

**ESTIMATING THE CHANCE OF SUCCESS
AND SUGGESTION FOR TREATMENT IN
IVF**

A THESIS

SUBMITTED TO THE DEPARTMENT OF COMPUTER ENGINEERING
AND THE GRADUATE SCHOOL OF ENGINEERING AND SCIENCE
OF BILKENT UNIVERSITY

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR THE DEGREE OF
MASTER OF SCIENCE

By

Gizem Mısırlı

August, 2013

I certify that I have read this thesis and that in my opinion it is fully adequate, in scope and in quality, as a thesis for the degree of Master of Science.

Prof. Dr. H. Altay Güvenir(Advisor)

I certify that I have read this thesis and that in my opinion it is fully adequate, in scope and in quality, as a thesis for the degree of Master of Science.

Assoc. Prof. Dr. Hakan Ferhatosmanođlu

I certify that I have read this thesis and that in my opinion it is fully adequate, in scope and in quality, as a thesis for the degree of Master of Science.

Prof. Dr. Serdar Dilbaz

Approved for the Graduate School of Engineering and Science:

Prof. Dr. Levent Onural
Director of the Graduate School

ABSTRACT

ESTIMATING THE CHANCE OF SUCCESS AND SUGGESTION FOR TREATMENT IN IVF

Gizem Mısırlı

M.S. in Computer Engineering

Supervisor: Prof. Dr. H. Altay Güvenir

August, 2013

In medicine, the chance of success for a treatment is important for decision making for the doctor and the patient. This thesis focuses on the domain of In Vitro Fertilization (IVF), where there are two issues: the first one is the decision on whether or not go with the treatment procedure, the second one is the selection of the proper treatment protocol for the patient.

It is important for both the doctor and the couple to have some idea about the chance of success of the treatment after the initial evaluation. If the chance of success is low, the patient couple may decide not to proceed with this stressful and expensive treatment. Once a decision for treatment is made, the next issue for the doctors is the choice of the treatment protocol which is the most suitable for the couple.

Our first aim is to develop techniques to estimate the chance of success and determine the factors that affect the success in IVF treatment. So, we employ ranking algorithms to estimate the chance of success.

The ranking methods used are RIMARC (Ranking Instances by Maximizing the Area under the ROC Curve), SVM^{light} (Support Vector Machine *Ranking Algorithm*) and RI k NN (Ranking Instances using k Nearest Neighbour). All of these three algorithms learn a model to rank the instances based on their score values. RIMARC is a method for ranking instances by maximizing the area under the ROC curve. SVM^{light} is an implementation of Support Vector Machine for ranking instances. RI k NN is a k Nearest Neighbour (k NN) based algorithm that is developed for ranking instances based on similarity metric. We also used RI w k NN, which is the version of RI k NN where the features are assigned weights by experts in the domain. These algorithms are compared on the basis of the AUC of 10-fold stratified cross-validation. Moreover, these ranking algorithms are

modified as a classification algorithm and compared on the basis of the accuracy of 10-fold stratified cross-validation.

As a by-product, the RIMARC algorithm learns the factors that affect the success in IVF treatment. It calculates feature weights and creates rules that are in a human readable form and easy to interpret.

After a decision for a treatment is made, the second aim is to determine which treatment protocol is the most suitable for the couple. In IVF treatment, many different types of drugs and dosages are used, however, which drug and the dosage are the most suitable for the given patient is not certain. Doctors generally make their decision based on their past experiences and the results of research published all over the world. To the best of our knowledge, there are no methods for learning a model that can be used to suggest the best feature values to increase the chance that the class label to be the desired one. We will refer to such a system as *Suggestion System*.

To help doctors in making decision on the selection of the suitable treatment protocols, we present three suggestion systems that are based on well-known machine learning techniques. We will call the suggestion systems developed as a part of this work as NSNS (Nearest Successful Neighbour Based Suggestion), k NNS (k Nearest Neighbour Based Suggestion) and DTS (Decision Tree Based Suggestion). We also implemented the weighted version of NSNS using feature weights that are produced by the RIMARC algorithm. Moreover, we propose performance metrics for the evaluation of the suggestion algorithms. We introduce four evaluation metrics namely; pessimistic metric (m_p), optimistic metric (m_o), validated optimistic metric (m_{vo}) and validated pessimistic metric (m_{vp}) to test the correctness of the algorithms.

In order to help doctors to utilize developed algorithms, we develop a decision support system, called RAST (Risk Analysis and Suggestion for Treatment). This system is actively being used in the IVF center at Etlik Zübeyde Hanım Woman's Health and Teaching Hospital.

Keywords: Prediction, Suggestion, Ranking, Classification, RIMARC, SVM, k NN, Decision Trees, Decision Support System.

ÖZET

TÜP BEBEK YÖNTEMİNDE TEDAVİ BAŞARI ŞANSINI TAHMİN ETME VE TEDAVİ YÖNTEMİ ÖNERME

Gizem Mısırlı

Bilgisayar Mühendisliği, Yüksek Lisans

Tez Yöneticisi: Prof. Dr. H. Altay Güvenir

Ağustos, 2013

Tıp alanında, bir tedavi sonucunda başarıya ulaşma şansının karar verilmesi çok önemlidir. Bu tez çalışması, tüp bebek tedavisinde dikkate alınması gereken iki önemli aşama üzerine odaklanmıştır. Bu aşamalardan birincisi gelen hastanın tüp bebek tedavisi için uygun olup olmadığıdır. Hastanın tedaviye uygun olduğu kararı verildikten sonra ikinci aşama hastaya uygulanacak olan en uygun tedavi yönteminin belirlenmesidir.

Hem doktorlar, hem de tedavi uygulanacak olan aday hasta çifti için ilk değerlendirmeden sonra hastaya uygulanacak olan tedavi sonucunda başarıya ulaşma şansı çok önemlidir. Eğer başarı şansı düşük ise, hasta çifti bu pahalı ve stresli tedaviye devam etmek istemeyebilir. Tedavi uygulama kararı verildikten sonra doktorlar için karar verilmesi gereken ikinci konu hasta çifti için en uygun olan tedavi yöntemini seçmektir.

Bu tez çalışmasındaki ilk amacımız tedavi için gelen bir hasta çifti için başarı şansını tahmin etme ve tüp bebek tedavisindeki başarı oranını etkileyen faktörleri bulmak ve amacıyla teknikler geliştirmektir. Bu amaçlar doğrultusunda sıralama algoritmaları kullanılmaktadır.

Kullanılan metodlar RIMARC (Ranking Instances by Maximizing the Area under the ROC Curve), SVM^{light} (Support Vector Machine *Ranking Algorithm*) ve RI k NN (Ranking Instances using k Nearest Neighbour)'dir. Bu algoritmaların her üçü de örnek hastaları onlara atanmış olan skor değerlerine göre sıralamaya dayalı bir model öğrenir. RIMARC, Receiver Operating Characteristics (ROC) eğrisi altında kalan alanı maksimize ederek örnekleri sıralayan bir metoddur. SVM^{light}, destek vektör makinesi algoritmasının örnek sıralaması

için geliştirilmiş bir versiyonudur. $RIkNN$, en yakın komşu algoritmasını esas alan ve örnek sıralamasında benzerlik ölçütünü kullanan bir algoritmadır. Bunlara ek olarak, bu tez çalışmasında $RIkNN$ algoritmasının bir versiyonu olan ve her bir öznitelik için konunun uzmanları tarafından belirlenmiş olan öznitelik ağırlıklarını da dikkate alan $RIwkNN$ algoritmasını da kullandık. Bu algoritmaları değerlendirmek için ROC eğrisi altındaki alan (AUC) değeri ve katmanlaştırılmış 10'lu çapraz geçerlilik yöntemlerini kullandık. Ek olarak, tasarlanan sıralama algoritmalarını birer sınıflandırma algoritması haline getirdik ve bu algoritmaları değerlendirmek için accuracy değeri ve katmanlaştırılmış 10'lu çapraz geçerlilik yöntemlerini kullandık.

Yan ürün olarak RIMARC algoritması tüp bebek tedavisinde başarı şansını etkileyen faktörleri öğrenmektedir. Bu amaçla öznitelik ağırlıklarını hesaplar ve insanların kolaylıkla anlayıp yorumlayabilecekleri kurallar üretir.

Gelen hasta çifti için ilk değerlendirmeden sonra tedavi sonrası şansının yüksek olduğuna karar verilir ise ikinci aşamaya geçilir. Bu aşama hasta için en uygun olan tedavi yönteminin belirlenmesi aşamasıdır. Tüp bebek tedavisi içerisinde çok sayıda ilaç yer almaktadır fakat bu ilaçlardan hangisinin hasta için en uygun olduğu kesin olarak bilinmemektedir. Doktorlar genellikle hasta için ilaç seçimi yaparken geçmişte tedavi ettikleri hastaların değerlerine bakarak karar verirler. Bu karar her zaman olumlu bir şekilde sonuçlanmayabilir çünkü insan hafızası gereği doktorların geçmişte tedavi ettikleri bütün hasta profillerini doğru bir şekilde hatırlayabilmeleri mümkün değildir. Bildiğimiz kadarıyla, istenilen sonucu elde etme şansını arttırmak amacıyla en iyi öznitelik değerini önermek için model öğrenen bir method bulunmamaktadır. Biz bu tür bir sistemi *Önerme Sistemi* olarak adlandıracağız.

Doktorlara, uygun tedavi yöntemlerini belirleme aşamasında yardımcı olmak için bilinen makine öğrenmesi tekniklerine dayalı üç önerme sistemi geliştirdik. Bu çalışmanın bir parçası olarak geliştirilen önerme sistemlerini NSNS (Nearest Successful Neighbour Based Suggestion), $kNNS$ (k Nearest Neighbour Based Suggestion) ve DTS (Decision Tree Based Suggestion) olarak adlandıracağız. Bunlara ek olarak, bu tez çalışmasında NSNS algoritmasının bir versiyonu olan ve her bir öznitelik için RIMARC algoritması tarafından belirlenmiş olan öznitelik ağırlıklarını da dikkate alan $wNSNS$ algoritmasını da kullandık. Ayrıca, önerme

algoritmalarının doğruluğunu deęerlendirmek için performans kriterleri tasarladık. Bu amaçla, bu tez çalışmasında, pessimistic metric (m_p), optimistic metric (m_o), validated optimistic metric (m_{vo}) ve validated pessimistic metric (m_{vp}) olarak adlandırılan dört adet deęerlendirme kriteri sunuyoruz.

Geliştirilen bu algoritmalarından doktorların faydalanmasını sağlamak amacı ile RAST (Risk Analysis and Suggestion for Treatment) adı verilen bir karar destek sistemi geliştirdik. Sistem şıanda Ankara Etlik Zübeyde Hanım Kadın Hastalıkları Eğitim ve Araştırma Hastanesi Tüp Bebek Merkezi'nde aktif olarak kullanılmaktadır.

