

SUPERVISED TECHNIQUES IN DATA MINING

**A Thesis Submitted to the
Graduate School of Natural and Applied Sciences of Dokuz Eylül University
In Partial Fulfillment of the Requirements for the Degree of Doctor of
Philosophy in Computer Engineering, Computer Engineering Program**

**by
Mehmet Seval KAYGULU**

February, 2009

İZMİR

Ph.D. THESIS EXAMINATION RESULT FORM

We have read the thesis entitled “**SUPERVISED TECHNIQUES IN DATA MINING**” completed by **MEHMET SEVAL KAYGULU** under supervision of **PROF.DR. ALP KUT** and we certify that in our opinion it is fully adequate, in scope and in quality, as a thesis for the degree of Doctor of Philosophy.

.....
PROF.DR. ALP KUT

Supervisor

.....
DOÇ.DR. YALÇIN ÇEBİ

Thesis Committee Member

.....
PROF.DR. HÜLYA İNANER

Thesis Committee Member

.....
YARD.DOÇ.DR. GÖKHAN DALKILIÇ

Examining Committee Member

.....
DOÇ.DR. BİRGÜL KUTLU

Examining Committee Member

Prof.Dr. Cahit HELVACI
Director
Graduate School of Natural and Applied Sciences
Fen Bilimleri Enstitüsü

ACKNOWLEDGMENTS

First of all I respectfully remember my deceased Prof.Dr. Esen Özkarahan. I would like to thank very much to Alp Kut who managed the studies, to Hülya İnaner and Yalçın Çebi for their views, warnings and helps for finding data. They were not only academic advisors, but were like friends who support me in everything I do. Also I would like to Prof.Dr. Eran Nakoman for his permission to use the data.

Mehmet Seval KAYGULU

SUPERVISED TECHNIQUES IN DATA MINING

ABSTRACT

Usage of Data Mining techniques is very common for reaching info on huge database. Techniques especially canalized by the user are used in this study. Theory of Data Mining is shortly described in first 6 chapters. Subjects are: learning and reaching info methods, Database Operational System types and selection, organizing data, removal of problems related data and presentation of obtained results.

Data mining application is very common on especially commercial and medical areas. However, known application has not been encountered in earth sciences. Therefore, data is being used which obtained from Seyitömer Coal Basin in this application. When data examined: it is noted that there is no standardization for material naming. First of all, it is tried to hinder to name material in different ways at the stage of forming database. Summarized info is being represented after entering the data. Even if summary is not canalized by the user, it is added to the application because it may help to searcher. User chooses the material. Finds the first layer met for the chosen material in bore-hole. Therefore, reaches the material list takes place above this layer. Besides finds the last layer met. And obtains the material list takes place under this layer.

User may wish to group some material under same name. And can re-organize the database according to this. The above described studies can be applied on this new database. This application also obtains vertical cross-section diagram drawing. At last, user can classify bore-holes according to code of layer which chosen material first met. The result of this procedure is represented on a plane by using different colored points to the researcher.

Keywords : Data mining, database, Seyitömer Coal Basin, application for coal beds.

VERİ MADENCİLİĞİNDE YÖNLENDİRİLMİŞ TEKNİKLER

ÖZ

Büyük boyuttaki veri tabanlarından bilgiye ulaşmak için veri madenciliği tekniklerinin kullanımı çok yaygınlaşmıştır. Bu çalışmada özellikle kullanıcı tarafından yönlendirilmiş teknikler üzerinde durulmuştur. İlk altı bölümde veri madenciliğinin teorisi kısaca açıklanmıştır. Üzerinde durulan konular öğrenme ve bilgiye ulaşma yöntemleri, veri tabanı işletim sistemi tipleri ve seçimi, verilerin düzenlenmesi, veriler ile ilgili sorunların giderilmesi, elde edilen sonuçların sunulmasıdır.

Özellikle tıp ve ticaret alanlarında veri madenciliği uygulamalarına çokça rastlanılmaktadır. Ancak, yer bilimlerinde bilinen bir uygulamasına rastlanılmamıştır. Bu nedenle, uygulamamızda Seyitömer Kömür Havzasından elde edilen veriler kullanılmıştır. Veriler incelendiğinde malzeme isimlendirmede bir standardın olmadığı görülmüştür. Öncelikle veri tabanının oluşturulması aşamasında bir malzemenin farklı şekillerde isimlendirilmesi engellenmeye çalışılmıştır. Verilerin girilmesinden sonra özet bilgiler sunulmaktadır. Her ne kadar özetleme kullanıcı tarafından yönlendirilmiyor ise de, araştırmacıya fayda sağlayabileceği düşünülerek uygulamaya eklenmiştir. Bunun dışında, kullanıcı bir malzeme seçer. Kuyularda seçilen malzemenin ilk rastlandığı katman bulunur. Böylece bu katmanın üstünde yer alan malzemelerin listesine ulaşılır. Ayrıca son rastlandığı katman bulunur. Ve bu katmanın altında yer alan malzemelerin listesi elde edilir.

Kullanıcı bazı malzemeleri aynı bir isim altında gruplamak isteyebilir. Veri tabanını da yaptığı gruplandırmaya göre yeniden düzenleyebilir. Bu yeni veri tabanı üzerinde de yukarıda anlatılmış olan çalışmaları uygulayabilir. Uygulama ayrıca kuyuların düşey kesit diyagramının çizilmesini sağlamaktadır. Son olarak kullanıcı, seçtiği bir malzemenin kuyularda ilk rastlandığı katmanın kotuna göre kuyuları sınıflandırabilir. Bu işlemin sonucu farklı renkli noktalar kullanılarak bir düzlem üzerinde araştırmacıya sunulmaktadır.

Anahtar sözcükler : Veri madenciliđi, veri tabanı, Seyitömer Kömür Havzası, kömür yatakları için bir uygulama.

CONTENTS

	Page
THESIS EXAMINATION RESULT FORM	ii
ACKNOWLEDGEMENTS	iii
ABSTRACT	iv
ÖZ	v
CHAPTER ONE – INTRODUCTION	1
1.1 Why Do We Need Data Mining?	1
1.2 Overview to Knowledge Discovery in Databases Process	2
CHAPTER TWO – TYPES OF LEARNING	6
2.1 Inductive Learning.....	7
2.1.1 Models	7
2.1.1.1 Environment	8
2.1.1.2 Classes.....	9
2.2 Learning	9
2.2.1 Supervised learning.....	10
2.2.2 Unsupervised learning.....	10
2.3 Quality	11
2.4 Machine Learning.....	11
CHAPTER THREE – DATA MINING	12
3.1 The Comparison of Machine Learning With Data Mining.....	12

3.2 The Training Set.....	13
CHAPTER FOUR – SEARCH ALGORITHMS.....	15
4.1 Search Space	15
4.1.1 Description space	15
4.1.2 Operations.....	16
4.1.3 Domains of the attributes.....	16
4.1.4 Quality function	17
4.2 Limitations on the Operations	21
CHAPTER FIVE – PROBLEMS.....	23
5.1 Limited Information	23
5.1.1 Incomplete information	23
5.1.2 Sparse data.....	24
5.2 Data Corruption.....	24
5.2.1 Noise.....	24
5.2.2 Missing attribute values.....	25
5.3 Databases	26
5.3.1 Size of database.....	26
5.3.2 Updates.....	27
CHAPTER SIX – KNOWLEDGE REPRESENTATION.....	28
6.1 Propositional-Like Representations.....	28
6.1.1 Decision trees.....	29

6.1.2 Production rules	29
6.1.3 Decision list	30
6.1.4 Ripple-down rule sets.....	30
6.2 First Order Logic	31
6.3 Structured Representations	32
6.3.1 Semantic nets	32
6.3.2 Frames and schemata	33
CHAPTER SEVEN – APPLICATION PROGRAM.....	35
7.1 Introduction	35
7.2 Data Examination	35
7.2.1 Introduction of data	35
7.2.2 Choosing the database management system.....	37
7.2.3 Introduction of data sheet	38
7.3 Windows of the Program	39
7.3.1 Main Window	39
7.3.2 Window of “ Malzeme Tanımlama ”	40
7.3.3 Window of “ Kuyular ”	45
7.3.4 Window of “ Katmanlar ”	47
7.4 Beginning of Supervised Data Mining	53
7.4.1 Beginning of data mining processes	53
7.4.2 Main Interface for Examination on Input Data.....	55
7.4.3 Reduction of Raw Data for Creation New Database	57
7.4.3.1 Reduction of material names.....	57
7.4.3.2 Creation new database with renamed materials	62
7.4.3.3 Removing materials rarely met	69

7.4.4 Plotting the location of bore-hole	69
7.5 Discussion	70
CHAPTER EIGHT – CONCLUSION	74
REFERENCES	75

CHAPTER ONE

INTRODUCTION

A database is a store of non-trivial information. Most important purpose of a database is the efficient retrieval of information. This retrieval information can be a copy of information stored in a database. Some important information can be hidden in a database, so that, this hidden information must be inferred from the database. This information is not only statistical, but a relation between attributes of the database.

Data mining is the automatic process of handling of information from databases which can not be seen directly. Data mining is used for finding useful trends and patterns. In some articles and documents, the term data mining is given the same meaning with knowledge discovery in databases (KDD), means an automatic process of non-trivial extractions, formerly unknown and useful information (including rules, constrains and regularities) from data in databases. There are many other terms, with similar meaning or small difference in meaning, such as knowledge mining from databases, knowledge extraction, data archaeology, data dredging, data analysis, etc. Some researchers (U. Fayyad, G. Piatetsky-Shapiro, P.Smyth) suggest that data mining is only one of the steps of KDD.

1.1 Why Do We Need Data Mining?

Traditionally, analysts used manual process for analysis. If statistical techniques are used to generate reports, analysts must familiar with the data. Data that are used in statistics, is a small part of whole knowledge and coincidentally chosen. But now, this process is very difficult, expensive and slow because of rapid growth of data and increasing number of attributes in databases. According to some observations, amount of data is being two times of old data at every eighteen months. On the other hand, this process, at the same time, is subjective.

We can store and access to reliable data efficiently and inexpensively with using current hardware and database technology. In raw form, datasets about business management, government administration, medicine, science or engineering have little value. Databases are calm resources that have potential to yield important benefits.

No one could organize billions of records, each having tens or hundreds of fields and, extract knowledge from such databases. These processes are over the human ability. We need new techniques and tools for knowledge discovery or extraction in databases.

1.2 Overview to Knowledge Discovery in Databases Process

In this section, it is accepted that knowledge discovery from databases (KDD) includes all steps for finding useful patterns in data. Data mining is only a particular step of KDD. Other steps are data preparation and selection, data cleaning, incorporation of prior knowledge, and interpretation of the result of mining.

KDD has evolved, and continues to evolve, from the intersection of research in such fields as databases, machine learning, pattern recognition, statistics, artificial intelligence and reasoning with uncertainty, knowledge acquisition for expert systems, data visualization, machine discovery, information retrieval, and high-performance computing. KDD software systems incorporate theories, algorithms, and methods from all of these fields. (Fayyad, U., Piatetsky-Shapiro, G., Smyth, P., 1996, p.29)

KDD is interested in the all processes of knowledge discovery from datasets, such that how data is stored (types of database such as relational database, hierarchical database, network database, types of data such as numbers, text, images and voice), which algorithms can be chosen that run efficiently on huge data sets, how results can be interpreted, how interpreted results can be visualized and how user interface can be modeled. And also, KDD interests in noise in data sets.

Statistics has much related with KDD. Handling patterns and knowledge inference has been a component of statistics. A statistician can find patterns if the statistician searches in any dataset sufficiently. These patterns can be seen significant from the view of statistics. But they are not significant in real world. To find nontrivial pattern is very important for KDD. Activity of understanding how to find these patterns correctly is data mining. KDD includes larger views of modeling than statistics. Aim of KDD is to provide tools that whole process of data analysis can be done. U. Fayyad defined the KDD process as: “The nontrivial process of identifying valid, novel, potentially useful, and ultimately understandable patterns in data” (Fayyad & his friends)

In above definition **data** is a set of facts, **pattern** is a description of a subset of the data. The steps of data preparation, search for patterns, knowledge retrieval and refinement (all these steps are in multiple iteration) are named as **process**. The process is assumed to be **nontrivial**. The found pattern should be **valid** for new data. It is preferable that these patterns are **novel** at least to the system and to the user, and **potentially useful** for the user. And the patterns should be **understandable**.

There are many interactive and iterative (because, user can make many decisions) steps.

i. *Learning the application domain:*

In this step, analyst should search and understand prior knowledge and the aim of the application.

ii. *Creating a target dataset:*

In this step, analyst should select a dataset or data samples on which knowledge discovery is performed.

iii. *Data cleaning and preprocessing:*

In this step, analyst should remove noise, collect the necessary information to model, decide on strategies for getting missing data and decide database management

system (DBMS) problems such as data types, schema and mapping of unknown values.

iv. *Data reduction and projection:*

In this step, analyst should find features to represent the data and, reduce the number of variables under consideration or to find constant representations for data, depending on the aim of the application.

v. *Choosing the function of data mining:*

In this step, analyst should decide the purpose of the model such as summarization, classification, regression, and clustering, that is derived by the data mining algorithm.

vi. *Choosing the data mining algorithm:*

In this step, analyst should search for patterns in the data in a particular representational form.

vii. *Data mining:*

In this step, analyst should decide which models and parameters are used for patterns in the data.

viii. *Interpretation:*

In this step, analyst should interpret the extracted patterns including visualization of these patterns and translating into terms understandable by users.

ix. *Using discovered knowledge:*

Incorporating the discovered knowledge into the performance system, trying to find out the conflicts between the knowledge acquired and the one previously extracted and taking action related with the knowledge which take place in making use of the discovered knowledge.

CHAPTER TWO

TYPES OF LEARNING

The first purpose of database which is store of true information is to retrieve efficient and useful knowledge. This knowledge sometimes can be of hidden form. Therefore, we must have some techniques to infer that hidden knowledge. From a logical point of view, there are two techniques to infer knowledge.

The first one of these two techniques is *deduction*. Inferred knowledge by deduction technique is a logical consequence of the information in the database. For example, many engineers work at region of coal bed. Each engineer manages to drill many bore-hole. Also, there are some kinds of drilling machines. One of those machines is used at any bore-hole. We can infer the list of the name of engineers and the brand of drilling machines related with the engineers. This knowledge can be inferred from the database with applying the join operation between two relational tables such as ENGINEER-BOREHOLE and MACHINEBRAND-BOREHOLE.

The second is *induction* technique. Generalized information can be inferred from the information in the database. For example, the knowledge “each drilling machine is used by at least one engineer” might be inferred from ENGINEER-BOREHOLE and MACHINEBRAND-BOREHOLE relational tables. This is higher-level knowledge than inferred knowledge from the database by induction technique. If we can formulate this higher-level knowledge, we can predict the value of an attribute in terms of other attributes.

The knowledge inferred by induction technique may not be always true in the real world; it is only supported by the database. By the knowledge inferred by deduction technique is probably correct in the real world that is provided that the database is correct. Therefore we must carefully select the regularities that they are plausible and supported by the database.

2.1 Inductive Learning

Humans try to understand their environment by simplifying it. Simplification of this environment is called a *model*. *Inductive learning* is the process of creation of a model of environment. During this process, cognitive system observes its environment. It recognizes similarities among objects and events in this environment. Cognitive system uses similar objects to make a class. It constructs rules for the behavior of the members of a class. The set of rules of a class is called *class description*.

There are mainly two learning techniques. In *supervised learning*, classes are defined and examples of each class are given to the cognitive system by someone, let's say a teacher. The system will construct the class descriptions by discovering common properties in the examples for each class. The form 'if < description > then <class >' is called a *classification rule*. Classification rules can be used to predict the class of new, previously unseen objects. This inductive learning technique is also known as *learning from examples*.

In *unsupervised learning*, there is not any teacher. Cognitive system has to discover classes and their descriptions itself. System observes its environment and recognizes common properties of objects. This inductive learning technique is also known as *learning from observation and discovery*.

