

KADİR HAS UNIVERSITY
GRADUATE SCHOOL OF SCIENCE AND ENGINEERING
PROGRAM OF COMPUTER ENGINEERING



SMART CHATBOTS

RECEP ÇİFÇİ

MASTER'S THESIS

İSTANBUL, JUNE, 2018

RECEP ÇİFÇİ

M. S. Thesis

2018





SMART CHATBOTS

RECEP IFI

MASTER'S THESIS

Submitted to the Graduate School of Science and
Engineering of Kadir Has University
in partial fulfillment of the requirements for the degree of
Master of Science in Computer Engineering

İSTANBUL, JUNE, 2018

DECLARATION OF RESEARCH ETHICS /
METHODS OF DISSEMINATION

I, RECEP ÇİFÇİ, hereby declare that;

- this Master's Thesis is my own original work and that due references have been appropriately provided on all supporting literature and resources;
- this Master's Thesis contains no material that has been submitted or accepted for a degree or diploma in any other educational institution;
- I have followed *Kadir Has University Academic Ethics Principles* prepared in accordance with *The Council of Higher Education's Ethical Conduct Principles*.

In addition, I understand that any false claim in respect of this work will result in disciplinary action in accordance with University regulations.

Furthermore, both printed and electronic copies of my work will be kept in Kadir Has Information Center under the following condition as indicated below:

The full content of my thesis/project will be accessible only within the campus of Kadir Has University.

RECEP ÇİFÇİ



KADİR HAS UNIVERSITY

GRADUATE SCHOOL OF SCIENCE AND ENGINEERING

ACCEPTANCE AND APPROVAL

This work entitled SMART CHATBOTS prepared by RECEP ÇİFÇİ has been judged to be successful at the defense exam on 07/06/2018 and accepted by our jury as Master's Thesis.

APPROVED BY:

Asst. Prof. Dr. ARİF SELÇUK ÖĞRENCİ (Advisor)

Kadir Has University

Asst. Prof. Dr. TANER ARSAN

Kadir Has University

Assoc. Prof. Dr. SONGÜL VARLI ALBAYRAK

Yıldız Technical University

I certify that the above signatures belong to the faculty members named above

Assoc. Prof. Dr. DEMET AKTEN AKDOĞAN

Dean of Graduate School of Science and Engineering

DATE OF APPROVAL:

Table of Contents

ABSTRACT	i
OZET	ii
ACKNOWLEDGEMENTS	iii
LIST OF TABLES	iv
LIST OF FIGURES	v
1. INTRODUCTION.....	1
2. ARTIFICIAL INTELLIGENCE MARKUP LANGUAGE	4
2.1 DEFINITION.....	4
2.2 MOST IMPORTANT TAGS	4
3. ZEMBEREK NLP LIBRARY	7
3.1 DEFINITION.....	7
3.2 ADDING ZEMBEREK TO .NET.....	8
3.3 HOW TO USE ZEMBEREK TO .NET	10
4. DATA MINING.....	11
4.1 Understanding Data Mining.....	11
4.2 WEKA	12
4.3 Data Mining Methods, The Example Data Set and Results	16
5. RUNNING THE PROGRAM	28
5.1 NEURAL NETWORK LIBRARY PROJECT.....	28
5.2 NEURAL NETWORK CONSOLE APPLICATION PROJECT.....	29
5.3 SMARTCHATBOT WEB PROJECT	29
5.4 IMPORTANT FUNCTIONS	35
6. CONCLUSION.....	43
REFERENCES.....	44


SMART CHATBOT

ABSTRACT

Chatbot is an abbreviation of chat and robot. Nowadays, we can see ChatBot as the primitive state of artificial intelligence. Chatbot is an algorithm based on software that we can do some special work, chat and talk for fun.

The emergence and implementation of the ChatBot idea begins with a theory of Alan Turing. The Turing test includes a user, a computer and a person which responding to this user. The computer and people answer the questions the user asks. If the user can not figure out which one is a real person, then the computer is said to have passed the test.

We are trying to create a chatbot adhering to Turing test, but there is a spesific difference between our chatbot and Turing Test. This difference is that in our case user knows that it talks to the computer. There are 3 major change in our chatbot, these are;

- 
- 1- It supports Turkish Language
 - 2- It follows a spesific workflows.
 - 3- It is smart. It means it uses Artificial Intelligence. Gets the question from user, edits the error if exists and response to user according to worklows.

SMART CHATBOT

ÖZET

Kelime anlamı olarak chat ve robot 'un kısaltmasıdır. Günümüzde ise chatbot 'u yapay zekanın ilkel hali yada ilk tohumları gibi görebilirsiniz. Chatbot bazı özel işlerinizi yaptırabileceğiniz, sohbet edebileceğiniz ve eğlence amaçlı konuşabileceğiniz algoritma bazlı bir yazılımdır.

ChatBot fikrinin ortaya çıkışı ve uygulanması Alan Turing teorisiyle başlar. Turing testi, bir kullanıcı ve bu kullanıcıya cevap veren bir bilgisayar ve bir insan bulunur. Bilgisayar ve insan, kullanıcının sorduğu sorulara cevaplar verir. Kullanıcı hangisinin bilgisayar hangisinin ise gerçek bir insan olduğunu anlayamaz ise bilgisayarın testi geçtiği söylenir. Biz Turing testine bağlı bir chatbot yaratmaya çalışıyoruz, ancak chatbot ve Turing Testimiz arasında büyük bir fark var. Bu fark, bizim durumumuzda kullanıcının bilgisayarla konuştuğunu biliyor olmasıdır. Chatbot umuzda 3 temel değişiklik vardır, bunlar;

- 1- Türkçe dil desteği vardır.
- 2- Belirli bir iş akışını takip eder.
- 3- Akıllıdır. Bunun anlamı Yapay Zeka' yı kullanır. Kullanıcıdan soruyu alır, varsa hatayı düzenler ve kullanıcıya iş akışlarına göre yanıt verir

ACKNOWLEDGEMENTS

In my planning, researching and realization of this study, I have benefited from his vigorous knowledge and experience, giving me valuable help, guiding me in terms of subject, source and method, guiding me with interest and support and sharing valuable information. I offer endless thanks to Selçuk Öğrenci.



LIST OF TABLES

Table 4.1: Sentences/Words List	17
Table 4.2: Naive Bayes Summary	19
Table 4.3: Naive Bayes Detailed Accuracy By Class	19
Table 4.4: Naïve Bayes Confusion Matrix	19
Table 4.5: Decision Tree Summary	22
Table 4.6: Decision Tree Detailed Accuracy By Class	22
Table 4.7: Decision Tree Confusion Matrix.....	23
Table 4.8: Neural Network Summary.....	26
Table 4.9: Neural Network Detailed Accuracy By Class.....	27
Table 4.10: Neural Network Confusion Matrix.....	27



LIST OF FIGURES

Figure 1.1: Turing Machine	2
Figure 3.1: Nuget Package	8
Figure 3.2: Add Zemberek to .Net	9
Figure 4.1: Weka UI.....	15
Figure 4.2: Weka Explorer	15
Figure 4.3: Weka Explorer after Data Import	16
Figure 4.4: Decision Tree.....	20
Figure 4.5: Neural Network Types.....	24
Figure 4.6: Neural Network Architectures	25
Figure 4.7: Sigmoid Function Values29	26
Figure 5.1: Smart Chatbot Main Page	29
Figure 5.2: Smart Chatbot Conversation Page	30
Figure 5.3: Smart Chatbot Error Page 1	31
Figure 5.4: Smart Chatbot Error Page 2.....	31
Figure 5.5: Smart Chatbot Success Page.....	31
Figure 5.6: Fraud Category Input Page	32
Figure 5.7: Fraud Category Success Page.....	32
Figure 5.8: Hesap Category Input Page	32
Figure 5.9: Fraud Category Success Page.....	33
Figure 5.10: Ödeme Category Input Page	33
Figure 5.11: Ödeme Category Invoice Types	33
Figure 5.12: Ödeme Category Success Page.....	34
Figure 5.13: Teslimat Category Input Page	34
Figure 5.14: Teslimat Category Success Page	34

1. INTRODUCTION

Chatbot is an abbreviation of chat and robot. Nowadays, we can see ChatBot as the primitive state of artificial intelligence. Chatbot is an algorithm based on software that we can do some special work, chat and talk for fun. For example, suppose that you have all the applications on your phones left and use ChatBot instead. In this case, if you are curious about the weather, you will not need to open the weather application. You will ask how the weather is and he will tell you. In a different way, you can think of chatbots as your personal advisor. You will be able to organize the necessary business meeting for you when the command is given verbally, inform you of your delayed flight and inform you of your cargo which is about to arrive at your house. Today, you'll be able to do these things on your own, using separate internet sites and a large number of different smartphone apps. Today, many large companies are investing heavily in their development. There are some example of ChatBots;

- Amazon – Echo
- Google– Google Home Voice Assistant
- Facebook – Messenger
- Washington Post – WaPo Bot
- Sephora – Kik uygulaması
- Lydia – Slack bot
- Microsoft – Xiaoice bot
- Baidu – Medical Bot

The emergence and implementation of the ChatBot idea begins with a theory of Alan Turing. The Turing test includes a user, a computer and a person which responding to this user. The computer and people answer the questions the user asks. If the user can not figure out which one is a real person, then the computer is said to have passed the test.

