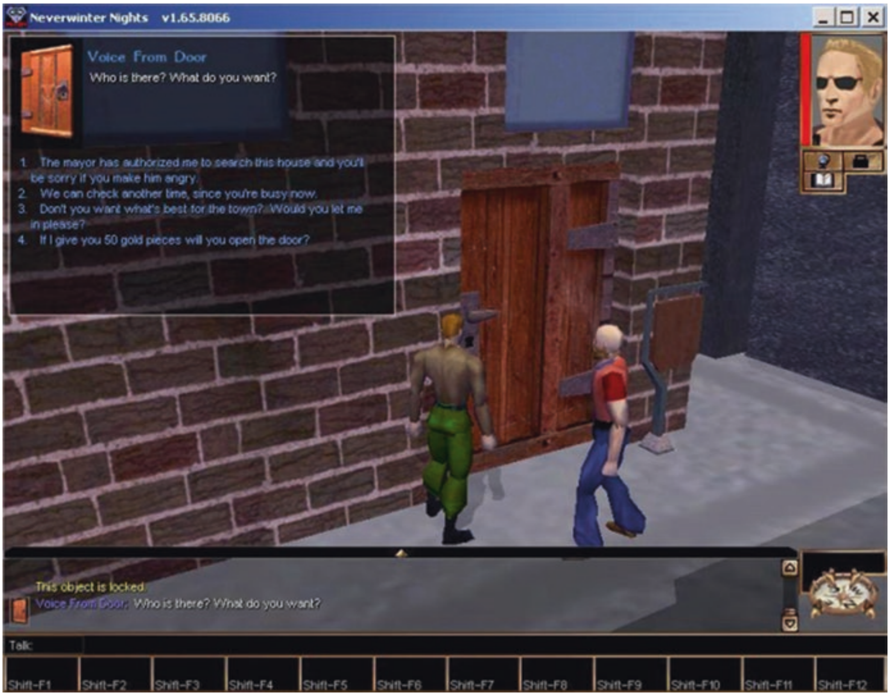

Fig. 7.2 A screenshot from the SABRE game based test-bed

each, were required to search for hidden weapons caches in an urban environment (town) while earning or loosing Goodwill points. Different amount of Goodwill points were earned depending on whether the weapons cache was found indoor or outdoor. Team also can lose points if for example they open a weapon-less container, etc. Players have a significant choice over the amount, timing, and type of interactions like chatting to specific individuals or broadcast, communication using structured formats using the journal-management or map-marking tools provided to the members. There were several phases in the game starting with Survey, followed by Training and Planning phases and finally, the Executing phase. It is the Execution phase, 1 h in length, where the four member teams search for the weapons and earn good will points.

7.2.2 Individual Level Metrics

In our analysis we develop two types of Individual Level Metrics from the SABRE dataset. The first are the **Role** type metrics. These are based upon the kind of role the individual is playing within the team. There are a total of seven Role Metrics for each individual member of a team:

1. Number of Tips from NPC (Non-Player Character--automated in the game)
2. Number of Conversations initiated with NPC
3. Number of Chats Sent
4. Number of Chats Received
5. Number of Buildings Entered
6. Number of Tips Sent
7. Number of Tips Received

These metrics try to quantify the Role an individual is playing within the team while keeping track of the various actions he or she performs or his/her in-game dynamics.

The second type of metrics are the **Skill** type metrics which reflect upon the skill of a team member. These were ascertained via a pre-game survey filled by each of the members for all the teams. In all we have 18 different kinds of Skill-type individual metrics (Table 7.2).

7.2.3 Constructing Group Level Metrics (Control Variables) from Individual Metrics

We now develop group or team level metrics using the two types of Individual Metrics discussed in the previous subsection. We construct the group level metrics by aggregating the individual level metrics for all the four individuals in each group. We aggregate in two ways to get two kinds of group level metrics. For the first kind, we take sum of values of an individual metric for all team members and we refer to these as the "TOTAL" group metrics. The second group metric is attained by taking

Table 7.2 List of skill type individual metrics with their type and range

Member Skill	Type	Value
English native	Yes or no	{1,2}
English ability	4 level choices	{1,2,3,4}
Stress in English environment	4 level choices	{1,2,3,4}
Reserve for English view	5 level choices	{1,2,3,4,5}
Computer expertise	3 level choices	{1,2,3}
Own computer	Yes or no	{1,2}
Email usage	5 level choices	{1,2,3,4,5}
Browser usage	5 level choices	{1,2,3,4,5}
Teleconference usage	5 level choices	{1,2,3,4,5}
Chat usage	5 level choices	{1,2,3,4,5}
Net-meeting usage	5 level choices	{1,2,3,4,5}
Own game console	4 category choices	{1,2,3,4}
Comp games time spent	Number of hours	Real
Multiplayer comp game	Yes or no	{1,2}
*Neverwinter Night*s	Yes or no	{1,2}
Comp game names	Yes or no	{1,2}
Game mods	Yes or no	{1,2}
Game list	Yes or no	{1,2}

into consideration the heterogeneity among the group members with respect to a given individual metric. We quantify this heterogeneity by employing the concept of Information Entropy (Teachman, 1980). We define the Information Entropy for a group of four members for a given individual metric "*x*" as:

$$H(x) = -\sum_{n=1}^{4}\left(p_n \log_2 p_n\right) \tag{7.1}$$

where

$$p_n = \frac{x_n}{\sum_{n=1}^{4}\left(x_n\right)} \tag{7.2}$$

is the fractional contribution of the member n for individual metric x and x_n is the value of the individual metric x for the member n of the group. As there are only four members in each group we have H in the range $[0, 2]$. The higher the entropy, the lower the heterogeneity. Table 7.3, illustrates the values for the values attained by "TOTAL" and "ENTROPY" metrics for some example values of the "Tips Sent" individual metric i.e. x = "Tips Sent".

Tables 7.4 and 7.5, show the Group Level Metrics corresponding to the Role and Skill Type Individual Metrics, respectively, along with their mean values across all the 56 Groups in the SABRE dataset.

Table 7.3 Four example teams with different kinds of variety with respect to tips sending behavior. Tips Sent Entropy and Total metrics are also shown

Attribute: tips sent					
Member 1	Member 2	Member 3	Member 4	Entropy metric	Total metric
1 (p1 = 1/8)	0 (p2 = 0/8)	1 (p3 = 1/8)	6 (p3 = 6/8)	1.06	8
6	6	5	6	1.99	23
0	0	0	1	0	1
6	6	6	6	2	24

Table 7.4 List of all the group level role type metrics along with their mean values across groups

Total role metric	Mean value	Entropy role metric	Mean value
Tips_from_NPC_Total	17.625	Tips_from_NPC_Entropy	1.770445
NPC_Interacted_Total	85.98214	NPC_Interacted_Entropy	1.711167
Chats_Received_Total	657.1607	Chats_Received_Entropy	1.982355
Chats_Sent_Total	657.1607	Chats_Sent_Entropy	1.87555
Buildings_Entered_Total	61.33929	Buildings_Entered_Entropy	1.847229
Tips_Received_Total	23.96429	Tips_Received_Entropy	1.492368
Tips_Sent_Total	23.96429	Tips_Sent_Entropy	1.537586
Total_Mean_Total	218.1709	Total_Mean_Entropy	1.7452

Furthermore, we also have information per team regarding the type of configuration they adopted while playing the game. There are five group configurations as follows:

1. {1-1-1-1}: All working separate.
2. {1-1-2}: Two working together and the other two separately.
3. {1-3}: One working separately and three together.
4. {2-2}: Working in groups of two.
5. {4}: All working together.

Corresponding to the above five group configurations we have define five TOTAL Group Level Metrics:

1. Group_Conf_1-1-1-1_Total: Percentage of time spent in configuration {1-1-1-1} configuration
2. Group_Conf_1-1-2_Total: Percentage of time spent in configuration in {1-1-2} configuration
3. Group_Conf_1-3_Total: Percentage of time spent in configuration in {1-3} configuration
4. Group_Conf_2-2_Total: Percentage of time spent in configuration in {2-2} configuration
5. Group_Conf_4_Total: Percentage of time spent in configuration in {4} configuration

Table 7.5 List of all the group level skill type metrics along with their mean values across groups