2.1.1 Models

Inductive learning is the process of creation of a model of the environment of the cognitive system. This model consists of classes which represent objects that have similar properties, and rules that describe properties of members of each class and changes in environment. Cognitive system uses the models to predict changes in the environment, and to interact with this environment.

2.1.1.1 Environment

Defining of environment depends on the context. Environment of a cognitive system may be defined in local terms such as students of a faculty, a football team, a chess board, or as the whole of the universe which includes the system itself.

The situation of the environment is described by a **state** S_t , at a specific time t . This state, S_t , has some rules which describe the properties of the objects in the environment and mutual relationships among the objects. But the state of the environment changes over time. At the time $t+1$, state S_{t+1} may have new objects and relationships, or some objects may have disappeared, and properties of objects may have changed. So that, we must have a function that describes how the environment changes over time. This function is called **state transition function** and it is represented by τ . Transition function maps from one state to another state.

Marcel Holsheimer and Arno Siebes have given a definition for the environment: “The environment is a state transition system, i.e., a pair (S, τ) , where S is the set of all possible states and τ is the function $\tau: S \rightarrow S$. τ defines the next state S_{t+1} for any state S_t ” (Holsheimer , M. & Siebes, A. p. 11)

EXAMPLE:

Assume that the state consists of a single object, with properties “name is Ali” and “second year student”. In the next state “name” of the object remains unchanged, but the property “second year student” has changed to “third year student” obeying the law that all second year students will be third year students if they achieve their courses at the end of the academic year.

To make a reliable internal copy of this state transition system is a straightforward way to create a model of the environment. Simply, all encountered states are stored and all transitions are recorded. The current state is compared with all stored states to predict the next state from the current state. But, this representation is suitable for

simple environments that have a small number of various states. Otherwise, for realistic environments, the enormous amount of storage is needed to represent all possible states that the current state will exactly match any of the previous states. And some times, it will be impossible to determine the all possible states.

Because of such difficulties, we must use abstractions instead of making a faithful internal copy of any state transition system. For abstraction, a small number of properties to characterize the objects in a state is used. Objects having the same subset of properties are mapped to the same internal representation.

2.1.1.2 Classes

We describe the state in the model with using a small number of properties. This may cause that distinct objects in the environment may be accepted as the same object. That means we collect the objects having same chosen properties in a group. This group is called *equivalence classes* of objects. The *class description* consists of the unique values of properties of the objects. Each class corresponds a class description.

We can construct a *classification function* $P:S \rightarrow C$ where C denotes the set of all classes, and to each class C_i corresponds a description D_i . The classification function P maps an object O in state S to class C_i if properties of O have the same values in the description D_i .

2.2 Learning

The cognitive system should adapt itself to its environment. This means that the system should *learn*. Learning is to find suitable classes (internal representation) and a model transition function that acts on these classes. There are various learning strategies such as *learning by being told* and *learning from analogy*. For example, in learning by being told, a teacher acquires the knowledge like a textbook. The system only translates this knowledge to an internal format. In learning from analogy, the

system changes existing rules to generate new rules which are applicable to new, similar situations. In inductive learning, there are mainly two strategies; supervised learning and unsupervised learning.

2.2.1 Supervised learning

In supervised learning or learning from examples, the teacher defines the classes and supplies pre-classified objects of each class. The system should only find the description for each class to construct the model. A ***single class*** or ***multiple classes*** can be defined by the teacher.

In single class learning, only one class C is defined by a teacher. This teacher also provides all examples. If an example is a member of the class, this example called ***positive*** example, otherwise it called ***negative*** example. Teacher may provide all positive examples. Or both positive and negative examples are provided. The negative examples can be seen as members of many other classes. With characteristic ***description***, positive examples, members of class C are separated from negative examples which are not instances of class C .

In multiple classes learning, a finite number of classes C_1, C_2, \dots, C_n are defined by a teacher. Characteristic description D_i distinguishes positive examples of C_i from other examples (negative examples). Alternatively, the system constructs discriminating descriptions that cover all objects. Discriminating description distinguishes an instance of a class from the instance of all other classes.

2.2.2 Unsupervised learning

In unsupervised learning or learning from observation and discovery, there is no teacher that defines the classes. The system has to find its own classes. In practice, the system has to construct some clusters of the set of states in the environment. Such as in supervised learning, objects or examples are known. The system has to observe the objects and constructs class descriptions or patterns. Class descriptions are

constructed for each discovered class. Discovered classes cover all objects in the environment. Set of class descriptions is the result of unsupervised learning process.

2.3 Quality

Created model may change with respect to set of examples, and multiple models can be constructed from the same set of examples. It can be said that all created models can be correct with respect to given set of examples. Models should correctly predict the next state $S_{t+1,i}$ for all environmental states $S_{t,i}$ that already known. Models should be used also for any unseen state when new, unseen states occur.

Discovered or apparent relationships among states are not generally valid. Because, the number of objects is limited. So that, apparent relationships can be different from really existing relationships among states in the environment.

The correctness of a model is not verified by checking for all possible states, because the number of possible states is infinite for most environments. If multiple models are constructed, the simplest model can be chosen. Because of the simplest model is more likely to handle the nature of the phenomenon. (Ockham's razor rule)

2.4 Machine Learning

Computers can be used for inductive learning processes. This process is called machine learning. Machine learning systems use a coded form of a finite set of examples and observations, and do not interact directly with its environment. This coded finite set is called *training set*. In supervised learning, classes are defined by a user and system searches descriptions for each class. In unsupervised learning, machine learning system constructs the set of new discovered classes and class descriptions.

CHAPTER THREE

DATA MINING

The methods for handling regularities and rules are named data mining when the data set is a database. The knowledge (data) stored by a database has a different purpose than a learning process. This data may have noise and some values of attributes may be lost. To discover descriptions from a database is harder than machine learning where the ideal conditions have already being defined. Because of the size of database, the cost of inferring rules and verification of hypotheses is high. This cost can be reduced by using browsing optimization and caching. To remove noisy and missing values, statistical techniques are used.

As it has already being seen, we can say that learning is the process to construct the rules for transitions from state S_t to state S_{t+1} which t represents time, based on objects in the environment and observations of states of the environment. Machine learning is an automatic learning process which uses computer. Machine learning systems use training set, instead of real environment. Automatic inductive learning process is called *data mining* when the training set is a database. Now we can say that data mining is a special kind of machine learning.

3.1 The Comparison of Machine Learning With Data Mining

In machine learning, environment represents a finite set of objects. These objects are encoded in some readable form for the machine by the encoder. The set of encoded objects is the training set for machine learning algorithm, as shown in the figure 3.1.

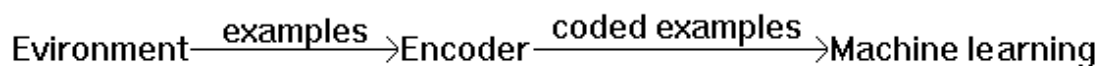


Figure 3.1 Diagram for machine learning



Figure 3.2 Diagram for data mining

In data mining, database is used instead of encoder (figure 3.2). Database consists of facts which are taken from the environment. It can be said that database is a small and simple model of the environment, because it has finite set of examples. Each state of the database represents a state of that environment. Each state transition of database represents a state transition of that environment. And data mining algorithm constructs a model from the database. That means, data mining algorithm infers classification rules that manage the classes of database objects. The rules for transition between classes should also be inferred from the transition in the database.

It may be seen that data mining and machine learning both have a similar framework. But there are important differences between them. First of all, in machine learning, training set is chosen suitable to help the learning process. On the other hand, database is not designed to help the data mining process. Objects of the database are chosen for the needs of applications. These objects may not meet the needs of data mining. In data mining, some attributes (or properties) may be chosen to simplify the learning process, but these attributes may not be in the database.

As a second and important difference, databases can contain some errors. In machine learning, learning algorithm often uses suitable examples which are chosen carefully and do not contain error or noise, but data mining algorithm has to cope with data which has noisy and contradictory data.

3.2 The Training Set

The training set of the learning algorithm of the data mining is database which contains non-trivial knowledge from the environment. There are many types of databases that database management systems (DBMS) support. We are interested in relational type of database. In a relational database, examples (or objects or

instances) are represented by tuples and properties of objects which are called attributes. Each tuple may have many attributes.

Each tuple can represent one or more objects in the environment. If tuples have at least one unique attribute, each tuple can represent only one object. Otherwise, each tuple may represent more than one object.

We can construct more than one table with using attributes and tuples. Each column of a table represents an attribute and each row of a table represents an example or object. We can recognize relationship between tables with common attribute (or attributes). Common attributes can be used for JOIN operations. In these tables, values of attributes may be NULL or unknown. If we use ***Universal Relation Assumption***, we construct a single table which contains all objects and their properties. Of course, the values of attributes in this table may be NULL or unknown.

Each attribute or property of objects is an element of set A , $A = \{A_1, A_2, \dots, A_n\}$. Distinct values of attributes from domains of attributes. We can say that domain of attribute A_1 is D_1 , domain of attribute A_2 is D_2 , and so on. Constructed table with attributes and their domains are called ***training set***. All relations over attributes of the table is called ***Universe U***. That means, $U = D_1 \times D_2 \times \dots \times D_n$. Now, we can say that, each training set is a finite subset of Universe U . Of course, we assume that each domain of attributes is finite and as a result, Universe is also finite.

CHAPTER FOUR

SEARCH ALGORITHMS

As we see before, data mining system constructs many classes and descriptions that describe these classes. Some of these descriptions can classify unseen examples correctly than others and can describe relationships between objects in the data used. The problem is to find the best descriptions among the possible descriptions set D constructed by the data mining system. This problem can be named as *search problem*.

Data mining systems has a quality function to measure the quality of a description. These systems initially choose a description, initial description, and iteratively modify the initial description by using quality function. Thus, data mining systems try to improve the quality of description and to get best description. Both, the set of descriptions and the quality function together, is called *search space*.

4.1 Search Space

We can define the search space as $(\mathcal{D}, f, \mathcal{O})$ where \mathcal{D} is a set of descriptions, f is a quality function and \mathcal{O} is a set operations on descriptions in the set \mathcal{D} .

4.1.1 Description space

The description space \mathcal{D} is the set of all possible descriptions constructed by the data mining system. A subset of training set S that defined by each description D in \mathcal{D} , is called *cover* $\sigma_D(S)$.

4.1.2 Operations

There are two types of operation: **Generalization and specialization**.

Generalization: If we apply a generalization operation to a description D in \mathcal{D} , we get a new description D' . D' contains more objects than the description D . If an object belongs to D , it also belongs to D' . But any object of D' may not belong to D .

So, we can say that $\sigma_D(S) \subseteq \sigma_{D'}(S)$. A rule can be correct or in other words, a rule can classify the objects correctly, but a generalization of this rule may not be correct. This means that the generalization operation is not truth preserving. But generalization rules are falsity preserving. If an object is covered by D , but it is not an example of class C , (i.e. the object is not correctly classified by the rule), and then the object falsifies the rule. If an object falsifies any rule, then it will also falsify any of generalizations of this rule.

Specialization:

If we apply a specialization operation to description D in \mathcal{D} , we get a new description D' . D' contains fewer objects from the description D . If an object belongs to D' , it also belongs to D . But any object of D may not belong to D' . So, we can say that $\sigma_{D'}(S) \subseteq \sigma_D(S)$. As it can be seen that the specialization operation is the inverse of the generalization operation.

4.1.3 Domains of the attributes

User should define the structure of the domain of the attributes in the database which generalization and specialization operation will be applied on. There are three basic types of structure of the domain.

Nominal (Categorical):

In this type of structure, symbols or names in the domain are independent and the values of an attribute are not ordered.

Linear:

In this type of structure, domain is totally ordered. Linear domain can be ***ordinal***. For example, values of the domain of an attribute can be low, medium or high. We can not use the mathematical operations such as summation or multiplication. The linear domain can be an ***interval*** domain such that we can apply summation operation on the elements of domain, but we cannot apply multiplication. The linear domain can be ***ratio*** domain. We can apply both summation and multiplication on the elements of the domain.

Partially ordered:

The domain is partially ordered. Partially ordered domain has a hierarchical form, where a parent node represents a more general concept than its children. Any symbol is smaller than the top symbol.

4.1.4 Quality function

The quality function produces a numeric values. Each value belongs to a specific description and indicates the quality of the description. A description should classify any new, unseen object correctly. This statement means that a description should be ***valid*** generally. And, in unsupervised learning, a description should be ***correct*** with respect to defined classes. So that, a description has two criteria; validity and correctness. We can assign a value to each criterion and these criteria can be combined with using a function to compute the quality of the descriptions

Validity:

It is accepted that the number of objects is limited in a database. But this cannot be seen in real world. So that the correctness of a rule cannot be verified for all possible situations. This means that we cannot prove the validity of a rule, in general. Most data mine systems rely on that the simpler description describes relationships between objects, approximately best. This rule is known as Ockham's razor. The quality function for validity, f_v , has higher value for simpler descriptions.

Correctness in supervised learning:

If all positive examples belong to class C and any negative example does not belongs to class C with respect to description D, it can be said that description D is correct. In other words, if $\sigma_D(S) = C$ then the description D is correct.

Two probabilistic concepts are used to determine a rule or a description is correct; classification accuracy and coverage. If a rule is not correct it may be complete or deterministic. These concepts are explained below.

The classification accuracy is the relative portion of the number of elements of the training set S which are covered by the description D that is also covered by the class C:

$$\text{classification accuracy} = \frac{|\sigma_D(S) \cap C|}{|\sigma_D(S)|}$$

The value of classification accuracy is an element of the interval [0,1]. The classification accuracy is the probability that an object covered by the description D belongs to the class C.

The coverage is the relative portion of the number of elements of class C which are also elements of the training set S covered by the description D :

$$\text{coverage} = \frac{|\sigma_D(S) \cap C|}{|C|}$$

The value of coverage is also an element of the interval $[0,1]$. The coverage is the probability that an object belongs to the class C is also covered by the description D .

If the coverage is equal to 1, the description is a necessary condition for the class. In this situation, any object belonging to the class is also covered by the description. The class C is a subset of $\sigma_D(S)$, $C \subseteq \sigma_D(S)$. And the rule is called as ***complete rule***.

If the classification accuracy is 1, the description is a sufficient condition for the class. In this situation, any object covered by the description belongs to the class. $\sigma_D(S)$ is a subset of class C , $\sigma_D(S) \subseteq C$. And the rule is called as ***deterministic rule***.

If the classification accuracy and coverage are equal to 1, the description is both a necessary and sufficient condition for the class. In this situation the class and the set $\sigma_D(S)$ are the same. And the rule is called as ***correct rule***.

The quality function for correctness, f_c , has a value which belongs to the interval $[0,1]$. When the description is correct, the value of f_c is 1. When the description is incorrect, the value of f_c is smaller than 1. G. Piatetsky-Shapiro and W. J. Frawley proposes some principles for the construction of the correctness criterion f_c such that:

1- If the classification accuracy is equal to the probability that any object in the training set S belongs to the class C then the description D and class C are statistically independent. The value of f_c is 0 and the description is wrong.

$$\frac{|\sigma_D(S) \cap C|}{|\sigma_D(S)|} = \frac{|C|}{|S|}$$

2- f_c monotonically increases with $|\sigma_D(S) \cap C|$ when $\sigma_D(S)$ and C remain the same.

3- f_c monotonically decreases with $|\sigma_D(S)|$ or $|C|$ when $\sigma_D(S) \cap C$ remains the same.

Combining criteria:

The criteria validity f_v and correctness f_c denote the quality of a description. In some problems, many other criteria such as the cost of evaluating the description or the cost of measuring attributes could be taken into account. For the overall quality, we should combine these criteria.