In the Turing Test, there are three terminals. Two of these terminals are controlled by person, one of is controlled by a computer. Each of these terminals are seperated from each other. One person is determined as the questioner, other person and computer are determined as the responder. The questioner ask the question to responders both of computer and person, about a specific topic and certain time like 5 minutes. After the specified time, the questioner tries to decide which responder is controlled by the person responder, and which responder is controlled by the computer. To simply specify;

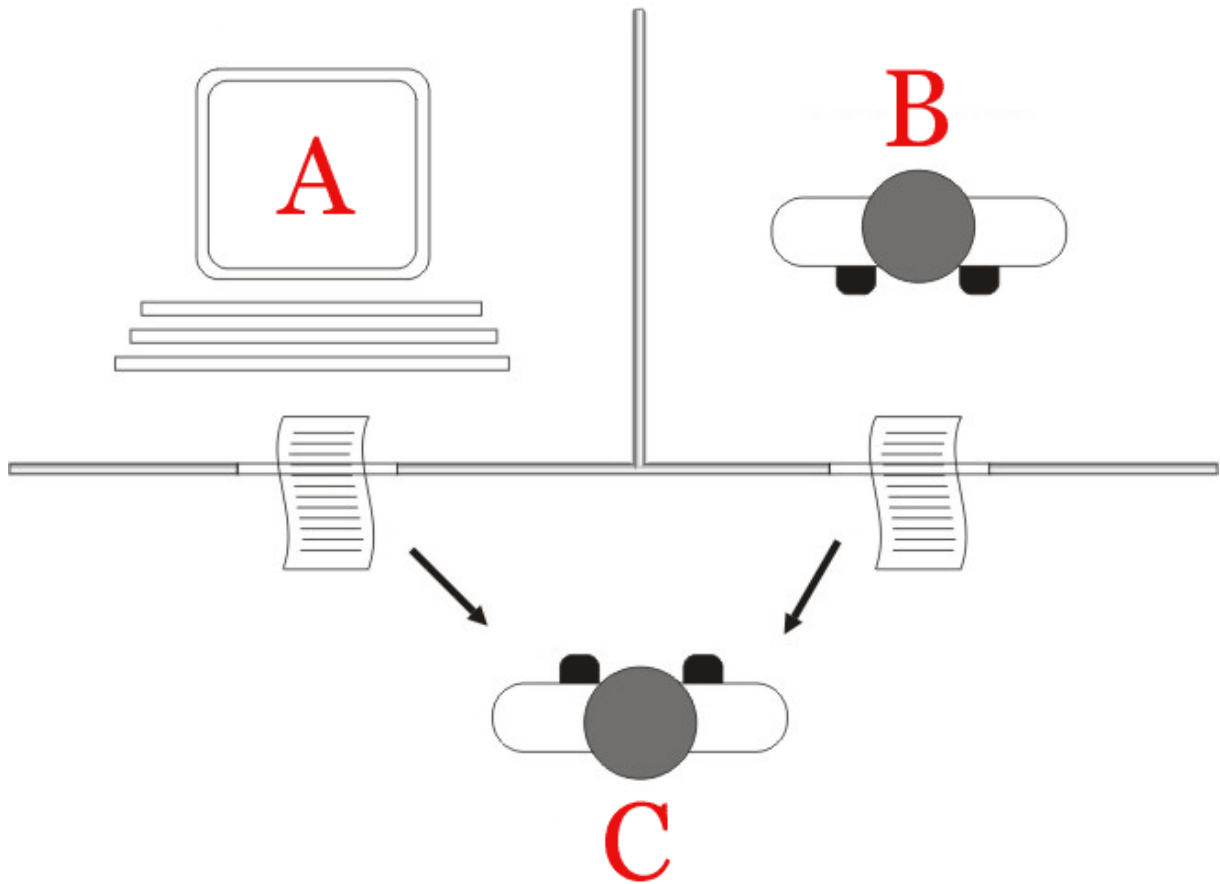


Figure 1.1: Turing Machine

We are trying to create a chatbot adhering to Turing test, but there is a specific difference between our chatbot and Turing Test. This difference is that in our case user knows that it talks to the computer. There are 3 major change in our chatbot, these are;

- 4- It supports Turkish Language
- 5- It follows a specific workflows.

- 6- It is smart. It means it uses Artificial Intelligence. Gets the question from user, edits the error if exists and response to user according to worklows.



2. ARTIFICIAL INTELLIGENCE MARKUP LANGUAGE

2.1 DEFINITION

Artificial Intelligence Mark-up Language, OR AIML is a programming language that is used in Artificial Intelligence and is structurally similar to XML. This language was developed between 1995-2000 years by Artificial Linguistic Internet Computer Entity (A.L.I.C.E).

AIML consists of categories. Each category contains question patterns and answers to those question patterns. Questions and answers can be in static or case sensitive or dynamic. Thanks to this, user can be given natural answers. (WALLACE, The elements of AIML style. Alice AI Foundation, 2003) (FONTE, 2009)

2.2 MOST IMPORTANT TAGS

The most important units of AIML are; (WALLACE, The elements of AIML style. Alice AI Foundation, 2003)

- 1- <aiml>: shows that aiml start and finish.
- 2- <category>: specifies categories of aiml. It contains questions and answers inside. This tag must contain pattern and template.
- 3- <pattern>: defines the pattern to match what a user may input.
- 4- <template>: contains responses that user asks. It can be found in category tag and can be more than one.
- 5- <star>: Used to match * characters in pattern tag.
- 6- <srai> allows to define different responses in same template.
- 7- <random>: gets the random responses. It uses with li tag, and gets the random li tag.
- 8- <set> and <get>: It is used for the same purpose in .Net platform. Set tag is used for setting a value in a variable in AIML, and Get tag is used for getting that value in variable.
- 9- <topic> Tag is used in AIML to store a context so that later conversation can be done based on that context.
- 10- <think> Set value without notifying the user. <set> will Display name, but to override display we use <think>.
- 11- <condition> Works like IF in .Net. It contains a condition and give responses according to conditions.

Example of AIML

```
<?xml version="1.0" encoding="ISO-8859-1"?>

<aiml>

<category>

<pattern>Kredikarti</pattern>

<template>

<random>

<li> Kart Ekleme </li>

<li> Son Kullanma Tarihi Guncelleme </li>

<li> Kart Limitini Ogrenme </li>

</random>

</template>

</category>

<category>

  <pattern> ALBERT EINSTEIN kimdir</pattern>

  <template>Albert Einstein Alman bir fizikçidir</template>

</category>

<category>

  <pattern> Isaac NEWTON kimdir</pattern>

  <template>Isaac Newton İngiliz Matematikçi ve Fizikçidir</template>

</category>
```


<category>

<pattern>* kimdir biliyor musun?</pattern>

<template>

<srai> <star/> kim</srai>

</template>

</category>

</aiml>



3. ZEMBEREK NLP LIBRARY

Before analyzing data, we separated the words from the sentences. After separating the words, to deduce the words, we had to find the roots of the words. To find to root of the words we use Zemberek NLP Library.

3.1 DEFINITION

Zemberek is an open source Turkish Natural language processing library and OpenOffice, LibreOffice plugin. The first version is distributed under BSD license. The entirely Java-developed library has features such as spell checking, suggestions for incorrect words, spelling and erroneous code clearing. In the second version with the code Zemberek2, MPL license was passed, and architectural changes were made in order to create a DDI infrastructure for all Turkish languages in general. A server written using Zemberek provides general spell checking support for Pardus. The server communicates with other applications via TCP-IP sockets via a simple ISpell-like protocol. In the new version, the DBUS interface is also added to the server.

The NZemberek project has been launched to create a .net version of the Zemberek library.

The Zemberek library and the LibreOffice plugin are platform independent as they are written in Java.

3.2 ADDING ZEMBEREK TO .NET

Zemberek NLP Library based on Java, but there is a Nuget Package for .Net. Thanks to that, we can use Zemberek in .Net.

To adding Zemberek into .Net, follow these steps;

- 1- Open the project in Visual Studio and follow these path: Tools -> Nuget Package Manager -> Manage Nuget Packages For Solution.

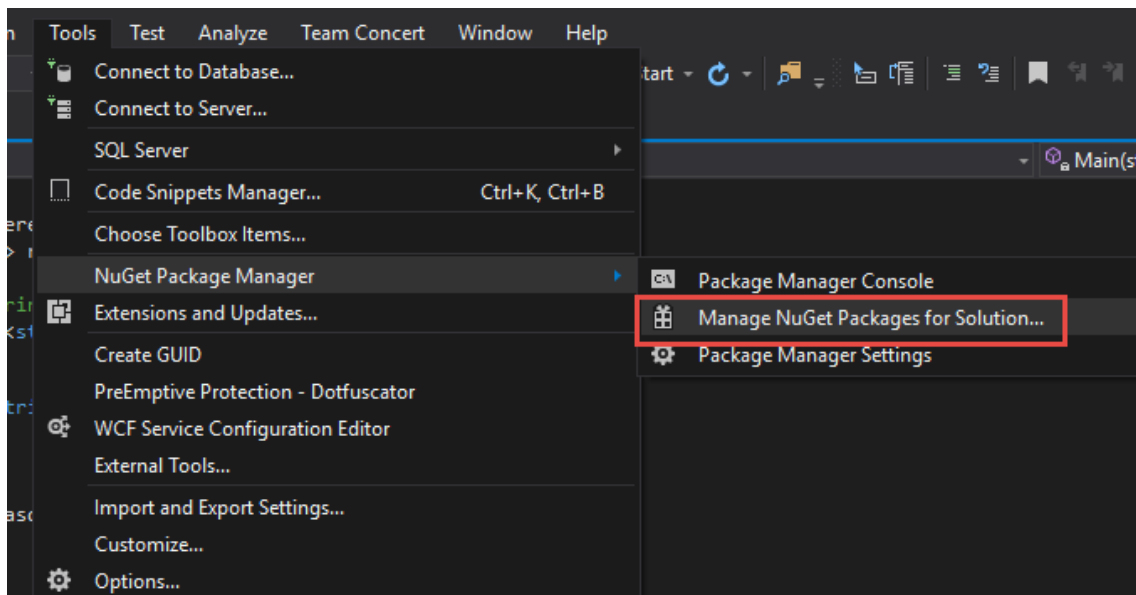


Figure 3.1: Nuget Package

- 2- After clicking Manage NuGet Packages For Solution, the below screen will open. In the pop-up screen, write and search ZEMBEREK, select the project that you use and click Install button. After that, Zemberek Library will download and become ready to use.