TOTAL skill metrics	Mean	Min	Max	ENTROPY skill metrics	Mean	Min	Max
English_Native_Total	**4.678571**	**4**	**8**	English_Native_Entropy	**1.986054**	**1.9219**	**2**
English_Ability_Total	**11**	**7**	**16**	English_Ability_Entropy	**1.967525**	**1.8911**	**2**
Stress_in_English_Total	**9.642857**	**6**	**14**	Stress_in_English_Entropy	**1.941323**	**1.5305**	**2**
Reserve_for_English_View_Total	**13.14286**	**10**	**18**	Reserve_for_English_View_Entropy	**1.962373**	**1.8352**	**2**
Game_Mods_Total	**4.142857**	**4**	**5**	Game_Mods_Entropy	**1.988843**	**1.9219**	**2**
Comp_Game_Names_Total	**3.178571**	**1**	**4**	Comp_Game_Names_Entropy	**1.612232**	**0**	**2**
Neverwinter_Nights_Total	**4.285714**	**4**	**6**	Neverwinter_Nights_Entropy	**1.980346**	**1.9183**	**2**
Multiplayer_Comp_Game_Total	**7.655357**	**0.2**	**27.2**	Multiplayer_Comp_Game_Entropy	**1.124491**	**0.10125**	**2**
Game_List_Total	**4.232143**	**4**	**6**	Game_List_Entropy	**1.9832**	**1.9183**	**2**
Email_Usage_Total	**19.44643**	**16**	**20**	Email_Usage_Entropy	**1.993227**	**1.8232**	**2**
Browser_Usage_Total	**19.08929**	**13**	**20**	Browser_Usage_Entropy	**1.980116**	**1.8232**	**2**
Teleconference_Usage_Total	**8.267857**	**4**	**15**	Teleconference_Usage_Entropy	**1.854605**	**1.65**	**2**
Chat_Usage_Total	**13.94643**	**5**	**20**	Chat_Usage_Entropy	**1.876964**	**1.65**	**2**
Netmeeting_Usage_Total	**6.482143**	**4**	**13**	Netmeeting_Usage_Entropy	**1.858314**	**1.5488**	**2**
Own_Game_Console_Total	**6.071429**	**4**	**11**	Own_Game_Console_Entropy	**1.928354**	**1.75**	**2**
Comp_Games_Time_Spent_Total	**15.54179**	**0.04**	**62.04**	Comp_Games_Time_Spent_Entropy	**1.147974**	**0.029089**	**2**
Computer_Expertise_Total	**9.482143**	**6**	**12**	Computer_Expertise_Entropy	**1.971993**	**1.8911**	**2**
Own_Computer_Total	**3.946429**	**3**	**4**	Own_Computer_Entropy	**1.977768**	**1.585**	**2**

We also define one ENTROPY metric for group configuration which captures the diversity in group configuration over time. We refer to it as, “Group_Conf_Entropy”.

7.2.4 Group Performance (Dependent Variables)

As the teams search for weapons they earn or lose goodwill points. We define Performance of a team as the Net Change in number of goodwill points earned by each team. The histogram of team performance is shown in Fig. 7.3. The middle of the three red vertical lines is the mean performance (**840.71**) and the other two denote the top and bottom 25 % performance cutoff for teams. We use these cutoffs to define three categories (0, 1 and 2) of team as follows:

Category 0—Low Performing teams (bottom 25 %): Net Goodwill points ≤500.
Category 1—Medium Performing teams: 500 < Net Goodwill points < 1150.
Category 2—High Performing teams (top 25 %): Net Goodwill points ≥ 1150.

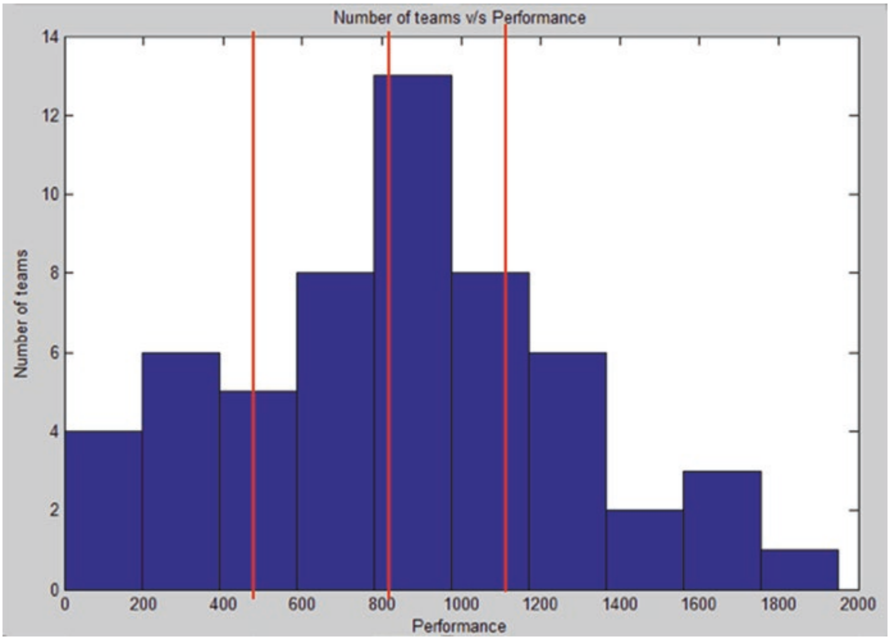

Fig. 7.3 Histogram of the group performance of 56 groups in SABRE dataset

7.3 Experimentation Methodology

Our experiments involve the application of machine learning methodologies described in Sect. 7.1 to perform group analysis of teams in the SABRE dataset. We divide the experiments into two types of major levels (see Fig. 7.4). First, is the Micro-Level analysis where we perform the group analysis using a single type of group metrics (variables). As we have three types (Role, Skill & Group Configuration) of group-level metrics, the Micro-Level contains three experiments where we only consider attributes from within each of these three types. Second, we have Macro-Level analysis where we consider all the three type of metrics simultaneously. Within the Macro-level we consider all the three metrics together.

As the reader can observe each of the just described experiments different in the type of group attributes employed for analysis. Each of these experiments is conducted in four phases (see Fig. 7.5). Each phase helps us understand, from a variety of perspectives, including insights from their attributes (features), their relationships, and their effects on the group performance. We start with simple correlation

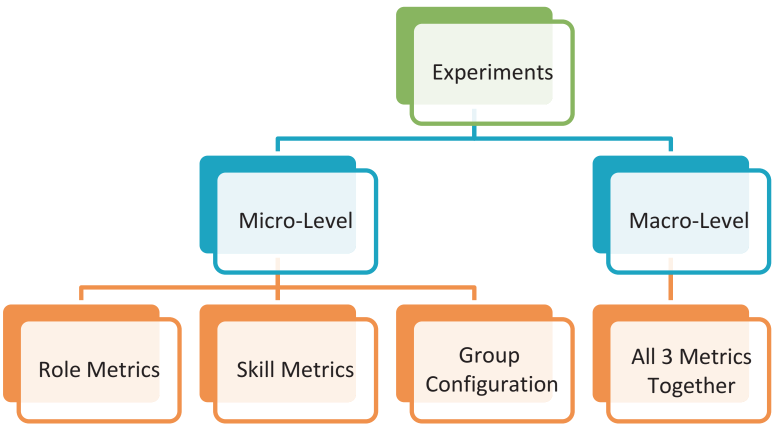

Fig. 7.4 Segregation of the different types of analysis conducted

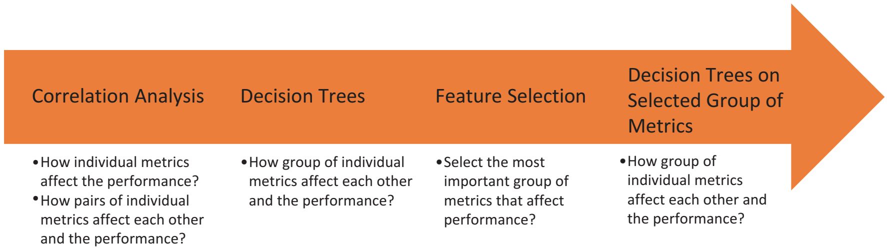

Fig. 7.5 Diagram showing the various analysis phases along with their purposes

analysis to find pair-wise dependence between all variables, both within and between each dependent and independent variable. This is followed by a decision tree, which explicitly highlights the patterns of relationships between different variables that may affect group performance. We perform feature selection next in order to focus on the dominating or most explanatory variables and discuss why the selected features can possibly be relevant. Finally, we again perform decision tree analysis using on the selected features from the previous phase and hope to find more strong and interesting patterns. We overall, therefore, have four sets of experiments and in each experiment we analyze groups from a series of four phases as we just described. Also within each of the four sets we consider both the TOTAL and ENTROPY variants of the group metrics.

7.4 Experiment 1: Group Analysis Using Role Based Metrics

7.4.1 Phase 1: Correlation Analysis

Table 7.6 shows the correlations with group performance among the different independent variables. The total amount of Tips Sent (total metric correlation of 0.43) and entropy of Tips Sent (entropy correlation of 0.30) were both significantly correlated. There was also a negative correlation with entropy regarding the number of buildings entered (negative entropy correlation of −0.22). Overall, it also seems that the TOTAL metrics are more related in general to the performance rather than the ENTROPY metrics.

The correlations between total group level metrics suggest some interesting and explainable dependencies (Table 7.7). For example, the more a team interacts with the NPCs the more likely the team gets more tips from them (correlation of 0.646). Also, as one of the team member gets tips from NPCs he or she is likely to forward them to other members, therefore, increasing the total tips flux within the group (observe the correlation 0.30).

Table 7.6 Correlation between independent variables and performance (dependent variable)

Total role metric	Correlation score	Entropy role metric	Correlation score
Tips from NPC	0.383145	Tips from NPC	−0.009487
NPC interacted	0.310966	NPC interacted	0.104403
Chats received	0.269815	Chats received	−0.102956
Chats sent	0.269815	Chats sent	**−0.221359**
Buildings entered	0.279464	Buildings entered	−0.221359
Tips received	0.430349	Tips received	0.081854
Tips sent	**0.430349**	Tips sent	**0.300998**
Total mean	0.339129	Total mean	0.0066

Table 7.7 Pair-wise correlation between total role type group metrics

Correlation	Got_Tip_from_NPC_Total	NPC_interacted_Total	Chats_Received_Total	Chats_Sent_Total	Buildings_Entered_Total	Tips_Received_Total	Tips_Sent_Total
Got_Tip_from_NPC_Total	1.000						
NPC_interacted_Total	**0.646**	1.000					
Chats_Received_Total	0.093	0.107	1.000				
Chats_Sent_Total	0.093	0.107	1.000	1.000			
Buildings_Entered_Total	0.187	0.128	−0.022	−0.022	1.000		
Tips_Received_Total	**0.300**	0.229	0.135	0.135	0.101	1.000	
Tips_Sent_Total	**0.300**	0.229	0.135	0.135	0.101	1.000	1.000

Let's focus now on the Entropy metrics and their pair-wise correlations, as depicted in Table 7.8. High Entropy for a given variable indicates that team members behave similarly with respect to that variable and Low Entropy indicates that there is a large variation among the team members for the given variable. Now we see a pretty high correlation between the entropies of interactions initiated with NPCs and the tips received by NPCs (correlation 0.595). This may make sense because if everyone initiates a conversation with NPCs (high entropy of initiation) everyone is likely to get a tip (high entropy of tips from NPC). Similarly, if only a few interact with NPCs (low entropy for initiation) only those few would receive tips from NPCs (low entropy). Although, this argument is straight forward, the point we wish to highlight is that this reasoning is not possible without a team diversity metric like entropy.

