A Decision Tree Model for Software Development Teams

Abdul Rehman Gilal, Mazni Omar, Ruqaya Gilal, Ahmed Waqas, Sharjeel Afridi, Jafreza Jaafar

Abstract: Different theoretical personality models for team composition proved to be inconsistent, posing validity challenges and missing guidance for personnel selection in software development. Due to these impacting issues, this study has produced a decision tree model for software team composition for effective team performance. The model is based on personality types (i.e., collected using Myer Briggs Type Indicator (MBTI)), gender and team role (i.e., only team leader and programmer) to predict team performance (i.e., effective or ineffective). Experimental data, collected from software engineering students of Universiti Teknologi Petronas (UTP) Malaysia, was used to develop and validate the model. In order to develop and validate the model, C4.5 algorithm and 10-fold cross validation methods were used respectively. The results indicate that Judging-Percieving (JP) personality pair is one of the significant attributes to identify the team performance. At the end, the model was observed acceptable during validation process by obtaining satisfactory prediction accuracy 70.48%.

Keywords: Team composition; decision tree; Personality; MBTI

I. INTRODUCTION

A plethora of studies has been carried out to explore the importance software development team composition by considering personality types of developers. But previous work has failed to adequately define the effective personality types for ideal and effective team composition[1]. The results extracted from different theoretical personality models proved to be inconsistent, posing validity challenges and missing guidance for personnel selection in software development. For instance, Nelson[2], concluded that “different results were obtained when models were used in practice to compose teams”. There has also been disagreement concerning the models that are suggested to compose teams as they did not yield positive results for organizations and scholars. Given this, it raises the need for more work in this area to identify the problem and solve contradictory fits. There can be many reasons for these problems but selecting a team member based on his personality without considering the exact team role is considered as one of major problems[3], [4].

Among the detrimental challenges, model development techniques and experimental setups are also considered on the top which drag the results to different directions[5]. In this point of view, classification technique plays an important role in generating effective results. Therefore, the aim of this study is to apply the most widely used classification technique, i.e. decision tree to extract the results. Actually, the results from decision tree algorithms are easy to comprehend, it is also superior in terms of rendering better performance while dealing with either categorical or discrete variables as compared to neural networks, k-Nearest Neighbourhood (kNN), Naive Bayes and Support Vector Machine (SVM)[6].

In order to develop the study model, software team roles (i.e., only team leader and programmer), gender and MBTI based personality types as independent variables which are used to measure the team outcome or team effectiveness. Several studies have studied personality types with software team roles but gender has been ignored [7]. In this study, gender is considered as an important element because personality traits of male and female are different. As in the field of psychology, there has been many debates on male personality preferences can never be same as female personality preferences. They both are at different scales in every trait of personality in psychology. Unfortunately, software engineering (SE) researchers have so far considered male and female personality profiles without any gender differences. Therefore, it is of great importance that one must not ignore gender factor if personality based research is being carried out in software engineering[8]. Based on the study conducted by Jayne[9], the issues get existence if the personality is being interpreted without considering gender differences. It leads us to the conclusion that gender needs to be examined as there are strong limitations in software development research which involves personality.

This study is a continuation of our previously published work [5] in which we mainly presented the results extracted from Johnson Algorithm (JA) under Rough set theory paradigms. One can refer to our published work to learn more about the study variables. Moreover, this study has covered the details of decision tree model development and validation. Therefore, following section only covers the related work on decision tree.

II. RELATED WORK

The concept of decision tree technique is centered on divide-and-conquer

Revised Manuscript Received on March 08, 2019.
Abdul Rehman Gilal, Sukkur IBA University, Pakistan,
Mazni Omar, Universiti Utara Malaysia,
Ruqaya Gilal, Universiti Teknologi Petronas, Malaysia,
Ahmed Waqas, Sukkur IBA University, Pakistan,
Sharjeel Afridi, Sukkur IBA University, Pakistan,
Jafreza Jaafar, Universiti Teknologi Petronas, Malaysia,
algorithm and it is generally integrated into two indispensable concepts i.e., entropy and information gain. The algorithm in decision tree technique locates the best predictor attributes so as to fragment its value according to its information value and thus, the highest information gain is denoted by the one having correct classification. Later on, the value an important attribute is marked on the top and root of the tree. Then, the value of next attribute is tested following its gain information based on leaf node that represents the outcome class. Whereas, entropy is defined as measurement of uncertainty of data and is labeled as equal to zero in a situation when all objects comes off with a single class.