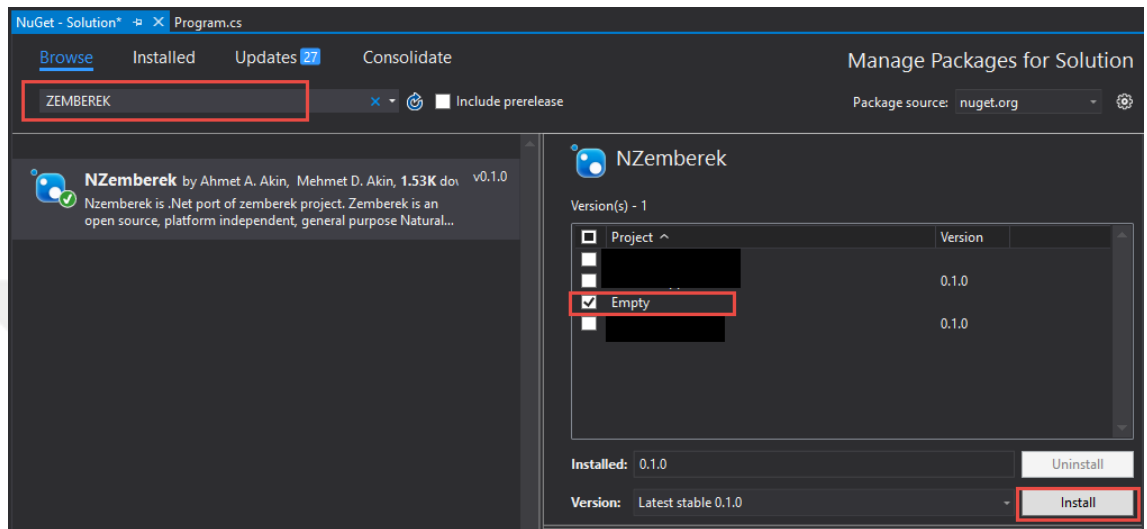


Figure 3.2: Add Zemberek to .Net

3.3 HOW TO USE ZEMBEREK TO .NET

To find root of a word, we should call kelimeCozumle method in Zemberek. There is a simple function about it.

```
public static string FindWordRoot(string word)
{
    try
    {
        Kelime[] kelimeler = zemberek.kelimeCozumle(word);

        if (kelimeler.FirstOrDefault() == null
            || kelimeler.FirstOrDefault().kok() == null
            || kelimeler.FirstOrDefault().kok().icerik() == null)
            return word;
        var asdasdasd = kelimeler[0].icerikStr();

        return kelimeler.FirstOrDefault().kok().icerik();
    }
    catch (Exception e)
    {

        throw new Exception("FindWordRoot: " + e.Message);
    }
}
```

kelimeCozumle methods gets the root, word type and word appendix. We just use word root in this part.

4. DATA MINING

4.1 Understanding Data Mining

4.1.1 Definition

Data Mining is defined as getting large amount of data, searching of the relations and rules by using computer programs like R, Weka etc, and let us to predict the future about this data. “Over 30 years old, well-paid computer engineers have more than one laptop in their homes” is an example about Data Mining. According to this information, a computer company can aim to increase its sales by making a laptop campaign over the age of 30.

4.1.2 Data Mining Techniques

There are two learning types about Data Mining;

- 1- Supervised Learning: In supervised learning, there are inputs and outputs that defined before. In this case, it is certain that one input has a specific output and the system is expected to examine the data accordingly. Because of all data labelled, this is called Classification methods.
- 2- Unsupervised Learning: In unsupervised learning, only inputs exist. The system is expected to find the output according to the given inputs. Because of data are not labelled and outputs is predicted, this is called Clustering methods.

Main difference between Supervised and Unsupervised are;

- 1- In Supervised Method, all inputs are labelled. This means all outputs are known by the system. In Unsupervised Method, the outputs are not known, the system is expected to find common points and cluster the data
- 2- In Supervised Method, there is a training set that can be train with rest of data. But in Unsupervised Method, there is no training set, because outputs are not known.

Most common methods are:

- Supervised:
 - a- Decision Tree
 - b- Naïve Bayes Classification
 - c- Neural Network
- Unsupervised
 - a- K Means Clustering
 - b- Hierarchical Clustering

4.2 WEKA

We used WEKA program to do data analysis.

4.2.1 Definition

Weka is a program used to analyze data. This program is developed in Java language and is open source code. Weka can be used by installing the application directly on the computer or by calling it from Java code.

Weka is licensed under the GNU General Public License. This means that it is a open source code and everyone can add a libabry on it.

Because of developing in Waikato University and it is an open source code, is called Waikado Environment for Knowledge Analysis shortly WEKA. (HALL, 2009) (HOLMES, 1994)

4.2.2 User Interface

User Interface of Weka is below. In this screen, there are several tabs on it;

- **Explorer:** The part in which the data is retrieved and examined.
- **Experimenter:** The area where statistical tests are performed.
- **KnowledgeFlow:** Machine learning experiments are created to run the domain.
- **Simple CLI:** The part where Weka commands are run directly

4.2.3 How to Analyze Data in Weka

In order to analyze a data in Weka, the file format must be arff. Long name of ARFF is Attribute Relationship File Format. This means arff file contains data and its attributes in this file.

There are 2 types in Weka;

- **Instance:** A row of data is called an instance, as in an instance or observation from the problem domain.
- **Attribute:** A column of data is called a feature or attribute, as in feature of the observation. There are 4 attributes types. These are;
 - **Real:** All numeric values like “1”, “2.5”
 - **Integer:** Numeric values without a fractional part like “2”, “7”
 - **String:** List of words. Same version on .Net Strings.
 - **Nominal:** Categorical data like “Kredikarti”, “Odeme”

4.2.4 Data in Weka

Weka accepts data in the ARFF format. It is an extension of the CSV file format where a header is used that provides metadata about the data types in the columns.

There are 3 important tag in arff:

- 1- **Relation:** Contains dataset name
- 2- **Attribute:** Contains the attribute which ,s used in dataset
- 3- **Data:** Contains the values of dataset

Here is a small part of arff file in our dataset.

@relation data

@attribute kredi {0,1}

@attribute kart {0,1}

@attribute hesap {0,1}

@attribute odeme {0,1}

@attribute teslimat {0,1}

@attribute calinti {0,1}

@attribute ekle {0,1}

@attribute spor {0,1}

@attribute uyelik {0,1}

@ATTRIBUTE class {kredikarti,hesap,teslimat,odeme,fraud,dummy}

@data

1,1,0,0,0,0,1,0,0,kredikarti

0,0,1,0,0,0,1,0,0,hesap

0,0,0,0,1,0,0,0,0,teslimat

0,0,0,1,0,0,1,0,0,odeme

1,1,0,0,0,1,0,0,0,fraud

0,0,0,0,0,0,0,1,1,dummy

4.2.5 Load CSV Files in the ARFF

You must follow the steps below to upload the file to Weka;

1- Start the Weka. The following screen will open when you open Weka

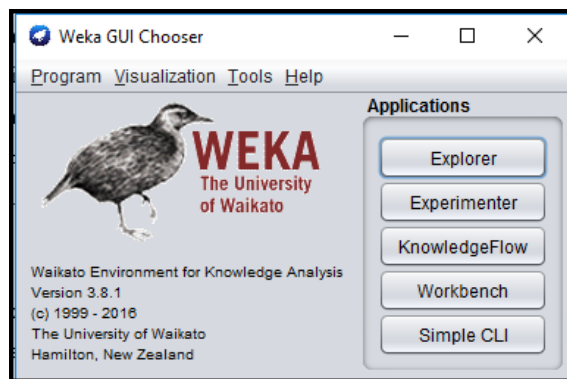


Figure 4.1: Weka UI

2- Click Explorer button, the below screen will open. In this screen, click Open File button and select the arff file.

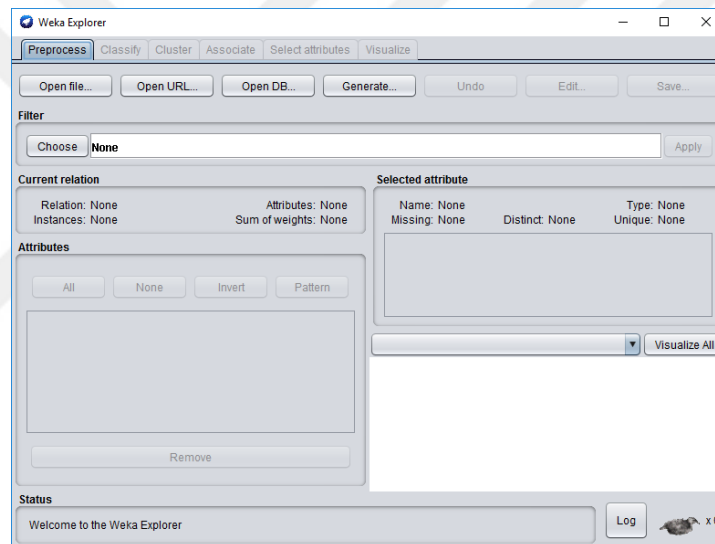


Figure 4.2: Weka Explorer

- 3- After selecting the arff file, Weka screen will be below. In this screen, we can see the Instances, attributes and its distributions.

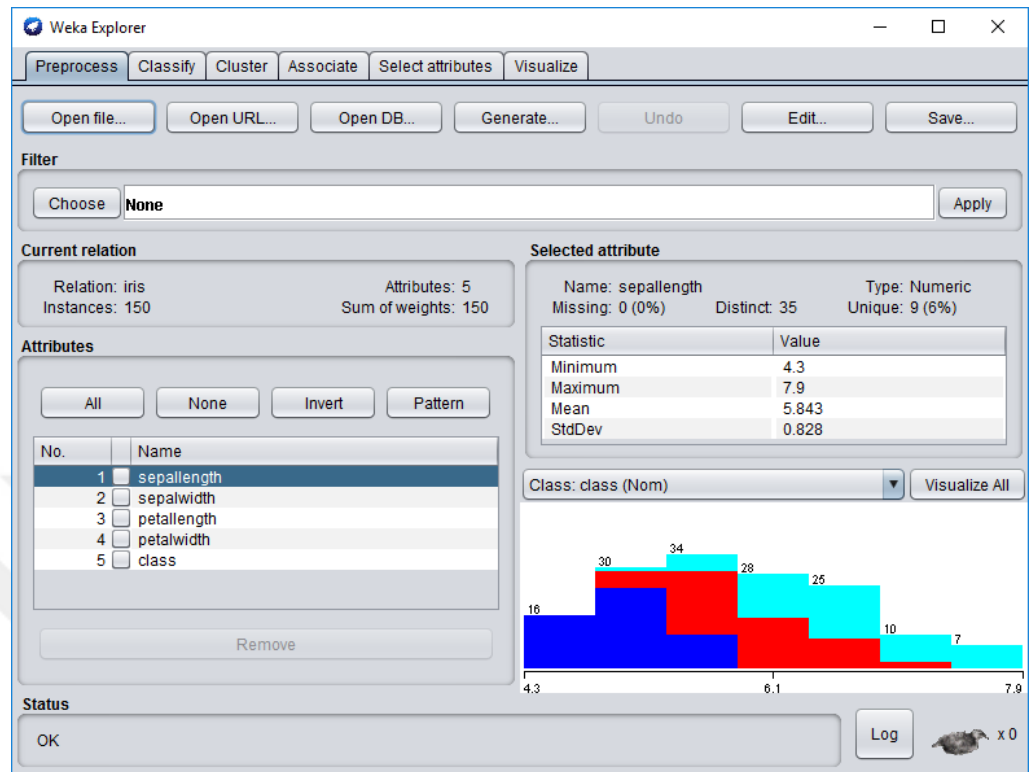


Figure 4.3: Weka Explorer after Data Import

There are several Data Mining methods in Weka. We can choose these methods under the Classify and Cluster tabs.

