International Journal of Advanced Smart Convergence Vol.11 No.4 68-80 (2022)
http://dx.doi.org/10.7236/IJASC.2022.11.4.68

IJASC 22-4-8

A Study on the Classification of Variables Affecting Smartphone Addiction in Decision Tree Environment Using Python Program

Seung-Jae Kim(s)

Professor, Department of Convergence Honam University, Korea
cdma1234@hanmail.net

Abstract

Since the launch of AI, technology development to implement complete and sophisticated AI functions has continued. In efforts to develop technologies for complete automation, Machine Learning techniques and deep learning techniques are mainly used. These techniques deal with supervised learning, unsupervised learning, and reinforcement learning as internal technical elements, and use the Big-data Analysis method again to set the cornerstone for decision-making. In addition, established decision-making is being improved through subsequent repetition and renewal of decision-making standards. In other words, big data analysis, which enables data classification and recognition/recognition, is important enough to be called a key technical element of AI function. Therefore, big data analysis itself is important and requires sophisticated analysis. In this study, among various tools that can analyze big data, we will use a Python program to find out what variables can affect addiction according to smartphone use in a decision tree environment. We the Python program checks whether data classification by decision tree shows the same performance as other tools, and sees if it can give reliability to decision-making about the addictiveness of smartphone use. Through the results of this study, it can be seen that there is no problem in performing big data analysis using any of the various statistical tools such as Python and R when analyzing big data.

Keywords: Decision Tree; Machine Learning; Classification: Python Code Analysis: R Code Analysis_

1. Introduction

As the 4th industry was discussed, a new paradigm called AI appeared in all industries. A new paradigm of AI is expected to revolutionize all industries with limitless energy and consequent capabilities throughout society as we know it. These AIs are currently showing groundbreaking achievements in each field of society around the world, and will be further developed and improved as time goes on. After the 4th industry, AI technologies that perfectly connect the virtual world and the physical world in real time, such as 'digital twin' and 'modeling and simulation', have been analyzed as major technologies for next-generation production innovation technologies [1]. AI technology, which surprised the world, is making rapid progress through technologies called machine learning (ML) and deep learning (DL). Applications of ML are routinely used in speech recognition, computer vision, and many other commercial systems. In Korea, an analysis was also conducted to classify subway stations according to their getting on and off patterns using ML [2,3]. As for the application of DL, research on predicting low blood pressure in high-risk groups that cause serious complications has been conducted with Random Forest, and research on natural language processing has also

Manuscript Received: October. 5, 2022 / Revised: October. 8, 2022 / Accepted: October. 11, 2022
Corresponding Author: cdma1234@hanmail.net
Tel: +82-62-940-5639, Fax: +82-62-940-5005
Professor, Department of Convergence, Honam University, Korea

been conducted to predict the next event in business processes [4,5]. Here, ML is an internal technology, which is supervised learning, unsupervised learning, and reinforcement learning, and each technology has the ability to produce excellent results that can be said to be very good. When a seller inputs product registration information in natural language, the 'KoNLPy' morpheme analysis process is performed, and supervised learning is used to implement a system that automatically recommends catalog information most suitable for the product by applying the 'Naïve Bayes' classification method [6]. As for the application of unsupervised learning, research has been conducted to predict the overall shape using cluster data rather than step-by-step data learning by applying neural networks to reflect data learning methods [7]. In addition, in the case of reinforcement learning, an autonomous parking simulator based on the Ackerman steering geometry model formula was developed to model the actual vehicle operation method, and then an autonomous parking simulator to which the Deep Deterministic Policy Gradient (DDPG) reinforcement learning algorithm was applied was studied [8]. In addition, DL is a technology that improves itself through self-directed learning renewal called reinforcement learning among ML, and has the ability to produce very excellent results. It is an obvious fact that these technologies will continue to be researched and developed even now.

In order for the above-mentioned technologies to derive more and more improved results, the results of big data analysis (BDA), which is the basis of this technology, must have high reliability. AI technology will be utilized not only in fragmentary cases or environments that are easy to use, but in highly complex and diverse environments. Recognizing and recognizing objects in such a fragmentary or complex environment can cause a big social problem if the function is implemented only through a few cases. BDA and accurate classification of data are very important in order not to cause social problems caused by AI functions. In order to utilize the data for the purpose, it must be possible to accurately classify the data, and the classified data must always have high reliability. There are various techniques for data classification (DC), but among them, various experiments were conducted on key factors such as the number of trees, feature selection, and learning set size in terms of classification performance for automatically assigning topic categories to domestic journal articles using random forest [9]. In addition, research was conducted to detect and classify erratic faults, drift faults, hard-over faults, spike faults, and stuck faults, which are typical types of faults that occur in sensors, by applying ML algorithms such as SVM and CNN [10].

To classify the data, you need to use data analysis tools (DAT). There are various statistical tools such as SPSS, SAS, STATA, and Excel for DAT, and R and Python are available for DAT that use computer languages. In the field where the R program was applied, social network analysis was performed using empirical data for researchers to conduct related research [11]. In addition, in the field of application of the python program, by proposing a learning procedure and teaching strategy for a python-based software education model, and a curriculum for one semester, it was applied to liberal arts classes, resulting in significant results in an effective aspect [12]. R and Python are widely used as computer languages for implementing AI technology today, and provide variety and many libraries for free in BDA.