Further more interesting would be to utilize the correlation between the entropy metrics and the total metrics as shown in Table 7.9. For example, we observe a negative correlation between Chats received as well as the Chats sent entropy and the total amount of buildings entered by the team. A possible explanation would be that team is busy in chatting and therefore, fail to enter several buildings. Also chat-receiving entropy is negatively correlated with the total amount of tips received from NPC (correlation –0.226). This suggests that possibly a few team members are busy getting tips from NPC (making high total NPC tips for team) and these members are not receiving much chats, as compared to other members (low entropy), because they are busy interacting with NPCs.

7.4.2 *Phase 2: Decision Tree Analysis*

Weka was employed for Decision Tree Analysis using the J48 Decision Tree implementation provided in the software. To give a more hands-on experience, Fig. 7.6 shows the "Preprocess tab" when we load the data (only the Role type group metrics) in the Weka software.

In order to perform decision tree analysis we move to the "Classify" tab (see Fig. 7.7) and choose using the "Choose" button the J48 (which can be found under weka >classifier –> trees) classifier. Run the classifier using the "Start" button on the left after choosing the "Use training set" option under the "Test options".

At this point, we would again highlight here that our major focus in these experiments is not to build strong predictive models where the only concern is to improve the prediction accuracy over the unseen examples as a test set. Contrary to this, our main focus is to perform feature space analysis which involves objectives like reducing the number of independent variables to a manageable set. Furthermore, we would like to understand how the various features interact and which are the most important features that can help us understand the given data samples sufficiently well, rather than the generalization power of model to unknown test samples.

In other words, we are satisfied if our model fits the training data sufficiently well and focus on interpretation of feature space. For this reason, we choose the "Use training set" option under the "Test options" on the left. This tells Weka to evaluate the accuracy of the learnt model on the training data itself.

Table 7.8 Pair-wise correlation between role type entropy group metrics

Correlation	NPC_tips_ Entropy	NPC_initiated_ Entropy	Chat_Received_ Entropy	Chat_Sent_ Entropy	Buldings_Entered_ Entropy	Tips_Revc_ Entropy	Tips_Sent_ Entropy
NPC_tips_Entropy	1.000						
NPC_initiated_Entropy	**0.595**	1.000					
Chat_Received_Entropy	0.324	0.114	1.000				
Chat_Sent_Entropy	0.031	−0.010	0.196	1.000			
Buldings_Entered_Entropy	0.072	−0.075	0.064	0.004	1.000		
Tips_Revc_Entropy	−0.142	−0.140	−0.098	−0.001	0.099	1.000	
Tips_Sent_Entropy	−0.076	0.051	−0.024	0.012	−0.141	0.346	1.000

Table 7.9 Correlation between role type entropy v/s total group metrics

Entropy v/s Total	Got_Tip_from_ NPC_Total	NPC_interacted_ Total	Chats_ Received_Total	Chats_ Sent_Total	Buildings_ Entered_Total	Tips_Received_ Total	Tips_Sent_ Total
NPC_tips_Entropy	−0.090	0.046	0.064	0.064	0.058	−0.097	−0.097
NPC_initiated_Entropy	−0.022	0.080	0.152	0.152	0.104	−0.063	−0.063
Chat_Received_Entropy	**−0.226**	−0.049	0.159	0.159	**−0.293**	−0.118	−0.118
Chat_Sent_Entropy	−0.018	0.014	0.324	0.324	**−0.197**	0.248	0.248
Buldings_Entered_Entropy	−0.226	−0.123	−0.068	−0.068	0.155	0.109	0.109
Tips_Revc_Entropy	−0.138	−0.116	−0.237	−0.237	0.158	0.369	0.369
Tips_Sent_Entropy	0.184	−0.031	0.300	0.300	0.005	0.567	0.567

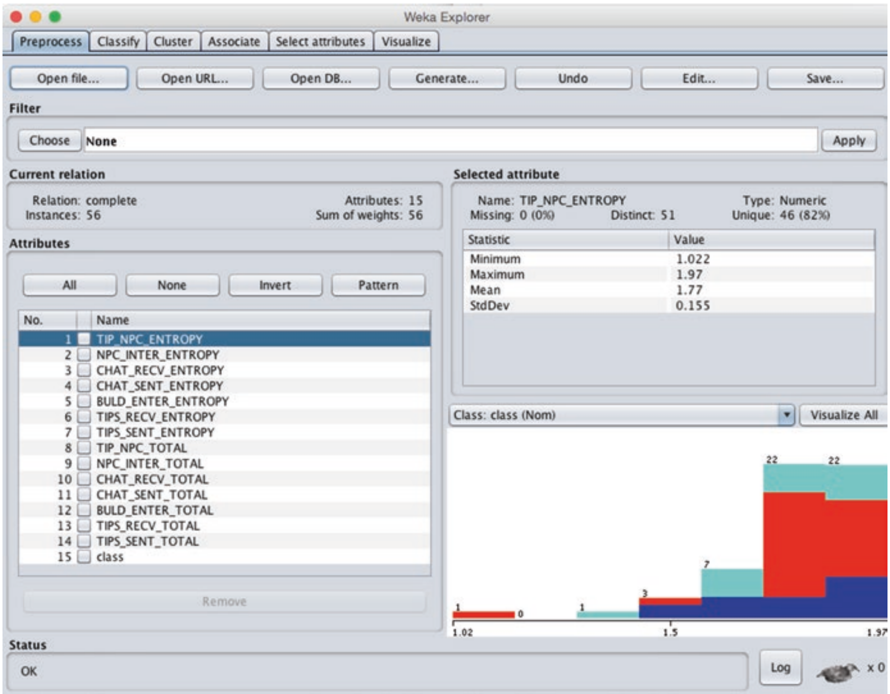

Fig. 7.6 Preprocess tab in Weka

After running the analysis, the output screen on right shows the results, as shown in Fig. 7.7. As we can observe the decision tree fits the 56 group samples fairly well. In order to visualize the tree, right click on the Result list at the bottom left and choose "Visualize Tree" option. Figure 7.8 shows the resultant tree for the total as well as entropy type group metrics together. Recall we had mentioned that we divide the teams in three categories: low (0), medium (1) and high (2), based upon their performance. Our aim in the Decision Tree analysis is to find those path ways or relationships between different variables starting from the *top* of tree that take us to high performing (labeled 2) *leaves* i.e. bottom-most nodes (dependent variable) in the tree. This helps us better understand the relationship in a visual fashion. Note the format of the leaves in the decision tree is of type x(y/z) where x is the class label (0: low, medium:1 or 2:high), y is the number of samples or instances correctly classified and z is the number of samples incorrectly classified. We would like to have the fraction (y/z) as high as possible for a reliable decision on the leaf node.

We observe in Fig. 7.8, that sub-tree to the right of the nodes: TIPS_RECV_TOTAL and TIPS_SENT_ENTROPY, contains mostly medium and high performing leaves. Therefore, higher tips circulated within the team and higher tips sent entropy are all related to team performance according to the model (i.e. everyone sending tips results in good team performance).

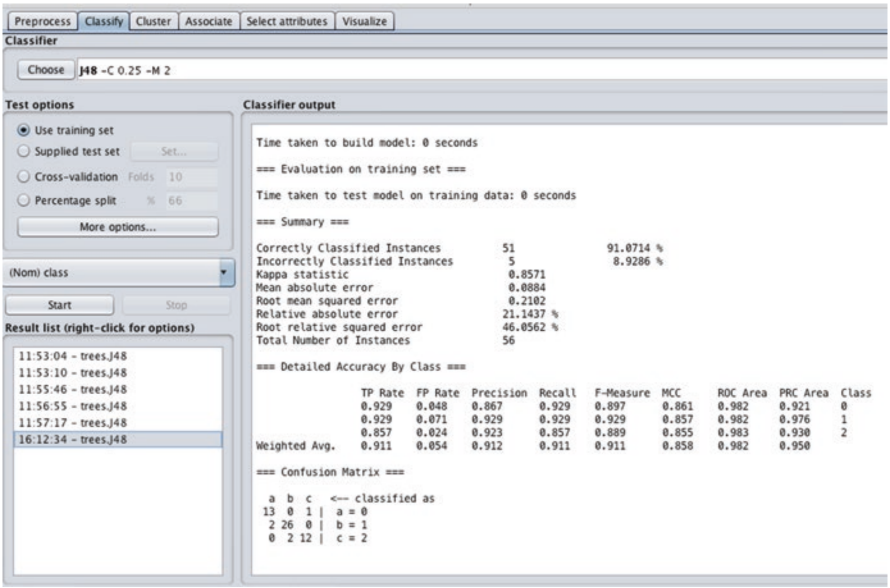

Fig. 7.7 Full role metric model fit statistics

Also, we observe that the high performing teams (leaves with label ‘2’) are either in the right sub-trees of TIPS_SENT_ENTROPY node or of the NPC_INTER_TOTAL node. But we can notice that even after this, if a group falls in the left sub-tree of NPC_INTER_TOTAL (≤91) node, it is still predicted to have medium performance by having sufficiently high (>18) total tips from NPCs (i.e. right of TIP_NPC_TOTAL is a medium (‘1’) leaf). This reflects the importance of tips from NPCs. However, we also find three high performing groups (leaf labeled ‘2 (3.0)’) to the left of TIP_NPC_ENTROPY. This means that if the tips receiving entropy of the group is less than 1.7 it is predicted to be high performing. For a four member team this typically should mean that only one or two members should be receiving those tips from NPCs. Readers are encouraged to see Table 7.3 to get a sense of the range of entropy and the type of values assumed by team members for a metric.