4.3 Data Mining Methods, The Example Data Set and Results

Our data has 6 categories. These are KrediKartı, Hesap, Teslimat, Odeme, Fraud and Dummy. We have 50 sentences for each category, totally 300 sentences. Before analyzing the data, we split the sentences into words, write each word in separate columns and score the sentences by containing the word. If a sentence contains a word, we write 1(one) under the word. If does not contain a word, we write 0(zero) under the word. This part was done in a project that works with .net code, but we will explain later. After all sentences is analyzis, there comes out a matrix like below;

Table 4.1: Sentences/Words List

Sentences/Words	Word 1	Word 2	Word 3	Word 4	Word 5
Sentence 1	1	1	0	0	1
Sentence 2	0	1	1	0	0
Sentence 3	0	0	0	1	1
Sentence 4	1	0	0	0	1
Sentence 5	0	0	1	1	1

While analyzing these 300 sentences, we choose 3 different Data Mining methods. These are;

- 1- Naive Bayes Classification
- 2- Decision Tree
- 3- Neural Network

4.3.1 Naive Bayes Classification

- Definition

Naive Bayes Classification was invented by Thomas Bayes who is an English Mathematician lived on 17. Century. In this classification, sample data has to be and their category has to be known. When guessing the category of new sample, probability of all categories are calculated. It is guessed that the data belongs to the category with the greatest probability. (KOHAVI, 1996) (KELLER, 2002) (KEOGH, 2006) (LEUNG, 2007)

The formula of Naive Bayes is below;

$$P(X|Y) = (P(Y|X) \cdot P(X)) / P(Y)$$

P(X): Probability of X

P(Y): Probability of Y

P(Y|X): Probability of Y, If X occurs

P(X|Y): Probability of X, If Y occurs

- The Algorithm

Naive Bayes algorithm consists of 4 steps:

- 1- Define the sample list
- 2- Find the probability of all categories.
- 3- Calculate the probability that the data belongs to each category
- 4- Find the category which is the biggest probability and guess that the data belongs to that category

- **Results**

13.3 percent data was incorrectly clustered in Naive Bayes Method. Details of results are below.

Table 4.2: Naive Bayes Summary

Correctly Classified Instances	260	86.6667 %
Incorrectly Classified Instances	40	13.3333 %
Kappa statistic	0.84	
Mean absolute error	0.0619	
Root mean squared error	0.1739	
Relative absolute error	22.2787 %	
Root relative squared error	46.6733 %	
Total Number of Instances	300	

Table 4.3: Naive Bayes Detailed Accuracy By Class

	TP Rate	FP Rate	Precision	Recall	F- Measure	MCC	ROC Area	PRC Area	Class
	0,88	0,048	0,786	0,88	0,83	0,796	0,984	0,944	kredikarti
	0,96	0,032	0,857	0,96	0,906	0,888	0,996	0,988	hesap
	1	0,016	0,926	1	0,962	0,955	0,999	0,997	teslimat
	1	0,008	0,962	1	0,98	0,977	1	1	odeme
	0,96	0,056	0,774	0,96	0,857	0,832	0,994	0,973	fraud
	0,4	0	1	0,4	0,571	0,598	0,99	0,956	dummy
Weighted Avg.	0,867	0,027	0,884	0,867	0,851	0,841	0,994	0,976	

Table 4.4: Naive Bayes Confusion Matrix

A	b	c	d	e	f	<-- classified as
44	0	0	1	5	0	a = kredikarti
0	48	2	0	0	0	b = hesap
0	0	50	0	0	0	c = teslimat
0	0	0	50	0	0	d = odeme
2	0	0	0	48	0	e = fraud
10	8	2	1	9	20	f = dummy

4.3.2 Decision Tree

- Definition

Decision Trees are a type of Supervised Machine Learning, where data is constantly divided according to a certain parameter. It is a frequently used method for classification problems. It is easy to understand. Classification accuracy is more effective than other learning methods. Decision Tree seems to flow diagram. A Decision Tree has 3 important parts. These are; (Srivastava, 1999) (SAFAVIAN & LANDGREBE, 1991)

- 1- **Root Node:** Test for the value of a specific attribute
- 2- **Edges:** Corresponds to the result of a test.
- 3- **Leaves:** Terminal nodes that predict the outcome

Nodes can be numeric or categoric data. If a node has numeric data, it should be an equation like $\text{age} = 30$ or $\text{salary} > 10000$. If terminal node is categoric data, it should be answer like Yes/No.

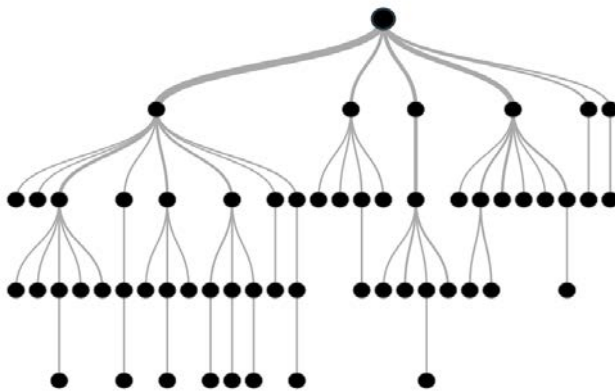


Figure 4.4: Decision Tree

The highest probability data is selected as Root Node. According to root node, edges and leaves is selected. For defining the Root Node we should calculate system entropy and entropies of each classes.

Let D is a class, p is probability of this class and n is a decision types of this class. Then entropy of this class is;

$$\text{Entropy}(D) : \text{Info}(D) = -\sum_{i=1}^n p_i * \log_2 p_i$$

Let n=2 and $p_1=5/9$ and $p_2=4/9$,

$$\text{Then Entropy}(D) = \text{Info}(D) = -(5/9 * \log_2 5/9 + 4/9 * \log_2 4/9)$$

If we calculate an entropy according to a class, we calculate like this;

Let D, E a class of dataset. Then entropy of E according to class D;

$$\text{Info}_E(D) = \sum_{c \in X} P(c) * E(c)$$

Information Gain = $\text{Info}(D) - \text{Info}_E(D)$ for each class.

An attribute is selected as root node which Information Gain is higher. (FREUND, 1999)

- **Advantages**

1. It is easy to understand even for people from non-analytical background. It does not necessary to know any statistical knowledge to read and interpret them.
2. It is the fastest way to define datas and their relations.
3. Do not need to clean data in Decision Trees.
4. Numeric and categoric data can be analyzis at the same time

- **Disadvantages**

1. **Over fitting:** Over fitting is one of the most practical difficulty for decision tree models. This problem gets solved by setting constraints on model parameters and pruning (discussed in detailed below).
2. **Not fit for continuous variables:** While working with continuous numerical variables, decision tree loses information when it categorizes variables in different categories.

- **Results**

17.3 percent data was incorrectly clustered in Decision Tree.

Table 4.5: Decision Tree Summary

Correctly Classified Instances	248	82.6667 %
Incorrectly Classified Instances	52	17.3333 %
Kappa statistic	0.792	
Mean absolute error	0.068	
Root mean squared error	0.2147	
Relative absolute error	24.4647 %	
Root relative squared error	57.6025 %	
Total Number of Instances	300	

Table 4.6: Decision Tree Detailed Accuracy By Class

	TP Rate	FP Rate	Precision	Recall	F-Measure	MCC	ROC Area	PRC Area	Class
	0,8	0,068	0,702	0,8	0,748	0,695	0,956	0,797	kredikarti
	0,98	0,016	0,925	0,98	0,951	0,942	0,987	0,974	hesap
	1	0,016	0,926	1	0,962	0,955	0,991	0,916	teslimat
	0,82	0	1	0,82	0,901	0,89	0,973	0,944	odeme
	0,9	0,092	0,662	0,9	0,763	0,719	0,954	0,734	fraud
	0,46	0,016	0,852	0,46	0,597	0,578	0,853	0,692	dummy
Weighted Avg.	0,827	0,035	0,844	0,827	0,82	0,796	0,952	0,843	

Table 4.7: Decision Tree Confusion Matrix

a	b	c	d	e	f	<-- classified as
40	0	0	0	8	2	a = kredikarti
0	49	0	0	1	0	b = hesap
0	0	50	0	0	0	c = teslimat
4	0	1	41	2	2	d = odeme
5	0	0	0	45	0	e = fraud
8	4	3	0	12	23	f = dummy

4.3.3 Neural Network

- Definition

Neural networks are structures that can be processed at the same time that many processing units come together. They are structurally similar to nerve cells (neurons). Because of the many units it contains and the connections between these units, it acts like a human brain, that is, it can behave like learning, remembering, and classifying. (HECHT-NIELSEN, 1988) (SANGER, 1989) (ODOM & SHARDA, 1990) (LAWRENCE, 1997) (KAWATO, FURUKAWA, & SUZUKI, 1987)

- Architecture of Artificial Neural Networks

There are 3 basic elements in a neural network. These are; (MARTINETZ, 1991)