In general, there are two ways to compute the overall quality. In first way, a weight to each criterion should be assigned and weighted sum of the qualities for these criteria gives the overall quality. Let f_1, f_2, \dots, f_n are the values of criteria and w_1, w_2, \dots, w_n are the weights of these criteria respectively,

$$\text{Overall quality} = f_1w_1 + f_2w_2 + \dots + f_nw_n$$

Second way of the computation of the overall quality is called ***lexicographic evaluation functional*** (LEF)¹. In this way, the criteria f_1, f_2, \dots, f_n are ordered and t_1, t_2, \dots, t_n are tolerances or thresholds of these criteria respectively. LEF of these criteria can be shown as,

$$\text{LEF} = ((f_1, t_1), (f_2, t_2), \dots, (f_n, t_n))$$

The LEF determines the most suitable description from the given set of descriptions in this way: at first, all descriptions are ordered based on the value of the

¹ Michalski, R.S. , Carbonell, J.G. & Mitchell, T.M. Machine Learning, an Artificial Intelligence Approach, volume 2. California, 1986. pp:83-134

first criterion. Only the descriptions within the range defined by the threshold t_1 from the best description are chosen. Best description in the process is top of the ordered list. Chosen descriptions are ordered based on the value of the second criterion and the best are retained. When these processes are applied to the last criterion, the best description is handled.

4.2 Limitations on the Operations

Heuristic knowledge is specific to a part of the domain. This information has to be supplied by the user. There are two forms of heuristic knowledge.

Irrelevant attributes:

Some attributes in the database can be chosen as the irrelevant attributes. For example, first name of the student is not important in the question “How many students have finished the Law Faculty?”. So, the user could define the relevant and the irrelevant attributes for the classification. By using this way, the number of descriptions can be reduced.

Some attributes depend on the value of other attributes. For example, military knowledge is necessary if the person is male. So that, the attribute “military” can not be regarded for some classes.

Interrelationships between attributes:

Some attributes value can be computed from other values of attributes. For example, the volume of a cube can be computed from length of the one edge of the cube. So that, the quality of a description will not increase when the condition “volume” is adding.

The heuristic knowledge can also be the previous knowledge of the user or previously constructed rules and classes. Some information may not be coded as

heuristic knowledge in the database, but this information can be known by the user, and the user uses this information in the search process. Also, previously discovered rules and classes can be used for further investigation by the system. This process is important when the set of examples is updated.

CHAPTER FIVE

PROBLEMS

In data mining process, we accept that descriptions or classification rules exist in the data set. This may be true for some artificial data sets which are used in machine learning. But it is not always true when databases are used. We come face to face with several problems when the training set is a database. These problems can be limited of the supplied information, missing data, the size of the database and the problems from dynamic behaviors of the database.

5.1 Limited Information

5.1.1 Incomplete information

In supervised learning, we choose predicted attributes to determine the classes. These attributes are relevant for classification. But some of these predicted attributes may not be recorded in the database. And to construct a rule for classification may not always be possible by using known predicting variables.

There are two approaches when we have unknown predicting variables: we can either restrict ourselves with known variables. By which, we can construct only deterministic rules, and cannot find some valuable information that is hidden in the database.

Or we can search rules that are not necessary to determine the classes correctly; because it is possible that an object, covered by the description, belongs to a class. Such rules can be called as probabilistic rules. We can obtain very important information about relationships between objects in the database. For example, smoking is not necessary and not sufficient condition for cancer. But, still, this relationship is considered very important.

5.1.2 Sparse data

A classification rule constructed by the data mine system has to set the class boundaries. We can investigate the quality of these boundaries if the database contains examples which are just within (near misses) or outside (near hits) the class. This means that a database must have facts which represent all possible behavior. But in real world, facts in a database represent only a small subset of all possible behavior. So that class boundaries can be incorrect or vague.

For a solution, we need additional information. The system might search additional information in the database for interesting examples.

5.2 Data Corruption

We assume that all examples in the data set have correct values. An object in the database has many properties or attributes. Some attributes may have values which are based on measurement or subjective judgments which may cause some errors in the value of attributes. Such errors are called noise. And also, some attribute values may be missing. Both cause misclassification.

5.2.1 Noise

We meet problems caused by noise, when the system construct the descriptions and classify the examples by using these descriptions.

Constructing descriptions:

In a noisy training set, constructed descriptions may cover corrupted examples. Therefore, the system should decide whether an example is corrupted or not. Corrupted examples should be ignored.

Classifying examples:

Previously constructed descriptions taken from a training set can be used to classify the previously unseen examples. Corrupted examples may cause misclassification. Of course, misclassification of unseen examples is expected at a low level. For the solution, we can compare the rules constructed from the noisy training set with the rules constructed from the same but noise free training set. If there is a small amount of this misclassification, the rules constructed from the noisy training set can be used in practice.

5.2.2 Missing attribute values

We meet two problems which missing attribute values cause, at two different levels of learning process.

Constructing descriptions:

The system may not take into consideration the examples with missing attributes. Or, the system can replace the missing value with the new approximate value. This value can be computed from the value of other attributes by the statistical methods. Or, simply, missing values are filled with the value 'unknown' and these values can be used in the descriptions.

Classifying examples:

Unseen examples with missing attribute values can be classified by previously constructed descriptions. If these descriptions contain conditions on some of these attributes, they cannot be applied.

5.3 Databases

Database is a training set used in data mining. This training set has some difference from the training set used in machine learning. The training set used in machine learning is constructed by the user for a special purpose.

5.3.1 Size of database

In machine learning, the training set is small, (for example, the training set which contains thousand objects, is considered to be a large training set) but on other hand the number of objects in a database and the amount of properties per objects are generally very large.

Information per object:

Most databases contain many attributes. For example in the database for students in Law Faculty, the number of attributes is approximately 200. In reality, much information per objects is an advantage. With more information per object, we can learn true relationships between objects, but the number of constructable description increases with the number of information. The number of description depends on the size of domain of attributes. Roughly, the number of constructable descriptions is 2^ℓ , where ℓ is the sum of the size of domain of attributes. To overcome this problem, we eliminate some attributes which are considered not necessary.

Number of objects:

The problem is faced when we try to verify the quality of each constructed description. We use statistical tests in this verifying process. This test needs information about the number of examples covered by the description, or the distribution of values in this set. This test is very expensive in huge databases. We can use two techniques to overcome this problem.

1. Multiple descriptions:

In a single iteration of the search process, multiple descriptions can be constructed. We can compute their quality simultaneously by a single but complex database access.

2. Windows:

We choose a subset of database as a representative sample. This sample is called a window. This sample is small with respect to the entire database. It contains a few thousand objects. We can compute the quality of a description using this window. Then, the best descriptions are tested on the real databases.

Of course, the actual probability of the rules may not be equal to the predicted probability. We choose some incorrectly classified examples. Then, we add these incorrectly classified examples to the window and modify the rules using this new window. This modification process is called *incremental learning*.

5.3.2 Updates

In the course of time, the database will change. Some properties of examples can change, some examples can be added or removed. Because of this reason, the quality of some rules will decrease. When we run these rules, some objects are classified incorrectly. The system should adapt to such changes, and rules should be adjusted.

When too many incorrect predictions are made, data mine systems start the process of rule adjustment. Some kinds of incremental learning can be used to overcome this problem. A kind of incremental learning is *learning with full memory*. In this type of learning, the system remembers all examples. Other kind of incremental learning is *learning with partial memory* which is opposite of first type of learning. It is obvious that new rules are guaranteed to be correct with respect to all old and new training examples.

CHAPTER SIX

KNOWLEDGE REPRESENTATION

In this chapter, we will discuss some kind of knowledge representations. In previous chapters, we used relational algebra (or selection conditions) to present the condition in supervised learning and to describe the database in unsupervised learning.

The other representation methods are First Order Logic (FOL), propositional representation, structured representation and neural networks.

6.1 Propositional-Like Representations

In propositional representations, we use logic operators to formulate the descriptions. These descriptions consist of the values of attributes. ‘(high school=Normal \vee high school=Anadolu) \wedge father’s education=University’ is an example. This formula is in Conjunctive Normal Form (CNF), conjunction of clauses, where clauses are disjunctions of attribute value conditions. We can re-write this formula as the set-description ‘high school \in {Normal, Anadolu} \wedge father’s education \in {University}’.

An alternative form of CNF is Disjunctive Normal Form (DNF), disjunction of clauses, where clauses are conjunctions of attribute value conditions. Previous studies indicated that, the generated descriptions with CNF representations are smaller than the DNF representations.

The above examples are not really propositional, because, in propositional representation, we must use variables. This is the reason why we call them propositional-like.

6.1.1 Decision trees

With a decision tree, the examples are classified to a finite number of classes. In the tree, nodes are labeled with attribute names, the edges are labeled with possible values for this attribute, and the leaves are labeled with the different classes. An object is classified from the bottom of the tree, by taking the values of the attributes written on the edges.

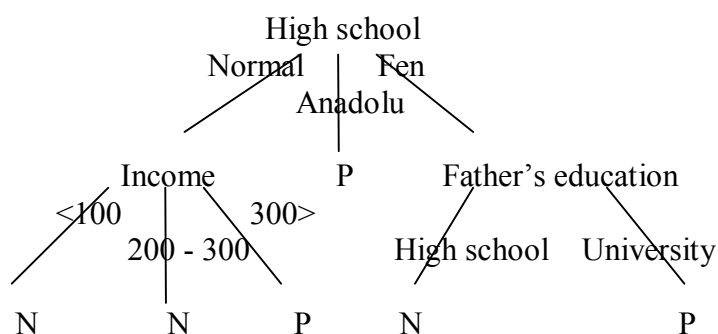


Figure 6.1 An example for the decision tree

In Figure 6.1, 'high school' is an attribute name and, 'Normal', 'Anadolu' and 'Fen' are the possible values of the attribute 'high school'. P represents positive examples and N represents negative examples.

Decision tree is suitable for supervised machine learning systems. But, for realistic application, the decision tree becomes very large. There has been some research to transform the decision tree into other representations.

6.1.2 Production rules

Production rules are a transformation of the decision tree and a propositional-like representation. In expert systems, production rules are widely used for representing knowledge. Production rules can easily be interpreted, because a single rule can be understood without reference to the other rules.

As an example, we can transform the decision tree in the Figure 6.1 to the propositional-like production rules. We will use the conjunctive normal form.

If high school=normal and income < 100 then class = N

If high school = Fen and father's education = University then class = P

If high school = Anadolu the class = P

6.1.3 Decision list

Decision list is another propositional-like representation. Any knowledge structure constructed as a decision list representation can be transformed to a decision list or DNF representation or CNF representation. A decision list is a list of pairs which first item of a pair is an elementary description ϕ_i and the second item is C_i .

$(\phi_1, C_1), (\phi_2, C_2), (\phi_3, C_3), \dots, (\phi_k, C_k)$

The last description has a constant value; true. If the last index of a description is j the ϕ_j covers object \bullet and \bullet belongs to class C_j . A decision list can be extended as a rule 'if ϕ_1 , then C_1 , else if ϕ_2 then C_2 ,else C_r '.

6.1.4 Ripple-down rule sets

Sometimes, we need to represent some exceptions in the rules, such that 'if ϕ_i then C_i unless ϕ_j '. We can add an exception rule 'if ϕ_j then C_j '. But, any object for which ϕ_j is true will be assigned to class C_j globally, whereas we want the exception to be local to the rule 'if ϕ_i then C_i '. As a result, the decision list will be difficult to understand.

Ripple-down rule sets represent exceptions in a more localized manner. These rules consist of conditions and exceptions to these conditions that are local to the rule. These rules are nested if-then statements such that;

```

if  $\emptyset_i$  then
    if  $\emptyset_j$  then  $C_j$ 
        else  $C_i$ 
    else  $C_j$ 

```

Above example is a ripple-down rule set with depth 2. Here, we do not need a global ordering of the rules, as in the decision lists.

6.2 First Order Logic

The propositional-like representation has some disadvantages. We cannot represent the patterns in terms of relationships among objects or attributes. For example, we can not construct a class which contains students that take same grade from 'Anayasa' and 'Hukuk Baslangici'. We need more powerful representation to state that any student where 'Anayasa = Hukuk Baslangici' belongs to the class.

In the learning process with the propositional-like representation, it is difficult to incorporate domain knowledge. It is accepted that domain knowledge consists of constraints on the descriptions, generated by the system. But, the domain knowledge is rarely complete and consistent.

Learning systems construct descriptions within the limits of a fixed vocabulary of propositional attributes. We can increase the set of patterns and the comprehensibility of the representation by using auxiliary predicates.

We need a more powerful representation to overcome these problems. Some kind of First Order Logic is used to represent knowledge. This type of representation is called ***Inductive Logic Programming***. The aim of the inductive logic programming is to construct a First Order Logic program. This program has the training set as its logical consequence.

When we find complex descriptions in a less powerful representation, First Order Logic allows us to find simple descriptions for classes. As a result the computational complexity of construction of description decreases. The set of possible descriptions gets larger. Larger set of descriptions may make learning easier. It may be easier for the learning algorithm to produce a nearly correct answer from a rich set of alternatives than from a small set of alternatives. But, it is difficult to select the best description. A solution is to search for particular descriptions only.

6.3 Structured Representations

There are mainly two types of structured representation; semantic nets and frames. These representations are not more powerful than First Order Logic, but they provide a more comprehensible representation. We can state subtype relationships among objects by structured representations. Structured representations can be expressed as a First Order Logic program.

6.3.1 Semantic nets

A semantic network is a directed graph. The nodes of this graph denote concepts and the arcs denote relationships between those concepts.

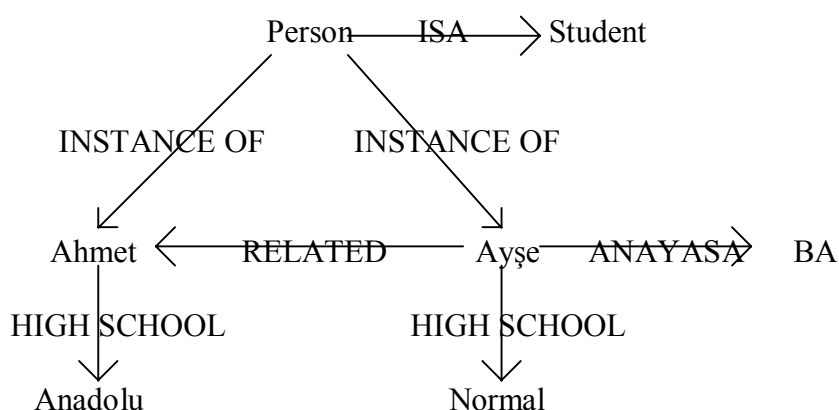


Figure 6.2 An example of semantic nets

In the Figure 6.2, we can see an example of the semantic network. In this example, ‘Brother’, ‘High school’, ‘Anayasa’ are the relationships between concepts and, ‘instance-of’ and ‘isa’ are the relationships between concepts and classes. If we represent these relationships with a different network, they can be more comprehensible.

As we stated before, we can express this semantic network in First Order Logic. Each arc can be expressed as a binary predicate and nodes can be expressed as terms:

High school (Ahmet, Anadolu).

High school (Ayse, Normal).

Related (Ahmet, Ayse).

Anayasa (Ayse, BA),

Person (Ahmet).

Person (Ayse).

$\forall x. \text{person}(x) \rightarrow \text{Student}(x)$.

With semantic nets representation, we can find all information related to a particular object. Each example or object is a semantic net for data mining. We use graph-manipulations to find patterns. These patterns are only subgraphs which are shared by all objects of the same class.

6.3.2 Frames and schemata

When we represent relationships in the semantic nets, as a schema, we get the new type of structured object which is named frame. A frame consists of a framename and slots. A framename is name of initial node and slots are the named attributes. Slots are filled with values for particular instances.

As an example, we can re-represent the information about Ayse that was used in the example at section 3.1., in Table 6.1 as a frame.

Table 6.1 A frame example

Frame name	person
slot 1	isa: Student
slot 2	name: Ayşe
slot 3	related: Ahmet
slot 4	high school: Normal
slot 5	anayasa: BA

We can incorporate subtype information in frames by using 'isa' slots. An 'isa' slot refers to another frame. In above example, 'isa' slot refers the 'student' frame. 'Student' frame stores information about students such as 'father name', 'address'.