Note that we chose the minimum number of classified instances as two using the “−M” option for our classifier as “**J48** –C 0.25 –M 2” (see top of Fig. 7.7). This means that our decision tree will assign a new variable node even if the instances it is able to split are as low two. Therefore, if the leaf format in the visualized tree is x(y/z) then y ≥ m if we select option “−M m”. In our case we observe this limit in the leaf “0(2.0/1.0)” where y = 2 as we chose m = 2. Notice that as we increase ‘m’, the misclassification instances i.e. z will also increase. We however, leverage the small size of our data to completely interpret our data by generating a tree node even if it is able to classify as low as two instances only.

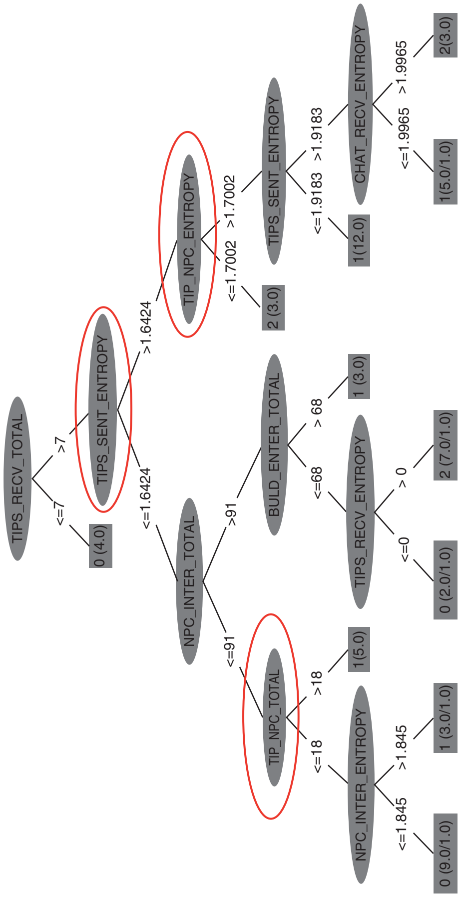

Fig. 7.8 Full role metric model decision tree

Adding to discussion on generalizability of the models we are using, we would also bring into notice that our dataset size is quite small and sparse with the sample size being the same as the number of dimension (56 sample size and 56 metrics with total and entropy types combined). Therefore, the generalizability and prediction on out of sample test cases for our models is not high. But they very well explain the training samples and how features affect the given data. We have chosen this smaller dataset in order to illustrate how beautifully we can zoom into the feature space. Our focus is therefore, how well can the group features explain the data samples. So in some sense we are fitting the machine learning model to the training set and care less about the prediction capability. If we have a larger dataset we can have more generalizability and less prediction error on testing this set as well.

7.4.3 Phase 3: Feature Selection

In the previous two sections, our analysis consisted of all the 16 available metrics of the role type. However, not all the metrics might be that relevant for a performance analysis of the teams. In machine learning, a subset of the most important variables and rank among them is done using feature selection methods (Guyon et al., 2010). Although there are a variety of feature selection methods, we will focus on of the powerful SVM classification based embedded method (Guyon et al., 2002) discussed earlier. In the Weka software this SVM based method is implemented under the name "SVMAttributeEval" in the Attribute Evaluators which is under "Select Attributes" tab (see Fig. 7.9). There are several options within SVMAttributeEval that we can play with, but for this illustration we restrict to the default options. Note, "attribute evaluator" scores the worth a subset of features and "search method" determines what kind of search is performed. We encourage readers to try different kinds feature selection methods.

After pressing the "start" button, the method returns a ranked list of all the attributes as per their relevance (as can be observed in the Attribute Selection output on the right). SVM-RFE algorithm implemented within "SVMAttributeEval" eliminates as well as rank the features iteratively. In each iteration the features are eliminated if required and are ranked as per their performance classification accuracy over the training set when used within the SVM classifier. We observe in the selected features, similar to the decision tree analysis in the previous section, that the tips exchanging behavior of members, captured in TIPS_SENT_ENTROPY and TIPS_SENT_TOTAL metrics, plays an important role in deciding team success. Furthermore, unlike any of the previous analysis, feature selection also indicates that chatting behavior of members also affects the performance.

Recall we are only concerned with the accuracy of the model on the training set and therefore, we choose the "Use training set" option under "Test options" on the left. If we have a larger sample size, then we can go for cross-validation as well. In fact, for our data, both for decision trees as well as feature selection, there was almost no difference between the models built using training set (with low error)

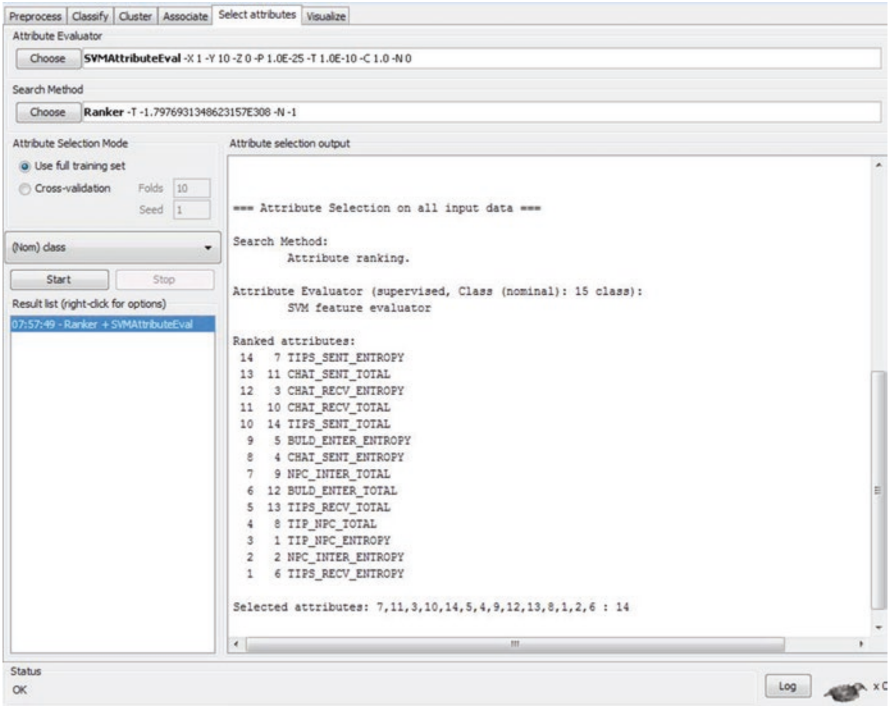

Fig. 7.9 Ranked attributes for role model using SVM

and via cross-validation (with less accuracy). This further confirms that our generalizability is restricted by lack of enough data samples. We therefore, focus on training set performance only.

7.4.4 Phase 4: Decision Tree Analysis over Selected Features

Notice that decision trees, as we saw in Phase 2, can tell us exactly whether it was the low or high value of a variable and in what context of other variable's values, affects group performance. This is in contrast to the black box approach of feature selection in Phase 3, which gives a list of highly important variables, but there is no way to ascertain what kind of values of these selected features affect the performance in what way.

In this phase we try to combine the best of both worlds. We use the top five highly ranked features, which in our case are the group level role metrics. In this way we leverage the ranking information from feature selection to lower the size of feature set from 16 to the five most important ones. We then build decision trees using only the top five role metrics just selected.