- **Input layer:** Dataset that is used in beginning of a program . This dataset contains data that is intent to be trained.
- **Output layer:** Specifies the classes that belongs to input layers.
- **Hidden layer:** These neurons are in between input and output layers. Gets the information from input layers and tries to find the correct outputs. There can be many number of hidden layer, but at least one hidden layer has to exist. If there exists only one hidden layer, there must be X neurons in this layer where

$$X = (\text{number of attributes} + \text{number of classes})/2$$

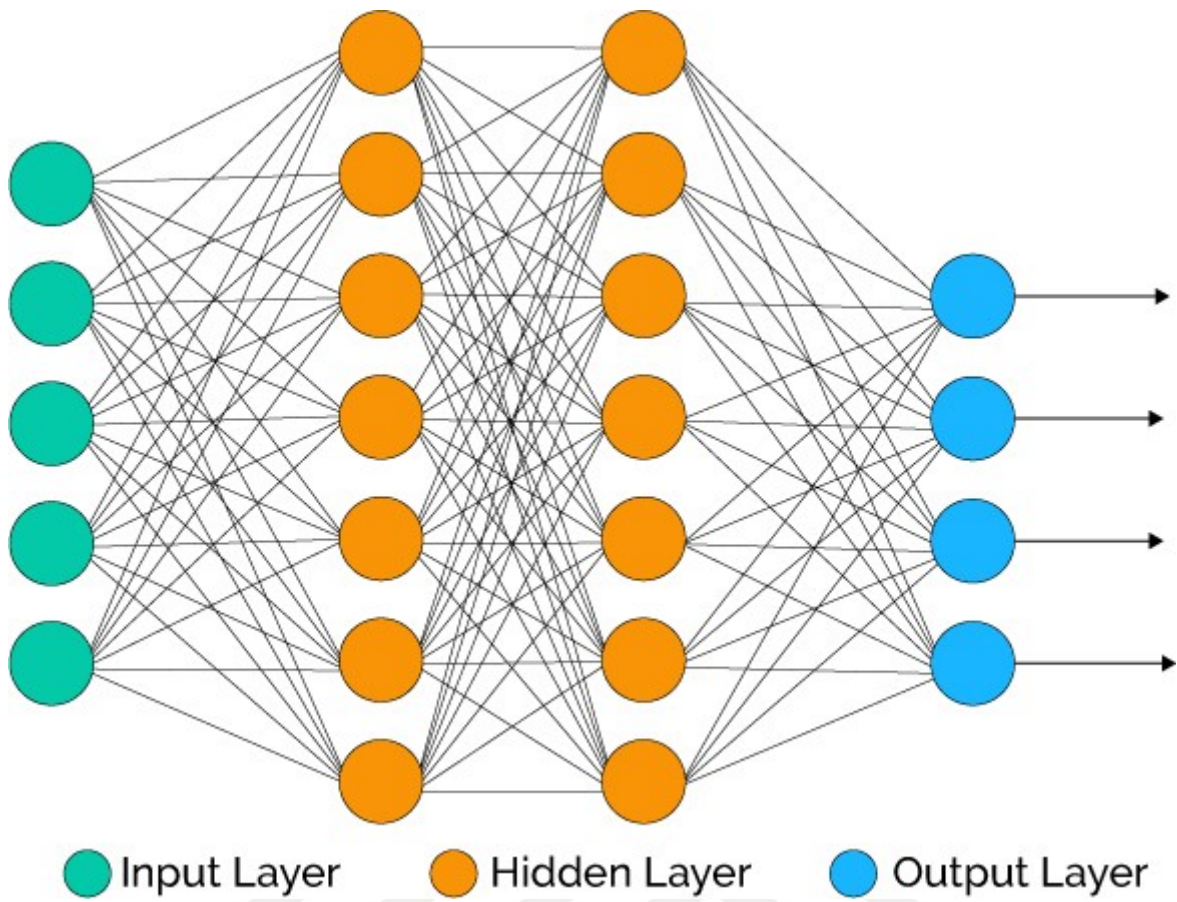


Figure 4.5: Neural Network Types

- **Popular Neural Network Architectures**

Here is some Neural Network types;

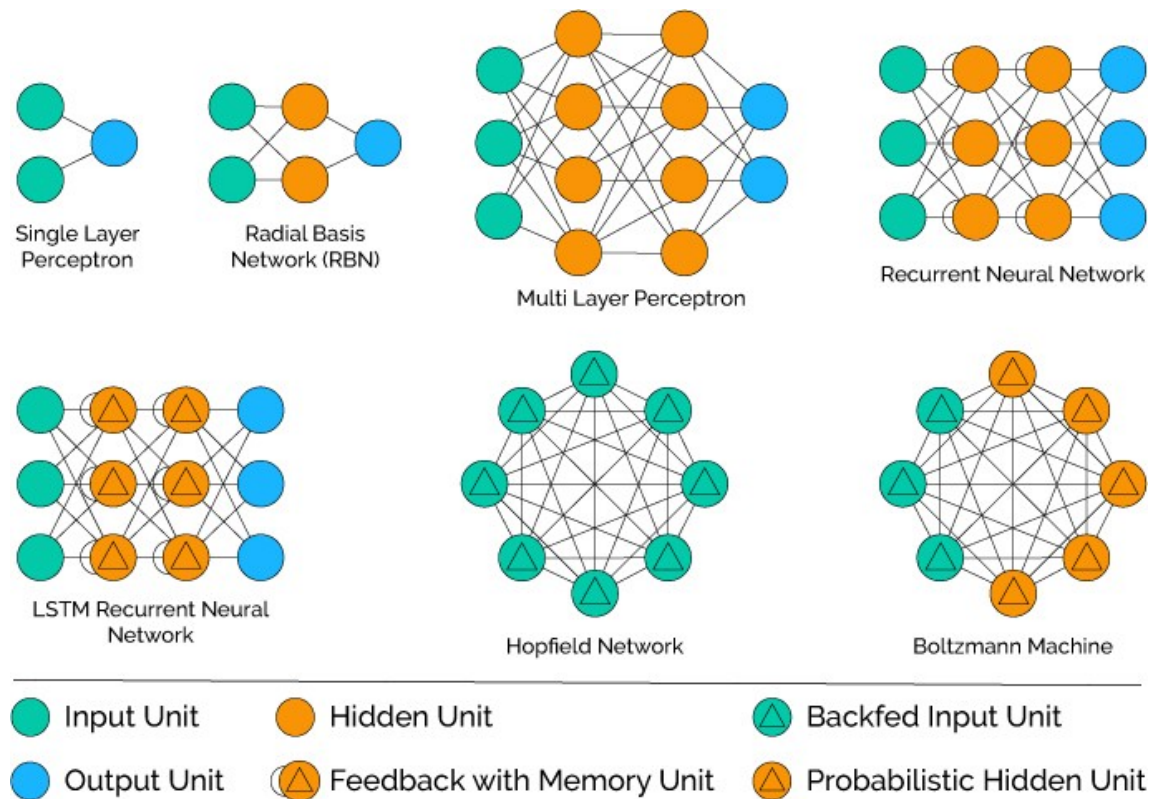


Figure 4.6: Neural Network Architectures

- **Keys For Neural Network**

There are some important keys to understand neural network

- **Neuron:** Main item of neural network that contains Synapse, Bias and Gradient
- **Synapse:** Part of Neurons that contains connections about inputs, outputs and their weights.
- **Weight:** Individual weights represent the strength of connections between units. If the weight from unit X to unit Y has greater magnitude, it means that X has greater influence over Y
- **Bias:** Value that allows you to shift the activation function to the left or right
- **Sigmoid Function:** This function is used for the activation of neurons in an artificial neural network. This function can take a result value between 0 and 1.

Formula of Sigmoid is;

$$f(x) = 1 / (1 + e^{-x})$$

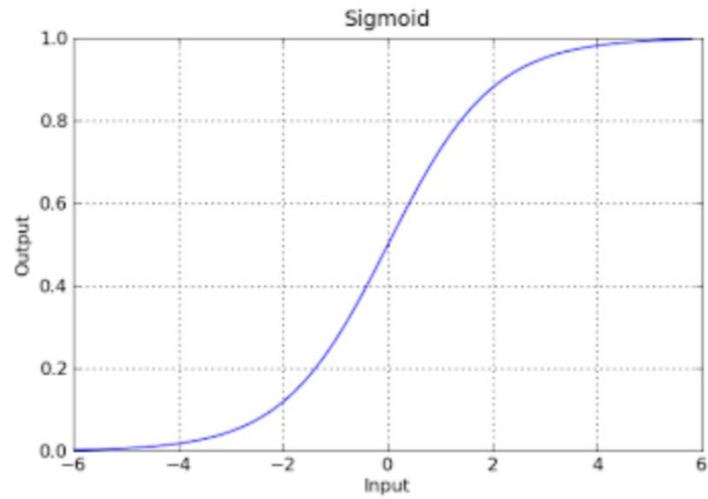


Figure 4.7: Sigmoid Function Values

Results

7.3 percent data was incorrectly classified in neural network

Table 4.8: Neural Network Summary

Correctly Classified Instances	278	92.6667 %
Incorrectly Classified Instances	22	7.3333 %
Kappa statistic	0.912	
Mean absolute error	0.0302	
Root mean squared error	0.1452	
Relative absolute error	10.8814 %	
Root relative squared error	38.9541 %	
Total Number of Instances	300	

Table 4.9: Neural Network Detailed Accuracy By Class

	TP Rate	FP Rate	Precision	Recall	F-Measure	MCC	ROC Area	PRC Area	Class
	0,88	0,044	0,8	0,88	0,838	0,805	0,973	0,914	kredikarti
	0,98	0,004	0,98	0,98	0,98	0,976	1	0,999	hesap
	0,98	0	1	0,98	0,99	0,988	1	0,999	teslimat
	0,98	0,008	0,961	0,98	0,97	0,964	0,999	0,996	odeme
	0,94	0,028	0,87	0,94	0,904	0,885	0,993	0,965	fraud
	0,8	0,004	0,976	0,8	0,879	0,864	0,991	0,965	dummy
Weighted Avg.	0,927	0,015	0,931	0,927	0,927	0,914	0,993	0,973	

Table 4.10: Neural Network Confusion Matrix

a	b	c	d	e	f	<-- classified as
44	0	0	1	5	0	a = kredikarti
1	49	0	0	0	0	b = hesap
0	0	49	0	0	1	c = teslimat
1	0	0	49	0	0	d = odeme
3	0	0	0	47	0	e = fraud
6	1	0	1	2	40	f = dummy

5. RUNNING THE PROGRAM

This solution contains 3 projects;

- 1- Neural Network Library Project
- 2- Neural Network Console Application Project
- 3- Smart Chatbot Web Project

5.1 NEURAL NETWORK LIBRARY PROJECT

This project contains the definitions of neural network. There are 3 sections;

- 1- Models
- 2- Helpers
- 3- Controllers

5.1.1 Models

Definitions of all neural network items. These are;

- **DataSet:** Contains Values and its targets
- **Network:** Contains the main Neural Network items. These are input layer, hidden alyer and output layer.
- **Neuron:** Contains Synapse, Bias, Gradient and unique id of Neuron
- **Sigmoid:** Contains the calculation of Sigmoid function
- **Synapse:** Contains input neuron, output neuron and their weights.