In this study, we try to analyze data by using Python program among various statistical tools. The data analysis used will use decision trees (DT). Various studies on DT have studied models that predict the probability of fire occurrence when weather conditions are given using DT to prevent catastrophic fires [13]. In addition, a study was conducted to classify and analyze patterns of mobile communication customers in order to increase customer credit prediction by applying a combined DT (C4.5) and neural network techniques [14]. In addition, a study was conducted to compare human empirical knowledge with collected data, convert it into weights, and use it to create a DT [15]. The purpose of the analysis is to find out what variables will lead to addiction in smartphone use through DC. In addition, based on the data classified by the Python code, we also examine the relationship between the variables. Through the results of this study, it will be confirmed that there is no problem in performing BDA using any of the various statistical tools such as Python and R when analyzing big data.

2. Data Classification

DC analysis is one of the ML techniques, and classification analysis (CA) is a technique that can classify

data by grouping collected data with different characteristics. Today, there are several statistical tools such as SPSS and SAS for DT analysis, but computer languages are used a lot from the point of view of implementing AI technology. As a computer language used, one of R Program and Python Program is used, and sometimes each may be used for the purpose.

2.1 DT Definition

DT have relatively fast, simple, and easy-to-understand rules compared to other CA techniques. A DT is a technique that can classify collected data into several groups and classify decision-making rules that appear between variables using a tree structure.

DT use the concept of impurity to select classification criteria in the DT, which means the complexity of the data. In other words, it means the degree to which different data are mixed in one category. When setting the classification standard, the impurity of the child node should be set to be reduced compared to the impurity of the current node. This difference is called information gain. Equation 1 is the impurity function. p is the proportion of classes belonging to each group.

$$G(S) = 1 - \sum_{i=1}^{c} p_i^2 \quad p_i (i = 1, 2, \ldots, k) \tag{1}$$

For sophisticated DC, a step-by-step analysis process is performed. At this time, the analysis process consists of five steps. Table 1shows the 5-step analysis process of the DT.

Table 1. Analysis process of 5 steps of DT

Analysis stage	Step-by-step analysis	
Step 1	Analysis process	Creation of decision tree
	According to the purpose of analysis, it has appropriate separation criteria and stopping rules.	
Step 2	Analysis process	Pruning
	Remove branches that have the potential to increase the classification error or have induction rules that are inappropriate.	
Step 3	Analysis process	Feasibility evaluation
	Analyze cross-validation using gains chart, risk chart, or verification data.	
Step 4	Analysis process	Interpretation and prediction
	Interpret the decision tree and establish a predictive model.	
Step 5	Analysis process	Decision tree formation
	In the above process, different decision trees are formed depending on how the separation criteria, suspension rules, and evaluation criteria are applied.	

Looking at the 5 stages of the DT, first, it has the appropriate separation criteria and stopping rules according to the purpose of analysis, which are processed by the internal function. The branch with the second highest possible misclassification potential or inappropriate inference rule is eliminated. Thirdly, cross-

validation is analyzed by benefit table, risk chart, or data for verification. Fourth, we interpret the DT and set up a prediction model. Finally, in the above process, decisions are made according to the separation standards and rules.

The pseudo-decision tree divides the entire data into a train (training) set and a test (test) set through an internal operation, learns each, and then calculates the classification rate. In addition, in order to classify data using DT analysis, the type of data used must be data with continuous information. If these data are simply defined using R and Python programs, it is as follows.

2.2 Decision Making by R Program

The R program has been used for a long time as a free program for public use, and its usage has increased rapidly since the importance of big data analysis was mentioned. DT analysis using an R program divides the entire data into a training set and a test set, learns it, and then calculates the classification rate.

DT analysis using an R program divides the entire data into a training set and a test set, learns it, and then calculates the classification rate. The R program uses the internal prune() function to classify data into tree types, which is called a branch function. In DT analysis, no matter which analysis tool is used, the classification criteria for pruning are determined by entropy and information gain from parent node to child node, and calculation is performed by 'installPackage(tree)'. The figure below Figure 1 shows the pruning after the pruning standard is set by the prune() function of the R program.