Anahtar sözcükler: Tahmin, Öneri, Sıralama, Sınıflandırma, RIMARC, SVM, k NN, Karar Ağaçları, Karar Destek Sistemi.

Acknowledgement

First of all, I would like to express by deep gratitude to a very special person, Prof. Dr. H. Altay Güvenir for his guidance, encouragement and suggestions throughout this study. During my master study, I realized that I could not have asked for a better person to guide me in my research. It was a great pleasure for me to work with him in this thesis. I had a chance to observe many supervisors during this three years and it is obvious that I am the luckiest graduate student because I have a great supervisor. I want to thank him with all my heart to give me a chance to work with him. I remain grateful to him during my life.

I would like to thank Assoc. Prof. Dr. Hakan Ferhatosmanoglu and Prof. Dr. Serdar Dilbaz for accepting to read and review the thesis. Moreover, I would like to thank Dr. Özlem Özdeğirmenci and Berfu Demir from Etlik Zübeyde Hanım Woman's Health and Teaching Hospital for providing us the dataset and their precious information about the IVF domain.

I would like to thank my parents, Adalet Mısırlı and Ali Mısırlı for their love and support that always kept me motivated. Also, thanks to my aunt Emine Mısırlı for her great emotional support. I would like to thank my aunt Nilgün Mumcuoğlu and my uncle Osman Mumcuoğlu, because thanks to them, Ankara became a better and lovely place for me.

I also would like to thank TUBITAK-BIDEB and Bilkent University Computer Engineering Department because of their financial support during my graduate study.

I would like to thank all of my friends; Bengü Kevinç, Can Telkenaroğlu, Elif Eser, Gökçen Çimen, Seher Acer, Sinan Arıyürek and Zeynep Korkmaz because of their support. I would like to thank my dear friend Gülden Olgun for her close friendship, love and support. In every difficult situation, she was with me and I am sure that she will do this during our lives. Finally, I would like to thank a very special person Nevzat Orhan for being in my life, for his suggestions and help. Everything would be difficult without him...

To My Family,

Contents

- 1 Introduction** **1**
 - 1.1 Estimation of the Chance of the Success 2
 - 1.2 Suggestion of the Best Treatment Protocol 3
 - 1.3 Decision Support System 4

- 2 Background** **6**
 - 2.1 Ranking 6
 - 2.2 ROC, AUC, AUC Maximization and Accuracy 7
 - 2.2.1 Receiver Operating Characteristics (ROC) 7
 - 2.2.2 Area Under the ROC Curve (AUC) 9
 - 2.2.3 The reason why AUC is more accurate than Accuracy . . . 11
 - 2.2.4 AUC Maximization 12
 - 2.3 Prediction of the Outcome in IVF 13
 - 2.4 Decision Support Systems 14

- 3 In Vitro Fertilization and IVF Dataset** **15**

<i>CONTENTS</i>	xi
3.1 IVF Domain Description	15
3.2 IVF Dataset	17
4 Ranking Algorithms	20
4.1 Ranking Algorithms Introduction	20
4.2 RIMARC: Ranking Instances by Maximizing the Area under the ROC Curve	22
4.3 SVM ^{light} : Support Vector Machine <i>Ranking Algorithm</i>	26
4.4 RI <i>k</i> NN: Ranking Instances using <i>k</i> Nearest Neighbour	27
5 Determining the Factors in the Success of IVF Treatment	30
6 Suggestion of the Best Treatment Protocol	37
6.1 Suggestion Introduction	37
6.2 NSNS: Nearest Successful Neighbour Based Suggestion	38
6.3 <i>k</i> NNS: <i>k</i> Nearest Neighbour Based Suggestion	39
6.4 DTS: Decision Tree Based Suggestion	42
6.5 Performance Evaluation Metrics	48
6.5.1 m_p : pessimistic metric	49
6.5.2 m_o : optimistic metric	50
6.5.3 m_{vo} : validated optimistic metric	51
6.5.4 m_{vp} : validated pessimistic metric	51

CONTENTS xii

7 Empirical Evaluation 53

7.1 Estimation of the Chance of Success 53

 7.1.1 Computation of the AUC metric for Prediction 54

 7.1.2 Computation of the Accuracy for Classification 59

7.2 Suggestion of the Best Treatment Protocol 65

8 Risk Analysis and Suggestion for Treatment (RAST) 73

8.1 RAST Introduction 73

8.2 Ensuring the Data Correctness 76

8.3 User Interface 77

9 Conclusion and Future Work 86

List of Figures

2.1	Confusion matrix of the binary classification outcomes.	9
2.2	Example ROC curve.	10
4.1	The first fold of the training.	21
4.2	The i th fold of the training.	22
5.1	Rule for categoric feature, Male_Female_Blood_Type.	31
5.2	Rule for numerical feature, Female_Age.	34
5.3	Rule for numerical feature, Total_Antral_Follicule_Count.	34
5.4	Rule for numerical feature, Sperm_Motility.	35
5.5	Rule for numerical feature, D3_FSH.	35
5.6	Rule for numerical feature, Weight.	36
6.1	Example of k NN classification.	40
6.2	Example for to suggest an alternative treatment protocol with score value in IVF treatment.	42
6.3	Example of training phase in Decision Tree.	44

6.4	Example of testing phase in Decision Tree.	45
6.5	Splitting the training files in DTS.	46
6.6	Generation of the training datasets for each fold in DTS.	47
7.1	First fold for testing instances using AUC.	54
7.2	i th fold for testing instances using AUC.	55
7.3	Computation of the AUC metric.	56
7.4	Experimental result for dataset IVFa based on AUC.	57
7.5	Experimental result for dataset IVFb based on AUC.	57
7.6	Experimental result for dataset IVFc based on AUC.	58
7.7	Creating sorted and scored training dataset.	61
7.8	Classification of the test instances.	62
7.9	Experimental result for dataset IVFa based on accuracy.	63
7.10	Experimental result for dataset IVFb based on accuracy.	63
7.11	Experimental result for dataset IVFc based on accuracy.	64
7.12	Experimental result for “Ovulation_Induction_Protocol” based on pessimistic metric (m_p).	66
7.13	Experimental result for “Ovulation_Induction_Dose_Protocol” based on pessimistic metric (m_p).	67
7.14	Experimental result for “Ovulation_Induction_Protocol” based on optimistic metric (m_o).	68
7.15	Experimental result for “Ovulation_Induction_Dose_Protocol” based on optimistic metric (m_o).	69

7.16	Experimental result for “Ovulation_Induction_Protocol” based on validated optimistic metric (m_{vo}).	69
7.17	Experimental result for “Ovulation_Induction_Dose_Protocol” based on validated optimistic metric (m_{vo}).	70
7.18	Experimental result for “Ovulation_Induction_Protocol” based on validated pessimistic metric (m_{vp}).	71
7.19	Experimental result for “Ovulation_Induction_Dose_Protocol” based on validated pessimistic metric (m_{vp}).	72
8.1	Administrator interface to RAST.	75
8.2	Editing variable details.	76
8.3	Searching for past cases and list of matching records.	79
8.4	Searching for similar records to the selected patient.	81
8.5	Chance estimation for the selected patient.	82
8.6	Suggestion for “Ovulation_Induction_Protocol” for the selected patient.	83
8.7	Data analysis by the RIMARC algorithm.	84
8.8	Feature weights that are produced by the RIMARC algorithm.	84
8.9	Rules that are produced by the RIMARC algorithm.	85

List of Tables

3.1	Summary of the IVF Datasets.	17
3.2	Features in the IVFa Dataset.	18
3.3	Additional Features in IVFb Dataset.	19
3.4	Additional Features in IVFc Dataset.	19
4.1	An example for chance estimation using RIMARC.	25
5.1	Feature weights learned by RIMARC on the IVF dataset.	32
5.2	Feature weights learned by RIMARC on the IVF dataset Cont.	33
6.1	Example of the k NNS.	43
6.2	Suggested treatment protocols with score values.	43
6.3	Example of the performance evaluation metrics calculation.	52
7.1	AUC values for ranking algorithms for datasets IVFa, IVFb and IVFc.	58
7.2	Accuracy values for ranking algorithm for datasets IVFa, IVFb and IVFc	64

7.3	Results of performance evaluation metrics for suggestible feature “Ovulation_Induction_Protocol”	71
7.4	Results of performance evaluation metrics for suggestible feature “Ovulation_Induction_Dose_Protocol”	72

Abbreviations

RIMARC	Ranking Instances by Maximizing the Area under the ROC Curve
SVM ^{light}	Support Vector Machine <i>Ranking Algorithm</i>
RI k NN	Ranking Instances using k Nearest Neighbour
RIw k NN	Ranking Instances using weighted k Nearest Neighbour
NSNS	Nearest Successful Neighbour Based Suggestion
wNSNS	weighted Nearest Successful Neighbour Based Suggestion
k NNS	k Nearest Neighbour Based Suggestion
DTS	Decision Tree Based Suggestion
RAST	Risk Analysis and Suggestion for Treatment

Chapter 1

Introduction

In Vitro Fertilization (IVF) is a major treatment in infertility, among the assisted reproductive technologies. The IVF treatment involves the use of many different drugs including hormones [1]. Further, it is a quite stressful procedure for both candidate mother and father. The cost of the IVF treatment is also high. If a try (cycle) fails, the couple has to wait for several months before the next try. It is important for both the doctor and the couple to have some idea about the chance of success of the treatment, since if the chance is low the couple may choose to adopt a baby, instead. On the other hand, estimating the chance of success for a given IVF patient constitutes a great challenge in obstetrics and gynecology.

Given a new candidate for IVF, there are two important questions that a doctor has to address. The first question is whether or not the patient should undergo the IVF treatment. If the chances of success are low, the couple may choose not to continue with the treatment. If the answer to this question is yes, then the second question is the treatment protocol to be applied. An IVF protocol specifies all of the steps of the treatment, including the hormones and the medicines to be used, and the way they are to be administered. Although, there are many protocols in common use, it is a difficult question for the doctors to choose the best protocol for a given patient.

In this thesis, several algorithms for predicting the chance of the success and

the suggestion for the best treatment protocol for a given patient are proposed. Also a web based decision support system is developed that implements these algorithms to help doctors in IVF treatment.

1.1 Estimation of the Chance of the Success

In IVF treatment, the most challenging question is whether or not the patient couple is a candidate for a successful treatment. To this end, it is important to estimate the chance of success of the treatment; since if the chance is low, a couple may decide not to continue with the treatment due to cost and side effects. For an IVF treatment, doctors generally make their decisions based on their past experiences. When a new patient couple applied to the clinic, the doctors consider the previous couples that are the most similar to the new one.

If the data about the previous patients, including clinical parameters, and the results of treatments are available, machine learning techniques could be of great value for doctors and medical personnel.

In this thesis, we show that a ranking algorithm that learns a model to rank instances based on a score value can be used to estimate the chance of success in an IVF treatment. Moreover, these ranking algorithms can be used for classify the instances as Successful or Failure.