5.1.2 Helpers

Contains the helper class of Network, Neuron and Synapse.

5.1.3 Controllers

There are 2 controller types;

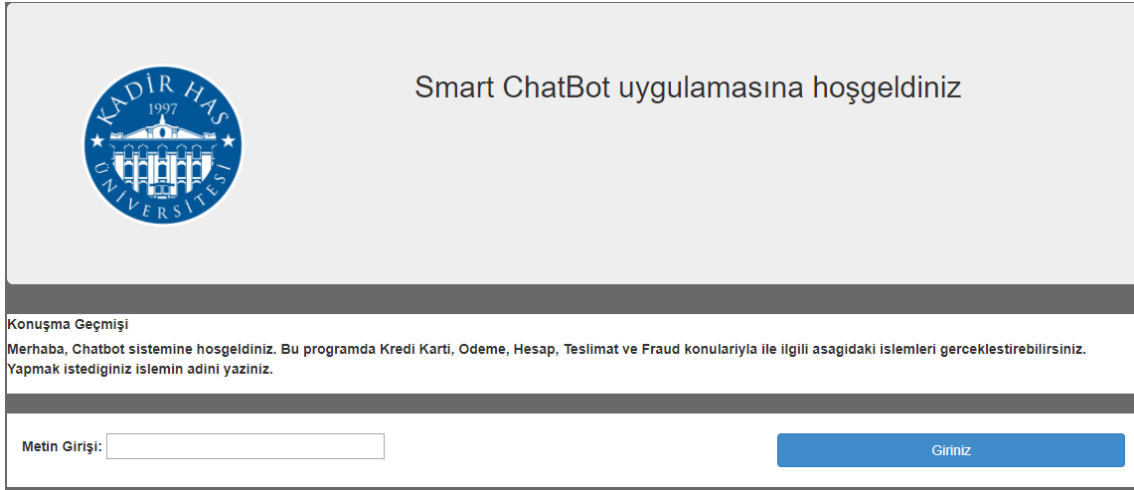
- 1- ExportController: Helps to export a simple network to solution. It creates the “network.dat” file under bin folder.
- 2- ImportController: Helps to import a simple network to solution by reading the “network.dat” document which is located in bin folder.

5.2 NEURAL NETWORK CONSOLE APPLICATION PROJECT

This is a console application that works about training data or testing the system by new inputs. This calls the Controller classes which is under Neural Network Library Project

5.3 SMARTCHATBOT WEB PROJECT

This is Chatbot's interface. This project is based on ASP MVC, AIML, Chatbot, Zemberek and main Asp Library are used in this project. Start page of project is below. "Konuşma Geçmişi" part contains all history of conversations. In this part, first sentence comes from AIML tags. When a user enters a text to "Metin Girişi" and clicks the "Giriniz" button, system gets the text, splits into words, finds these words roots and trains according to Network that we create before.



KADIR HAS
1997
UNİVERSİTESİ

Smart ChatBot uygulamasına hoşgeldiniz

Konuşma Geçmişi
Merhaba, Chatbot sistemine hoşgeldiniz. Bu programda Kredi Kartı, Odeme, Hesap, Teslimat ve Fraud konularıyla ilgili asagidaki işlemleri gerçekleştirebilirsiniz. Y yapmak istediğiniz işlemin adını yazınız.

Metin Girişi:

Giriniz

Figure 5.1: Smart Chatbot Main Page

5.3.1 Chatbot Categories

There are 6 categories in chatbot. These are

- 1- Kredi Kartı
- 2- Ödeme
- 3- Hesap
- 4- Teslimat
- 5- Fraud
- 6- Dummy

First five categories can be processed categories in chatbot. “Dummy” category contains the sentences which does not belongs to first five categories. We will explain the chatbot system in Kredi Kartı category and show screenshot about other topics in other categories

- 1- Kredi Kartı

If user writes a sentence like “kredi kartımın son kullanma tarihini güncellemek istiyorum”, the following screen appears. System keeps the information about conversation and gets the Partial View that users demand. After that user should fill the new page that inside the conversation page.

KADIR HAS
1997
UNİVERSİTESİ

Smart ChatBot uygulamasına hoşgeldiniz

Konuşma Geçmişi
Merhaba, Chatbot sistemine hoşgeldiniz. Bu programda Kredi Kartı, Ödeme, Hesap, Teslimat ve Fraud konularıyla ilgili aşağıdaki işlemleri gerçekleştirebilirsiniz. Yapmak istediğiniz işlemin adını yazınız.
Kullanıcı: Kredi kartımın son kullanma tarihini güncellemek istiyorum

Kredi Kartı Numarasını Giriniz
Kredi Kartı Son Kullanma Tarihini Giriniz

Onayla

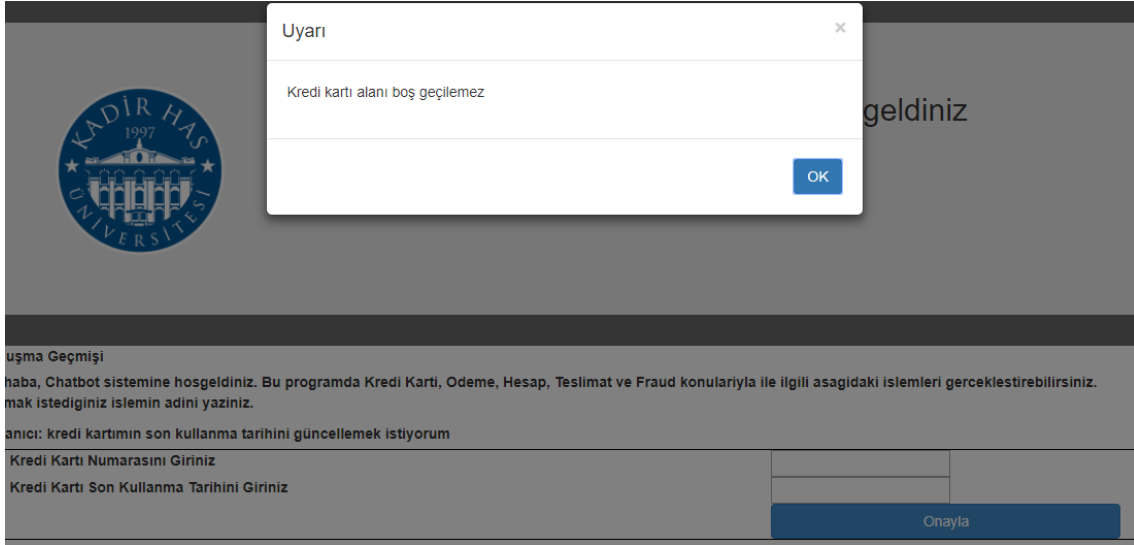
Metin Girişi: kredi kartımın son kullanma tarihini güncelle

Giriniz

Figure 5.2: Smart Chatbot Conversation Page

If user does not fill the textboxes, there comes out an warning page.

- If Credit Card Number is empty



Uyarı

Kredi kartı alanı boş geçilemez

OK

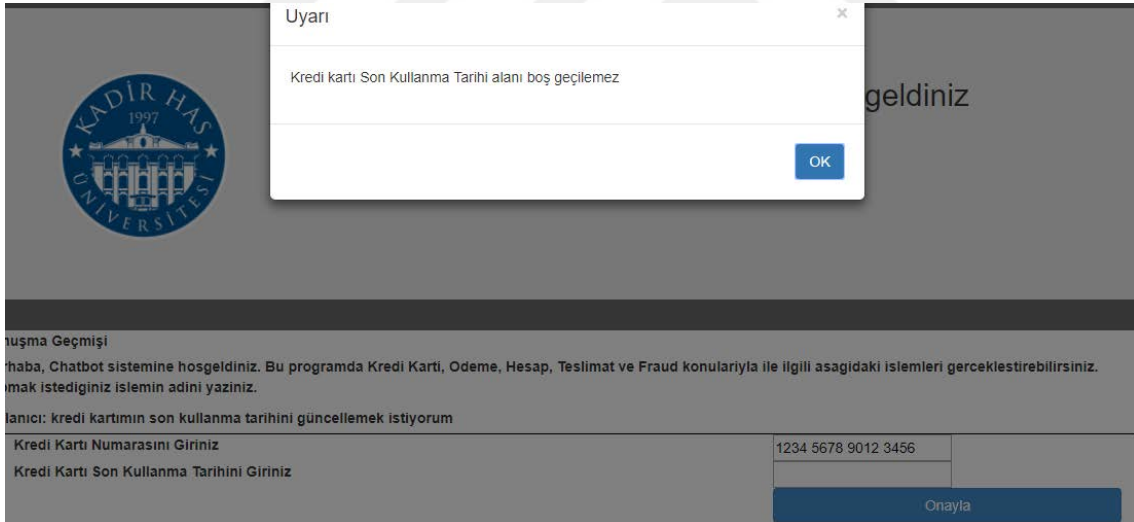
Kredi Kartı Numarasını Giriniz

Kredi Kartı Son Kullanma Tarihini Giriniz

Onayla

Figure 5.3: Smart Chatbot Error Page 1

- If Credit Card Expiration Date is empty



Uyarı

Kredi kartı Son Kullanma Tarihi alanı boş geçilemez

OK

Kredi Kartı Numarasını Giriniz

Kredi Kartı Son Kullanma Tarihini Giriniz

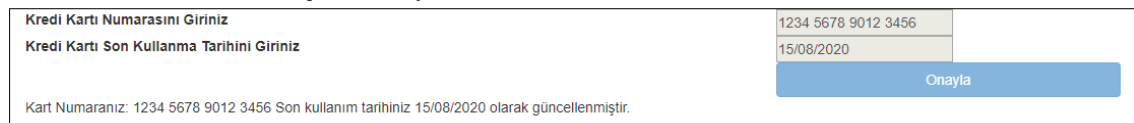
1234 5678 9012 3456

15/08/2020

Onayla

Figure 5.4: Smart Chatbot Error Page 2

If all textboxes are filled and there is no error, successful message is written under the page.