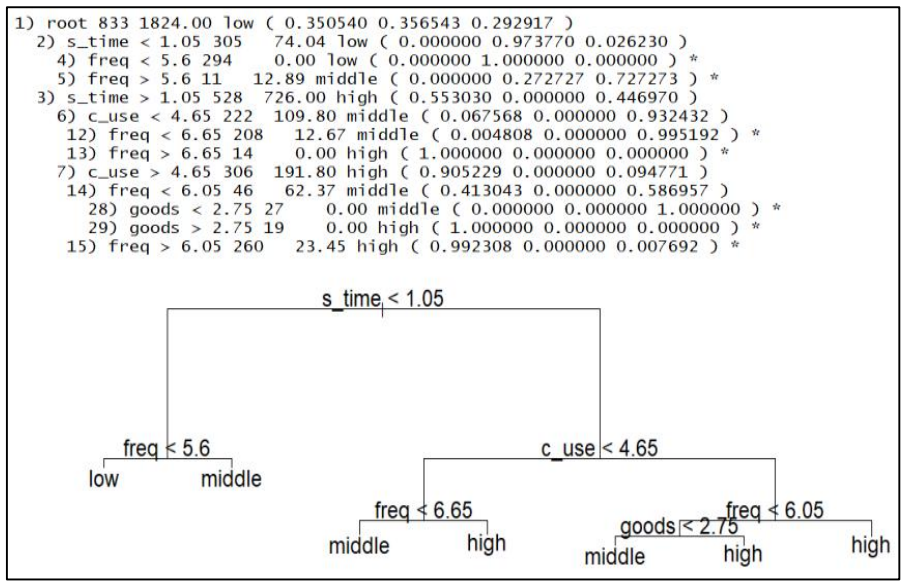

Figure 1. Pruning by R program

2.3 Decision Making by Python Program

The Python program has been used for a long time as a free program for public use, and its usage has increased rapidly since the importance of BDA was mentioned. In addition, in order to implement AI functions today, it is necessary to build a complex system using Python and various libraries. When performing DT analysis using a variety of Python programs, the classification rate is calculated after learning by dividing the entire data into a training set and a test set.

The Python program's pruning criteria are also determined by entropy and information gain, and the algorithms include "ID3, C4.5, C5.0" based on ML and "CART, CHAID" based on statistics. Among them, C5.0 is a supervised learning algorithm that has improved the previous two types.

C5.0 is based on the concept of entropy and information gain, and if the data of the initial target variable (explanatory variable) is mixed, the impurity increases, and at this time, entropy becomes large. In the process

of classifying each input variable (dependent variable) data, the data of the target variable are grouped by similar characteristics, lowering entropy, and at this time, information gain occurs. The upper layer of the value keeper is determined by the information gain, and the variable that makes the information gain the greatest is selected.

In C5.0, the classification forecast is clf. The predictor() function is used, and the entropy model uses the tree. DecisionTree() function by the target variable. Figure 2 shows the entropy and tree structure by the Python program.

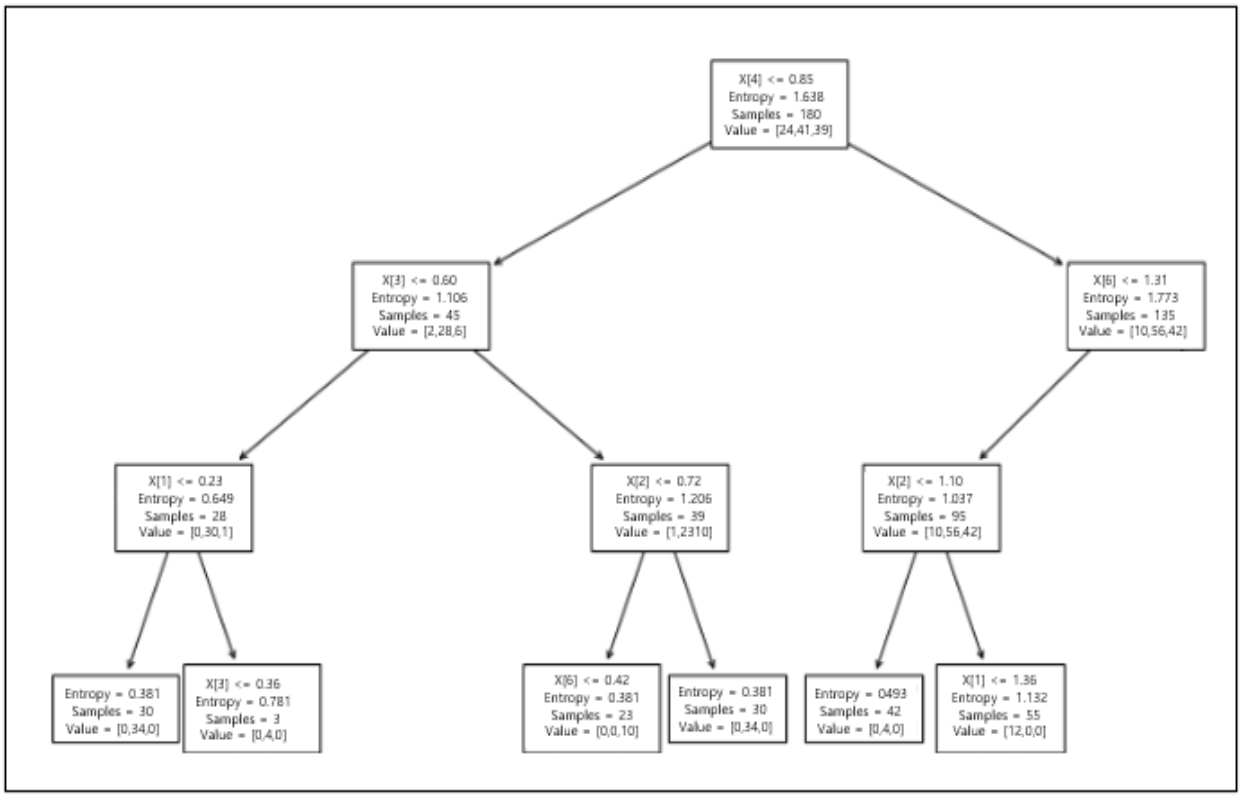

Figure 2. Tree structure by Python program

In the DT analysis by Python and R programs, when the entropy becomes 0, the impurity concentration becomes 0%, so it can be said that it is classified with only one piece of information.