CHAPTER SEVEN

APPLICATION PROGRAM

7.1 Introduction

In Chapter Seven, a sample program is presented to demonstrate the results of explained subjects in previous chapters. Data used in this application program are obtained from Seyitömer Coal Basin. The reasons why these data chosen are, explained below.

- A) Especially at the field commerce and medical science, there are many samples of data mining applications. But any sample related with earth science has not been encountered.
- B) We wish to show that data mining can have value at the field of geology and mine engineering.
- C) Data are not suitable for unsupervised techniques such as clustering.

7.2 Data Examination

7.2.1 Introduction of data

Data are related with earth materials that obtained from bore-holes. Material names and the thickness of layers related with this material are shown on the vertical cross-sectional drawings of bore-holes. An unique number is written at the top of this figure as the bore-hole number.

Near of this figure, there is a table that includes place of bore-hole, drilling machine brand, beginning and ending date of drilling job. There are two more tables. These tables include some physical specifications of flammable materials and team information who works at that bore-hole. An example of the data sheet of a bore-hole

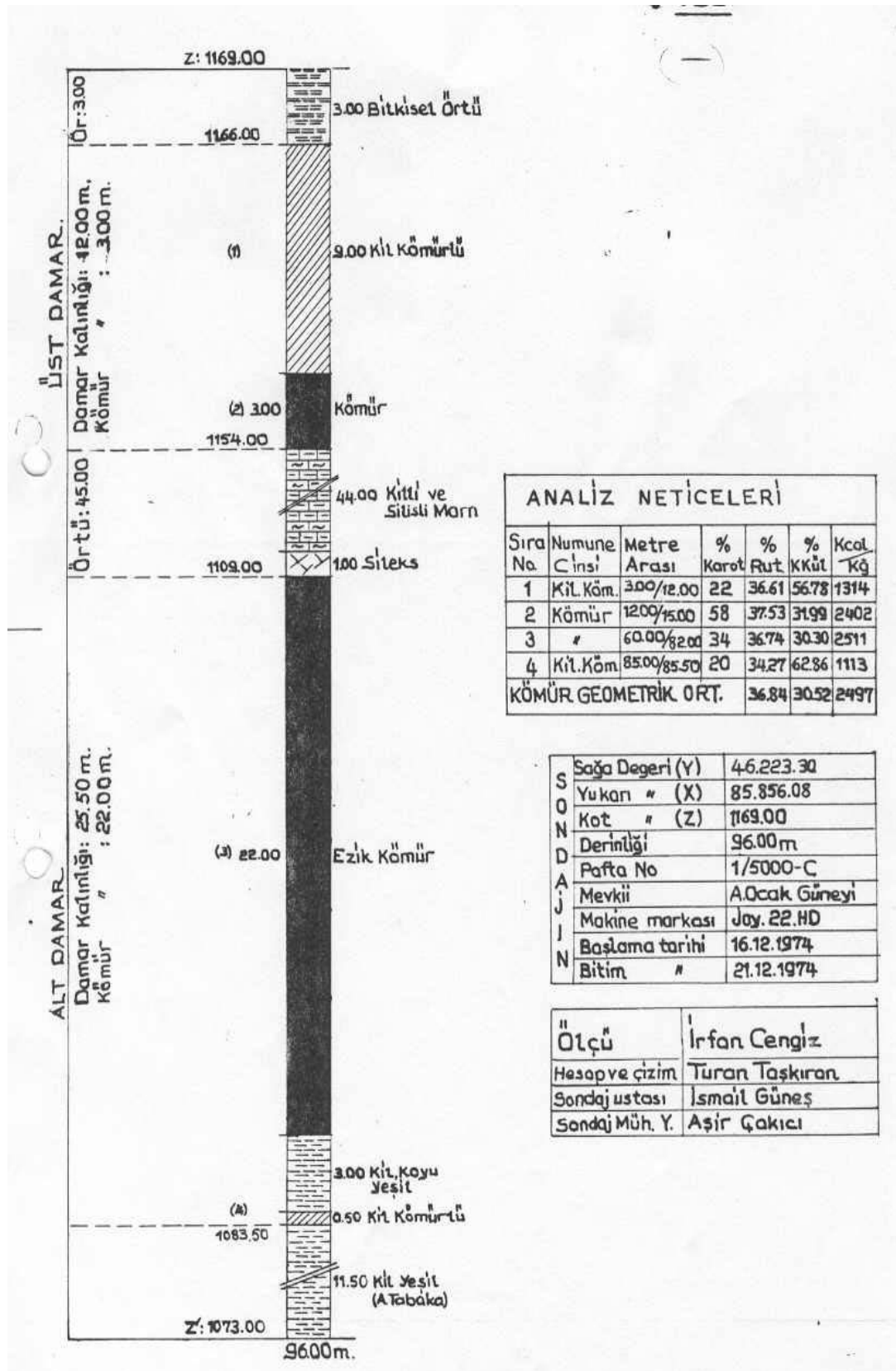


Figure 7.1 Data sample

is shown at Figure 7.1.

Used data in this study is quite different than the data used in previous applications. For example, it is possible to find out the names of goods and marketing frequency of these goods that purchased by any customer of a market, from data related with the customer. Also, we can predict which kind of goods will be purchased by the customer in the future. Or, purchasing habit of people who belong to the specific age group may be decided. If we examine the data used in this study, there are great differences between bore-holes which are very close to each other. When thick coal layers are met a bore-hole, it is possible not to meet any coal at any bore-hole just 200 meters faraway from the first bore-hole.

Basic knowledge about coal is; a big forest must be covered with clay for formation of coal. The possibility of coal existence under clayed region is accepted high.

7.2.2 Choosing the database management system

Several Database Management Systems are described according to types of the data and procedures to be applied. These are “Relational Database”, “Transactional Database”, “Object Oriented Database”, “Spatial Database”, “Data Warehouse” and “Data Cube”. Relational Database is preferred in our sample. Reasons for this are explained below.

Our data can be gathered under four main headings which are bore-hole specifications, layer specifications, analyze results and bore-hole crew information. The last one has not been used because it is out of procedure goal. Data are two types as numerical and character. There are no other data types as figure, graphic, audio and video. Therefore, there is no need to use the data warehouse techniques. Since map descriptions are not used, “Spatial Database”, and structures of holes are not suitable to describe the object “Object Oriented Database” is not used. Since layer quantities in holes changes approximately between 10 and 60 and attribute quantity

is high, “Data Cube” usage will not be proper. If “Data Cube” is used more than three dimensions will be needed and a lot of cells will be null. “Relational Database” preferred instead of “Transactional Database” since “Standard Query Language” (SQL) can be used more easily.

First only one huge table usage thought. Since, NULL values will be very high smaller tables are used. Three main tables are defined according to characteristics of the data. These are “Kuyu” (Bore-hole), “Katman” (Layer) and “Analiz” (Analyze). There are no empty NULL cells in these tables. Besides, the necessary tables prepared for securing easiness to user, saving the results and giving ability for reaching info whenever needed.

7.2.3 Introduction of data sheet

The data which data mining will apply on was the result of drilling bore-hole on coal-bed of the region of *Seyitömer*. First, bore-hole properties are taken place on data sheet as a table that can be seen at Figure 1. These properties are “*Sj-Sondaj Numarası* ” that will be named as “*Kuyu Numarası* ” (Bore-Hole Number), “*Sağa Değeri (Y)*” (Right Value), “*Yukarı Değeri (X)*” (Up Value), “*Kot Değeri (Z)*” (Altitude Value), “*Derinliği*” (Depth), “*Pafta No*” (Section Number), “*Mevkii*” (Location), “*Makine Markası*” (Machine Brand), “*Başlama Tarihi*” (Beginning Date) and “*Bitiş Tarihi*” or “*Bitim Tarihi*” (End Date), that are given on table. Bore-hole number is an unique value. Right value, up value and altitude value give the coordinates of bore-hole opening in three dimensional space. Depth value is the sum of the thicknesses of the layers. On the data sheet, materials names of ground layers and thickness of layers are shown on a vertical cross sectional diagram of a bore-hole. Analysis results of each coal layers in bore-hole are shown at another table. At this table, “*Sıra No*” (Order Number) is used to set relation between table and cross-sectional diagram. “*Numune Cinsi*” means name of the material. Only flammable materials are analyzed. “*Metre Arası*” includes beginning and end altitudes of related layer. “*%Karot*” expresses the percentage for the full section of the “karot”. “*%Rut*” is humidity percentage, “*%K.Kül*” is the percentage of dried ash and “*Kcal/kg*” is the

heat energy as Kcal/Kg when the material is burned. At last, there is a table which contains names and tasks of personals who were worked about related bore-hole.

7.3 Windows of the Program

7.3.1 Main Window

Window projected on the screen when the program run, is main window. The name of this window is “*Kömür Kuyularını Sınıflandırma Programı*” (Classification Program for Bore-Hole of Coal). There are three menu names on the menu bar. These are “*Tanımlamalar*” (Definitions of Data), “*İşlemler*” (Procedures) and “*Pencere*” (Window) that can be seen at Figure 7.2. Data input can do with using the

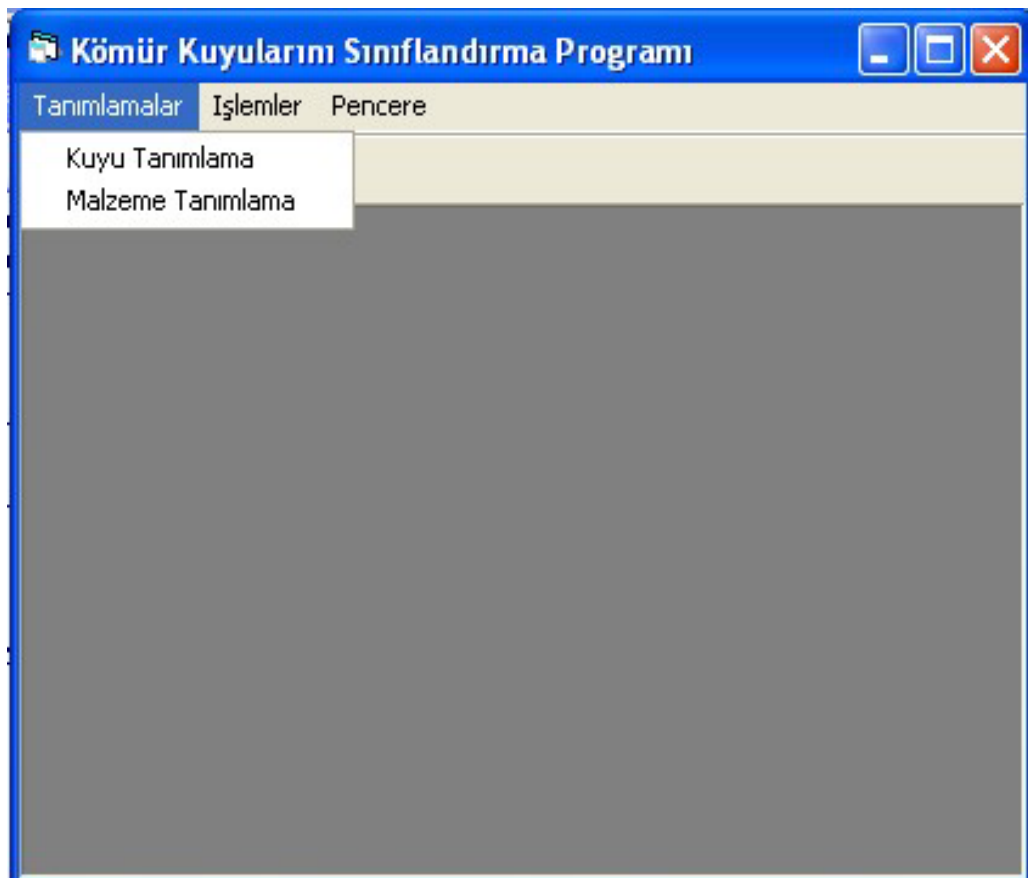


Figure 7.2 Main window of program

menu named “*Tanımlamalar*”. There are two procedure commands at this menu; “*Kuyu Tanımlama*” (Data Input Procedure which Related Bore-Hole) and “*Malzeme Tanımlama*” (Input Material Names and Codes). Procedures about these commands were explained at section 7.3.2 and 7.3.3.

7.3.2 Window of “*Malzeme Tanımlama*”

Material samples are taken from the bore-hole with a tool that is named “karot”. There exist different material samples at any one of bore-holes, even at a “karot”. Some of material names are very long and there is no standardization about names of materials. Different team has given different name to the same material. For example, on one data sheet, a layer was named as “*Killi Kömür*” (*Clayey Coal*), on the other one, the same material was named as “*Kömür Killi*” (*Coal with Clay*), or “*Kil, kahverengi*” (*Clay, brown*) was changed with “*Kahverengi kil*” (*Brown clay*). These names cannot be accepted as the same properties of the tuples by classification program and, as a result, some trivial patterns may come out.

To solve the problems that pointed out above, different code numbers were given to different materials. If the materials are the same, same code number was given to them. These material names and their codes were entered with using the window of “*Malzeme Tanımlama*” shown at Figure 7.3. This window can visualize on the screen by selection of “*Malzeme Tanımlama*” command from menu “*Tanımlamalar*” on the main window “*Kömür Kuyuları Sınıflandırma Programı*” (Classification Program for Bore-Hole of Coal). The value of “*Malzeme Kod*” (Material Code) increases when “*Yeni*” (New) button is selected which is taken right side of this window. “*Tamam*” (OK) button must be selected after material name was written at the data input box which is right side of the “*Material name*” and, the code number and name of the material is added to the below list. That list shows codes and related material names added before.

Malzeme Tanımlama

Malzeme Kod: 13

Malzeme Adı: Kil, sert

Görünüm Rengi: 3 - Açık Gri

Tarama Şekli: 7 - Çapraz Taram

Tarama Rengi: 15 - Siyah

Kapat

Yeni

Değiştir

Sil

Tamam

İptal

MALZEMEKOD	MALZEMEADI	RU
3	Gre, killi	
4	Kalker, killi	
5	Kalker, silisli	
6	Kil	
7	Kil, beyaz	
8	Kil, kahverengi	
9	Kil, kömürlü	
10	Kil, kumlu	
11	Kil, marnlı	
12	Kil, sarı	
13	Kil, sert	
14	Kil, çakıllı	
15	Kil, silisli	
16	Kil, siyah	
17	Kil, siyah yanmıyor	
18	Kil, şistli	
19	Kil, şistli siyah	
20	Kil, yanık	
21	Kil, yeşil	
22	Kil, yumuşak	
23	Kömür	

Figure 7.3 Window of Malzeme Tanımlama.

New material names can be entered after pre-research of the data sheets. Or, user can begin with “Kuyu Tanımlama” to enter the data related with bore-hole properties. User will return to “Malzeme Tanımlama” to enter material name and material code before to enter layer properties at “Katmanlar” interface, which explained at section 7.3.4. Before the new material name is entered, list of material name must be controlled, if there is an equivalent name or not. If an equivalent material name can be found, its code number must be chosen as the code number of

Malzeme Tanımlama

Malzeme Kod: 13

Malzeme Adı: Kil, sert

Görünüm Rengi: 3 - Açık Gri

Tarama Şekli: 0 - Beyaz

Tarama Rengi: 3 - Açık Gri

0 - Beyaz
1 - Pembe
2 - Açık Sarı
3 - Açık Gri
4 - Açık Kırmızı
5 - Turuncu
6 - Sarı
7 - Kırmızı

Kapat

Yeni

Değiştir

Sil

Tamam

İptal

MALZEMEKOD	MALZEMEADI	RU
3	Gre, killi	
4	Kalker, killi	
5	Kalker, silisli	
6	Kil	
7	Kil beyaz	

Figure 7.4 Window for choosing base color.