Kredi Kartı Numarasını Giriniz

Kredi Kartı Son Kullanma Tarihini Giriniz

1234 5678 9012 3456

15/08/2020

Onayla

Kart Numaranız: 1234 5678 9012 3456 Son kullanım tarihiniz 15/08/2020 olarak güncellenmiştir.

Figure 5.5: Smart Chatbot Success Page

2- Fraud

The credit card is canceled on this page. To cancel the Credit Card, Credit Card Number, Full Name of Credit Card Owner and Government Id should be known.

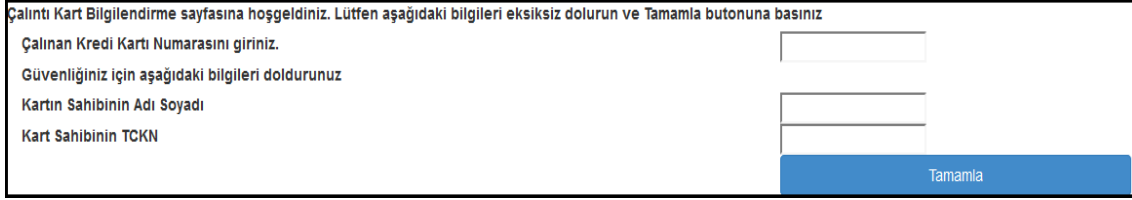


Figure 5.6: Fraud Category Input Page

If the credit card is successfully canceled, below screen will be come out.

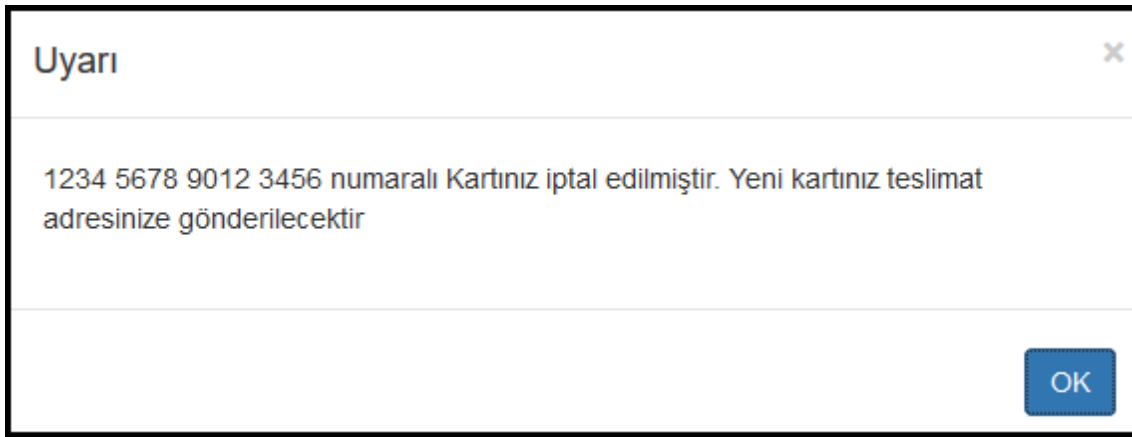


Figure 5.7: Fraud Category Success Page

3- Hesap

From this page you can create a new account. To create an account, Full name, Birthdate, Birth Place, Government Id and Father Name should be known.

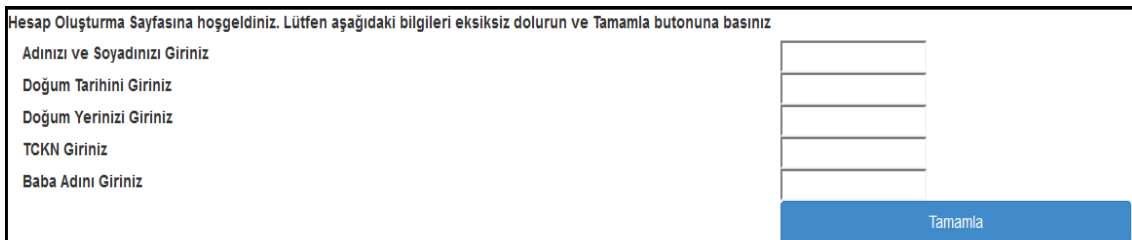


Figure 5.8: Hesap Category Input Page

If account is successfully created, below screen will be come out.



Figure 5.9: Fraud Category Success Page

4- Ödeme

From this page you can give payment order for 4 type of invoice. To give a payment order, Credit Card Number, Invoice Type, Institution Name and Subscriber Number should be known.

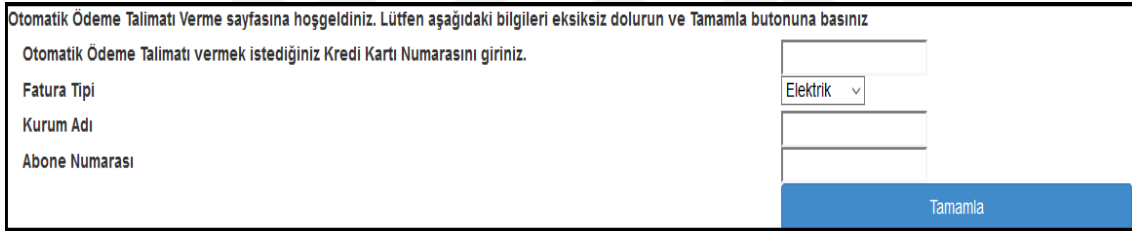


Figure 5.10: Ödeme Category Input Page

These are Invoice Types that can vbe given a payment order.

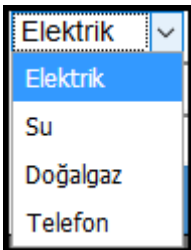


Figure 5.11: Ödeme Category Invoice Types

If a payment order is successfully given, below screen will be come out.

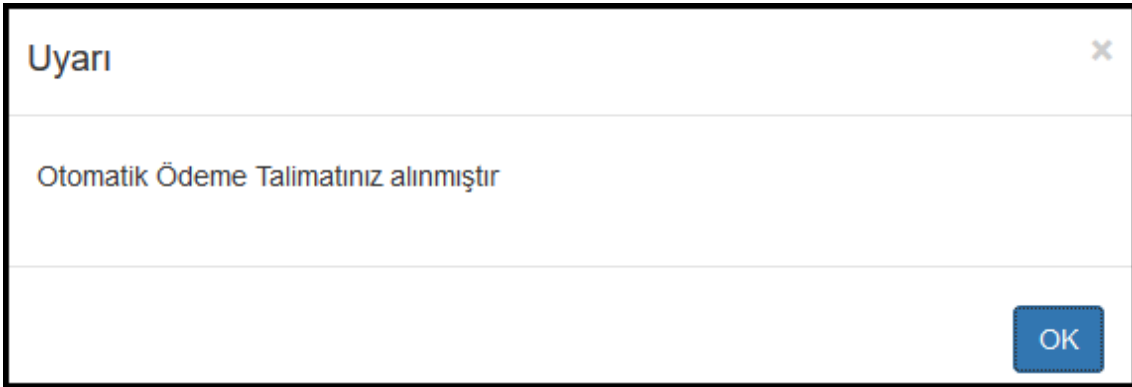


Figure 5.12: Ödeme Category Success Page

5- Teslimat

From this page you can give payment order for 4 type of invoice. To give a payment order, Credit Card Number, Invoice Type, Institution Name and Subscriber Number should be known.

Ekstre bilgileri Gönderim Adresini Değişirme sayfasına hoşgeldiniz. Lütfen aşağıdaki bilgileri eksiksiz doldurun ve Tamamla butonuna basınız

Ekstre adresini güncellemek istediğiniz Kredi Kartı Numarasını giriniz.

Adres Bilgileri

Ülke

İl

İlçe

Figure 5.13: Teslimat Category Input Page

If a payment order is successfully given, below screen will be come out.

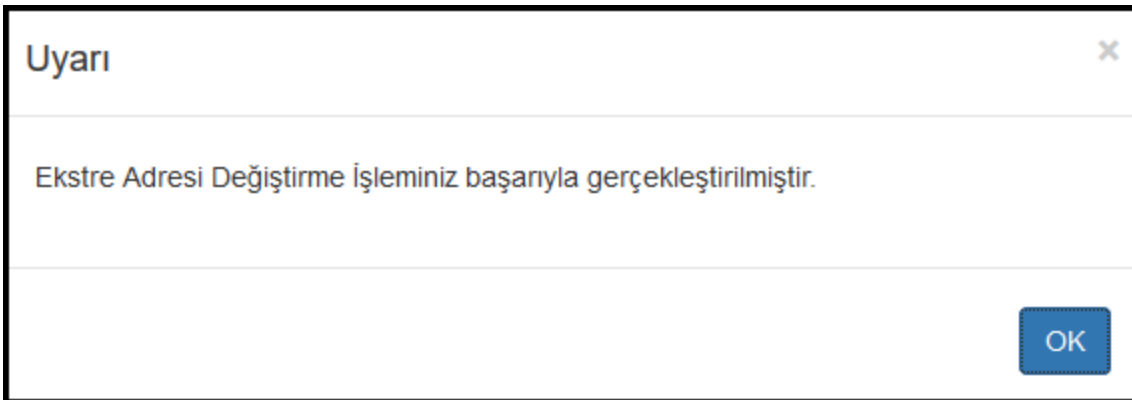


Figure 5.14: Teslimat Category Success Page

5.4 IMPORTANT FUNCTIONS

This project is a project where many different libraries are run together. For this reason, I explain some important functions so that the codes can be understood more easily.