The structure in Figure 2 shows the visualization of the DT according to the results after pruning according to the internal criteria based on the data when using the Python program. As for the meaning of the text in the square box, square brackets ([]) mean each variable, and information on entropy information and branching point settings based on entropy values is shown below. If you use the R program, it will show a slightly different shape and text.

3. Experiments

In this study, the DT analysis by the Python program is pruned through the DC process, and by finding out which variables are influential according to the decision-making process, we find out whether the DT analysis by the Python program is also a reliable result. see. The experiment used a Python program to find out which variables determine the addictiveness of smartphone use, and a total of 8 variables were used, and the number of data was 232. In this experiment, in order to evaluate whether smartphones are addictive, we will

sequentially classify data that have a lot of influence on smartphone addiction through DT analysis using the values of each variable. By classifying data that affect addiction, we find out which variables have an effect. At this time, it is checked whether the information classified by the DT analysis is a reliable classification result. Therefore, in this experiment, based on the prepared data, DT analysis is used to classify the data according to whether or not the smartphone is Middle Eastern, and confirm the classification criteria.

3.1 DT Experiment

Previously, we looked at the pruning model according to data classification when using R code and Python code. Here, we classify the data using Python code based on the actual data to be analyzed, and check the pruning process according to the pruning criteria.

We use a Python program to check which variables have an influence on smartphone addiction. First, the data structure for 8 variables of 232 data that will determine smartphone addiction is confirmed in matrix and table structures. Figure 3 shows the summary information of the data output by the Python code and the command to show the data structure with information summarized based on the entire data.

```
uci_path = open('dt_corr_exdata(eng).csv')
df = csv.reader(uci_path)
next(df)
df = pd.read_csv(uci_path, header=None)
print(df)
```

```
     0  1  2  3    4    5    6      7
0    1  1  2  2   60   30   no  1.000
1    1  1  3  2  180   60   no  2.375
2    2  1  2  2   60  120   no  2.000
3    1  1  1  2   30   30   no  1.750
4    2  1  1  2  120   60   no  2.500
..  .. .. .. ..  ...  ...  ...    ...
227  1  2  2  3  420  300  yes  2.875
228  1  2  3  3  240  240  yes  2.250
229  2  2  2  3  240  240  yes  2.375
230  2  1  2  3  240   30  yes  2.625
231  1  2  2  2   60   30  yes  2.000

[232 rows x 8 columns]
```

Figure 3. Python code and summary information

First, after importing the data file to be analyzed using Python code, it is set to be advantageous for analysis through a simple pre-processing process. Then, the file compressed in CSV format is read in CSV format. The CSV file is a file format in which all variables to be used for analysis are separated with commas (,). If you issue a Python code command up to this point and then output it, all the data read from the data file will be spread out on the monitor screen. When the screen space is insufficient, a dot (...) is displayed.

Figure 4 shows factor variables that affect smartphone addiction, order in memory for each variable, variable name, missing value, data type, data structure, etc. As for the overall data type, float64(1) indicates that there is one real number variable and int64(7) indicates that there are 8 integer type variables. At this time, the size of the memory used is additionally notified that 14.6 KB is being used.

```
df.columns = ['S_type', 'Gender', 'Std_grade', 'S_living', 'S_time', 'SNS_time', 'Addiction', 'Impulsiveness']
print(df.info())
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 232 entries, 0 to 231
Data columns (total 8 columns):
 #   Column          Non-Null Count  Dtype
---  ------          --------------  -----
 0   S_type          232 non-null    int64
 1   Gender          232 non-null    int64
 2   Std_grade       232 non-null    int64
 3   S_living        232 non-null    int64
 4   S_time          232 non-null    int64
 5   SNS_time        232 non-null    int64
 6   Addiction       232 non-null    int64
 7   Impulsiveness   232 non-null    float64
dtypes: float64(1), int64(7)
memory usage: 14.6 KB
```

Figure 4. Python code and data structure information

Looking at the composition of each number, the names of the eight variables are S_type, Gender, Std_grade, S_living, S_time, SNS_time, Addiction, and Impulsiveness. For each variable, Impulsiveness is a float64 real number variable, and all other variables are int64 integer type numeric variables.