Given a new patient couple, such a ranking method assigns a score to the new couple and determines its rank for success among the training instances. Then, the chance of the success of the treatment for the new couple can be estimated as the ratio of successful training instances among the ones with similar score values.

We briefly sketch three ranking algorithms, namely RIMARC (Ranking Instances by Maximizing the Area under the ROC Curve), SVM^{light} (Support Vector Machine *Ranking Algorithm*) and RIKNN (Ranking Instances using k Nearest Neighbour). We also implemented the weighted version of R k NN that is R wk NN

(Ranking Instances using weighted k Nearest Neighbour). RIMARC is a recently introduced method that learns to rank instances by aiming to maximize the area under the ROC curve [2]. It is shown that RIMARC is a simple yet efficient and fast algorithm. SVM^{light} is an implementation of Vapnik's Support Vector Machine [3] for the problem of pattern recognition, for the problem of regression, and for the problem of learning a ranking function. RI k NN is a k Nearest Neighbor (k NN) based algorithm that is developed for ranking instances based on similarity metric. We also implemented RIw k NN, which is a version of RI k NN, where the features are assigned weights by experts in the domain. According to our experimental results, it is clearly shown that RIMARC outperforms other methods in terms of AUC. As a classification algorithm, RIMARC again outperforms other methods in terms of accuracy on the average.

1.2 Suggestion of the Best Treatment Protocol

After a decision is made for a given patient couple, if the IVF treatment is decided to start, doctors have to decide on the most suitable treatment protocol, which includes the types of the drugs and the way they are to be applied.

The goal of this research is to develop machine learning algorithms that learn models to suggest best values for selected features in a way that the chance of achieving the desired result will be maximized. Therefore, we aimed to suggest the best value for the selected feature especially the treatment protocol for our problem, since, if the suggested feature is the most valuable one for the patient, than the chance of achieving the desired result will be maximized.

As it is known from classical machine learning techniques, if there exists a data about the previous patients that include clinical parameters, applied treatment protocols and the results of the treatment, these techniques can be used by their help, while deciding the treatment protocol doctors can be more self-confident and the chance of acquiring positive result can be increased.

In this thesis, we propose three suggestion algorithms. They are NSNS (Nearest Successful Based Suggestion), k NNS (k Nearest Neighbour Based Suggestion) and DTS (Decision Tree Based Suggestion). We also propose the weighted version of NSNS called wNSNS, using feature weights that are produced by the RIMARC algorithm.

Evaluating the correctness of suggestion is also a challenge. Since there is no suggestion system in the literature, there are no methods proposed to be used as an evaluation metric. In this thesis, we introduce four performance evaluation metrics that are pessimistic metric (m_p), optimistic metric (m_o), validated optimistic metric (m_{vo}) and validated pessimistic metric (m_{vp}). According to the performance evaluation metrics, DTS outperforms other algorithms in overall evaluation.

The most important contribution of this thesis is the definition of suggestion as a machine learning problem. Here we defined the problem, proposed three machine learning algorithms, and formulated four metrics for the evaluation of these algorithms. To the best of our knowledge, there are no algorithms in the literature for suggestion. It is a newly defined problem and this thesis will be the first academic work that contributes to the literature for suggestion.

1.3 Decision Support System

Medical domains are among the areas where decision support systems are applied successfully. Making a diagnosis based on the symptoms seen in a patient or deciding on the best treatment for a given patient is the most challenging part in medical domains. Doctors generally make their decisions based on their experiences; however, these decisions may not always be successful as expected. In order to increase the chance of achieving the desired results, decision support systems are developed to help doctors. These systems provide doctors with alternatives that are more likely to result in successful treatment.

As it is mentioned above, our aim is to develop algorithms to predict the outcome of treatment, and give suggestions about the treatment protocol to achieve the desired result for the IVF patient. We want to allow doctors to take the advantage of these methods because the results are really valuable. If the doctors take into consideration the results of prediction and suggestion algorithms, the success rate of the IVF treatment increases. So, in order to bring our algorithms into use, we developed a web based decision support system called RAST (Risk Analysis & Suggestion for Treatment). The RAST system also helps in the data entering by checking the plausibility of the values. We provide data correctness by defining limitations. Doctors can observe how the process will continue and they can compare patients and judge about them.

In the next chapter, literature summary about the ranking algorithms, ROC, AUC maximization, accuracy, prediction, classification and decision support systems are given. Chapter 3 covers the IVF domain and the dataset. Chapter 4 introduces the theoretical background of the ranking algorithms that are RIMARC, RIKNN and SVM^{light}, their implementation details and how they are used to predict the chance of success in the IVF treatment. The RIMARC algorithm also learns rules and weights about the factors affecting the outcome of an IVF treatment. Chapter 5 gives information about the rules and weights learned by RIMARC. Chapter 6 covers the suggestion algorithms that are NSNS, kNNS and DTS. It also presents performance evaluation metrics namely; pessimistic metric (m_p), optimistic metric (m_o), validated optimistic metric (m_{vo}) and validated pessimistic metric (m_{vp}). In Chapter 7, empirical evaluation of prediction and suggestion algorithms are presented. Chapter 8 gives information about the decision support system, namely RAST. Finally, Chapter 9 concludes with some directions for future work.

Chapter 2

Background

This chapter starts with a background on the ranking problem. Evaluation metrics such as ROC, and AUC are detailed. The ROC, AUC, AUC Maximization and accuracy subjects are given since they are essential for ranking algorithms RIMARC, SVM^{light} and RIKNN. Then, prediction and classification subjects are determined. Next, the intelligent decision support systems for IVF are outlined.

2.1 Ranking

The ranking problem can be classified as a binary classification problem with additional ordinal information. In the binary classification problems, a finite sequence of training examples $z = ((x_1, y_1), \dots, (x_n, y_n))$, where the instances are x_i in some instance space X and with their class labels y_i belongs to $Y = \{s, f\}$. Here s and f are two possible class labels. In our examples, s will stand for successful and f will stand for failure cases. The aim in binary classification problems to learn a binary-valued function $h: X \rightarrow Y$ that predicts the class labels for future instances [2].

In the machine learning literature, the problem of learning a real-valued function that induces a ranking over an instance space is very important. Information

retrieval, estimation of risks associated with a surgery or credit-risk screening are some examples of the application domains. The problem of learning a ranking function from a training set of examples with binary labels to rank positive instances higher and negative instances are lower is known as *bipartite ranking problem* [4], [5], [6]. Agarwal and Roth [4] worked on to learn a bipartite ranking function and showed that learning linear ranking functions is NP-hard.

Different ranking functions have been developed for particular domains such as information retrieval [7], [8]. In medicine, Conroy et al. [9] developed ranking function to estimate ten-year risk of fatal cardiovascular disease. Also, Agostino et al. [10] and Provost et al. [11] proposed ranking functions in medical domain. In the field of insurance, Kevin et al. [12] worked on insurance applications of some risk measures. In addition to them, there exist research areas where different ranking functions are developed such as finance and fraud detection [13], [14].

2.2 ROC, AUC, AUC Maximization and Accuracy

ROC curves, AUC and Accuracy metrics are popular due to the fact that their application to the machine learning techniques. AUC and Accuracy are used in order to evaluate machine learning algorithms as a learning criterion. We explain these subjects in this section. The reason why AUC is more accurate than Accuracy and AUC Maximization subjects are determined in this section.

2.2.1 Receiver Operating Characteristics (ROC)

A ROC curve is a graphical plot that illustrates a performance of a classifier system as its discrimination threshold is varied. The first application of ROC graphs were used to analyse radar signals [15]. After that, the usage of it expanded in different areas such as medicine and signal detection [16], [17], [18]. Spackman has done the first application of ROC graphs in machine learning [19]. ROC

graphs become popular as a performance evaluation measure in the machine learning community after realizing that accuracy is not an accurate metric to evaluate classifier performance [20], [21], [11].

ROC curves are more proper to binary classification problems than multi ones. At the end of the classification phase, each instance is mapped to a class label that is a discrete output. On the other hand, some classifiers such as Neural Networks and Naive Bayes are able to predict a probability value for an instance that belong to a specific class label. This kind of outputs are known as continuous valued output or score. Classifiers that produce a discrete output represented as a single point in the ROC space because only one confusion matrix is produced from their classification output. Classifiers that produce continuous output can have more than one confusion matrix by applying different thresholds to predict class membership. For ranking algorithms in this thesis, instances who have a higher score value than the threshold are predicted to be **s** class and all others are predicted to be **f** class.

ROC space is a two dimensional space with a range of (0.0, 1.0) on both x and y axes. A ROC space is defined by *True Positive Rate (TPR)* and *False Positive Rate (FPR)* as x and y axes. The *TPR* defines how many correct positive results occur among all positive instances during the test. On the other hand, *FPR* defines how many incorrect positive results occur among all negative instances during the test.

Let us consider a binary classification problem where the outcomes are classified as **s** (Successful) and **f** (Failure) and in order to calculate the *TPR* and *FPR*, we need to know four possible outcomes of a binary classifier. If the outcome from a prediction is **s** and the actual value is also **s**, then it is called a true positive (*TP*); however if the actual value is **f** then it is said to be a false positive (*FP*). Conversely, a true negative (*TN*) has occurred when both the prediction outcome and the actual value are **f**, and false negative (*FN*) is when the prediction outcome is **f** while the actual value is **s**. These outcomes constitute the parts of the confusion matrix that can be showed in Figure 2.1.

TPR and *FPR* values are calculated by using Equation 2.1. The number of

		<u>Actual Value</u>	
		s	f
<u>Prediction Outcome</u>	s	<i>TP</i>	<i>FP</i>
	f	<i>FN</i>	<i>TN</i>
<u>Total:</u>		<i>S</i>	<i>F</i>

Figure 2.1: Confusion matrix of the binary classification outcomes.

s labelled instances is indicated by S and that number of **f** labelled instances is by F .

$$\begin{aligned}
 TPR &= TP/S \\
 FPR &= FP/F
 \end{aligned}
 \tag{2.1}$$

As it is mentioned before, the classifiers that produce continuous output can form a curve because they are represented with more than one point in the ROC graph. As a result, to draw the ROC graph different threshold values are selected and different confusion matrices are formed.

2.2.2 Area Under the ROC Curve (AUC)

The area under ROC (receiver operating characteristic) is a widely used an accepted performance evaluation metric for evaluating machine learning algorithms and quality of a ranking function [22], [23].