5.4.1 Functions In Neural Network Library Project

5.4.1.1 GetHelperNetwork

This method gets the existing network, deserialize the json to .Net code and returns it. The network name is “network.dat” and under the bin folder

```
private static HelperNetwork GetHelperNetwork(string location = null)
{
    try
    {
        string filelocation = "network.dat";

        using (var file = File.OpenText(filelocation))
        {
            return JsonConvert.DeserializeObject<HelperNetwork>(file.ReadToEnd());
        }
    }
    catch (Exception)
    {
        return null;
    }
}
```

Figure 24: GetHelperNetwork

5.4.1.2 ImportNetwork

This method imports the network to system. It calls “GetHelperNetwork”, gets the value of all neurons and sets their values into network.

```
public static Network ImportNetwork()
{
    var dn = GetHelperNetwork();
    if (dn == null) return null;
}
```

```

var network = new Network();
var allNeurons = new List<Neuron>();

network.LearnRate = dn.LearnRate;
network.Momentum = dn.Momentum;

//Input Layer
foreach (var n in dn.InputLayer)
{
    var neuron = new Neuron
    {
        Id = n.Id,
        Bias = n.Bias,
        BiasDelta = n.BiasDelta,
        Gradient = n.Gradient,
        Value = n.Value
    };
    network.InputLayer.Add(neuron);
    allNeurons.Add(neuron);
}

//Hidden Layers
foreach (var layer in dn.HiddenLayers)
{
    var neurons = new List<Neuron>();
    foreach (var n in layer)
    {
        var neuron = new Neuron
        {
            Id = n.Id,
            Bias = n.Bias,
            BiasDelta = n.BiasDelta,
            Gradient = n.Gradient,

```

```

        Value = n.Value
    };

    neurons.Add(neuron);
    allNeurons.Add(neuron);
}

network.HiddenLayers.Add(neurons);
}

//Export Layer
foreach (var n in dn.OutputLayer)
{
    var neuron = new Neuron
    {
        Id = n.Id,
        Bias = n.Bias,
        BiasDelta = n.BiasDelta,
        Gradient = n.Gradient,
        Value = n.Value
    };

    network.OutputLayer.Add(neuron);
    allNeurons.Add(neuron);
}

//Synapses
foreach (var syn in dn.Synapses)
{
    var synapse = new Synapse { Id = syn.Id };
    var inputNeuron = allNeurons.First(x => x.Id == syn.InputNeuronId);
    var outputNeuron = allNeurons.First(x => x.Id == syn.OutputNeuronId);
    synapse.InputNeuron = inputNeuron;
    synapse.OutputNeuron = outputNeuron;
}

```



```

        synapse.Weight = syn.Weight;
        synapse.WeightDelta = syn.WeightDelta;

        inputNeuron.OutputSynapses.Add(synapse);
        outputNeuron.InputSynapses.Add(synapse);
    }

    return network;
}

```

5.4.1.3 ExportNetwork

This method exports the network from system. It calls “GetHelperNetwork”, sets the value of all neurons and writes to text which name is “network.dat”.

```

public static void ExportNetwork(Network network)
{
    var dn = GetHelperNetwork(network);

    using (var file = File.CreateText("network.dat"))
    {
        var serializer = new JsonSerializer { Formatting = Formatting.Indented };
        serializer.Serialize(file, dn);
    }
}

```

Figure 25: ExportNetwork

5.4.2 Functions In Smart Chatbot Project

5.4.2.1 GetResponseFromAIML

This method gets the AIML tag response according to request. AIMLBot DLL is used in this method.

```

public static string GetResponseFromAIML(string request)
{
    AI = new Bot();
    AI.loadSettings();
}

```

```
AI.loadAIMLFromFiles();

AI.isAcceptingUserInput = false;

myUser = new User("KHas Chatbot", AI);

AI.isAcceptingUserInput = true;

Request r = new Request(request, myUser, AI);

Result res = AI.Chat(r);
return res.ToString();
}
```

Figure 26: GetResponseFromAIML

5.4.2.2 FindWordRoot

This method finds the root of word by using Zemberek Library.

```
public static string FindWordRoot(string word)
{
    try
    {
        Kelime[] kelimeler = zemberek.kelimeCozumle(word);

        if (kelimeler.FirstOrDefault() == null
            || kelimeler.FirstOrDefault().kok() == null
            || kelimeler.FirstOrDefault().kok().icerik() == null)
            return word;
        var asdasdasd = kelimeler[0].icerikStr();

        return kelimeler.FirstOrDefault().kok().icerik();
    }
    catch (Exception e)
    {
        throw new Exception("FindWordRoot: " + e.Message);
    }
}
```

Figure 27: FindWordRoot

5.4.2.3 FindCategory

This method finds the related category according to user's sentences. By doing this, method follows these steps;

- 1- Imports the network that we train before
- 2- Gets the word list of user's sentence.
- 3- Find the root words that we find Step 2
- 4- Gets the attributes where the input layer labels on neural network
- 5- Creates an input array according the features of data.
- 6- Compute the input data and gets the double array.
- 7- Find the index of max value and according to that, gets the process name that user wants.

```
public string FindCategory(string sentences)
{
    NeuralNetwork.Models.Network network =
    NeuralNetwork.Controllers.ImportController.ImportNetwork();

    List<string> wordListOfSentences = sentences.Split().ToList();
    List<string> rootwordListOfSentences = new List<string>();
    foreach (var item in wordListOfSentences)
    {
        rootwordListOfSentences.Add(FindWordRoot(item));
    }

    List<string> attributeList = new List<string>();
    using (var file = new
    StreamReader(@"C:\Users\RecepÇifci\Desktop\Tez\SmartChatbot\SmartChatbot\Smart
    Chatbot\AttributeList.txt"))
    {
        string line;
        while ((line = file.ReadLine()) != null)
        {
            attributeList.Add(line);
        }
    }
}
```

```

}

double[,] input = new double[339, 1]; //feature count
for (int i = 0; i < attributeList.Count; i++)
{
    if (rootwordListOfSentences.Contains(attributeList[i]))
        input[i, 0] = 1.0;
    else
        input[i, 0] = 0.0;
}

    double[] output = network.Compute(input);
    double maxValue = output.Max();
    int maxIndex = output.ToList().IndexOf(maxValue);
    string category = String.Empty;
    switch (maxIndex)
    {
        case 0: category = "Kredikarti"; break;
        case 1: category = "hesap"; break;
        case 2: category = "teslimat"; break;
        case 3: category = "odeme"; break;
        case 4: category = "fraud"; break;
        case 5: category = "dummy"; break;
    }
    return category;
}

```

6. CONCLUSION

In today's world, technology is developing at an unbearable pace. Human power now places its place on machine power and even on a machine power similar to human intelligence. Artificial intelligence is one of the most obvious examples of this change. Along with scientific research in the field of artificial intelligence and simultaneous technological developments, it is making good progress. In artificial intelligence processing, artificial neural networks are used similar to human intelligence and intelligent technology is produced. Thanks to this technology, we provide a machine learning, logicalization and self-correction. Thus, it is inevitable that we use the machine's power more accurately and we get better results.

This project is also designed as an example of artificial intelligence. The project aims to perceive the user of the program through these artificial neural networks and direct them to the related workflow. During this process, complex data is analyzed and analyzed first. It then interrogates in the direction of a predetermined parameter and processes for the correct result. Artificial neural networks have the ability to experience similar things as people benefit from their experience. It has the ability to learn in the same way and to question the knowledge it learns. Together with these features, the system will allow the next process to be more accurate when it runs.

The program has a dynamic structure. New words can be defined for the specified categories, so the function of the program can be increased. Since this project is written on the .Net platform, new screens can be designed and quickly integrated into the system. New categories can also be defined according to the need, but for this it is necessary to pass the Weka analysis again, including the new categories of cues.

REFERENCES

- COURSEY, Kino (2004). Living in cyn: mating aiml and cyc together with program n
- DE GASPERIS, Giovanni (2009). Building an aiml chatter bot knowledge-base starting from a faq and a glossary. Journal of e-Learning and Knowledge Society
- FONTE, F. A (2009). Journal of Universal Computer Science
- FREUND, Yoav; MASON, Llew (1999). The alternating decision tree learning algorithm
- HALL, Mark (2009). The WEKA data mining software: an update
- HECHT-NIELSEN, Robert (1988). Theory of the backpropagation neural network
- HOLMES, Geoffrey; DONKIN, Andrew; WITTEN, Ian H. (1994). Weka: A machine learning workbench
- KAWATO, Mitsuho; FURUKAWA, Kazunori; SUZUKI, Ryoji (1987). A hierarchical neural-network model for control and learning of voluntary movement
- KELLER, Frank (2002). Naive Bayes Classifiers. Connect. Stat. Lang. Process. Course Univ. Saarlandes
- KEOGH, Eamonn (2006). Naive bayes classifier
- KOHAVI, Ron (1996). Scaling up the accuracy of Naive-Bayes classifiers: a decision-tree hybrid
- LAWRENCE, Steve (1997). Face recognition: A convolutional neural-network approach
- LEUNG, K. Ming (2007). Naive bayesian classifier. Polytechnic University Department of Computer Science/Finance and Risk Engineering
- MARTINETZ, Thomas (1991). A "neural-gas" network learns topologies
- MIKIC, Fernando (2009). Charlie: An aiml-based chatterbot which works as an interface among ines and humans. In: EAEEIE Annual Conference
- MIKIC, Fernando (2008). T-Bot and Q-Bot: A couple of AIML-based bots for tutoring courses and evaluating students

ODOM, Marcus D.; SHARDA, Ramesh (1990). A neural network model for bankruptcy prediction. In: Neural Networks, 1990., 1990 IJCNN International Joint Conference on. IEEE

SAFAVIAN, S. Rasoul; LANDGREBE, David (1991). A survey of decision tree classifier methodology. IEEE transactions on systems, man, and cybernetics

SANGER, Terence D (1989). Optimal unsupervised learning in a single-layer linear feedforward neural network. Neural networks

(Srivastava, 1999)

WALLACE, Richard (2003). The elements of AIML style. Alice AI Foundation