Second, the training set and the test set are separated based on the entire data. The training set to be separated is set to 70% (162) of the total, and the test set is set to 30% (70). Figure 5 means a command to set the separation criterion to 70% for training set and 30% for test using Python code. It shows the calculation of classification criteria for training and test data corresponding to each variable based on the entire data.

```
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3, random_state=10)
print(X_train, X_test, y_train, y_test)
[[-1.06228074  0.98319208  1.83345889 ...  0.29714654  0.9327269
  -0.21824767]
 [ 0.94137074 -0.98319208  1.83345889 ... -0.87526837 -0.67494502
  -1.17273839]
 [ 0.94137074 -0.98319208 -1.0406118  ... -0.28906092  0.28965813
  -1.80906553]
 ...
[-1.06228074 -0.98319208  0.39642354 -1.62076803 -0.28906092 -0.35341064
  1.05440662]
[-1.06228074  0.98319208 -1.0406118  -0.29676034  0.88335399  0.9327269
  0.0999159 ]] 111    1
```

Figure 5. Train set, test set classification standard value

Figure 6 shows the results of separating data for training and testing by the DC standard value according to Figure 5. Looking at Figure 6, it can be seen that the instruction set is divided into 162 out of 232 data sets, and the test set is divided into 70 out of 232 data sets. As an additional study, in actual R code, it is necessary to check how many data are divided into each when data is separated by the same criterion. Looking at the additional information in Figure 6, there are numerical expressions represented by 1 and 2, which contain information on smartphone addiction. 1 means not poisoned, 2 means poisoned.

```
198     2
91      1
120     1
194     2
       ..
64      1
15      1
228     2
125     1
9       1
Name: Addiction, Length: 162, dtype: int64 26      1
220     2
61      1
161     2
165     2
       ..
231     2
39      1
201     2
159     2
83      1
Name: Addiction, Length: 70, dtype: int64
```

Figure 6. Separation of training set (70%) and test set (30%)

The upper part of Figure 6 is the training set, and among the total 232 data, 162 data were classified as training data by the pruning function. The lower part is the test set, which informs that 70 data out of a total of 232 data are classified as test data.

Third, check the input values and target values for the training set and test set. Each value can be checked using 'train.shape' and 'test.shape'. Figure 7 shows the input values and target values of the training set (left) and test set (right), respectively.

```
(162, 7)
[[-1.06228074  0.98319208  1.83345889 ...  0.29714654  0.9327269
  -0.21824767]
 [ 0.94137074 -0.98319208  1.83345889 ... -0.87526837 -0.67494502
  -1.17273839]
 [ 0.94137074 -0.98319208 -1.0406118  ... -0.28906092  0.28965813
  -1.80906553]
 ...
 [-1.06228074  0.98319208  1.83345889 ...  0.88335399  1.57579567
   0.0999159 ]
 [ 0.94137074  0.98319208  1.83345889 ...  0.29714654 -0.35341064
  -0.85457482]
 [ 0.94137074  0.98319208 -1.0406118  ... -0.87526837 -0.67494502
  -0.21824767]]

(162,)
111    1
198    2
91     1
120    1
194    2
      ..
64     1
15     1
228    2
125    1
9      1
Name: Addiction, Length: 162, dtype: int64
```

(a)train set

```
(70, 7)
[[-1.06228074 -0.98319208  0.39642354 -1.62076803 -1.1683721  -0.78212315
  -0.53641124]
 [ 0.94137074  0.98319208  1.83345889  1.02724734 -1.26607334 -0.67494502
  -0.53641124]
 [ 0.94137074  0.98319208  0.39642354 -0.29676034 -0.28906092 -0.35341064
  -0.85457482]
 [-1.06228074 -0.98319208  0.39642354 -0.29676034  1.46956145 -0.67494502
  -0.21824767]
 [-1.06228074  0.98319208 -1.0406118  -0.29676034  1.46956145  2.21886443
   0.41807947]
 [-1.06228074  0.98319208 -1.0406118  -0.29676034  0.88335399  0.9327269
   0.0999159 ]]

(70,)
26     1
220    2
61     1
161    2
165    2
      ..
231    2
39     1
201    2
159    2
83     1
Name: Addiction, Length: 70, dtype: int64
```

(b) test set

Figure 7. Input value and target value of training set

Fourth, set the entropy for the evaluation index of the target variable and proceed with modeling. Modeling is a step in finding a value with less entropy because the information gain increases as the entropy decreases. The section where the range of change in the entropy value increases becomes the depth of pruning. Figure 8 is the code that proceeds with modeling based on the value of entropy and the result of the code.

```
from sklearn import tree
tree_model = tree.DecisionTreeClassifier(criterion='entropy', max_depth=5)
print(tree_model)
```

```
DecisionTreeClassifier(criterion='entropy', max_depth=5)
```

Figure 8. Entropy Calculation and Value Keeping Depth Settings

Executed by DecisionTreeClassifier(), entropy is calculated internally, and the maximum depth of pruning is limited to 5 steps.

Fifth, training data is learned as input and target values to predict the model and determine the depth of pruning. Since the entropy calculation and pruning depth were set in Figure 8, the pruning model set based on the addiction variable is predicted here. Figure 9 shows the steps of predicting the model based on addiction variables according to the calculated pruning criteria.

```
DecisionTreeClassifier(criterion='entropy', max_depth=5)

[1 1 1 2 2 1 2 2 2 2 1 1 1 1 1 1 1 2 1 1 1 1 1 2 2 1 1 2 2 1 1 1 1 1 1 2
 1 2 1 2 1 1 1 1 1 1 1 2 1 2 1 1 2 1 1 1 1 1 1 1 1 2 1 1 1 1 1 1 1 2]
```

Figure 9. Prediction of pruning model based on addiction variables

Sixth, the presence or absence of smartphone addiction is predicted by the output of the confusion matrix for the test set. In the confusion matrix, rows represent actual values and columns represent predicted values. Then, since the result of the confusion matrix in this experiment is '[[35 9][16 10]]', the position of each value is '35:[0][0], 9:[0][1], 16:[1][0], 10:[1][1]', and it is the same as the matrix structure. Please note that the preceding symbol ([]) is a mark indicating the index of memory allocation criteria by the Python program, and does not mean references. In other words, if the model is predicted based on the total 70 data in the test set, the row is the actual value, so 35 cases were selected as no when not addicted to smartphones, and 9 cases were incorrectly selected as yes. Also, when smartphone addiction is correct, there are 16 wrongly selected no and 10 wrongly selected yes.