Decision tree application is mostly avoided to put into practice for having a key drawback as its applications are based on the concepts of C4.5 algorithm for having pruning facility that become ineffective in handling over-fitting issue of data analysis. Pruning can be defined as the process that lessens and curtails size of the trees by removing least significant attributes that facilitate researchers to make predictions for outcome. It also causes over-fitting that alters sample data with the emergence of noise and it can amplify accuracy of testing data sets by lessening the prediction accuracy related to new data sets. Figure 1 shows the pseudo code for building decision tree.

1. Check for base cases
2. For each attribute a
3. Find the normalized information gain from splitting on a
4. Let a_best be the attribute with the highest normalized information gain
5. Create a decision node that splits on a_best
6. Recur on the sub-lists obtained by splitting on a_best, and add those nodes as children of node

Fig. 1 Pseudo Code for Building Decision Tree [6]

The use of decision tree application is applied in different disciplines as well, i.e., in medical field [10], manufacturing [11] and bioinformatics [12]. The past literature review also indicates that decision tree applications have also been applied in the field of educational data mining due to their ease and simplicity that enables educational researchers to sort out different patterns of data for future analysis. For instance, an empirical research study conducted by Fang and Liu [13] in which he tried to foresee the academic performance of students and obtained accuracy (83.3 - 85.9%) that was quite better and accurate as compared to the results obtained using linear regression (66.7-71.9%). In another similar kind of research conducted by Janecek and Haddawy[14] that also aimed to predict academic performance of students, it was proclaimed that the efficiency of the decision tree application was far more better and accurate than Bayesian network. The researchers further stated that decision tree technique became more effective when the number of outcome classes was lessened from 4 to 0 that brought significant change in accuracy that enhanced from 71-73% to 93-94%. In another empirical study researched by Al-Radaideh, Ananbeh, and Al-Shawakfa[15] that aimed to predict favourable education tracks for students also brought significant and accurate results using decision tree application with accuracy of results (87.9 %). This evidence shows that the effectiveness of using decision tree applications can be measured keeping in view the nature of data employed in study.

WEKA tool’s concept is based on the machine learning algorithms i.e., decision tree, rules, Bayesian classifier, and lazy classifier. This tool works better on decision tree as for providing it with various algorithms like simple CART, ID3 and C4.5. WEKA tool basically works on four interfaces, such as Experimenter, Explorer, simple CLI and knowledge flow. It is better for new users of this tool to use Explorer for it facilitates the users to analyse and pre-process the data. Or else, other interfaces of this tool are complex enough to understand their functions and purposes in less time. This study uses the guidelines set by Witten and Frank [16] to use Explorer to handle WEKA adroitly.

III. METHODOLOGY

UTP students were involved in the controlled experiments to develop software projects (i.e., the details are already published in [17]-[19]). With 105 sample size, team role, MBTI personality pairs (i.e., IE, SN, TF and JP) and gender were the predicting variables for measuring the outcome variable team performance. Table 1 contains the predictor and outcome variables with their respective values, which are used in the model development.
Table. 1 Predictors and outcome variables for model development

<table>
<thead>
<tr>
<th>Variable</th>
<th>Input</th>
</tr>
</thead>
<tbody>
<tr>
<td>Role</td>
<td>team lead programmer</td>
</tr>
<tr>
<td>IE</td>
<td>introvert extrovert</td>
</tr>
<tr>
<td>SN</td>
<td>sensing intuiting</td>
</tr>
<tr>
<td>TF</td>
<td>thinking feeling</td>
</tr>
<tr>
<td>JP</td>
<td>judging perceiving</td>
</tr>
<tr>
<td>Gender</td>
<td>male female</td>
</tr>
<tr>
<td>Outcome</td>
<td>ineffective effective</td>
</tr>
</tbody>
</table>

Decision tree uses tree-like graphs to visualize the decision classes. It is based on divide-and-conquer algorithm which recursively break down the classes into subclasses for new input until the effective decision class is found. Furthermore, decision tree approach is free from data normality assumptions, hence it is also suitable for small data sets. Therefore, following were the reasons to use decision tree:

- It is an efficient approach for nominal and ordinal data.
- It is faster and easy to interpret the graph-like outputs.
- It does not require data normality assumptions, therefore, it allows small size of data sets.
- It is named as the mostly used approach in data mining [20].

This study used k-fold cross validation method. The K-fold cross validation method is computationally expensive than hold-out (divide data into training and testing subsets) but it produces precise results. [25]. This study used k=10 cross validation experiments to implement as it is a most experimented folds in data mining [16]. Additionally, Hubert and Engelen[26] also suggest that k-fold cross validation is suitable if the data size is small.