Malzeme Tanımlama

Malzeme Kod: 13

Malzeme Adı: Kil, sert

Görünüm Rengi: 3 - Açık Gri

Tarama Şekli: 7 - Çapraz Taram

Tarama Rengi: 0 - İçi Dolu

0 - İçi Dolu
1 - Şeffaf
2 - Yatay Çizgili
3 - Dikey Çizgili
4 - Sola Yatık Çizgili
5 - Sağa Yatık Çizgili
6 - Taramalı
7 - Çapraz Taramalı

Kapat

Yeni

Değiştir

Sil

Tamam

İptal

MALZEMEKOD	MALZEMEADI	RU
3	Gre, killi	
4	Kalker, killi	
5	Kalker silisli	

Figure 7.5 Window for choosing type of lineated.

the newest material name. Other wise, this material name and its new code number must be added to the list. “Görünüm Rengi”, “Tarama Şekli” and “Tarama Rengi”

will be used to draw the cross-sectional diagrams of bore-holes. User can symbolize any chosen material by choosing base color (Tarama Rengi), cross-hatching (Tarama Şekli) and base color (Tarama Rengi) of lineated that used as shadow, which can be seen at Figure 7.4, Figure 7.5 and Figure 7.6.

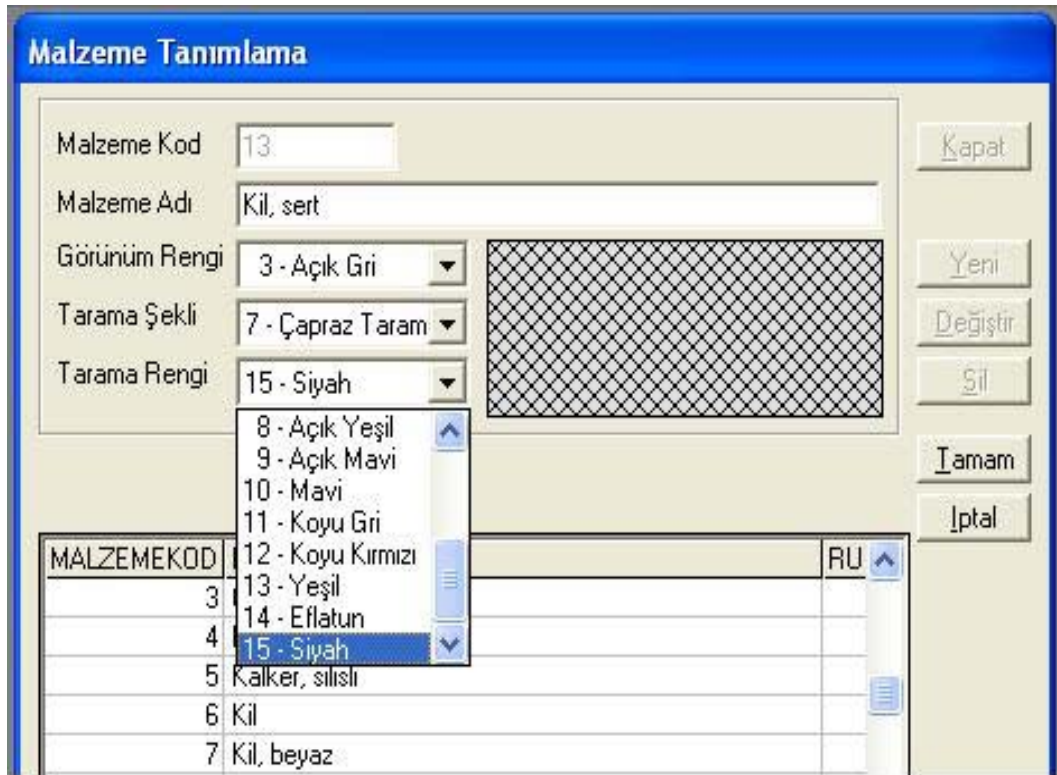


Figure 7.6 Window for choosing color of lineated.

Of course, there may be some mistakes. If any mistake occurs, we must choose the tuple from the list which has wrong value, and then select the “*Değiştir*” (Change) button. Now, program will allow us to change wrong value with the right one. Maybe, some material names can be written more than one. At this time, unnecessary tuple is selected and than user must be selected “*Sil*” (Delete) button, and selected tuple can be erased. Some material names and their codes are shown at Table 7.1.

Table 7.1 List of the material names and codes.

MALZEMEKOD	MALZEMEADI	MALZEMEKOD	MALZEMEADI
0	YOK	23	Kömür
1	Bitkisel örtü tabakası	24	Kömür, ezik
2	Gre, beyaz	25	Kömür, killi
3	Gre, killi	26	Kömür, killi ve şistli
4	Kalker, killi	27	Kömür, şistli
5	Kalker, silisli	28	Kömür, şistli ezik
6	Kil	29	Marn
7	Kil, beyaz	30	Marn, killi
8	Kil, kahverengi	31	Marn, silisli
9	Kil, kömürlü	32	Marn, yanıcı
10	Kil, kumlu	33	Marn, sert
11	Kil, marnlı	34	Serpantin
12	Kil, sarı	35	Sileks
13	Kil, sert	36	Şist, yanmıyor
14	Kil, çakıllı	37	Kum
15	Kil, silisli	38	Opal (Konglomera)
16	Kil, siyah	39	Tabaka tenavübü
17	Kil, siyah yanmıyor	40	Yanık Toprak
18	Kil, şistli	41	Dolgu Toprak
19	Kil, şistli siyah	42	Killi Bitkisel örtü
20	Kil, yanık	43	Kalker
21	Kil, yeşil	44	Killi Stok Kömür

A relational table named “*Malzeme*” that can be seen at Table 7.2, is created by using “*Malzeme Tanımlama*” window. This table has five attributes. Both attributes “*Malzeme Adı*” (Material Name) and “*Malzeme Kodu*” (Material Code) are unique and not permitted NULL value, so that both attributes can be selected as key attributes. One of the targets of the creation of this table is to assign the material code for each material name. Two benefits are expected by using code number.

1- If the attribute material name is selected as key attribute instead of material code, material name must be used in most of data tables to set relation between tables. As a result, more memory space must be needed.

2- It may force on the user to standardize the material name.

Table 7.2 Form of the table of “Malzeme”.

MALZEMEKOD	MALZEMEADI	GORUNUMRENGI	TARAMASEKLI	TARAMARENGI

7.3.3 Window of “Kuyular ”

The user can visualize “Kuyular” and “Katmanlar” windows on the screen by selection of “Kuyu Tanımlama” command from menu named “Tanımlamalar” on the main window. The window of “Kuyular” can be seen at Figure 7.7. “Kuyular” window is related with bore-hole properties which are explained before at the part of Introduction. Created table “Kuyu” that can be seen at Table 7.3, is the same with the table that is on the data sheet. The attributes of this table is “Kuyu No” (Bore-Hole Number), “Sağa Değeri (y)” (Right Value), “Yukarı Değeri (x)” (Up Value), “Kot Değeri (z)” (Altitude Value), “Kalınlık” (Thickness), “Pafta No” (Section No), “Mevki” (Location), “Makine Markası” (Machine Brand), “Başlama Tarihi” (Beginning Date), “Bitiş Tarihi” (End Date). The attribute “Kuyu No” is selected as key attributes, and it is used to set relationship with other tables. Also, there is a list of values of attributes that were entered by the user, at the lower part of window.

The values of attributes “Pafta No”, “Mevki”, “Makine Markası” were not found on the some of the data sheet in the main database. Indeed, we do not need any one of these values for the data mining purpose. The attributes which names were given above, and the attributes “Başlama Tarihi” and “Bitiş Tarihi” are used for only archives. Values of attributes “Sağa Değeri” and “Yukarı Değeri” will be used to show location of the members of classes after the classification. The values of attribute “Kalınlık” is the total thickness of the layers which gives the depth of the

Kömür Kuyularını Sınıflandırma Programı
Tanımlamalar İşlemler Pencere

Ölçek Aralık: 100 5

Kuyu Cizim Yazdır Kuyuyu Göster

Katmanlar

Kuyular

Kuyu No: 639 Pafta No: 1/5000-C

Sağla Değeri: 47750 Mevki: Eraklı kar T. Bahi

Yukarı Değeri: 82514 Makine Markası: JOY BUDA 22 HP

Kot Değeri: 1163.36 Başlama Tarihi: 28/4/1973

Kalınlık: 92 Bitiş Tarihi: 6/5/1973

Kapat Yeni Değiştir Sil Tamam İptal

KUYUNO	SAGADEGERI	YUKARIDGERI	KOTDEGERI	KALINLIK	PAFTANO	MEVKI	MAKINE
639	47750	82514	1163.36	92	1/5000-C	Eraklı kar T. Bahi	JOY BU
640	45499	86301	1193.52	74	1/5000-C	Açık ocak güneyi	3 HQ 6E
641	48001	82761	1150.45	48	1/5000-C	Kocadüz T. Güneydoğu	JOY BU
642	45401	86303	1187.86	80	1/5000-C	Açık ocak güneyi	3 HQ 6E
643	47503	82514	1167.79	77	1/5000-C	Kocadüz T. Güney batısı	JOY BU
644	45482	86194	1185.26	86	1/5000-C	Külüçek T. doğusu	3 HQ 6E
645	47262	82512	1165.22	58	1/5000-C	Kocadüz T. Güneybahi	JOY BU
646	45203	86203	1196.51	92	1/5000-C	Külüçek T. Doğusu	3 HQ 6E
647	47750	82752	1176	45	1/5000-C	Kocadüz T. Güneyi	JOY BU
648	45101	86103	1198.66	92	1/5000-C	Açık ocak güneyi	3 HQ 6E
649	47509	82748	1175.9	70	1/5000-C	Kocadüz T. Güneybatısı	JOY BU
650	45199	86102	1192.7	106	1/5000-C	Açık ocak güneyi	3 HQ 6E
651	47254	82765	1180.92	85	1/5000-C	Kocadüz T. Güney batısı	JOY BU
652	45600	86199	1191.24	106	1/5000-C	Açık ocak güneyi	3 HQ 6E
653	47003	82777	1175.72	60	1/5000-C	Kocadüz T. Güneybatısı	JOY BU
654	45705	86199	1193.35	96	1/5000-C	Açık ocak güneyi	3 HQ 6E
655	46496	82747	1173.69	85	1/5000-C	Gülbek T. Güneyi	JOY BU
656	45797	86198	1192.3	91	1/5000-C	Açık ocak güneyi	3HQ 6E
657	46248.69	82750.17	1169.29	80	1/5000-C	Gülbek T. güneyi	JOY BU
658	45900	86200	1191.87	81	1/5000-C	Açık ocak güneyi	3HQ 6E

Figure 7.7 Window of “Kuyular”

bore-hole. The values of “*Kalınlık*” can be used to control partly that the values of thickness were entered correctly or not.

Table 7.3 Form of the “Kuyu” table.

KUYUNO	SAGA DEGERI	YUKARI DEGERI	KOT DEGERI	KALINLIK	PAFTA NO	MEVKI	MAKINE MARKASI	BASLAMA TARIHI	BITIS TARIHI

7.3.4 Window of “*Katmanlar*”

When the user select the tab named “*Katmanlar*” at the “*Kuyular*” window which can be visualized by selection of “*Kuyu Tanımlama*” window from the menu named “*Tanımlamalar*”, “*Katmanlar*” window takes place on the screen that can be seen at Figure 7.8. Window of “*Katmanlar*” are formed five parts. There are properties of bore-hole which was chosen from window of “*Kuyular*”, at left and up side of window. This part was taken place only for knowledge to the user. Knowledge which can be seen on this part can not be changed, deleted, and can not be entered new data. The section which is under that part explained above and placed middle of the window, is used for entering data to the relational table “*Katman*” which related with properties of layers which taken from diagram on the data sheet. This table that can be seen at Table 7.4, includes attributes of “*Kuyu No*” (Bore-Hole Number), “*Katman No*” (Layer Number), “*Malzeme Kod*” (Material Code), “*Bitiş Kotu*” (End Altitude) and “*Kalınlık*” (Thickness). “*Kuyu No*” and “*Katman No*” are selected as key attributes.

Table 7.4 Form of the “*Katman*” table.

KUYUNO	KATMANNO	MALZEMEKOD	BITISKOTU	KALINLIK

Kuyular Ölçek Aralık 100 Kuyu Göster 10 Kuyu Çizim Yazdır Kuyu Göster 10

Katmanlar

Kuyular

Kuyu No: 659 Pafta No: 1/5000-C
 Sağa Değeri: 46236 Mevki: Gülbek T Güneyi
 Yukarı Değeri: 82994 Makine Markası: JOY BUDA 22 HD2/5/18
 Kot Değeri: 1193.5 Başlama Tarihi: 2/5/1973
 Kalınlık: 88 Bitiş Tarihi: 4/5/1973

Katman Bilgileri

Katman No: 5 Bitiş Kotu: 0
 Malzeme Kod: 23 Kömür Kalınlık: 1

Analiz Sonuçları

Karot Yüzdesi: 70
 Rutubet Yüzdesi: 26.26
 Kuj Yüzdesi: 27.59
 Kalori: 3179

KATMANNO	MALZEMEKOD	BITİSKOTU	KALINLIK
1	12	0	21
2	6	0	12.5
3	23	0	1.5
4	6	0	1
5	23	0	1
6	6	0	7
7	23	0	1.5
8	16	0	0.5
9	23	0	1.5
10	16	0	1.5
11	25	0	1.5
12	21	0	37.5

Kapat **Yeni** **Değiştir** **Sil** **Tanımla** **[F2]**

Figure 7.8 Window of Katmanlar.

Kömür Kuyularını Sınıflandırma Programı
Tanımlamalar İşlemler Pencere

Kuyular **Katmanlar**

Ölçek Aralık 100 10

Kuyu Çizim Yazdır Kuyu Göster

Kuyu Bilgileri

Kuyu No 639 Pafta No 1/5000-C
Sağa Değeri 47750 Mevki [Erikli kır T.Bah
Yukarı Değeri 82514 Makine Markası JOY BUDA 22 HP
Kot Değeri 1163,36 Başlama Tarihi 28/4/1973
Kalınlık 92 Bitiş Tarihi 6/5/1973

Katman Bilgileri

Katman No 13 Kıl. yeşil
Malzeme Kod 21


MALZEME

MALZEME KOD	MALZEME ADI
1	Bitkisel örtü tabakası
2	Doğru Toprak
3	Gre. beyaz
4	Gre. kıllı
43	Kalker
4	Kalker, kıllı
5	Kalker, silisli
6	Kil
7	Kil, beyaz
14	Kil, çakıllı
8	Kil, kahverengi
9	Kil, kömürlü
10	Kil, kumlu
11	Kil, marnlı
12	Kil, sarı
13	Kil, sert

Ok Cancel

KATMANNO	MALZEMEKOD	BITİSKOTU	KALINLIK
1	6	0	27
2	23	0	0,5
3	22	0	4,5
4	24	0	3,25
5	25	0	1,5
6	23	0	0,75
7	25	0	1,5
8	25	0	4,5
9	23	0	1
10	6	0	3
11	23	0	0,5
12	13	0	26,5
13	21	0	17,5

Figure 7.9 "Katmanlar" window with the list of material names and codes.

The values of three of this attributes can be entered by using this part of the window. These are “*Katman No*”, “*Malzeme Kod*” and “*Kalınlık*”. When code of material is entered, related material name takes place on the box which is right side of material code box, automatically. But, there are only material names on data sheet and it is not always possible to remember the code numbers. On the other hand, sometimes new material names can be met. Because of that reasons, there is  button on a level with “*Malzeme Kod*”. When this button is selected, the list of “*Malzeme Kod*” and “*Malzeme Adı*” appears that can be seen Figure 7.9. User can look for material name and related material code that was met on data sheet. If material name can not be found, it means that this material name is a new one or it is only written at different form of any of material in the list. If the name is a new one, user must select the tab “*Malzeme Tanımlama*” after selection of the tab “*Tanımlamalar*” on the main window. Thickness of layer is a knowledge taken from data sheet. However, altitude of the end of layer “*Bitiş Kotu*” will be calculated by thickness of the layer is subtracted from end altitude of the preceding layer and added to the table. Entered values of attributes will be added to the list which is below of data box when the user select or click the “*Tamam*” button. And data is also added to the table.