Seventh, the results of DC with a matrix structure are confirmed with statistics. The table according to the DC results contains various information about smartphone addiction. Among them, addiction information (1,2), accuracy (0.64), other statistics, and test set were analyzed, so support information is marked as 70. Figure 10 shows the results of DC based on the test set as a table.

	precision	recall	f1-score	support
1	0.69	0.80	0.74	44
2	0.53	0.38	0.44	26
accuracy			0.64	70
macro avg	0.61	0.59	0.59	70
weighted avg	0.63	0.64	0.63	70

Figure 10. DC result table based on test set

After training 162 data based on the total data, a test was conducted using 70 data to classify according to 'addiction' and 'non-addiction'. As a result, the accuracy of the DC rate was 0.64%, indicating an accuracy of

64%. In Figure 11, the DC rate is 0.64% even by the Python code, showing the same result as Figure 10. This can cause accuracy fluctuations depending on the amount of data, whether large or small. However, from the perspective of the AI system, it can be said that the degree of accuracy that is difficult to apply directly to the AI system has come out, so it can be seen that there is a need to improve the classification rate.

Eighth, the accuracy is calculated and displayed based on the results of the DC. Accuracy calculation in Python code is done with the same command as in Figure 11.

Figure 11. Accuracy calculation of Python code

As can be seen in Figure 11, the accuracy of the measured DC is 0.642%, which is about 64% accuracy. This cannot be seen as a high level of accuracy from the point of view of accuracy, so it can be seen that the accuracy is somewhat low. Low accuracy cannot be directly transferred to AI function implementation, and additional data purification and collection will have to be done. In addition, at least for this part, the need to analyze again using other statistical tools has been raised, so related research will be conducted in the future. Ninth, pruning is performed based on the classified results, and a tree structure is drawn based on a value with a large information gain as a factor influencing smartphone addiction. In the tree structure, the upper node is the parent node, and the lower node is the child node. The factor variables of the parent node can be interpreted in the same meaning and are composed of variables that can exert important influence on the child nodes. The entropy of the parent node has a greater value than the entropy of the child node. In other words, entropy increases as you go up, so the information gain decreases. However, on the contrary, as the entropy goes down, the information gain increases as the entropy decreases, so it is possible to set a reference value for classifying data. Therefore, among all nodes, the parent node is composed of variables that can exert influence on child nodes. (Figure 12) is a tree structure created by calculating entropy and information gain at each step after pruning according to DC criteria by Python commands. In the tree structure of (Figure 12), the X[number] mark means each variable used in smartphone addiction analysis, and the X[number] mark according to each variable can be expressed as in Table 2.

Table 2. Display X[number] according to each variable

variable name	X[number]표시	variable name	X[number]표시
S_type	X[0]	S_time	X[4]
Genger	X[1]	SNS_time	X[5]
Std_grade	X[2]	Impulsiveness	X[6]
S_living	X[3]		

The tree structure in Figure 12 must have been created by repeating the process of selecting the parent node by issuing the highest information gain at the lowest value of entropy. Then, in the tree structure, the factor variable that has the highest influence on smartphone addiction is X[4], which points to the S_time variable,

which means smartphone use time. In other words, it can be seen that the greatest influence on addiction is that the possibility of falling into addiction is the highest when a smartphone is used for a long time regardless of time and place. Second, X[6] refers to the impulsiveness variable that impulsively uses a smartphone. Therefore, the most influential variable is the S_time variable, and the second most influential variable is the Impulsiveness variable. The third variable that has an impact is the SNS_time variable, and it can be seen that the variable that uses a lot of various social networks such as KakaoTalk, Messenger, FateBock, Twitter, and Instagram has an effect on addiction. Next, according to S_type, addiction is relatively higher in high school students than in middle school students, and in the case of gender variables, differences in addiction also occur in male and female cases. Also, between the S_type variable and the Gender variable, there may be a change in influence due to the S_living variable.