IV. RESULTS AND DISCUSSION

Prior to decision tree implementation, zeroR classifier was used to predict the model by ignoring the all predictors. Based on the zeroR classifier results, the classification accuracy 57.1% correctly classified instances by the null model. Hence, Null model was set with 57.1% classification accuracy for C4.5 (J48 in Weka) results to be considered effective. Eventually, after including predictors along with outcome variables in the model, J48 obtained 70.48% of classification accuracy. In other words, 29.52% instances were classified incorrectly by the decision tree model. Based on the set null model’s benchmark, these results of decision tree can be used for measuring team performance. Table 2 shows the prediction accuracy of decision tree used in this study.

![Decision Tree Diagram](image-url)

Fig. 2 Pruned Decision Tree using C4.5 algorithm in Weka

The results of the tree state that team leader with perceiving personality trait were appeared ineffective (i.e., IF JP="Perceiving" AND Role="team-lead" THEN Class="ineffective"). Both judging (J) and perceiving (P) individuals are more likely related with the world outside them [27]. Judging individuals are more comfortable living in organized, tidy, and distinct kind of way where their lives are carefully regulated and controlled. Whereas, perceiver individuals are more comfortable for them living in versatile, and unplanned kind of way which they themselves adapt to the situation of life encountered. It is also mentioned that perceiving people are flexible and spontaneous.

Table. 2 Predication accuracy of decision tree

<table>
<thead>
<tr>
<th>Observed</th>
<th>Predicted</th>
<th>Percentage</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Ineffective</td>
<td>Effective</td>
</tr>
<tr>
<td>Outcome</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Ineffective</td>
<td>52</td>
<td>8</td>
</tr>
<tr>
<td>Effective</td>
<td>23</td>
<td>22</td>
</tr>
<tr>
<td>Overall Percentage</td>
<td>70.48</td>
<td></td>
</tr>
</tbody>
</table>
But, based on the results, it was found that team leaders with judging trait (planned and organized) tend to be more effective than perceiving. On the other hand, perceiving trait appeared suitable for male programmers (i.e., IF \(JP='Perceiving'\) AND \(Role='programmer'\) AND \(Gender='male'\) THEN \(Class='effective'\)). Female programmers also performed effective within perceiving trait provided they were composed with extrovert trait (i.e., IF \(JP='Perceiving'\) AND \(Role='programmer'\) AND \(Gender='female'\) AND \(IE='extrovert'\) THEN \(Class='effective'\)). It could have happened as Extroverts (Es) provide opportunities for communication among team members. Otherwise, introvert and perceiving composition was not shown suitable for female programmers (i.e., IF \(JP='Perceiving'\) AND \(Role='programmer'\) AND \(Gender='female'\) and \(IE='introvert'\) THEN \(Class='ineffective'\)). Similarly, combination of judging and introvert personality traits appeared ineffective (i.e., IF \(JP='judging'\) AND \(IE='introvert'\) THEN \(Class='ineffective'\)). At the same time, judging and extrovert personality traits were found effective for team lead role (i.e., IF \(JP='judging'\) AND \(IE='extrovert'\) AND \(Role='team-lead'\) THEN \(Class='effective'\)). However, the same judging and extrovert personality traits were not suitable for programmer role (i.e., IF \(JP='judging'\) AND \(IE='extrovert'\) AND \(Role='programmer'\) THEN \(Class='ineffective'\)).

The decision tree results were validated through 10-fold cross validation method. It divides the training set in equal 10 pieces and trains the model from 9 and use remaining 1 for testing. The results would be considered biased if the same 9 pieces are always used for training and 1 for testing. Therefore, to reduce the biasness, the model was trained and tested by using different seeds (pieces) each time for 10 times. This option is supported in Weka tool by changing random seed value (i.e., 1, 2, 3…10) instead of pre-assigned 1. The mean of these each values can be calculated to see the average point. In this study, the mean accuracy of 10-fold method validation was 65.05.

![Fig. 3 Cross validation of Decision Tree at 10-fold method](image)

However, 2.70 standard deviation was calculated in the 10 accuracies. Although, the standard deviation was not much high but the average predication accuracy is little lower than the initial prediction accuracy. The results of this model were appeared significantly improved if these were compared with baseline (i.e., null model) accuracy which was 57.1%.

**V. CONCLUSION**

Based on the results, the developed model appeared to be better than the null (zeroR) model as it improved the prediction accuracy with the predictor variables. Basically, team role, gender, and IE variables were found significant for team performance. On the other hand, SN and TF variables did not contribute significantly to the model. Moreover, although the model performance is better than the null model but our previous study [5] got 79.04% prediction accuracy on the same dataset. However, this study had generated 7 decision rules whereas our previous study’s model produced 24 decision rules. It shows that decision tree is better in terms of reducing complexity but it compromises the prediction accuracy.

**REFERENCES**