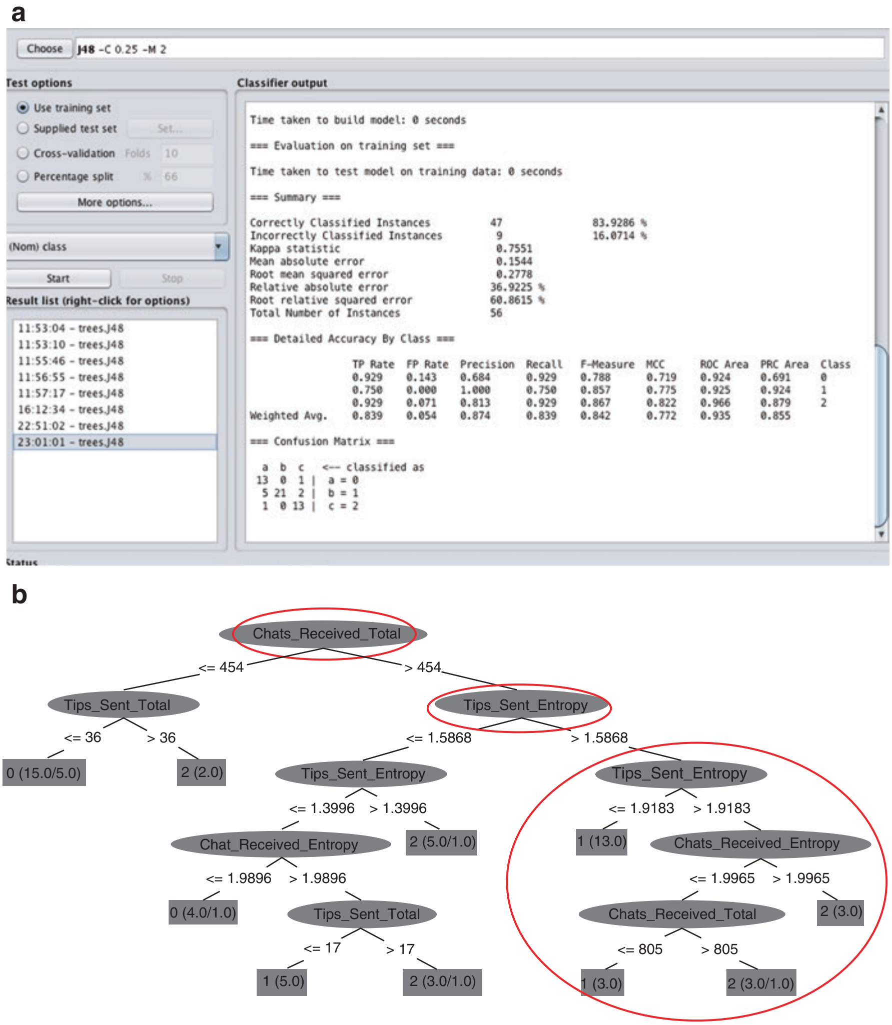

Fig. 7.10 (**a**) Reduced role metric model fit statistics. (**b**) Reduced role metric model decision tree

Before we go ahead with analysis of decision tree, we would like to comment on how to choose number of top ranked features. This choice is more of an art, especially if our focus is on feature space interpretation. Now as we increase the K top attributes, the training error on the decision tree built on it decreases. On the other hand, the number of attributes increases, making the tree possibly cumbersome to analyze. However, the latter is not always the case. Therefore, it becomes more of a subjective choice of K, which gives an interestingly interpretable decision tree and might have a sufficiently low training error as well.

For our choice of top five, the resultant tree is shown in Fig. 7.10b above and the model fit on training data is shown in Fig. 7.10a. As we can see that we now have a tree of just four metrics out of the 5 previously selected in Phase 3. This tree is sufficiently detailed and precisely tells us which kind of groups fall in intersection of which values of just these four group metrics. The big marked circle on the right contains a sub-tree whose leaves are either medium or high performing, implying that if a team falls in this sub-tree it is highly probable that it would perform well (at least medium if not high). In order to fall in this sub-tree, the team members should be chatting a lot and should have a similar tip sending behavior among the members (see the nodes in the two small circles).

Also if we observe the root node (CHAT_RECV_TOTAL), the left of root occurs if a group is chatting quite a bit less (<454). This value is significantly lower than the mean total chat across groups (see Table 7.4). If group members chat less and do not also send tips much, i.e. fall on left of TIPS_SENT_TOTAL node (left of root node), this group is more likely to perform low. As we can see the label of the leaf to the left of TIPS_SENT_TOTAL as “0(15.0/5.0)”. There are 15 low performing groups out of the total 19 low performing groups that were predicted to fall in this leaf. Now, on the other hand, notice if we concentrate on the right of TIPS_SENT_TOTAL. This happens if a low chatting group has significantly high (>36) tips circulated in group. Note the mean TIPS_SENT_TOTAL, from Table 7.4, is approximately equal to 24. So what this tells is that even a very low chatting group, if its members are circulating large volume of tips (> 36 much greater than mean of 24), it is predicted to perform well. As the label leaf on the right of TIPS_SENT_TOTAL says there are only two such cases seen so far i.e. “2 (2.0)”. Therefore, although such events are possible, they are very unlikely. So it is best for the team members to chat more (i.e., over 454).

Also if we observe the two TIPS_SENT_TOTAL nodes (one on top left and one bottom of the tree), we realize that higher total sent tips results into high performance even if the team is chatting less and has less tip sending entropy (i.e. only a few members send a large number of tips). As such, this indicates that high tip sending behavior may be favorable in the absence of chat receiving.

Summarizing this example, we found through four different types of analysis that for good performance, everyone in team should be communicating via both chatting as well as exchanging tips, but only a few members should be receiving a lots of tips from NPC and entering buildings.

7.5 Experiment 2: Group Analysis Using Skill Based Metrics

7.5.1 Phase 1: Correlation Analysis

We shall proceed for the group analysis using Skill metrics in a fashion similar to Role metrics performed in the last experiment. The Skill metric largely refers to the diversity of skills that make up each team and can be important regarding the assembly of teams. However, this time we assume that, with the detailed description in previous example, the reader is acquainted with the interpretation of entropy metrics as a variety quantifier. Firstly, we will see the correlation with the performance variable on different independent variables (both total and entropy metrics for all the skill type variables) as shown in Table 7.10. All the interesting correlations are highlighted using bold font. Overall, total English and Computer expertise as well as Native English speaking ability in the team are good predictors (positive corr. = 0.396 between total English native speaking ability of team with performance) of group performance. However, only "few" Native English speakers are better (negative corr. = −0.186 between Native English speaking Entropy and Performance). Teams having the most members with knowledge of Computer (positive corr. = **0.408** between Computer Expertise and performance) and spending time on Computer games (positive corr. = 0.334 between Comp. Games Time spent and performance) had a positive relationship with team performance.

The correlations between Entropy and Total skill metrics are shown in Table 7.11. The diagonal of this table is pretty important and interesting. All the interesting correlations are highlighted with bold font in Table 7.11. If a particular diagonal element is highly positive, it implies that the variable representing this row/column is high for all the individuals (high entropy) if total sum of all the team members for this variable is high (high total). On contrary if this diagonal element is highly negative, then it suggests that when the total group metric for this variable is high (high total metric), then only few (possible 1 or 2 in our four team member case) members are responsible or have high value for this variable (low entropy). Let us explain this with an example. Observe that Browser_Usage_Total is highly correlated with Browser_Usage_Entropy (positive corr. = 0.885 highlighted on the diagonal). This means that if the total browser usage in a team is high then the entropy with respect to browser usage in the team is also high. High entropy means that all the members of the team exhibit similar behavior. Given that team has high total browser usage, this indicates that all the team members are equally contributing to this high browser usage of the team. Note that it could have been possible that only a single member is responsible for all or most of the browser usage. If this would have been the case, this cell corresponding to Browser_Usage_Total and Browser_Usage_Entropy would have been dark green (i.e. highly negatively correlated). In fact such is the case for the pair of Neverwinter_Nights_Entropy and Neverwinter_Nights_Total, which is highly negatively correlated with a value of −0.949. This indicates that if the total team's score for playing Neverwinter Night is high, then it is highly likely, in our four team member case, that it was possibly just single member responsible for this score (very low entropy).

This diagonal element property that we just stressed is very important as it highlights the importance of the two group level metrics Total and Entropy. This fine grained description that we are able to achieve just at the level of correlation analysis, shows the value of these group level metrics.

Table 7.10 Correlation between independent variables and performance (dependent variable)

Correlation with performance	English native	English ability	Stress in English environment	Reserve for English view	Game mods	Comp game names	Neverwinter Nights	Multiplayer comp game	Game list	Email usage	Browser usage	Teleconference usage	Chat usage	Net meeting usage	Own game console	Comp games time spent	Computer expertise	Own computer
Total metric	**0.396**	**0.435**	0.115	−0.011	0.078	0.345	0.186	**0.320**	0.246	0.017	0.071	−0.189	0.333	−0.001	0.324	**0.334**	**0.408**	0.108
Entropy metric	**−0.186**	0.021	0.029	0.244	−0.078	0.352	−0.229	−0.332	−0.257	0.019	0.079	−0.245	0.053	−0.150	−0.214	**0.240**	**0.238**	0.108