ROC graphs are proper to use in order to visualize the performance of a classifier, however, to compare classifiers a scalar value is needed. In the literature,

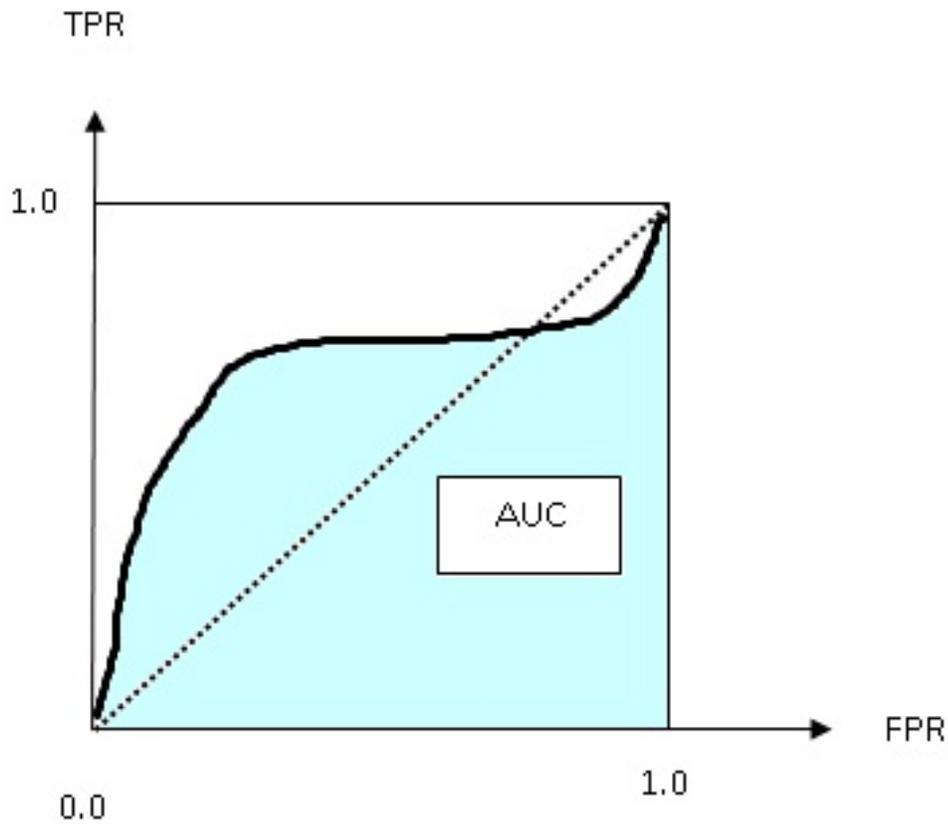


Figure 2.2: Example ROC curve.

ROC curve is intended as a performance evaluation metric by Bradley [22]. The classifier that has a higher AUC value is approved by having a better performance in general. In spite of having a higher AUC value, a classifier can be outperformed by another one in some regions of ROC space for particular threshold values.

The ROC graph space is a one-unit square. So, the maximum AUC value is 1.0 that also means the perfect classification. In ROC graphs, a 0.5 AUC value represents random guessing and values lower than 0.5 are not realistic. An example ROC curve is shown in Figure 2.2.

The AUC value of a classifier is the same as the probability that the classifier will rank a randomly chosen positive instance higher than a randomly chosen negative instance. It is shown that this is equal to the Wilcoxon test of ranks [24].

AUC has important characteristics such as insensitivity to class distribution and cost distributions [22], [23], [21]. Moreover, in the literature there are studies that show what kind of classification algorithms can be used for ranking problems [25].

2.2.3 The reason why AUC is more accurate than Accuracy

Accuracy has been widely used as the main criterion for comparing the predictive ability of classification systems. Most of these classifiers also produce probability estimations of the classification, but they are completely ignored in the accuracy measure. This is often taken for granted because both training and testing sets only provide class labels [26].

There are several reasons why AUC outperforms accuracy. The first one is the independence of the decision threshold of the AUC. AUC has the ability to measure the quality of ranking so it is a better performance evaluation metric in this domain.

Second reason is the discrimination power of the accuracy and AUC metrics. In the literature, AUC metric is recommended instead of accuracy for classifier algorithms by Bradley [22]. Also, for classification algorithms, ROC analysis is suggested as a powerful tool instead of the applicability of the accuracy by Provost et al. [11]. It is claimed that by Rosset [27], if the aim is to get the maximum accuracy, AUC may be better than empirical error for discriminating between models. Huang and Ling [21] give the formal proof of the superiority of the AUC. They showed that AUC is more discriminating and statistically consistent than accuracy. All of these studies prove the discriminatory power of the AUC metric.

The third reason to prefer AUC as a metric is the skewed (unbalanced) datasets. A dataset becomes an unbalanced when the difference between class distribution is high. Datasets in the areas like medicine [28], [29] and fraud detection [14] are the examples of unbalanced datasets. As an example, if a classifier

predicts the class labels as negative for all instances despite the fact that a few of the instances have very high accuracies, there exists an inaccurate and misleading situation [30].

2.2.4 AUC Maximization

The aim of the classification algorithms is to achieve the maximum accuracy value. Since accuracy is a performance evaluation metric for classification, when the classification algorithm maximizes the accuracy, it means that the algorithm gives a better predictive performance. Due to the fact that accuracy metric has some substantial drawbacks in some domains, AUC metric is preferred as a performance evaluation metric. In the literature, it is shown the maximizing accuracy does not outperform maximizing AUC [31], [32]. As a result, new algorithms that aim to maximize AUC have been developed.

Researchers have proposed some approximation methods that aim to maximize AUC value directly [33], [34], [32]. For example, Ataman et al. [35] proposed a ranking algorithm that maximizes AUC using linear programming. Brefeld and Scheffer [36] presented an AUC maximizing Support Vector Machine. Rakotomamonjy [30] proposed a quadratic programming based algorithm for AUC maximization and showed that under certain conditions 2-norm soft margin Support Vector Machines can also maximize AUC. Toh et al. [37] developed an algorithm in order to optimize the ROC performance directly for the fusion classifier. Ferri et al. [38] presented a method to optimize AUC locally in decision tree learning. Cortes and Mohri [31] proposed boosted decision stumps. Several algorithms have been proposed in order to maximize AUC in rule learning [39], [40], [41]. A nonparametric linear classifier based on the local maximization of AUC was proposed by Marrocco et al. [42]. Sebag et al. [43] presented a ROC-based genetic learning algorithm. Marrocco et al. [44] used linear combinations of dichotomizers for the same purpose. Freund et al. [6] proposed a boosting algorithm that combines multiple rankings. Cortes and Mohri [31] showed that this approach also aims to maximize AUC. Tax et al. [29] proposed a method that weighs features linearly by optimizing AUC to the detection of interstitial lung disease. Ataman

et al. [35] proposed an AUC-maximizing algorithm with linear programming. Joachims [45] introduced a binary classification algorithm by using SVM that can maximize AUC. Ling and Zhang [46] compared AUC-based Tree-Augmented Naive Bayes (TAN) and error-based TAN algorithms. The results showed that the AUC-based algorithms produce more accurate rankings. More recently, Calders and Jaroszewicz [47] suggested a polynomial approximation of AUC in order to optimize it efficiently. Linear combinations of classifiers are also used to maximize AUC in biometric scores fusion [37]. Han and Zhao [48] proposed a linear classifier based on active learning that aims to maximize AUC.

2.3 Prediction of the Outcome in IVF

Although, in the literature there are some intelligent decision support systems for IVF process, the related literature is limited. In the literature, it is seen that early studies that are case-based reasoning systems and neural networks have been constructed in order to predict the outcome of IVF [49], [50]. Sait et al. [51] and Trimarchi et al. [52], proposed decision tree models for predicting the outcome of IVF treatment . The most recent studies on IVF propose Naive Bayes, Bayesian Classification and Support Vector Machines in order to increase the chance of having a baby after IVF treatment. Uyar et al. [53] studied for implantation prediction on IVF embryos using Naive Bayes classification. In another study, the embryo implantation prediction is defined. In this study, embryo based prediction is identified in order to predict the outcome of IVF treatment and SVM based learning system is used [54]. Also, there is a study related to predicting implantation potentials of IVF embryos [55]. Predicting the IVF outcome is really challenging process so generally many researches aim to handle this problem [56], [57].

The area under the ROC curve (AUC) is a widely accepted performance measure for evaluating the quality of ranking. It has become a popular performance measure in the machine learning community after it was realized that accuracy is often a poor metric to evaluate classifier performance [21], [20], [11].

2.4 Decision Support Systems

As huge amounts of data are stored in medical databases, decision support systems (DSS) could be equipped with intelligent tools for efficient discovery and use of knowledge. Many hospitals have equipment for monitoring and data collection devices that provide inexpensive data collection and storage for hospital information systems. Decision support systems (DSS) are designed to assist physicians and other health professionals with decision making tasks, such as determining diagnosis from patient data. In the literature, examples of these kinds of systems can be seen. For example, Berner et al. [58] developed a clinical decision support system called Isabel in order to predict the correct diagnosis in medical cases. Another example for these systems was developed for dietary analysis and suggestions for Chinese menus [59]. Also, in order to improve abdominal aortic aneurysm in a primary care practice, a web based CDSS is designed [60].

In hospitals or medical research centres, patient records collected for diagnosis and prognosis typically encompass values of clinical and laboratory parameters, as well as treatment procedures and drugs that are used. Such datasets usually contain missing or noisy data [61]. Therefore, DSS that are designed to learn from past examples have to be able to cope with noise and missing values.

Chapter 3

In Vitro Fertilization and IVF Dataset

In this section, we give the domain description of the In Vitro Fertilization. Detailed information about the IVF dataset that is gathered from IVF center at Etlik Zübeyde Hanım Woman’s Health and Teaching Hospital is given.

3.1 IVF Domain Description

Infertility can be defined as a couple’s biological inability to have a baby. Various international studies have estimated that between 9% and 14% of couples will have difficulties in conceiving during their reproductive life [62]. If the infertility factor of a couple is identified, an appropriate treatment should be applied in order to conceive a successful pregnancy.

In Vitro Fertilization (IVF) is a major treatment for infertility when other methods of assisted reproductive technology have failed. It is a process by which an egg is fertilized by sperm outside the body. IVF gives the couples a chance of becoming parents. There are five basic steps in the IVF and embryo transfer process: Stimulating and monitoring the development of healthy eggs in the

ovaries, collecting the eggs, collecting the sperm, combining the egg and sperm together in the laboratory and providing the appropriate environment for fertilization and embryo growth, transferring the embryos into the uterus. Fertility medications are prescribed to control the timing of the ovulation and to increase the chance of collecting multiple eggs during one of the woman's cycles. Clinical pregnancy, which is the main outcome measure of an IVF program, is defined as a positive intrauterine gestational sac with fetal heart beat visible by ultrasound. However, the final goal is achieving and maintaining pregnancy in which there are many factors affecting the outcome. The prediction of a successful outcome during IVF critically depends on many parameters that are aimed to provide good-quality embryos. However, the parameters for predicting pregnancy rates after IVF are still lacking. Since the first birth by IVF was achieved in 1978, the techniques involved in assisted reproductive technology have grown at an enormous rate. Nevertheless, there are inconsistencies in the available clinical studies and endpoints. As a result, there are continuous efforts to find parameters that can detect the outcome earlier. It is very likely that the individual prognosis of the couple influences the outcome. Individual patient data analysis will allow us to take the prognostic factors into account and to evaluate their effects on the outcome of the treatment. In a prediction model, factors such as age of the couple, reason and duration of infertility, previous gynecologic surgery, tests for the ovarian reserve of the female and sperm parameters should be included. After the baseline characteristics of the couple, the next step is the decision of the ovulation induction protocol. Several protocols have been described for ovarian stimulation and generally the selection of the stimulation protocol depends on the individual characteristics of the patient.