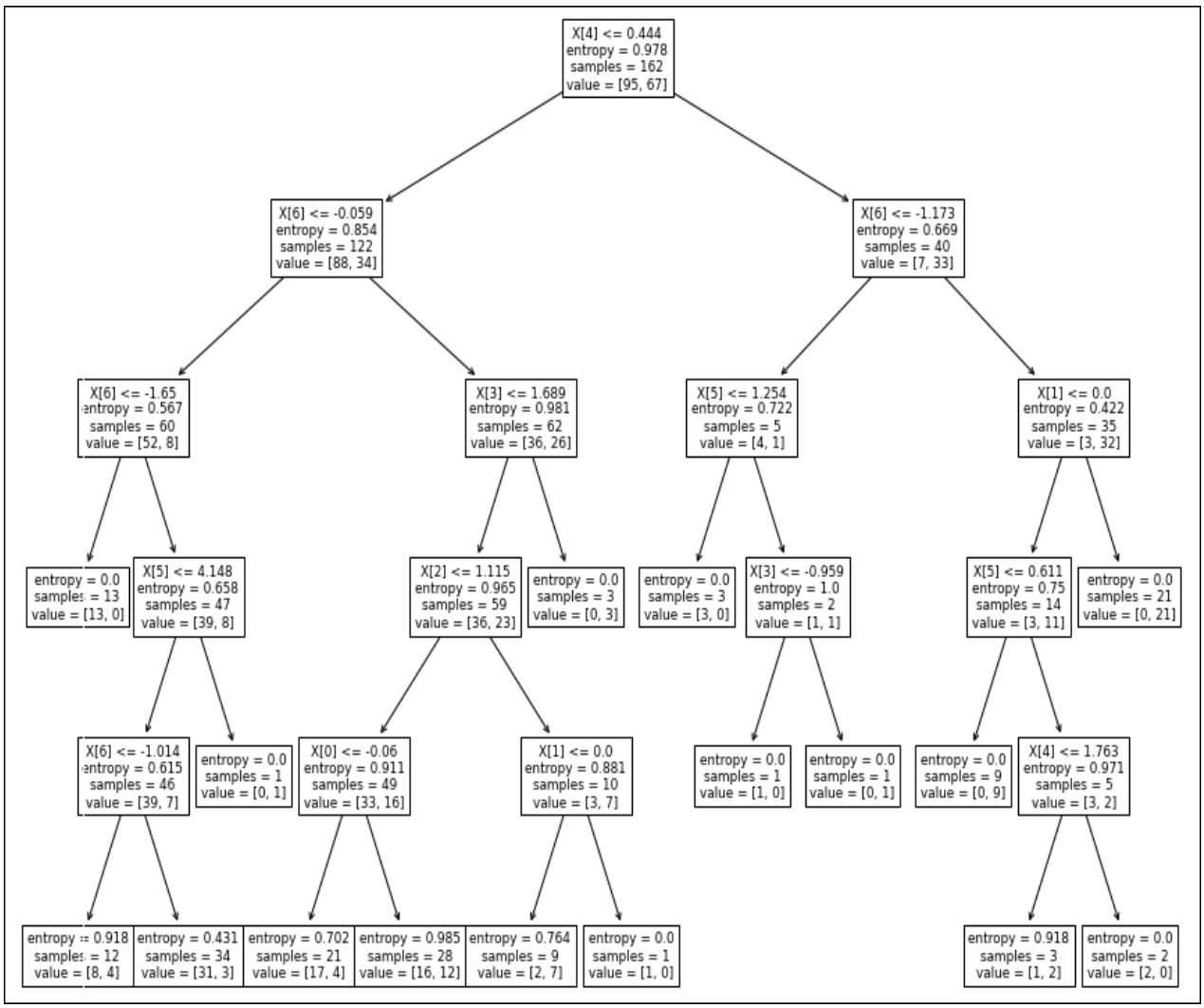

Figure 12. Creation of tree structure by entropy and information gain

So far, we have applied DT analysis to find out what variables have an influence on smartphone addiction using a Python program. In conclusion, the accuracy of the DT analysis using the Python program was about

65%, and although the pruning value by DC showed a slightly lower accuracy, the data were classified meaningfully in selecting factor variables that affect smartphone addiction. That is, as the factor variable having the greatest influence, the S_time variable was selected and selected as the topmost root node. As a child node, the Impulsiveness variable was set as a child node, and the lower child nodes were set as SNS_time, S_type, and S_living variables, so that the model was appropriately estimated according to the influence on smartphone addiction.

4. Conclusion

AI technology, which surprised the world, is making rapid progress through technologies called ML and DL. AI technology will be utilized not only in fragmentary cases or environments that are easy to use, but in highly complex and diverse environments. In order to utilize the data for the purpose, it must be possible to accurately classify the data, and the classified data must always have high reliability. In this study, among various statistical tools, a Python program was used to find out what variables would lead to addiction in smartphone use through DC.

First, after classifying the data using a DT, the tree structure pruned by the classification criteria and the corresponding variables were checked. As a result of the DT analysis process, pruning was performed between the Impulsiveness variable, which means addiction among 8 variables, and the other 7 variables according to the DC criteria. As a result of the pruning of the DT analysis, it was determined that the S_time variable had the greatest influence on smartphone addiction and was selected as the highest root node. Next, SNS_time, a variable with high social network usage, was selected by the root node as a child node. Next, the variables S_type, Std_grade, and S_living were selected as child nodes. Therefore, based on the data classified by the Python code, it is possible to know what kind of relationship the variables have. Through the results of this study, it was confirmed that DC is very important in implementing AI functions and that there is no problem in performing BDA using any of the various statistical tools such as Python and R when analyzing big data.