Table 7.11 Correlation between skill type entropy v/s total group metrics

Entropy v/s total	English_ Native_ Total	English_ Ability_ Total	Stress_ in_ English_ Total	Reserve_ for_ English_ View_ Total	Game_ Mods_ Total	Comp_ Game_ Names_ Total	Never-winter_ Nights_ Total	Multiplayer_ Comp_ Game_Total	Game_ List_ Total	Email_ Usage_ Total	Browser_ Usage_ Total	Telecon-ference_ Usage_ Total	Chat_ Usage_ Total	Netmeeting_ Usage_Total	Own_ Game_ Console_ Total	Comp_ Games_ Time_ Spent_ Total	Computer_ Expertise_ Total	Own_ Computer_ Total
English_ Native_Entropy	−0.114	−0.040	−0.158	−0.014	0.057	0.046	−0.102	0.164	0.032	−0.074	0.089	−0.005	0.096	0.075	0.047	0.069	0.151	−0.111
English_ Ability_Entropy	0.301	0.385	−0.129	0.000	0.031	0.158	0.195	0.159	0.120	0.216	0.414	0.010	0.134	−0.127	0.352	0.165	0.297	0.312
Stress_in_ English_ Entropy	0.205	0.266	0.052	−0.268	−0.028	0.113	0.171	0.134	0.058	0.284	0.295	0.112	0.078	−0.140	0.046	0.113	0.144	0.121
Reserve_for_ English_View_ Entropy	0.288	0.269	−0.031	−0.143	0.117	0.091	0.245	0.116	0.182	0.032	0.397	0.127	0.259	−0.033	0.302	0.154	0.244	0.265
Game_Mods_ Entropy	0.017	−0.110	−0.120	−0.103	**−1.000**	−0.036	0.125	−0.099	−0.678	0.136	−0.111	0.077	−0.130	−0.133	−0.013	−0.145	−0.314	0.130
Comp_Game_ Names_Entropy	0.107	0.126	0.034	0.043	0.073	**0.982**	0.090	0.458	0.156	−0.067	0.091	0.121	0.368	0.170	0.203	0.471	0.482	0.047
Neverwinter_ Nights_Entropy	−0.546	−0.509	−0.104	−0.154	0.119	−0.129	**−0.949**	−0.446	0.024	−0.245	−0.104	−0.054	−0.037	0.081	−0.434	−0.471	−0.315	−0.137
Multiplayer_ Comp_Game_ Entropy	−0.136	−0.273	−0.247	−0.215	−0.189	−0.354	−0.158	−0.272	−0.104	0.194	−0.162	−0.188	−0.339	−0.272	−0.076	−0.285	−0.354	−0.013
Game_List_ Entropy	−0.293	−0.337	0.095	0.057	−0.660	−0.154	0.036	−0.066	**−0.965**	0.060	−0.056	0.098	−0.121	0.017	−0.105	−0.191	−0.413	0.068
Email_Usage_ Entropy	0.133	0.162	0.206	0.085	−0.274	0.051	0.126	0.175	−0.203	**0.687**	0.272	0.062	0.117	−0.161	0.246	0.120	−0.043	0.543
Browser_ Usage_Entropy	0.126	0.191	0.105	0.011	−0.018	0.122	0.040	−0.070	0.044	0.091	**0.885**	0.285	0.159	0.173	0.193	0.081	0.190	0.475

Teleconference_ Usage_Entropy	−0.080	0.093	0.176	0.019	−0.046	−0.145	0.073	−0.003	−0.085	0.213	0.174	−0.062	−0.210	−0.027	−0.029	−0.066	0.078	−0.066
Chat_Usage_ Entropy	−0.064	0.007	−0.060	0.047	0.006	0.176	−0.185	0.356	0.038	0.062	−0.040	0.009	**0.749**	0.163	0.087	0.251	0.120	0.073
Netmeeting_ Usage_Entropy	0.006	0.108	−0.071	−0.049	−0.104	−0.181	−0.090	−0.078	0.069	0.297	−0.084	−0.339	−0.300	**−0.675**	0.146	−0.026	−0.257	−0.007
Own_Game_ Console_ Entropy	−0.071	−0.245	−0.199	−0.175	−0.203	−0.075	−0.064	−0.062	−0.096	−0.311	−0.025	0.233	0.026	0.176	−0.290	−0.110	−0.251	−0.110
Comp_Games_ Time_Spent_ Entropy	−0.071	−0.060	−0.006	0.147	0.050	0.485	0.148	0.278	−0.028	−0.128	−0.161	−0.159	0.106	−0.032	0.045	0.311	0.046	−0.274
Computer_ Expertise_ Entropy	0.210	0.280	−0.149	0.003	0.021	0.083	0.212	0.181	0.158	0.277	0.129	−0.271	−0.144	−0.325	0.274	0.139	0.261	0.069
Own_ Computer_ Entropy	0.123	0.170	0.238	0.101	−0.130	0.053	0.130	0.148	−0.052	0.358	0.537	0.255	0.288	0.116	0.057	0.184	0.237	**1.000**

7.5.2 Phase 2: Decision Tree Analysis

The decision tree using the Skill based metrics is shown in the Fig. 7.11a and the corresponding model accuracy on the training instances is shown in Fig. 7.11b. We know from Table 7.4 that the average Teleconference_Usage_Total across all the groups is around 8. The leaf right of the root node (Teleconference_Usage_Total) is attained if the group has very high (>13) Teleconference usage relative to the mean of 8. Unfortunately, this leaf is labeled "0 (3.0)", meaning three low performing groups have been observed with such high Teleconference usage. Here, Daily

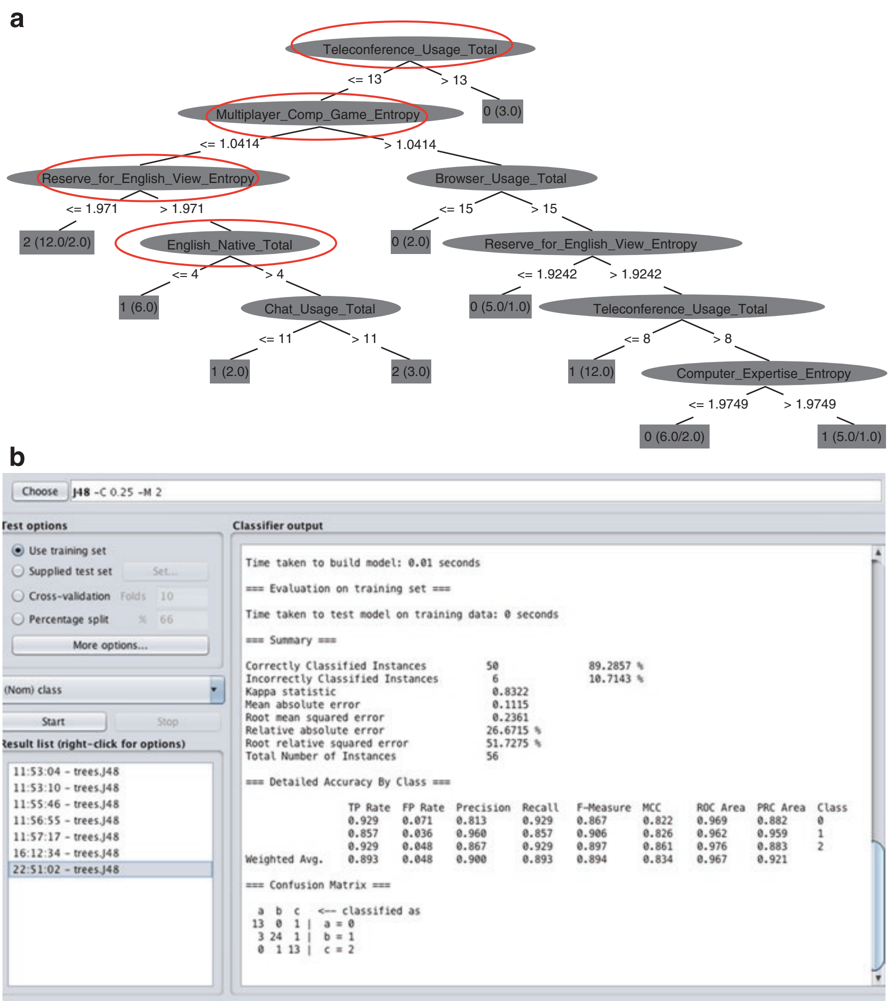

Fig. 7.11 (**a**) Full skill metric model decision tree. (**b**) Full skill metric model fit

or Weekly Teleconference usage was predicted as something agnostic to group performance.

Moreover, low entropy in multi-player game playing and low entropy in native English speakers had a positive relationship with high group performance. That is, groups with low entropy values on these two variables were overwhelmingly predicted to be in the high performance class. By following the parent nodes of these two variables (i.e. Multiplayer_Comp_Gam_Entropy ≥ Reserve_for_English_View_Entropy) to the leaves, it shows that 23 (e.g., add up all the predicted cases in the left, 12 + 6 + 2 + 3) groups fall in these leaves. Out of these 23 groups, 15 (12+3, ~65 %) were predicted as high performing groups (only two were incorrect). As such, out of the 14 high performing groups, these rules correctly classified 13 of them (~93 %), leaving only one false negative (i.e., a high performing group incorrectly predicted as not high performing).

7.5.3 Phase 3: Feature Selection

Similar to previous example, using Weka we performed the SVM based feature selection using the SVMAttributeEval functionality provided in Weka. The Attribute selection output contains the ranked list of various skill type group metrics is shown in Fig. 7.12.

7.5.4 Phase 4: Decision Tree Analysis over Selected Features

Finally, we perform a decision tree analysis using the selected features. This time we chose the top ten features out of the total 36 skill metrics which are shown in decreasing ranks in Table 7.12. We also tried other values for the number of top attributes to use, but they did not generate useful trees. In fact, the resulting J48 decision tree shown in the Fig. 7.13a employs only five features out of the ten selected features. However, as we can observe in Fig. 7.13a the left sub-tree of root has similar relationships to those in the tree built in Phase 2 (Fig. 7.10a). The left sub-tree highlighted with a red circle is the most interesting as it contains only high or medium performing groups. The corresponding model fit is given in the Classification output in Fig. 7.13b.

To summarize this example, the best predictor of high performing teams is a combination of low values regarding entropy in skill related to multiplayer computer games, total teleconference usage, and entropy in Reserve in English View presentation (12 predicted to be high performing, only two were incorrect). Likewise, high performing teams tended to have high entropy in Computer expertise and Game Mods (4 predicted to be high performing, one incorrect).