A cross sectional diagram can be seen at right side of the “*Kuyular*” window when the data related with layers of a bore-hole is entered and the “*Kuyu Göster*” button selects. Each layer is shown different color and type of lineated that chosen before, such as explained at section 7.3.2. User can change the scale and separate the layer from each other with “*Ölçek*” and “*Aralık*” boxes respectively. Cross sectional diagram of selected bore-hole can be shown separately and printed by “*Kuyu Çizim Yazdır*” button. Related window is shown at Figure 7.10.

Some materials can burn. Such as marn and coal, these materials are analyzed. The results of the analyzed layer can be entered by using “*Analiz Sonuçları*” which is the right side of the list. At first, a layer that analysis results exist is chosen from the list and then “*Analiz Sonuçları*” button is selected. Related window can be seen at Figure 7.11. Now, karot percentage (*Karot Yüzdesi*), humidity percentage (*Rutubet*

Yüzdesi), ash percentage (Kül Yüzdesi) and heat energy (Kalori) can be entered to the related box. When the “*Tamam*” button is selected, the values in the boxes are added the relational table named “*Analiz*” (Analysis Results) that can be seen at Table 7.5. This table includes two attributes besides results of analysis; Bore-Hole Number and Layer Number, that these two attributes are selected as key attributes.

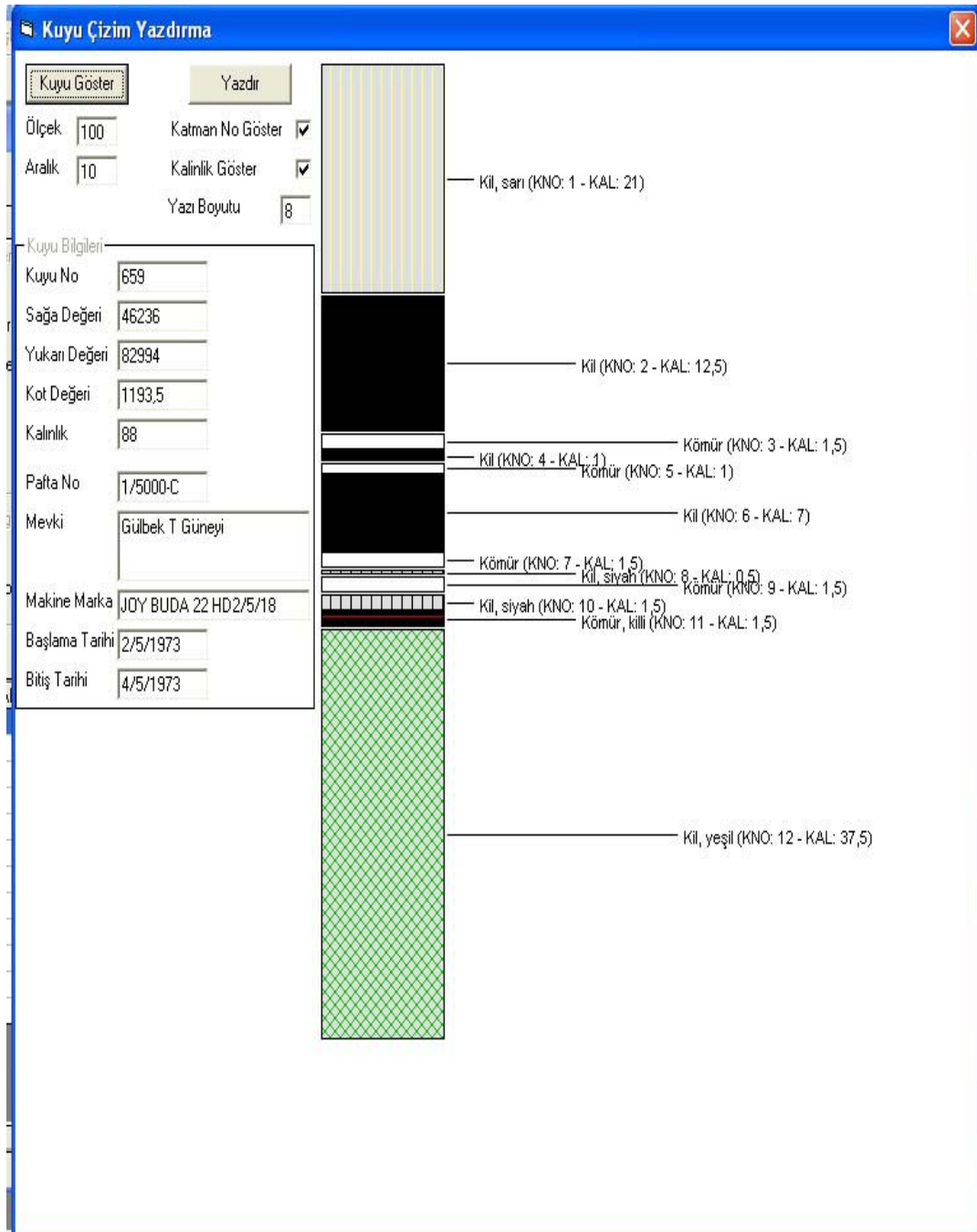


Figure 7.10 Cross sectional diagram of a bore-hole.

Kuyular Ölçek Aralık 100 10

Kuyuyu Göster Kuyuyu Gizim Yazdır

Katmanlar

Kuyuyu Bilgileri

Kuyuyu No: 639 Pafta No: 1/5000-C

Sağa Değeri: 47750 Mevki: Erikti İkn T. Batı

Yukarı Değeri: 82514 Makine Markası: JOY BUDA 22 HP

Kot Değeri: 1163,36 Başlama Tarihi: 28/4/1973

Kalınlık: 92 Bitiş Tarihi: 6/5/1973

Katman Bilgileri

Katman No: 9 Bitiş Kotu: 0

Malzeme Kod: 23 Kalınlık: 1

Malzeme Adı: Kömür

Analiz Sonuçları

Katman Analiz Sonuçları

KATMAN NO	MALZEME KODU	BITİSKOTU	KALINLIK
1	6	0	
2	23	0	
3	22	0	
4	24	0	
5	25	0	
6	23	0	
7	25	0	
8	25	0	
9	23	0	
10	6	0	
11	23	0	
12	13	0	
13	21	0	

Kuyuyu No: 639

Katman No: 9

Malzeme Kodu-Adı: 23 Kömür

Karot Yüzdesi: 100

Rutubet Yüzdesi: 28

Kül Yüzdesi: 46

Kalori: 2134

Kapat Yeni Değiştir Sil Tamam İptal

Kapat Değiştir Sil Tamam İptal

Figure 7.11 Window of Analiz.

Table 7.5 Form of the “Analiz” table.

KUYUNO	KATMANNO	KAROTYUZDESI	RUTUBETYUZDESI	KULYUZDESI	KALORI

“*Yeni*” button must be selected after the “*Tamam*” button was selected, when the user needs to enter new layer properties and/or new results of analysis. “*Değiştir*” and “*Sil*” buttons can be used to change and delete the knowledge of selected layer, respectively. “*İptal*” button can prevent the user to add the data to the table at the database. The window will close when “*Kapat*” button is selected. Later, if the user wants to enter new data, this window can be opened such as explained before.

7.4 Beginning of Supervised Data Mining

The original data which be used, was handled from Seyitömer coal-bed region. This data contains cross sectional diagram, knowledge of location, and result of analysis as shown at Figure 7.1. Thicknesses and types of earth materials can be seen at the cross sectional diagram. Some materials can be available repeatedly at different layers.

Location knowledge contains coordinates of bore-hole in three dimensional spaces, depth of bore-hole end, region name, machine brand used for boring, beginning and finishing date to bore. Knowledge about result of analysis contains humidity percentage and ash percentage of layer of coal, and heat energy getting out of coal as Kcal/Kg.

In this step of our study, special properties that are important for data mining process were chosen. New database that is necessary for data mining were begun to create. And we began to form conditions of data mining.

7.4.1 Beginning of data mining processes

Process menu (İşlemler) can be seen on the menu bar at main interface screen from the Figure 7.12. There are three process: “Malzeme Yüzdeleri” (percentage of


material), “Malzeme Üstündeki ve Altındaki Malzemeler” (materials that on top of and below the chosen material) and “İnceleme” (research). The amount of the bore-hole, that any types of materials were met, was calculated proportionally with the process “Malzeme Yüzdeleri”. Result of process can be seen at Figure 7.13.

In Figure 7.13, the meaning of the first row of the table, for example, cod number (kod numarası) of material is 10, name of the material (malzeme adı) is clay with sand (kil, kumlu), this material was seen only one bore-hole (kuyu sayısı) and the last number is the percentage of bore-hole which the material named clay with sand was come into view (yüzdesi). This process is applied on the whole data automatically and the user can not have any choice.

Second process of “İşlemler” is about the material of the above layer of a chosen material when the chosen material is met at first vertically and the material of below layer of chosen material when the chosen material is met at last. It is shown at Figure 14 that there are two choices. The first is “Kuyular Olarak Göster” and the second is “Toplam Olarak Göster”. With the process “Kuyular Olarak Göster”, two tables are shown that can be seen left side of Figure 7.14.



Figure 7.12 The menu of the processes.



MALZEMEKOD	MALZEMEADI	KUYUSAYISI	YUZDE
16	Kil, siyah	99	32,5657894736842
17	Kil, siyah yanmıyor	1	1,328947368421053
18	Kil, şistli	4	1,31578947368421
20	Kil, yanık	12	3,94736842105263
21	Kil, yeşil	277	91,1184210526316
22	Kil, yumuşak	2	1,657894736842105
23	Kömür	222	73,0263157894737
24	Kömür, ezik	71	23,3552631578947
25	Kömür, killi	180	59,2105263157895
26	Kömür, killi ve şistli	2	1,657894736842105
27	Kömür, şistli	5	1,64473684210526
28	Kömür, şistli ezik	1	1,328947368421053
29	Marn	44	14,4736842105263
30	Marn killi	57	18,75

Figure 7.13 Result of the process “Malzeme Yüzdeleri”.

The table which is left and upper part shows the material of the above layer of a chosen material when the chosen material is met at first vertically with the bore-hole number. The table which is left and lower part shows the material of below layer of chosen material when the chosen material is met at last with the bore-hole number.

As shown at Figure 7.15, this time we get the above and below materials of chosen material that we met first and last time respectively, in a bore-hole. “Kuyu Sayısı” gives us the total number of bore-hole which the same material is met at upper and lower layer of chosen material. The parts of Figure 7.14 and Figure 7.15 that named “Kuyular ve katmanlar” contain the list of bore-holes and their layers. The aim of these parts is to check of the results of above applications.

7.4.2 Main Interface for Examination on Input Data

User may want to make some operations on raw data after the end of data input. And also, user may want to investigate result of old examination. Because of this reason, we had necessity of screen named “İnceleme”. This screen can be seen in Figure 7.16.

Malzeme Üstündeki ve altındaki Malzemeler

Malzeme Kod: 23

Kuyular Olarak Göster
 Toplam Olarak Göster

Malzemenin İlk Raslandığı Katmanın İlk Üstünde Yer Alan Malzemeler

MALZEMEKOD	MALZEMEADI	KUYUNO	KATMANNO
10	Kil, kumlu	938	3
11	Kil, marnlı	683	3
11	Kil, marnlı	694	4
11	Kil, marnlı	715	2
11	Kil, marnlı	732	6
11	Kil, marnlı	750	2
11	Kil, marnlı	755	3
11	Kil, marnlı	762	2
11	Kil, marnlı	773	4
11	Kil, marnlı	775	7
11	Kil, marnlı	870	2
11	Kil, marnlı	911	3

Malzemenin Son Raslandığı Katmanın İlk Altında Yer Alan Malzemeler

MALZEMEKOD	MALZEMEADI	KUYUNO	KATMANNO
16	Kil, siyah	744	17
16	Kil, siyah	765	10
16	Kil, siyah	848	6
16	Kil, siyah	856	3
16	Kil, siyah	885	6
16	Kil, siyah	899	6
16	Kil, siyah	905	8
16	Kil, siyah	908	6
16	Kil, siyah	914	5
16	Kil, siyah	918	4
16	Kil, siyah	919	12
17	Kil, siyah, uarmık	673	9

Kuyular ve Katmanlar

KUYUNO	KATMANNO	MALZEMEKOD	MALZEMEADI
701	4	12	Kil, sarı
701	5	23	Kömür
701	6	25	Kömür, kıllı
701	7	21	Kil, yeşil
701	8	25	Kömür, kıllı
701	9	6	Kil
701	10	25	Kömür, kıllı
701	11	21	Kil, yeşil
701	12	25	Kömür, kıllı
701	13	21	Kil, yeşil
701	14	23	Kömür
701	15	21	Kil, yeşil
702	1	1	Bitkisel örtü tab.
702	2	12	Kil, sarı
702	3	24	Kömür, ezik
702	4	21	Kil, yeşil
702	5	34	Serpantin
703	1	1	Bitkisel örtü tab.
703	2	12	Kil, sarı
703	3	5	Kalker, silisli
703	4	12	Kil, sarı
703	5	30	Marn, kıllı
703	6	31	Marn, silisli
703	7	24	Kömür, ezik
703	8	6	Kil
703	9	21	Kil, yeşil
704	1	1	Bitkisel örtü tab.
704	2	12	Kil, sarı

Figure 7.14 List of the material of the above layer of a chosen material that is met at first vertically and the material of below layer of chosen material that is met at last with the bore-hole number.

User can give a sequence number at “İnceleme No” as a research number. Under “İnceleme No” with the text box “Açıklama”, some explanation can be made as a pre-knowledge about examination made. User may remember subject of that examination with this explanation.

7.4.3 Reduction of Raw Data for Creation New Database

In data mining, main goal is to handle meaningful patterns. If there are so many attributes or/and each attribute contains a lot of different values, huge number of classes, in other word, patterns will be formed. When the number of patterns will be getting higher value, to choose meaningful patterns will be harder, even impossible. To reduce number of patterns, we need some reduction operations on raw data. These operations will be explained below.

7.4.3.1 Reduction of material names

Some materials at bore-holes may not be valuable for some researchers. For example, plant cover layer is available most of the bore-hole as the first layer. So that, plant cover layer will not be used in new database as a material name. Some materials can be met a few number of bore-hole. These materials keep out of new database. Sometimes, we can meet very thin clay layers between two coal layers. This type of layers will be added to coal layers.

Some materials are very similar, but they are named differently, such as “Green Clay” and “Light green clay”. Users may want to gather these two materials under same code number and material name. Using this method, many values of some attributes and tuples will be eliminated and number of patterns can be reduced. These operations can be made with using screen named “Malzeme Birleştirme” as shown at Figure 7.17.