In the future research, we will proceed with the study of the part where additional experiments should be continued in this study and other methods such as DC analysis, correlation degree between variables and difference analysis between data distribution standards. In addition, we will proceed with consideration of various methods and alternatives for sophisticated DC.

References

[1] Suchul Lee and Mihyun Ko, “Exploring the Key Technologies on Next Production Innovation”, Journal of the Korea Convergence Society, Vol. 9, No. 9, pp. 199-207, 2018.
DOI: https://doi.org/10.15207/JKCS.2018.9.9.199

[2] T. M. Mitchell, “The discipline of machine learning(Vol. 9)”, Carnegie Mellon University, Shcool of Computer Science, MachineLearning Department, 2006.
DOI: https://www.researchgate.net/publication/268201693_The_Discipline_of_Machine_Learning

[3] Min, Meekyung, “Classification of Seoul Metro Stations Based on Boarding/ Alighting Patterns Using Machine Learning Clustering,” The Journal of The Institute of Internet, Broadcasting and Communication (IIBC), Vol. 18, No. 4, pp. 13-18, 2018(8).
DOI: https://doi.org/10.7236/JIIBC.2018.18.4.13.

[4] Jihyun Lee, Jiyoung Woo, Ah Reum Kang, Young-Seob Jeong, Woohyun Jung, Misoon Lee and Sang Hyun Kim, “Comparative Analysis on Machine Learning and Deep Learning to Predict Post-Induction Hypotension”, 2020 by the authors. Licensee MDPI, Basel, Switzerland, Sensors 2020, 20(16), 4575, 2020.
DOI: https://doi.org/10.3390/s20164575

[5] Joerg Evermann, Jana-Rebecca Rehse and Peter Fettke, “Predicting process behaviour using deep learning”, Decision Support Systems, Vol. 100, pp. 129-140, 2017(8)
DOI: https://doi.org/10.1016/j.dss.2017.04.003

[6] Hyung-Woo, Lee, "Development of Supervised Machine Learning based Catalog Entry Classification and Recommendation System", Journal of Internet Computing and Services, Vol. 20, Issue 1, pp. 57-65, 2019.
DOI: http://doi.org/10.7472/jksii.2019.20.1.57

[7] Se Hoon Jung, Jong Chan Kim, Kim Cheeyong, Kang Soo You and Chun Bo Sim, "A Study on Classification Evaluation Prediction Model by Cluster for Accuracy Measurement of Unsupervised Learning Data", Journal of Korea Multimedia Society Vol. 21, No. 7, pp. 779-786, 2018(7).
DOI: https://doi.org/10.9717/kmms.2018.21.7.779

[8] Hayoung Eom, Jeonghwan Kim, Seungyun Ji and Heeyoul Choi, "Autonomous Parking Simulator for Reinforcement Learning", Journal of Digital Contents Society, Vol. 21, No. 2, pp. 381-386, 2020(2)
DOI: DOI : 10.9728/dcs.2020.21.2.381

[9] Kim, Pan Jun, "An Analytical Study on Automatic Classification of Domestic Journal articles Using Random Forest", Journal of the Korean Society for information Management , Vol. 36. Issue. 2, pp. 57-77, 2019(6).
DOI: https://doi.org/10.3743/KOSIM.2019.36.2.057.

[10] Yang, Jae-Wan, Lee, Young-Doo and Koo, In-Soo, "Sensor Fault Detection Scheme based on Deep Learning and Support Vector Machine", The Journal of The Institute of Internet, Broadcasting and Communication (IIBC), Vol. 18, No. 2, pp. 185-195, 2018(4).
DOI:http://doi.org/10.7236/JIIBC.2018,18.2.185.

[11] Kyoungho Choi and Jin Ah Yoo, "A reviews on the social network analysis using R", Journal of KIISE, JOK:software and application, Vol. 6, No. 1, 2015.
DOI: http://dx.doi.org/10.15207/JKCS.2015.6.1.077

[12] Youngseok Lee, "Python-based Software Education Model for Non-Computer Majors", Journal of the Korea Convergence Society, Vol. 9, No. 3, pp. 73-78, 2018.
DOI: https://doi.org/10.15207/JKCS.2018.9.3.073

[13] Y. J. Kim, J. W. Ryu, W. M. Song and M. W. Kim, "Fire Probability Prediction Based on Weather Information Using Decision Tree", Journal of KIISE, JOK:software and application, Vol. 40, No. 11, 2013(11).
DOI: http://www.dbpia.co.kr/journal/articleDetail?nodeId=NODE02283313

[14] K. N. Lee and H. C. Lee, "A Study on the Combined Decision Tree(C4.5) and Neural Network Algorithm for Classification of Mobile Telecommunication Customer", Korea Intelligent Information Systems Society, Vol. 9, No. 1, pp. 139-155, 2003(6).
DOI: http://jiisonline.evehost.co.kr/files/DLA/8-9-1.pdf

[15] T. B. Yoon and J. H. Lee, "Design of Heuristic Decision Tree (HDT) Using Human Knowledge", Korean Institute of Intelligent Systems, Vol. 19, No. 1, pp. 161-164, 2009(4)
DOI: https://doi.org/10.5391/JKIS.2009.19.4.525.