According to the doctors, the most preferred protocols are long luteal agonist and antagonist protocol. For patients with diminished ovarian reserve, micro-dose agonist and antagonist protocols can be selected. The initial dose of gonadotrophin is tailored to the needs of the individual with typical starting doses range between 150-300 IU. In the decision of dosage, female age, ovarian reserve and body mass index are the main parameters. The decision of protocol and dosage generally depends on clinician expertise. A computerized system could

help to improve care, pre-IVF counselling for patients and most importantly, the outcome.

3.2 IVF Dataset

A dataset of 2,020 patients has been compiled by the IVF unit at Etlik Zübeyde Hanim Women’s Health and Teaching Hospital. For each patient, the dataset contains demographic features, 64 clinical features, and 77 treatment features and the result of the treatment.

In order to evaluate the success of the ranking based prediction algorithms on different states of the treatment process, the IVF dataset is divided into three groups as summarized in Table 3.1. Each dataset contains one dependent feature called Result, that has the value **s** (Successful) if the female patient had the clinical pregnancy 28 weeks after the treatment. It has the value **f** (Failure) if the female patient had only chemical pregnancy or no pregnancy, at all.

Table 3.1: Summary of the IVF Datasets.

Dataset	#instances	#categorical	#numeric	#missing
IVFa	1,801	43	21	15,782
IVFb	1,801	51	50	46,288
IVFc	1,801	78	63	70,693

The first group of the IVF dataset, called IVFa, contains only the clinical features that are known before making a decision on whether to apply the IVF treatment or not. The dataset contains 64 independent features; 52 of them are related to the female and 12 are related to the male. The independent features included in the IVFa dataset are summarized in Table 3.2. Among the independent features, 43 of them take on categorical values and 21 of them are numerical. Categorical features are indicated with a (C) and numerical ones are indicated with a (N). Features that take on only binary values, such as Yes/No or True/False are treated as categorical.

Table 3.2: Features in the IVFa Dataset.

Variables from Female		Variables from Male
Female_Age(N)	Laparoscopy(C)	Male_Factor(C)
Female_Blood_Type(C)	Hysteroscopy(C)	Male_Age(N)
Height(N)	Laparoscopic_Surgery(C)	Male_Blood_Type(C)
Weight(N)	Hysteroscopic_Surgery(C)	Male_Genital_Surgery(C)
BMI(N)	Abdominal_Surgery(C)	Semen_Analysis_Category(C)
Tubal_Factor(C)	Abdominal_Surgery_Category(C)	Male_FSH(N)
Age_Related_Infertility(C)	Gynecologic_Surgery(C)	Sperm_Count(N)
Ovulatory_Dysfunction(C)	Ovarian_Surgery(C)	Sperm_Motility(N)
Unexplained_Infertility(C)	Tubal_Surgery(C)	Total_Progressive_Sperm_Count(N)
Severe_Pelvic_Adhesion(C)	Uterine_Surgery(C)	Sperm_Morphology(N)
Endometriosis(C)	Duration_Infertility(N)	Testicular_Biopsy(C)
Cycle_No(N)	PCOS(C)	TESE_Outcome(C)
D3_FSH(N)	HSG_Cavity(C)	Male_Karyotype(C)
D3_LH(N)	HSG_Tubes(C)	
D3_E2(N)	Hydrosalpinx(C)	
Gravida(N)	Office_Hysteroscopy(C)	
Abortus(N)	Office_Hysteroscopic_Incision(C)	
Alive(N)	Office_Hysteroscopic_Procedure(C)	
DM(C)	Total_Antral_Follicle_Count(N)	
HT(C)	Right_Ovarian_Antral_Follicle_Count(N)	
Thyroid_Disease(C)	Left_Ovarian_Antral_Follicle_Count(N)	
Anemia(C)	Myoma_Uteri(C)	
Hyperprolactinemia(C)	Localization_Myoma_Uteri(C)	
Hepatitis(C)	Endometrioma_Surgery(C)	
Embryocryo(C)	Cyst_Aspiration(C)	
Laparotomy(C)		

The treatment phase is analyzed in two steps: the period up to and including the embryo transfer and the period after the embryo transfer. The second dataset, called IVFb, contains all the features in IVFa and 101 features involving the first phase of the treatment. IVFb dataset contains 51 categorical and 50 numerical features. Finally the IVFc dataset includes all features of IVFb and further features related with the final phase of treatment. IVFc dataset contains 78 categorical and 63 numerical features.

Table 3.3: Additional Features in IVFb Dataset.

Variables		
Ovulation_Induction_Protocol(C)	FSH_Brand_Name(C)	E2_Day2v3(N)
GNRH_Brand_Name(C)	HMG_Brand_Name(C)	E2_Day4v6(N)
GNRH_Duration(N)	HMG_Start_Day(N)	E2_Day7v8(N)
Antagonist_Day(N)	HMG_Dose(N)	E2_Day9v10
Antagonist_Duration(N)	Final_HMG_Dose(N)	E2_Day11v12(N)
Supressed_E2(N)	HMG_Duration(N)	E2_Day13v14(N)
Supressed_FSH(N)	Oral_Contraceptive_Brand_Name(C)	E2_Day15v16
Supressed_LH(N)	Ovulation_Induction_Dose_Day3(N)	E2_Max(N)
Supressed_Progesteron(N)	Ovulation_Induction_Dose_Day6(N)	Follicle_Count_17mm(N)
Supressed_Endometrial_Thickness(N)	Ovulation_Induction_Dose_Final(N)	Follicle_Count_15_17mm(N)
Supressed_Antral_Follicle_Count(N)	Ovulation_Induction_Dose_Protocol(C)	Follicle_Count_10_14mm(N)
Ovulation_Induction_Type(C)	Ovulation_Induction_Duration(N)	HCG_Dose(C)
Ovulation_Induction_Dose_Initial(N)	Ovulation_Induction_Total_Dose(N)	HCG_Cycle_Day(N)
HCG_Endometrial_Thickness(N)		

Table 3.4: Additional Features in IVFc Dataset.

Variables		
OPU_Procedure(C)	Quality_Score_Day2(N)	Catheter_Control(C)
OPU_E2(N)	Quality_Score_Day3(N)	ET_Progesteron(N)
OPU_LH(N)	Quality_Score_Day5(N)	ET_E2(N)
OPU_Progesteron(N)	Number_Embryo_Transferred(N)	ET_Endometrial_Pattern(C)
OPU_Endometrial_Pattern(C)	Number_Embryo_Gr1(N)	ET_Endometrial_Thickness(N)
OPU_Endometrial_Thickness(C)	Number_Embryo_Gr2(N)	Distance_Embryo_Fundus(N)
Method_Sperm_Retrieval(C)	Number_Embryo_Gr3(N)	Freezing_Embryo_Procedure(C)
Total_Oocyte_Count(N)	Number_Embryo_Gr4(N)	Number_Freezing_Embryo(N)
Mature_Oocyte_Count(N)	Blastocyst_Transfer(N)	Lutheal_Support(C)
Number_Inseminated_Oocytes(N)	Assisted_Hatching(C)	Hospitalization_OHSS(C)
Oocyte_Quality_Index(N)	Embryo_Transfer_Procedure(C)	Cycle_Cancellation(C)
Pronuclear2_No(N)	Embryo_Transfer_Type(C)	Result_BHCG(N)
Day_Embryo_Transfer(N)	End_thick_HCG(N)	

Chapter 4

Ranking Algorithms

This chapter presents detailed information about ranking algorithms that are RIMARC (Ranking Instances by Maximizing the Area under the ROC Curve), SVM^{light} (Support Vector Machine *Ranking Algorithm*) and RIKNN (Ranking Instances using k Nearest Neighbour).

4.1 Ranking Algorithms Introduction

In medicine, the chance of success for a treatment is important and risky for decision making for both doctor and the patient. In this thesis, the first problem is to predict the outcome of the IVF treatment. It is very crucial in the decision on proceeding with the treatment. It is very important for the doctor and the patient couple in the beginning stage of the treatment because this gives some idea about the chance of success of the treatment after the initial evaluation. As a result of this, if the chance of success is low, the patient couple may decide not to proceed with this stressful and expensive treatment.

In this research, the aim is to determine the factors that affect the success in IVF treatment and develop techniques that can be used to estimate the chance of success and classify the given patient as it will be successful or failure at

the beginning. The objective in developing the techniques for estimation is to employ ranking based algorithms where the ranking criterion ranks the instances according to their chance of success.

The methods used are RIMARC, SVM^{light} and RIKNN. Also, the weighted version of the RIKNN is used namely, RIwKNN where the features are assigned weights by experts in the domain. All of these algorithms learn a model to rank the instances based on their score values and these algorithms are compared on the basis of the AUC of 10-fold stratified cross-validation.

Ranking algorithms include two steps that are train and test. For computing AUC, 10-fold cross-validation technique is applied on the dataset. That means, the dataset is partitioned into 10 equal size sub-datasets. Among 10 sub-datasets, a single dataset is retained as the test dataset for testing the model, and the remaining 9 sub-datasets are used as training data. The cross-validation process is repeated 10 times, with each of the 10 sub-datasets are used exactly once as the test dataset. For each fold, ranking algorithms take the training datasets as an input and produce a model. The training operations are shown in Figure 4.1, Figure 4.2.

The first fold of train

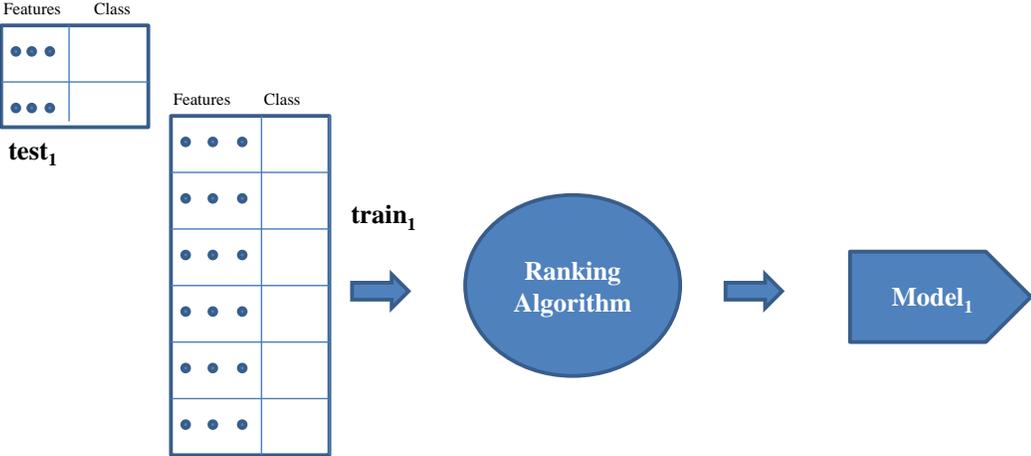


Figure 4.1: The first fold of the training.