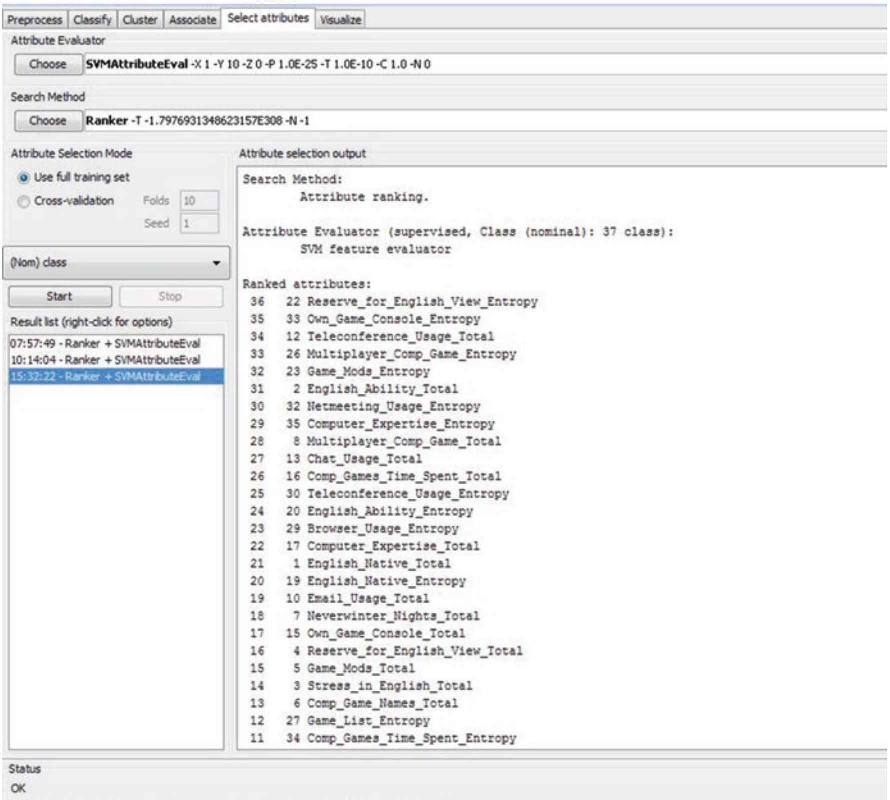

Fig. 7.12 Ranked attributes for skill model using SVM

7.6 Experiment 3: Group Analysis Using Group Configuration Metrics

In this section we focus on the effect of the group configuration metrics on the group's performance. Table 7.12 shows the correlation score of the different group configuration metrics with group performance (the dependent variable). The correlation of Group_Conf_1-1-1-1_Total with performance reflects that working separately is correlated with good performance, suggesting a division of labor may be beneficial rather than working collectively at the same time. To see this more visually we plot the linear regression curve in Fig. 7.14a where the line has a positive slope.

If we focus on the Group_Conf_Entropy, we observe a negative correlation with performance. Note that the Group_Conf_Entropy variable reflects homogeneity with respect to the different possible group configurations over time. It is high when a group spends equal time in each of the five configurations and lowest when the team is just playing in a single configuration during the entire playing time. The negative correlation therefore, suggests that in general, spending time in fewer

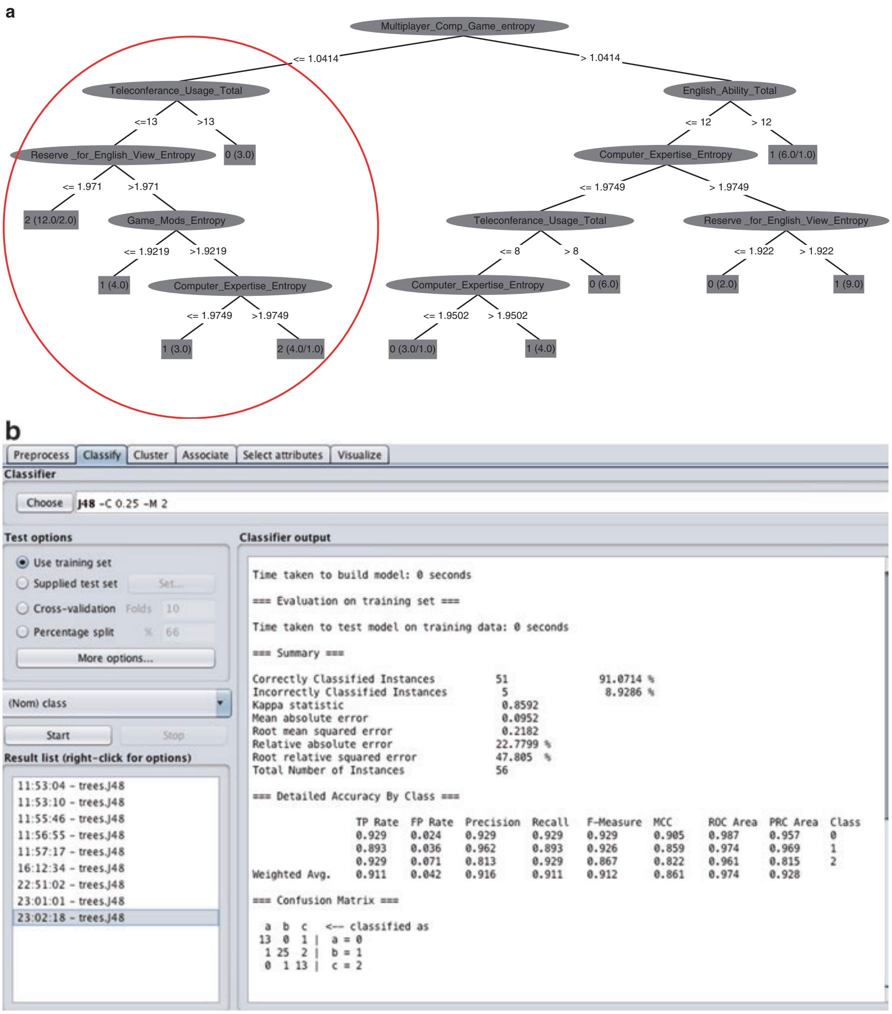

Fig. 7.13 (**a**) Reduced skill metric model decision tree. (**b**) Reduced skill metric model fit

Table 7.12 Correlation scores of the different group configuration metrics with group performance

Total metrics	Performance
Group_Conf_1-1-1-1_Total	**0.314**
Group_Conf_1-1-2_Total	−0.136
Group_Conf_1-3_Total	**−0.369**
Group_Conf_2-2_Total	−0.185
Group_Conf_4_Total	−0.177
Entropy metrics	
Group_Conf_Entropy_Entropy	**−0.349**

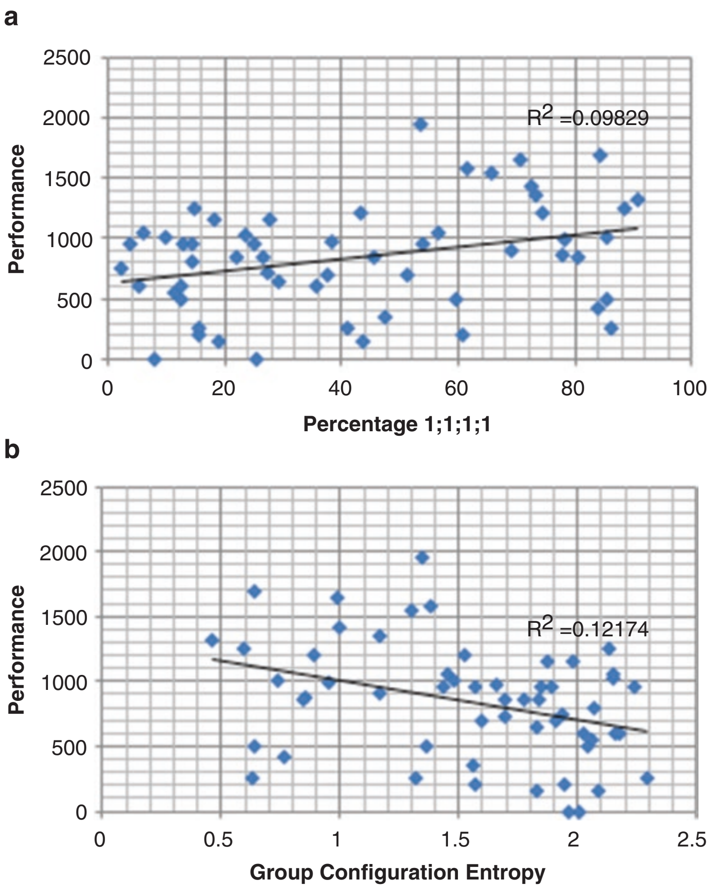

Fig. 7.14 (**a**) Plot of group performance and working separately. (**b**) Plot of group performance and Group Configuration Entropy

different configurations has a positive relationship with performance (rather than being equally distributed in all the configurations). This can be visually seen in the linear regression plot in Fig. 7.14b. Hence, analyzing both Entropy and Total metrics is important because both had significant relationships with group performance.