User can choose new code number and material name; even they may be the old ones that were found in raw data, with using text boxes named “Yeni Malzeme Kod”

Malzeme Üstündeki ve altındaki Malzemeler

Malzeme Kod: Bul

Kuyular Olarak Göster
 Toplam Olarak Göster

Kuyular ve Katmanlar

MALZEMEKOD	MALZEMEADI	KUYUSAYISI	KATMANNO	MALZEMEKOD	MALZEMEADI
1	Bitkisel örtü tab.	6	4	701	12 Kil, sarı
4	Kalker, killi	1	5	701	23 Kömür
5	Kalker, silisli	4	6	701	25 Kömür, killi
6	Kil	74	7	701	21 Kil, yeşil
7	Kil, beyaz	3	8	701	25 Kömür, killi
8	Kil, kahverengi	2	9	701	6 Kil
9	Kil, kömürlü	6	10	701	25 Kömür, killi
10	Kil, kumlu	2	11	701	21 Kil, yeşil
11	Kil, marnlı	12	12	701	25 Kömür, killi
12	Kil, sarı	7	13	701	21 Kil, yeşil
13	Kil, sert	6	14	701	23 Kömür
16	Kil, siyah	8	15	701	21 Kil, yeşil
0	YOK	1	1	702	1 Bitkisel örtü tab.
5	Kalker, silisli	1	2	702	12 Kil, sarı
6	Kil	39	3	702	24 Kömür, ezik
7	Kil, beyaz	1	4	702	21 Kil, yeşil
9	Kil, kömürlü	7	5	702	34 Serpantin
10	Kil, kumlu	2	1	703	1 Bitkisel örtü tab.
11	Kil, marnlı	2	2	703	12 Kil, sarı
12	Kil, sarı	2	3	703	5 Kalker, silisli
13	Kil, sert	1	4	703	12 Kil, sarı
16	Kil, siyah	17	5	703	30 Marn, killi
17	Kil, siyah yanmış	1	6	703	31 Marn, silisli
21	Kil, vesil	116	7	703	24 Kömür, ezik
			8	703	6 Kil
			9	703	21 Kil, yeşil
			1	704	1 Bitkisel örtü tab.
			2	704	12 Kil, sarı

Figure 7.15 List of the material of the above layer of a chosen material that is met at first vertically and the material of below layer of chosen material that is met at last with the total number of bore-hole number.

İnceleme No: 2
Açıklama: DENEME-1

İNCELEME NO	AÇIKLAMA
1	DENEME 1 2
2	DENEME-1
3	İd
4	Örnek-1

Kapat
Yeni
Değiştir
Sil
Tamam
İptal

Figure 7.16 “İncelemeler”; first window when user select the tab “İnceleme” from the list “İşlemler” at main window.

Malzeme Birleřtirme

İncelemeler

İnceleme Bilgileri:

İnceleme No | 2

Açıklama | DENEME-1

Yeni Malzeme Kodları

Yeni Malzeme Kod | 2

Yeni Malzeme Adı | Kil

Yeni Malzeme İçindekiler

MALZEMEKOD	YMALZEMEADI
1	Bitkisel örtü tabakası
2	Kil

Katmanları İndirgeme

Katmanları Düzenleme

Kapat

Yeni

Değiřtir

Sil

İtamam

İptal

Figure 7.17 “Malzeme Birleřtirme” window for reduction of material names.

Yeni Malzeme İçerikleri

İnceleme No: 2
 Açıklama: DENEME-1
 Yeni Malzeme Kodu: 2
 Yeni Malzeme Adı: Kil

Malzemeler

Malzeme Adı	Malzeme Kodu
YOK	0
Kil, kumlu	10
Kil, marlı	11
Kil, çakıllı	14
Kil, silisli	15
Kil, siyah yarıymıyör	17
Kil, şistli	18
Kil, şistli siyah	19
Gre, beyaz	2
Kil, yanık	20
Kil, yeşil	21
Kil, yumuşak	22
Kömür	23
Kömür, ezik	24
Kömür, kılı	25
Kömür, kılı ve şistli	26
Kömür, şistli	27
Kömür, şistli ezik	28
Marı	29
Gre, kılı	3

Yeni Malzeme İçindekiler

Malzeme Adı	Malzeme Kodu
Kil, sarı	12
Kil, sert	13
Kil, siyah	16

Katmanları İndirgeme

Kapat

Kayıdet

Yeni

Değiştir

Sil



İtamam

İptal

Figure 7.18 Selection material names to create new material group.

and “Yeni Malzeme Adı” respectively. These data will take part in the list that is the lower part of text boxes. After that, user must select “Yeni Malzeme İçindekiler” to choose materials for combining under given material name and code number. For this operation, necessary user interface is shown at Figure 7.18.

The text boxes “İnceleme No” (Research Number), “Açıklama” (Explanation), “Yeni Malzeme Kod” (New Material Code) and “Yeni Malzeme Adı” (New Material Name) are given as knowledge only. User can not change contents of these boxes.

User can select material name from the list at left part, then select the button  to send the selected material name to the right part for create a group of material names that combined under new code and name. If user wants to drop any one of the material name from the left part list, user must select that name first, and than button  must be selected. Finally “Kaydet” (Save) button is selected to save the choice. User will return the “Malzeme Birleştirme” screen when “Kapat” button is selected. These operations must do repeatedly for creation of every new group code and name. A sample screen for the result of operations that are explained below can be seen at Figure 7.19.

7.4.3.2 Creation new database with renamed materials

After the material combined process, user must select the tab named “Katmanları Düzenleme” to open the screen shown at Figure 7.20. User must select the “Düzenle” button that is on right side of the window. When this button is selected new reduced database will be created automatically, with new code numbers and names of associated materials. The left window of bottom of the screen shows new code numbers of materials. It can be seen that same code numbers come one under the other. In the left window, layers that have same code number and name are combined and they become only one layer which thickness of this new layer is total thicknesses of combined layers.

Inceleme Katmanları İndirgeme Katmanları Düzenleme

Malzeme Birleştime

Inceleme Bilgileri

Inceleme No

Açıklama

Yeni Malzeme Kodları

Yeni Malzeme Kod

Yeni Malzeme Adı

Yeni Malzeme İçindekiler

MALZEMEKOD	YMALZEMEADI
0	YOK
1	Bitkisel örtü tabakası
2	Gre
3	Kalker
4	Kil
5	Kömür
6	Marn
7	Serpantin
8	Sileks
9	Şist
10	Kum
11	Opal
12	T abaka Tenavübü

MALZEMEKOD	MALZEMEADI
6	Kil
7	Kil, beyaz
8	Kil, kahverengi
10	Kil, kumlu
11	Kil, marnlı
12	Kil, sarı
13	Kil, sert
14	Kil, çakıllı
15	Kil, silisli
16	Kil, siyah
17	Kil, siyah yanmıyor
18	Kil, şistli
19	Kil, şistli siyah
20	Kil, yanık
21	Kil, yeşil
22	Kil, yumuşak

Kapat

Yeni

Değiştir

Sil

Tamam

İptal

Figure 7.19 A sample for grouped materials.

İnceleme Katmanları İndirgeme

Katmanları Düzenleme

İnceleme Bilgileri
 İnceleme No: Düzenle
 Açıklama: Malzeme Üst-Alt

Katmanlar Birleştirme

Malzeme Birleştirme

Katmanlar Birleştirilmiş

Malzeme Yüzdeleri

Kıyı Harita

Katmanlar Düzenlenmiş		Katmanlar Birleştirilmiş		Malzeme Üst-Alt		Kıyı Harita					
KUYUNO	KATMANNO	MALZEMEKOD	YMALZEMEAD	KALINLIK	KUYUNO	KATMANNO	MALZEMEKOD	YMALZEMEAD	BASKOTU	BITISKOTU	KALINLIK
639	1	4 Kil		27	639	1	4 Kil		1163,36	1136,36	27
639	2	5 Kömür		0,5	639	2	5 Kömür		1136,36	1135,86	0,5
639	3	4 Kil		4,5	639	3	4 Kil		1135,86	1131,36	4,5
639	4	5 Kömür		3,25	639	4	5 Kömür		1131,36	1118,86	12,5
639	5	5 Kömür		1,5	639	10	4 Kil		1118,86	1115,86	3
639	6	5 Kömür		0,75	639	11	5 Kömür		1115,86	1115,36	0,5
639	7	5 Kömür		1,5	639	12	4 Kil		1115,36	1071,36	44
639	8	5 Kömür		4,5	640	1	1 Birkisel örtü tab.		1193,52	1187,52	6
639	9	5 Kömür		1	640	2	5 Kömür		1187,52	1162,52	25
639	10	4 Kil		3	640	3	4 Kil		1162,52	1160,77	1,75
639	11	5 Kömür		0,5	640	4	6 Marn		1160,77	1134,52	26,25
639	12	4 Kil		26,5	640	8	5 Kömür		1134,52	1119,27	15,25
639	13	4 Kil		17,5	640	9	4 Kil		1119,27	1118,77	0,5
640	1	1 Birkisel örtü tab.		6	640	10	5 Kömür		1118,77	1114,77	4
640	2	5 Kömür		25	640	11	4 Kil		1114,77	1113,77	1
640	3	4 Kil		1,75	640	12	5 Kömür		1113,77	1107,77	6
640	4	6 Marn		4	640	16	4 Kil		1107,77	1094,77	13
640	5	6 Marn		1,5	641	1	1 Birkisel örtü tab.		1150,45	1150,2	0,25
640	6	6 Marn		2	641	2	4 Kil		1150,2	1102,45	47,75
640	7	6 Marn		18,75	642	1	1 Birkisel örtü tab.		1187,86	1184,36	3,5
640	8	5 Kömür		15,25	642	2	4 Kil		1184,36	1168,36	16

Figure 7.20 New database with new material names and coeds.

Malzemenin Üstündeki ve Altındaki Malzemeler

Malzeme Kod: 17

Malzemenin İlk Rastlandığı Katmanın İlk Üstünde Yer Alan Malzemeler

Kuyular Olarak Göster
 Toplam Olarak Göster

Bul

Kuyular ve Katmanlar

MALZEMEKOD	YMALZEMEAD	KUYUNO	KATMANNO
1	Bitkisel örtü tab.	640	1
1	Bitkisel örtü tab.	640	1
1	Bitki örtüsü	640	1
1	Bitki örtüsü	684	1
1	Bitkisel örtü tab.	684	1
1	Bitkisel örtü tab.	684	1
1	Bitkisel örtü tab.	743	1
1	Bitkisel örtü tab.	743	1
1	Bitki örtüsü	743	1
1	Bitki örtüsü	767	2
1	Bitkisel örtü tab.	767	2
1	Bitkisel örtü tab.	767	2

Malzemenin Son Rastlandığı Katmanın İlk Altında Yer Alan Malzemeler

MALZEMEKOD	YMALZEMEAD	KUYUNO	KATMANNO
13	Kil renkli	656	17
13	Kil renkli	658	12
13	Kil renkli	661	15
13	Kil renkli	662	9
13	Kil renkli	663	7
13	Kil renkli	664	12
13	Kil renkli	665	17
13	Kil renkli	666	6
13	Kil renkli	667	9
13	Kil renkli	669	12
13	Kil renkli	670	7
13	Kil renkli	677	9

Malzemenin İlk Rastlandığı Katmanın İlk Üstünde Yer Alan Malzemeler

MALZEMEKOD	YMALZEMEAD	KUYUNO	KATMANNO
13	Kil renkli	639	1
17	Kömür	639	2
13	Kil renkli	639	3
17	Kömür	639	4
18	Kömür killi	639	5
17	Kömür	639	6
18	Kömür killi	639	7
17	Kömür	639	9
13	Kil renkli	639	10
17	Kömür	639	11
13	Kil renkli	639	12
1	Bitkisel örtü tab.	640	1
1	Bitki örtüsü	640	1
1	Bitkisel örtü tab.	640	1
17	Kömür	640	2
13	Kil renkli	640	3
4	Kil	640	4
4	Marm	640	4
4	kil kömürü	640	4
17	Kömür	640	8
13	Kil renkli	640	9
17	Kömür	640	10
13	Kil renkli	640	11
17	Kömür	640	12
18	Kömür killi	640	13
17	Kömür	640	14
18	Kömür killi	640	15
13	Kil renkli	640	16

KALINLIK

27
0,5
4,5
3,25
1,5
0,75
6
1
3
0,5
44
6
25
1,75
26,25
15,25
0,5
4
1
1,25
2

Düzenle

Kuyu Harita

Figure 7.22 The screen after “Malzeme Alt-Üst” button was selected.

Inceleme **Katmanları İndirgeme**

Inceleme Bilgileri

Inceleme No: Katmanları Düzenleme

Açıklama: Çıkarılacak Malzemelerin Yüzdesi: Katman İndirge

Malzeme Birleştirme

KUYUNO	ATMANNO	LZEMEKOD	BITISKOTU	KALINLIK
641	1	1	1150,2	0,25
906	1	1	1151,36	12
899	1	1	1200,86	32
796	1	1	1136,52	15
809	1	1	1164,02	0,25
798	1	1	1140,65	16
649	1	1	1175,4	0,5
898	1	1	1198,63	33
787	1	1	1181,59	1,5
794	1	1	1170,75	8
780	1	1	1178,29	1
647	1	1	1175,75	0,25
875	2	3	1133,33	2
843	15	3	1098,69	0,5
712	5	3	1168,48	1
744	12	3	1092,3	11
874	3	3	1142,75	4
877	3	3	1123,64	3
714	4	3	1172,01	0,5
834	4	3	1181,14	1
835	2	3	1162,87	1
899	3	3	1199,86	1
754	12	3	1133,95	1,5
693	5	3	1215,68	2,5
887	2	3	1138,17	1
703	3	3	1194,81	2
829	2	3	1144,87	1,5

Figure 7.23 Screen for reduction some materials according to chosen threshold.

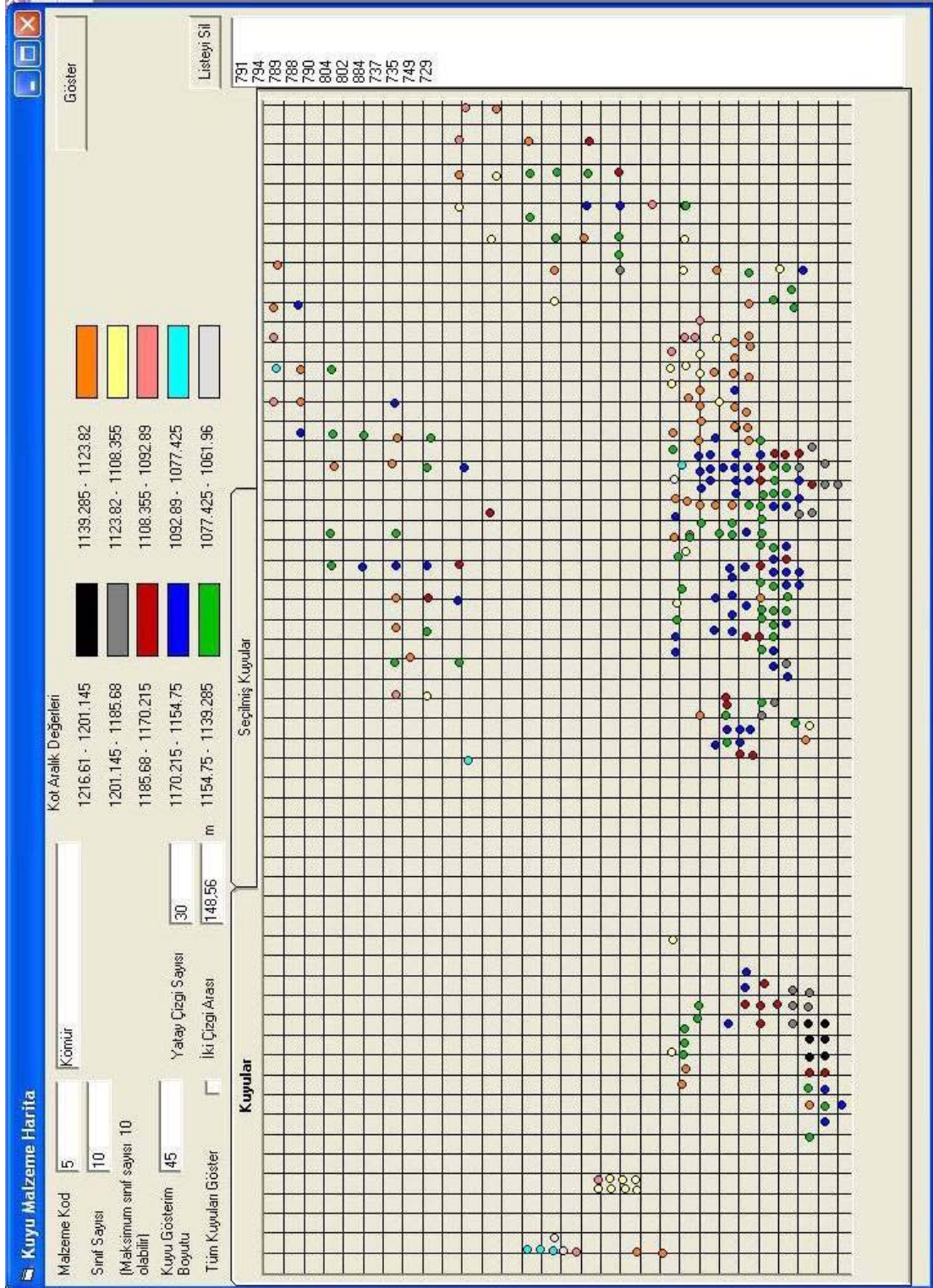


Figure 7.24 Bore-holes location on plane area.