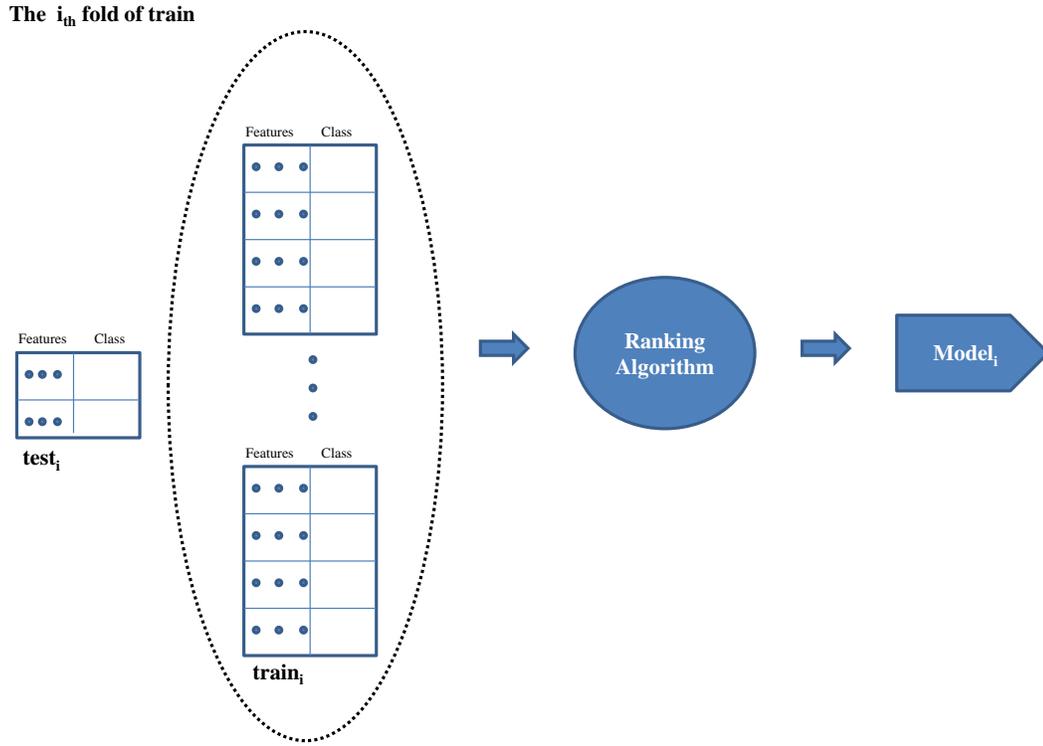


Figure 4.2: The i th fold of the training.

In the following sections, the details of the proposed ranking algorithms are given.

4.2 RIMARC: Ranking Instances by Maximizing the Area under the ROC Curve

RIMARC is a supervised, non-parametric algorithm that learns a ranking function [2]. The RIMARC algorithm aims to maximize the AUC value, since the area under the ROC curve (AUC) has become a widely accepted performance evaluation metric in order to evaluate the quality of ranking.

It learns a ranking function which is a linear combination of non-linear score functions constructed for each feature separately. Each of these non-linear score

functions aims to maximize the AUC by considering only the corresponding feature in ranking. It has been shown that, for a single categorical feature, it is possible to derive a scoring function that achieves the maximum AUC [2]. Therefore the RIMARC algorithm first discretizes all continuous features into categorical ones, in a way that optimizes the AUC, using the MAD2C algorithm proposed by Kurtcepe and Güvenir [63].

A categorical feature f has a finite set of values. Let

$$V_f = v_1, v_2, \dots, v_k \quad (4.1)$$

be the set of values for a given categorical feature f . Consider a dataset that includes only this feature and a class value for each instance. That is, an instance is represented by two values: f value and class label. A scoring function $s_f()$ can be defined to rank the elements of V_f . According to this scoring function

$$v_i \preceq v_j \quad (4.2)$$

if and only if

$$s_f(v_i) \leq s_f(v_j) \quad (4.3)$$

.Note that, the problem of ranking the instances in a dataset is reduced to the problem of ranking the values of a feature. Guvenir and Kurtcepe showed that a scoring function has to satisfy the following condition in order to achieve the maximum AUC [2].

$$s_f(v_i) \leq s_f(v_j) \quad \text{iff} \quad \frac{P_i}{N_i} < \frac{P_j}{N_j} \quad (4.4)$$

This newly defined scoring function satisfies the condition in Equation 4.4 and further it is interpretable since it is simply the probability of the p label among all

instances with value v_i . This probability value is easily interpretable by humans. The instances of the dataset, that has a single categorical feature f , are sorted by the scoring function $s_f()$, and the AUC is computed. The AUC obtained by such a scoring function is guaranteed to be between 0.5 and 1.0 [2]. If the feature f is irrelevant, the AUC will be 0.5. On the other hand, if the single feature f is sufficient to predict the class label, that is all positive and negative instances will be separated by the scoring function $s_f()$, the AUC will be 1.0. The RIMARC algorithm uses the AUC value to measure the weight (relevancy) of the feature f , as:

$$W_f = 2(AUC_f - 0.5) \quad (4.5)$$

where AUC_f is the AUC obtained for feature f . The RIMARC algorithm computes the weight of each feature by setting up a sub-dataset, which is composed of only that feature and the class label.

As an example, suppose that if the AUC computed for the feature f is 1, that means perfect ordering and this is the maximum value that AUC can have. That is, all instances in the training set can be ranked by using only the values of feature f . Therefore, we expect that query instances can be ranked correctly by using feature f only, as well.

The rule model learned by the RIMARC algorithm is used to compute the score for a given query patient q as:

$$score(q) = \frac{\sum_f w_f s_f(q)}{\sum_f w_f} \quad (4.6)$$

$$W_f = \begin{cases} 2(AUC_f - 0.5) & q_f \text{ is known} \\ 0 & q_f \text{ is missing} \end{cases} \quad (4.7)$$

Here w_f represents the weight of the feature f , and $s_f(p)$ represents the score

associated with the value of feature f for the patient couple p . For example, consider a 25 years old female, whose BMI is 25.7 and she does not have age related infertility and the semen analysis category for her partner is astheno; and the values of all other features are missing. Then the chance of the outcome of IVF treatment can be computed as shown in Table 4.1.

Table 4.1: An example for chance estimation using RIMARC.

Feature	Feature weight w_f	Feature value	Score value $s_f(q)$	$w_f.s_f(q)$
Female_Age	0.1753	25	0.2374798	0.04163021
BMI	0.1443	25.7	0.21691176	0.03130037
Semen_Analysis_Category	0.1407	astheno	0.35714287	0.05025000
Age_Related_Infertility	0.1178	no	0.22451456	0.02644782
Sum	0.5781			0.1496284
$score(p) = 0.1496/0.5781 = \mathbf{0.2587}$				

The ranking score value is used to locate the query patient among the training cases. However, what is needed is the chance of success of the treatment for a new query patient couple. On the other hand, semantically, the word chance refers to the probability. In order to report the chance of success of IVF treatment for a query patient q , we select the first 100 past (training) patients whose ranking scores are closest to $score(q)$. If the number of successful cases among these 100 training cases is P_{count} , then the chance of success for q is reported as

$$chance(q) = \frac{P_{count}}{100} \quad (4.8)$$

That is, $chance(q)$ represents the probability of success considering the most similar 100 past cases.

Such a ranking algorithm can also be used for binary classification, where the class labels are **s** and **f**. The class label of a query instance q , can be predicted as **s** if the $chance(q)$ is more than or equal to 0.5 as it is shown in Equation 4.9.

$$class(q) = \begin{cases} \mathbf{s} & chance(q) \geq 0.5 \\ \mathbf{f} & otherwise \end{cases} \quad (4.9)$$

4.3 SVM^{light}: Support Vector Machine *Ranking* Algorithm

SVM^{light} is an implementation of Support Vector Machine (SVM) in C [3]. It is designed for ranking problems. It is an implementation of Vapniks Support Vector Machine for the problem of regression, pattern recognition, and for the problem of learning a ranking function. It has many versions. New in this version is an algorithm for learning ranking functions. The goal is to learn a function from preference examples, so that it orders a new set of objects as accurately as possible.

SVM^{light} includes two modules that are learning module (svm_learn) and classification module (svm_classify). The classification module is used for applying the learned model to the new examples. In order to run the algorithms two input files are needed (train and test files). In the classification mode, the target value denotes the class of the example. A +1, as the target value, marks a positive example, -1 a negative example respectively. In our IVF data set, +1 is used to represent a successful instance, and a -1 is used to denote a failure.

The result of the svm_learn algorithm is the model which is learned from the training data in training file. The model is written to model file. To make predictions on test examples, svm_classify reads this file. For all test examples in test file the predicted values are written to the output file. There is one line per test example in the output file containing the value of the decision function on that example. The result of the decision function is real value that can be used as the rank score of the corresponding query instance in test file.

The SVM^{light} algorithm can be used for estimating the chance of success and predicting the class label of a given query instance as for the RIMARC algorithm.

4.4 **RI k NN: Ranking Instances using k Nearest Neighbour**

The k Nearest Neighbour (k NN) is one of the well-known classification methods in machine learning and pattern recognition. The k NN algorithm is a kind of lazy learning algorithm, where the training instances are simply stored and all computation is deferred until classification. It is among the simplest, yet effective, of all machine learning algorithms. The k NN algorithm classifies a query instance by a majority vote of its neighbours. That is, the query instance is assigned to the class most common among its k nearest neighbours. The k parameter is a positive, typically small, integer, indicating the number of nearest neighbours to be considered in the classification. If the value of k is 1, then the query instance is simply assigned to the class of its nearest neighbour [64], [65], [66], [67], [68].

Datasets used in classification methods have several parameters; also called features. These features are the variables that are believed to affect the result of the event. In medical domain, features can be symptoms of an illness, drugs that are applied to the patient and factors that are influential on the result of the treatment. The result of the treatment is called the class variable. Classification algorithms try to generate a model and predict the outcome of the event. In the nearest neighbour approach, this prediction, so called classification, is done based on cases that have been found similar to the queried case. The underlying bias is that, the classification of an instance should be similar to the classification of similar cases. In order to accomplish this goal, all instances are represented as a point in the n -dimensional space where n is the number of features. Since, nearest neighbour approach is a lazy learner, there is no calculation done in the training phase. When test starts, the algorithm tries to classify the query instance as correctly as possible. To find cases that are similar to the case that is being classified, distances to all other instances are computed. Class of the query instance is predicted to be the most frequently occurring class among the k nearest neighbours.