7.7 Experiment 4: Using All Types of Metrics Combined for Group Analysis

In this section we shall consider a mixed model that combines the set of all the three metric types: 16 role types, 36 skill types and 6 group configuration types together. Given we had already analyzed the correlation of these three types separately in the

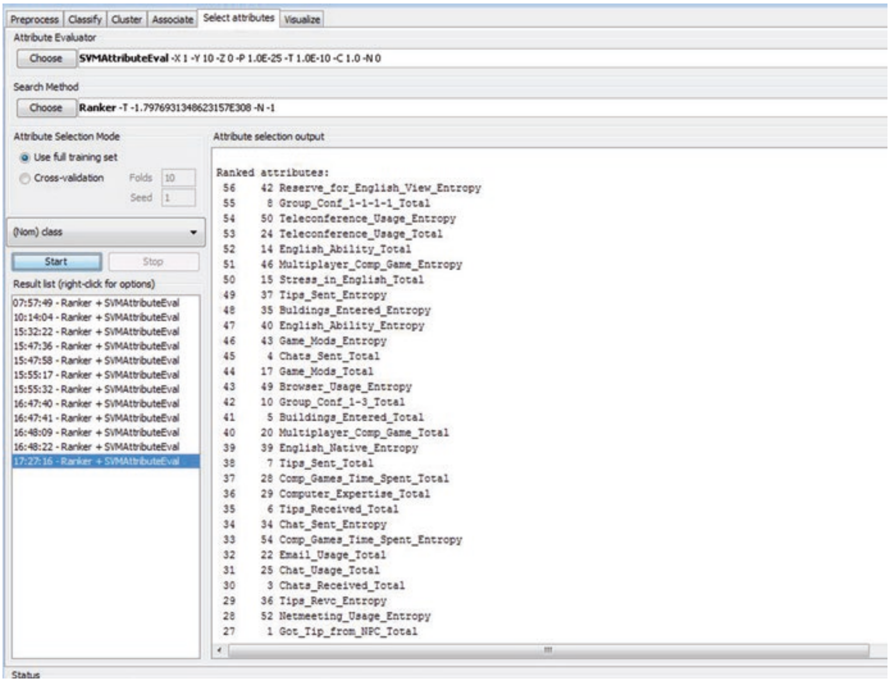

Fig. 7.15 Ranked attributes for mixed model using SVM

previous three experiments, we would not mention it further. We would also skip the pair-wise correlation analysis, although there are several interesting dependencies across different type metrics, but the across-type pairs are simply too many to analyze and describe. In fact, as the combined set of all the three types has a large number of metrics, we would directly perform decision tree analysis on SVM selected features (i.e. Phase 4). The attribute ranking of this combined set of metrics using the SVMAttributeEval in Weka is shown in the Fig. 7.15.

In this case we again go with top ten metrics, in decreasing order of rank. Moreover, the J48 decision tree classifier output tree is shown in Fig. 7.16b and the classification accuracy output from Weka is shown in Fig. 7.16a. The important point here is that now we are at the stage where we are considering all the 56 different metrics together. Therefore, we would now be able to compare and select metrics that are important across all the types. This should help us understand which are the most globally important metrics.

Although we had selected the top ten group metrics from the SVM ranking list, the decision tree only used eight out of these ten. We observe that among Role Metrics, Total Tips Sent, and Tips Entropy seem to play a very important role. That is, low performing groups tend to send few tips, while high performing groups tend to have higher levels of Total Tips and Tips Entropy. Within the Skill Metrics,

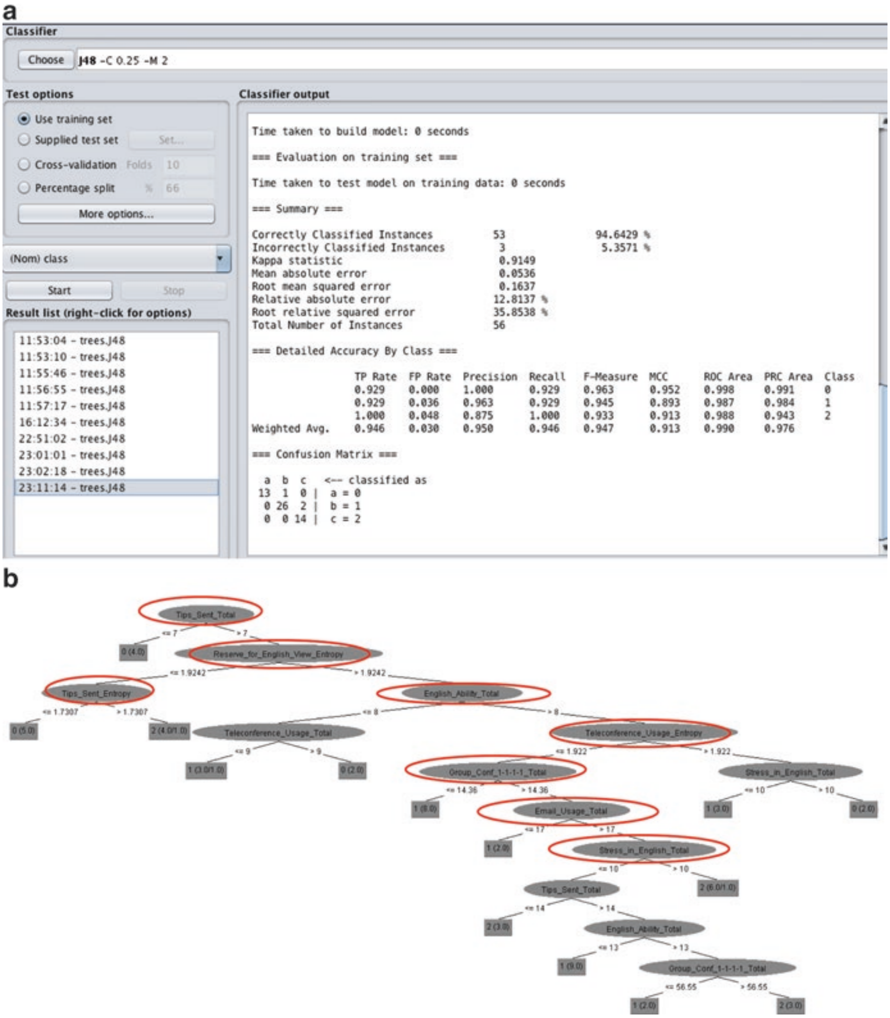

Fig. 7.16 (**a**) Mixed model fit. (**b**) Mixed model decision tree

heterogeneity of being Reserve in English View presentation and high English Ability tend to predict high performing groups. Teleconference Usage Entropy, Total Email, and Chat Usage turn out to be key factors as well.

As has been observed in previous analyses, total amount of time spent in {1,1,1,1} type Group Configuration is one the very crucial factors for team success. In fact, if we observe the pair-wise correlation matrix (see Table 7.13), we observe that when members work separately they chat less and spend more time in interaction with NPCs and gather tips from NPCs. This knowledge gathered from NPCs, we can hypothesize, may be highly influential for group success.

Table 7.13 Pair-wise correlation for mixed model variables

Correlation between total metrics	Got tip from NPC	NPC interacted	Chats received	Chats sent	Buildings entered	Tips received	Tips sent	Group_Conf_1-1-1-1_Total
Got tip from NPC	1.000							
NPC interacted	0.646	1.000						
Chats received	0.093	0.107	1.000					
Chats sent	0.093	0.107	1.000	1.000				
Buildings entered	0.187	0.128	−0.022	−0.022	1.000			
Tips received	0.300	0.229	0.135	0.135	0.101	1.000		
Tips sent	0.300	0.229	0.135	0.135	0.101	1.000	1.000	
1-1-1-1	**0.330**	**0.420**	**−0.344**	**−0.344**	0.243	0.071	0.071	1.000

Overall, the mixed model, beyond just being the best in terms of model fit, demonstrates how complex the interactions are amongst the different sets of variables. Following the different paths along the decision tree can yield important insights into how these variables moderate one another. As to which model is best depends on the goals of the researcher. All the models ran had overall accuracy levels nearing 90 %. As such, a parsimonious model, though slightly less accurate, may be useful for those attempting to seek out which are the "big" factors discriminating between high and low performing teams. On the other hand, a more complex, less parsimonious predictive model may be useful if the goal is to "predict at all costs", which may be useful developing predictive applications (e.g., a team assembly application).

7.8 Conclusion

In this work we illustrated how to analyze small group behavior using individual level data. In this direction we show two possible ways of aggregating individual level information to generate group level metrics. Further, we show how traditional correlation analysis can substantially be supplemented with the help of the proposed metrics. In this sense, the techniques are not competing, but complementary. Finally, we employ these metrics within existing machine learning and data-mining techniques and illustrate, with the help of Weka data-mining software, how group performance can be analyzed using data-mining.

References

Chandrashekar, G., & Sahin, F. (2014). A survey on feature selection methods. *Computers & Electrical Engineering*, *40*(1), 16–28.

Guyon, I., & Elisseeff, A. (2003). An introduction to variable and feature selection. *Journal of Machine Learning Research*, *3*(Mar), 1157–1182.

Guyon, I., Saffari, A., Dror, G., & Cawley, G. (2010). Model selection: Beyond the bayesian/frequentist divide. *Journal of Machine Learning Research*, *11*(Jan), 61–87.

Guyon, I., Weston, J., Barnhill, S., & Vapnik, V. (2002). Gene selection for cancer classification using support vector machines. *Machine Learning*, *46*(1–3), 389–422.

Hall, M., Frank, E., Holmes, G., Pfahringer, B., Reutemann, P., & Witten, I. H. (2009). The WEKA data mining software: An update. *ACM SIGKDD Explorations Newsletter*, *11*(1), 10–18.

Quinlan, J. R. (1986). Induction of decision trees. *Machine Learning*, *1*(1), 81–106.

Quinlan, J. R. (1993). *C4. 5: Programs for machine learning*. San Mateo: Morgan Kaufmann.

Teachman, J. D. (1980). Analysis of population diversity measures of qualitative variation. *Sociological Methods & Research*, *8*(3), 341–362.

Warren, R., Diller, D., Leung, A., Ferguson, W., Sutton, J.L (2005). *Simulating scenarios for research on culture and cognition using a commercial role-playing game*. Proceedings of the 2005 Winter Simulation Conference, Orlando, FL, December 4-7, 2005.