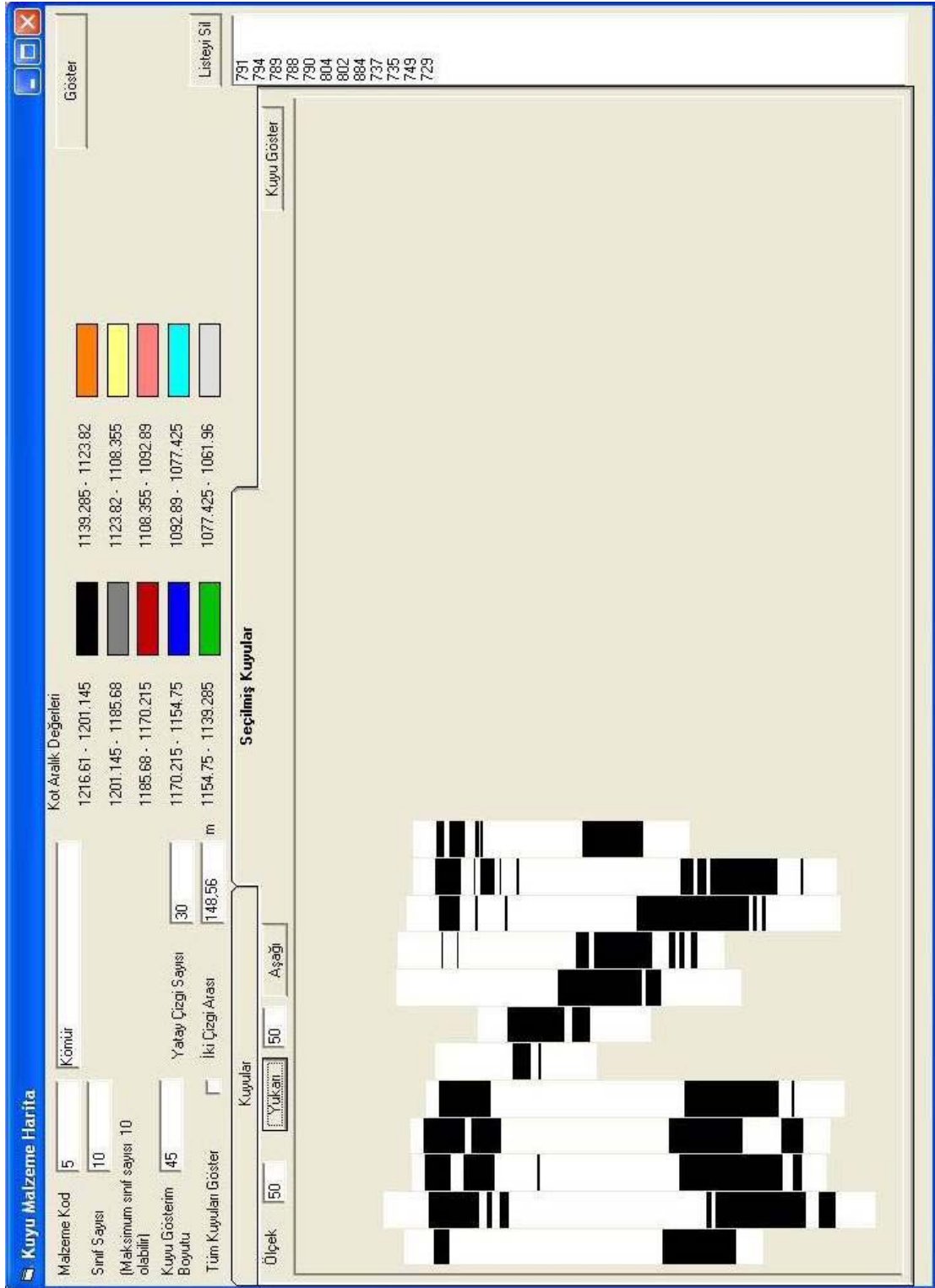


Figure 7.25 Cross sectional diagram of selected bore-holes.

User can apply two operations that applied on row data explained at section 3.1, with “Malzeme Yüzdeleri” button, new percentage of material can be shown. If any one want to see which materials taking place over where the chosen material in each bore-hole first met and materials taking place under where the chosen material in each bore-hole, “Malzeme Üst-Alt” button must be selected. Results of these operations are shown Figure 21 and Figure 22 respectively.

7.4.3.3 Removing materials rarely met

Some materials can be found in a few numbers of bore-holes. Because of that reason, their percentages are very low. Or some materials may not be important for the user. So that, user may want to remove such materials from database.

For that process, user can choose a percentage as a threshold, and write that percentage in the text box named “Çıkarılacak Malzemelerin Yüzdesi” at “Katmanları İndirge” screen shown at Figure 23. When “Katman İndirge” button is selected, materials that have percentage less than threshold are removed from new database created at the screen “Katmanları Düzenleme”.

7.4.4 Plotting the location of bore-hole

If the user wants to show the bore-holes that dept of the chosen material is the same interval of altitude when met first a bore-hole vertically, user must select the “Kuyu Harita” button. Related screen is shown at Figure 24 that has two tabs; “Kuyular” and “Seçilmiş Kuyular”. At the tab “Kuyular”, material code number must be written to the “Malzeme Kod” box and number of classes to the “Sınıf Sayısı” box. Maximum class number is chosen as ten to handle meaningful patterns. Dimension of points can be defined by “Kuyu Gösterim Boyutu”. Number of horizontal lines can be defined by user with “Yatay Çizgi Sayısı” box. When the user selects “Göster” button, points that represent bore-hole display on the grid. Intervals of depth of material are shown with same color.

The list of bore-hole number can be displayed at the right side of the window when user selects related points. Figure 7.25 belongs to the tab “Seçilmiş Kuyular”. This figure shows cross-sectional diagram of selected bore-holes. If user wants to examine new bore-holes, “Listeyi Sil” button must be selected to erase the previous list.

7.4 Discussion

It can be seen that ground has various properties even distance between two bore-holes than 200 meters after examination on the data. For this reason, inductive learning model is used in this study. Bore-holes are examined one by one to obtain general knowledge about ground and layer tendencies. Each bore-hole has very different properties from the others. We used supervised techniques as summarization and classification instead of clustering to reduce the number of classes. To find the above and below materials of chosen material that is met the first and the last respectively is the example of summarization technique. Bore-holes are gathered into the groups with respect to depth of coal layer that is met the first, as can be seen at the Figure 7.24. This operation is the example of classification technique.

There are two models at inductive learning; ‘Environment’ and ‘Class’. We have chosen class model. Because, data used in application do not change with respect to time. Classification functions are the algorithms that used to determine the description and classes. An algorithm example related with the Figure 7.14 is given below.

```
Select ‘Amalzeme Kod’ // User selects a material code
Create table MalzemeUst (KuyuNo, KatmanNo, MalzemeKod).
Get the values of attributes KuyuNo, KatmanNo, MalzemeKod from table
‘Katman’ (rst)
Sort table MalzemeUst with respect to ‘KuyuNo’ then ‘KatmanNo,’
Onceki KuyuNo = 0
```



```

Onceki KatmanNo = 0
Onceki MalzemeKod = 0
While NOT rst.EOF
    IF rst.MalzemeKod = AmalzemeKod
        THEN
            IF rst.KuyuNo <> Onceki KuyuNo
                THEN
                    MalzemeUst.KuyuNo = rst.Kuyu No
                    MalzemeUst.Katman No = rst.KatmanNo
                    MalzemeUst.MalzemeKod = rst.MalzemeKod
                    OncekiKuyuNo = rst.KuyuNo
                    Move next bore-hole
                ELSE
                    MalzemeUst.KuyuNo = OncekiKuyuNo
                    MalzemeUst.KatmanNo = OncekiKatmanNo
                    MalzemeUst.MalzemeKod = OncekiKatmanNo
                    OncekiKuyuNo = rst.KuyuNo
                    Move next bore-hole
                ENDIF
            ELSE
                OncekiKuyuNo = rst.KuyuNo
                OncekiKatmanNo = rst.KatmanNo
                Onceki MalzemeKod = rst.MalzemeKod
                Move next bore-hole
            ENDIF
        LOOP

```

Multiple classes are determined as it can be seen from given algorithm. Classification operation applied on all bore-holes. Simple descriptions are chosen to identify the classes. With these reasons, if any bore-hole is a negative example for a class it must belongs to any other class. On the other hand, if user chooses different material code, number of classes and descriptions can also be different. In addition at

the ‘İşlemler’, set of attributes do not change, but values of attributes are changed by the user.

Both generalization and specification operations are used for classification. At the ‘İşlemler’, the procedure of combining materials is an example of generalization operation. When the classification is applied on database obtained from real world number of elements of classes greater than that same operation is applied on database with combined materials. Of course these operations are not automatic but choice of user. Classification with respect to depth of the chosen material that first met at the bore-hole can be represented as an example of specialization operation.

Classes presented at Figure 7.14 and 7.24 can be used for correctness value of class descriptions. At Figure 7.14, classification is made according to the formation type that is just above the chosen material (for example coal) is met first at the bore-hole. ‘Malzeme Adı’ gives the rule and names of the class and, ‘Kuyu No’ gives elements of relevant class at the left and above side of the figure. At the right side, the list of the bore-holes is shown. Number of bore-holes that material code no 23 can be found, is 222. Sum of the elements number of classes is also 222 at the left and above side of the figure. This result shows that each element belongs to only one class. Besides, inquiries are built on concrete reality, all classes cover only positive examples. For these reasons both formulas given at Chapter Four must be equal to;

$$classification_accuracy = \frac{|\sigma_D(S) \cap C|}{|\sigma_D(S)|} = 1$$

$$coverage = \frac{|\sigma_D(S) \cap C|}{|C|} = 1$$

These results show that necessary conditions are satisfied for the classes.

Irrelevant attributes and knowledge did not used at these operations. Knowledge such that drilling machine brand, date of drilling, is gathered at the distinct table.

When depth of the formation is necessary, it is calculated as summation of the thicknesses above layers consecutively. Because of the main database includes elevation of the beginning of the bore-hole and thicknesses of the layers. Subtraction between elevation of bore-hole beginning and thickness of layer gives the elevation of end of the first layer. The elevation of end of each layer is the elevation of the beginning of the below layer. Because they can be calculated, these values are not used at the main table.

Each analysis is saved with name at the part 'İşlemler'. In this way users can retrieve earlier knowledge when they want. And they can use this knowledge for new analysis.

It is accepted that data is complete and correct to form the descriptions. Only deterministic rules are used to extract classes correctly. We do not meet with missing data during the loading knowledge to the computer. In addition, formations are represented with name and graphically at cross-sectional diagrams of bore-holes. It can be found the name of the formation from the cross-hatching form even if the name is forgotten. Because of the samples that are taken with "karot" are thrown after the measuring, to find and correct mistakes that can be occurred during the measuring of the thicknesses of the layer is almost impossible. For this reason, it is accepted that there is no measuring mistake. It is only the problem about the data, depths of layers show great differences. This problem is solved by using intervals. It is possible to enter the system new data. New bore-holes can be added related class to execute the program after new data input.

Tables and graphs are used to represent the retrieval knowledge. The aim of the usage tables and graphs is to simplify for the user to understand the results. If the retrieval knowledge getting from the main table is very large to understand easily, user can eliminate or combine unnecessary values for more comprehensible table.

CHAPTER EIGHT

CONCLUSION

To reduce the chaos at the stage of obtaining data, naming standardization is needed. But this application presented does not claim to build up this standardization. However, it is noted that this will have an accordance (harmony) in between teams. Bore-hole cross-sections for material knowledge taken off by “karots” from drill holes are shown by computer assistance will ease and accelerate the team’s works.

Field engineers and/or other related personnel can have a view by using examinations which permits application of the program on data for inclination and base condition of the target gem. Therefore, location of the new drill holes can be determined more accurately. At the stage of management, by examining the material depth may especially shown in Figure 3, it will be easier to decide in which areas open and covered mines to be used in basin.

The listed above are the results which could be seen by the searcher. Specialists of the subject may reach to other results. Besides co-worked searches done by the computer programmers together with the same people may enable more clear and useful results for examinations.

At the first sight, application might have some missing points. For example, coal amount taken from basin has not been calculated. However, this program is suitable for advancing and adding this or similar calculations. Main purpose of the program is sampling the DM in earth sciences and show that a profit can be gained. Therefore, the subject is not transferred to the program with all aspects. At the other hand time limitations created an obstacle.

It can be seen clearly that studies can not be limited by what’s done. The necessary studies will start as soon as possible for the program advancement.

REFERENCES

- Brachman, R.J., Khabaza, T., Kloesgen, W., Piatetsky-Shapiro, G. & Simoudis, E. (1996). Mining business databases. *Communication of the Acm.* 39, 42-48.
- Chen, M.S., Han, J. & Yu, P.S. (1996). Data mining: An overview from a database perspective. *IEEE Transaction on Knowledge and Data Engineering.* 8, 866-883.
- Dunham, M. (2002). *Data mining: Introductory and advanced topics*. NJ, USA: Prentice Hall.
- Elmasri, R., & Navathe, S. B. (2004). *Fundamentals of database systems*. United States of America: Addison-Wesley.
- Fayyad, U., Haussler, D. & Stolorz, P. (1996). Mining scientific data. *Communication of the Acm.* 39, 51-57.
- Fayyad, U., Piatetsky-Shapiro, G. & Smyth, P. (1996). The KDD process for extracting useful knowledge from volumes of data. *Communication of the Acm.* 39, 27-34.
- Fayyad, U. & Uthurusamy, R. (1996). Data mining and knowledge discovery in databases. *Communication of the Acm.* 39, 24-26.
- Glymour, C., Madigan, D., Pregibon, D. & Smyth, P. (1996). Statistical inference and data mining. *Communication of the Acm.* 39, 35-41.
- Han, J., & Kamber, M. (2001). *Data mining concepts and techniques*. United States of America: Morgan Kaufmann Publishers.
- Hand, D., Mannila, H., & Smyth, P. (2001). *Principles of data mining*. United States of America: The MIT Press.
- Holsheimer, M. & Siebes, A. (1991). *Data minig: The search for knowledge in databases*. Amsterdam: CWI.

- Imielinski, T. & Mannila, H. (1996). A database perspective on knowledge discovery. *Communication of the Acm.* 39, 58-64.
- Inmon, W.H. (1996). The data warehouse and data mining. *Communication of the Acm.* 39, 49-50.
- Jain, A.K. & Dubes, R.C. (1988). *Algorithms for clustering data*. New Jersey: Prentice-Hall.
- Knorr, E.M. & Ng, R.T. (1996). Finding aggregate proximity relationships and commonalities in spatial data mining. *IEEE Transaction on Knowledge and Data Engineering.* 8, 884-897.
- Özkarahan, E. (1997). *Database Management : Concepts, Design, and Practise* (2nd ed). İzmir: Saray Medical Publication.
- Tan, P., Steinbach, M., & Kumar, V. (2005). *Introduction to data mining*. USA: Addison Wesley.
- Rud, O. P. (2001). *Data mining cookbook*. United States of America: John Wiley & Sons, Inc.
- Westphal, C., & Blaxton, T. (1998). *Data mining solutions*. United States of America: John Wiley & Sons, Inc.