The number of neighbours to be taken into consideration during classification

is a controversial issue. k Nearest Neighbour, shortly k NN, is a well-known algorithm that implements the nearest neighbour approach. K is the number of neighbours to be considered in the overall classification.

In this thesis, we want to estimate the chance of success of the treatment for a given patient. That is, instead of a class value, the estimation algorithm has to return a real value indicating the chance of success. Therefore, in our implementation, the k NN algorithm returns the ratio of positive instances among all k nearest neighbours. That is the probability of success among the k nearest neighbours.

Medical science is one of the most related domains. In medicine, chance of an operation and chance of success for a treatment are all points that need to be handled carefully. Without help of data mining techniques, physicians infer from their past knowledge and conclude accordingly. Nonetheless, machine learning and data mining techniques find correlations in data that are not easily recognizable by human beings. Finding unknown relationships among features and learning dynamically from the dataset facilitate interpretation of the data. Nearest neighbour used in classification problems has been used extensively.

In this thesis, we used a modification of the k nearest neighbour algorithm for predicting the chance of success. Although this research describes the application on IVF treatment, the developed algorithm can be used in all domains in which a chance/probability of success is present.

For IVF treatment, doctors generally make their decisions based on their past experiences. When a new patient couple applied to the clinic, the doctors consider the past couples that are the most similar to the new one. It is easily understood that, this method is similar to the k NN algorithm which is very popular in data mining and machine learning domains. Due to the fact that k NN is easy to interpret for doctors, we developed a new algorithm based on it called RIk NN in order to rank instances.

The similarity between the query patient q and the past patient p , is defined as $s(q,p)$, which returns a real value between 0 and 1; here 1 represents the

exactly same values, while 0 represents a completely different case. The similarity function is defined as

$$s(q, p) = 1 - d(q, p) \quad (4.10)$$

where $d(q, p)$ represents the distance between two records, and returns a real value between 0 and 1. As a distance metric, Euclidean distance is used.

$$d(q, p) = \sum_{f=1}^n \delta(f, q_f, p_f) \cdot w_f \quad (4.11)$$

$$\delta(f, x, y) = \begin{cases} 0.25 & \text{if at least one of } x \text{ or } y \text{ is missing} \\ (x - y)^2 & \text{if } f \text{ is nominal or ordinal} \\ (x == y) & \text{if } f \text{ is categoric} \end{cases} \quad (4.12)$$

Here, w_f is the weight assigned to the attribute f by the doctors. Label values of the ordinal attributes are replaced by their ordinal (integer) values. Then, all numerical and ordinal attributes are normalized using the min-max normalization. While determining the difference on a variable, if at least one of the values is missing, the distance is assumed as 0.25 between two variables.

Having computed the similarity between the query patient couple and the records, starting with the instance that has highest similarity value, all k nearest neighbours are determined. In order to calculate the chance for a query instance, the following formula is used directly.

$$chance(q) = score(q) = \frac{P_{count}}{k} \quad (4.13)$$

Here P_{count} represents the number of instances whose class label is Successful and k represents the number of neighbours considered. That is, $chance(q)$ represents the probability of success considering the most similar k past cases.

Chapter 5

Determining the Factors in the Success of IVF Treatment

In the IVF dataset, there are many features about the patients. Each of them has an affect on the result however, their importance are not equal. Doctors have a general idea about which features are mostly effective on the outcome. According to the gynocologist, the most important factor in IVF is the female age. When a patient comes to the clinic, doctors firstly ask for the age of the patient. If the age is under a threshold, than achieving a positive result at the end of the IVF treatment is high. However, making a decision based on only one feature is not reliable. There are so many important features that affect the result. In order to make a good decision, importance weights of all features and their importances must be determined.

RIMARC learns feature weights and creates rules that are in a human readable form and easy to interpret. For example, a high feature weight value indicates that the corresponding feature is a highly effective factor in IVF. On the other hand, features with low weights may be ignored by doctors. These rules and weight values may be very useful for determining the chance of success since each rule has its own score value and weight. Listing the effects of features based on feature weights and how their particular values affect the ranking.

In Table 5.1 and 5.2, features and their weight values that are learned by the RIMARC algorithm are given. In Figure 5.1 to Figure 5.6, some rules that are learned by RIMARC are illustrated. It is obvious that, these rules are very easy to understand and interpret by domain expert.

There seems to be a strong correlation with the male blood type and the success of the treatment, in favour of B Rh-, see Figure 5.1. In our dataset the blood type of 1881 patient is give. Among them, there are 9 cases where male blood type is B RH- and the result of the treatment is Successful. This may be by chance or there may be a medical explanation. It deserves further investigation by the IVF community.

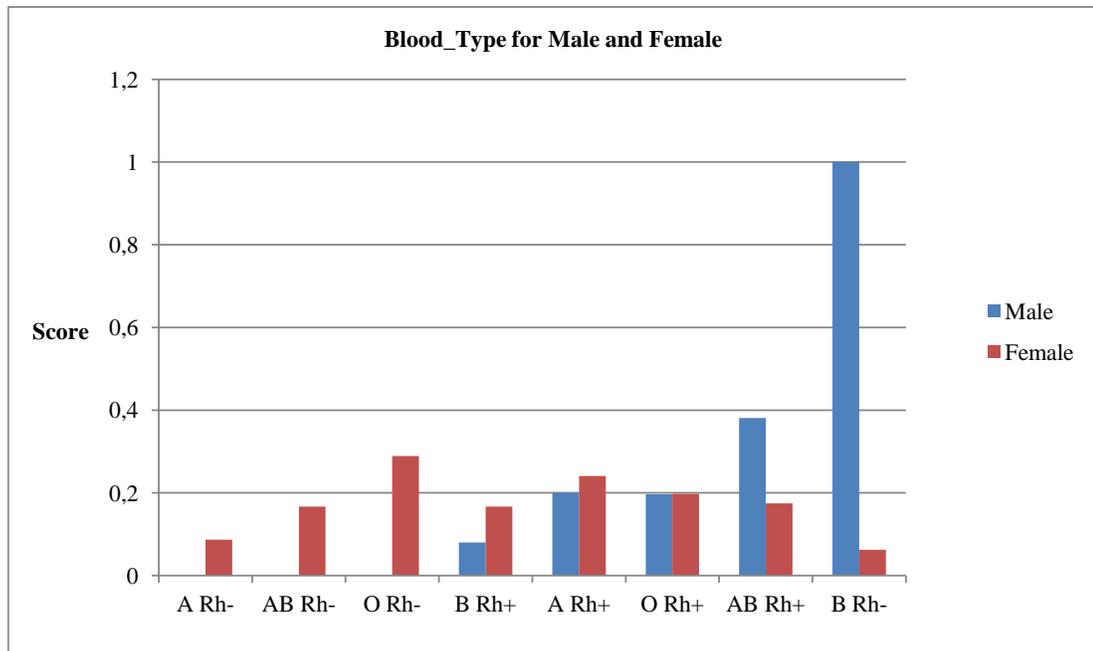


Figure 5.1: Rule for categoric feature, Male_Female_Blood_Type.

Table 5.1: Feature weights learned by RIMARC on the IVF dataset.

Feature	Weight	Feature	Weight
Result_BHCG	0,9905	End_thick_HCG	0,5689
E2_Day15v16	0,5579	Laparoscopic_Surgery	0,5363
TESE_Outcome	0,3936	Embryo_Transfer_Type	0,3871
Embryo_Transfer_Procedure	0,3743	Quality_Score_Day5	0,3421
E2_Day13v14	0,3334	Male_FSH	0,3160
Male_Blood_Type	0,2899	Pronuclear2_No	0,2898
E2_Day2v3	0,2887	Hysteroscopic_Surgery	0,2854
Mature_Oocyte_Count	0,2843	Quality_Score_Day3	0,2829
Number_Inseminated_Oocytes	0,2745	Ovulation_Induction_Dose_Final	0,2417
Total_Oocyte_Count	0,2396	Number_Embryo_Gr1	0,2389
Ovulation_Induction_Dose_Initial	0,2261	Ovulation_Induction_Dose_Day3	0,2253
Total_Antral_Follicle_Count	0,2202	Right_Ovarian_Antral_Follicle_Count	0,2131
Localization_Myoma_Uteri	0,2123	Ovulation_Induction_Dose_Day6	0,2079
Quality_Score_Day2	0,2068	E2_Max	0,2038
Left_Ovarian_Antral_Follicle_Count	0,1984	Ovulation_Induction_Protocol	0,1960
HCG_Endometrial_Thickness	0,1874	Supressed_FSH	0,1865
E2_Day11v12	0,1811	Oocyte_Quality_Index	0,1794
E2_Day4v6	0,1788	E2_Day9v10	0,1781
HMG_Start_Day	0,1763	Follicle_Count_15_17mm	0,1752
Supressed_LH	0,1744	E2_Day7v8	0,1737
ET_Progesteron	0,1696	HMG_Brand_Name	0,1686
Sperm_Count	0,1684	Follicle_Count_17mm	0,1677
Blastocyst_Transfer	0,1667	ET_E2	0,1664
Antagonist_Duration	0,1662	Number_Freezing_Embryo	0,1652
Female_Age	0,1629	Lutheal_Support	0,1586
OPU_Endometrial_Thickness	0,1576	Total_Progressive_Sperm_Count	0,1575
D3_FSH	0,1564	OPU_E2	0,1562
Supressed_Progesteron	0,1507	Follicle_Count_10_14mm	0,1487
Gynecologic_Surgery	0,1466	Day_Embryo_Transfer	0,1462
Unexplained_Infertility	0,1440	Ovulation_Induction_Type	0,1420
Semen_Analysis_Category	0,1411	Number_Embryo_Transferred	0,1402
Ovulation_Induction_Total_Dose	0,1390	Female_Blood_Type	0,1380
Ovulation_Induction_Dose_Protocol	0,1350	GNRH_Duration	0,1325
Freezing_Embryo_Procedure	0,1315	OPU_Progesteron	0,1313
OPU_Procedure	0,1305	Ovulation_Induction_Duration	0,1303

Table 5.2: Feature weights learned by RIMARC on the IVF dataset Cont.

Feature	Weight	Feature	Weight
Supressed_Antral_Follicle_Count	0,1285	BMI	0,1272
Duration_Infertility	0,1217	ET_Endometrial_Pattern	0,1214
Cycle_Cancellation	0,1179	Age_Related_Infertility	0,1151
Final_HMG_Dose	0,1133	Sperm_Motility	0,1130
Male_Age	0,1119	Method_Sperm_Retrieval	0,1118
Height	0,1094	GNRH_Brand_Name	0,1071
Weight	0,1049	Male_Genital_Surgery	0,1035
HMG_Duration	0,1016	Supressed_E2	0,0973
Distance_Embryo_Fundus	0,0945	Office_Hysteroscopy	0,0910
D3_LH	0,0909	Number_Embryo_Gr36	0,0902
OPU_LH	0,0886	HSG_Cavity	0,0869
Office_Hysteroscopic_Procedure	0,0865	Male_Karyotype	0,0865
Sperm_Morphology	0,0861	Supressed_Endometrial_Thickness	0,0833
ET_Endometrial_Thickness	0,0759	Ovulatory_Dysfunction	0,0756
PCOS	0,0751	Antagonist_Day	0,0747
Laparotomy	0,0740	D3_E2	0,0691
Number_Embryo_Gr4	0,0661	Number_Embryo_Gr2	0,0592
HMG_Dose	0,0583	HCG_Cycle_Day	0,0536
Uterine_Surgery	0,0476	Catheter_Control	0,0474
HSG_Tubes	0,0447	Cycle_No	0,0433
Abdominal_Surgery	0,0386	HCG_Dose	0,0382
Tubal_Factor	0,0350	Myoma_Uteri	0,0335
G	0,0300	Ovarian_Surgery	0,0290
Thyroid_Disease	0,0249	Testicular_Biopsy	0,0245
Abdominal_Surgery_Category	0,0245	FSH_Brand_Name	0,0243
Cyst_Aspiration	0,0217	Endometrioma_Surgery	0,0212
Male_Factor	0,0199	Assisted_Hatching	0,0198
Laparoscopy	0,0198	Oral_Contraceptive_Brand_Name	0,0192
OPU_Endometrial_Pattern	0,0120	DM	0,0120
Tubal_Surgery	0,0102	A	0,0093
Endometriosis	0,0084	Hydrosalpinx	0,0078
Hyperprolactinemia	0,0071	HT	0,0070
Embryocryo	0,0065	Hepatitis	0,0057
Office_Hysteroscopic_Incision	0,0033	Y	0,0027
Severe_Pelvic_Adhesion	0,0026	Hysteroscopy	0,0019
Hospitalization_OHSS	0,0015	Anemia	0,0001

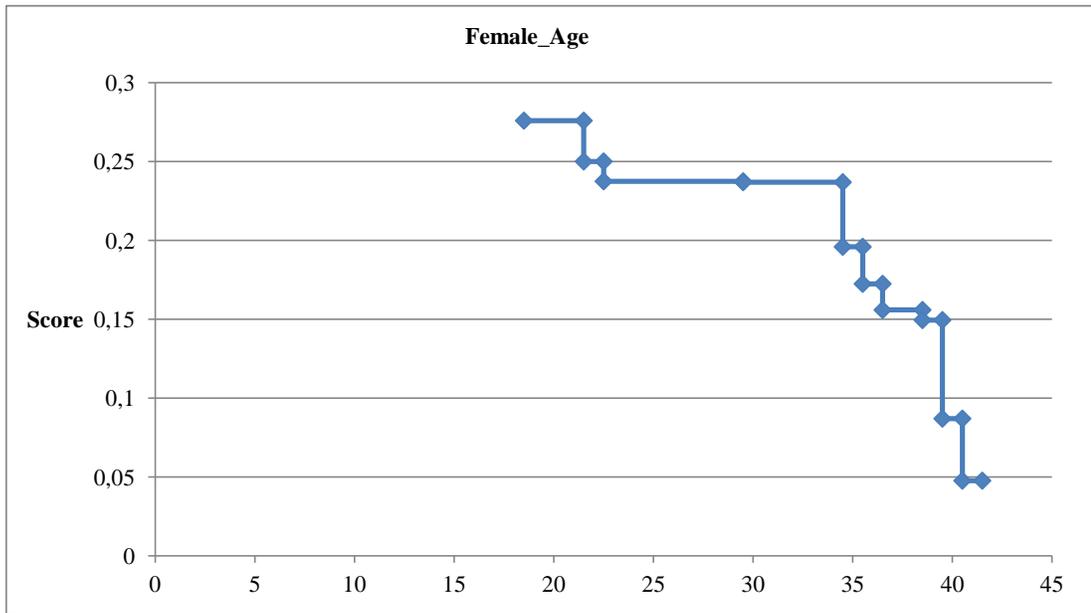


Figure 5.2: Rule for numerical feature, Female_Age.

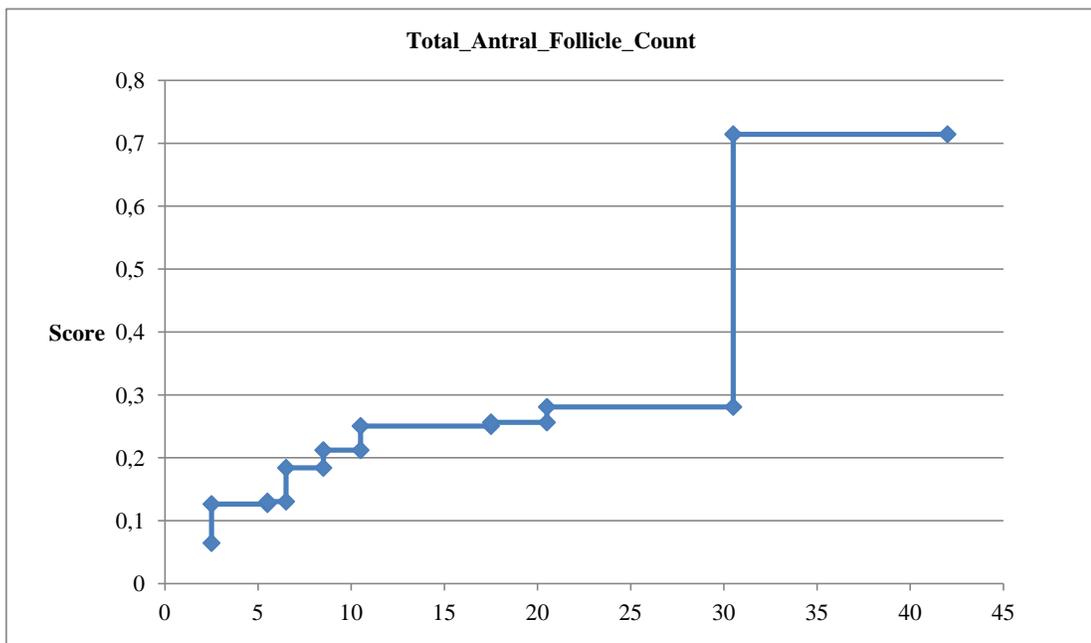


Figure 5.3: Rule for numerical feature, Total_Antral_Follicle_Count.

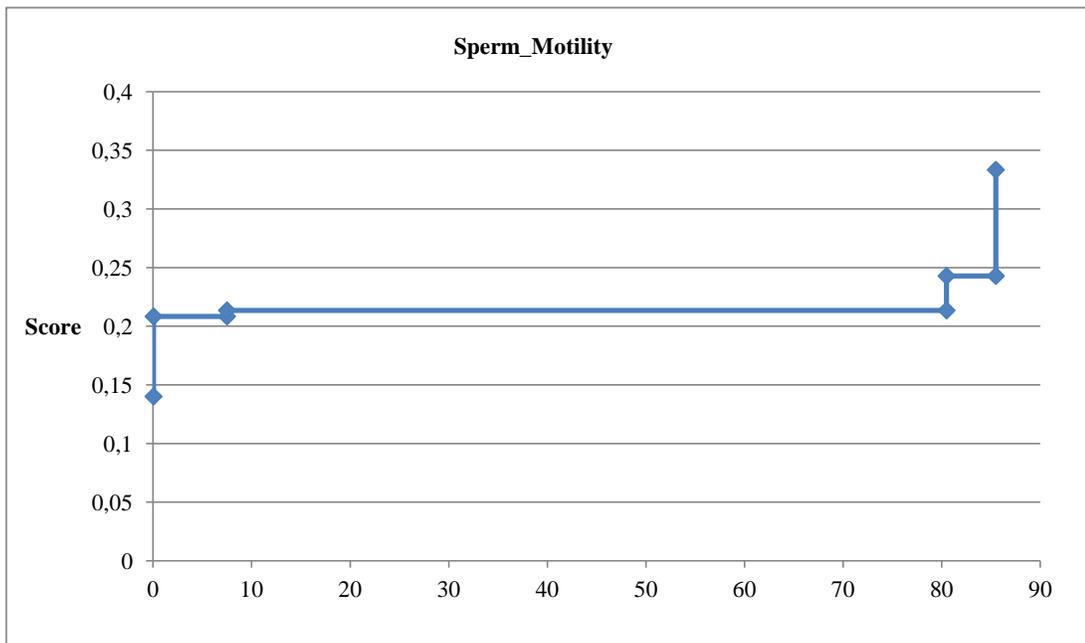


Figure 5.4: Rule for numerical feature, Sperm_Motility.

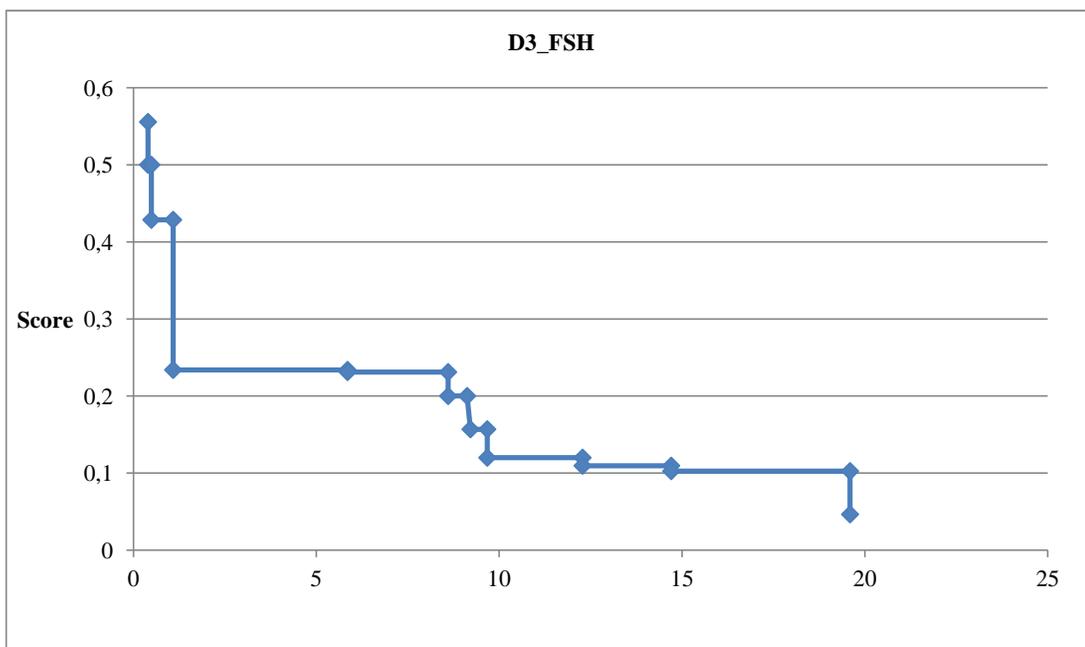


Figure 5.5: Rule for numerical feature, D3_FSH.

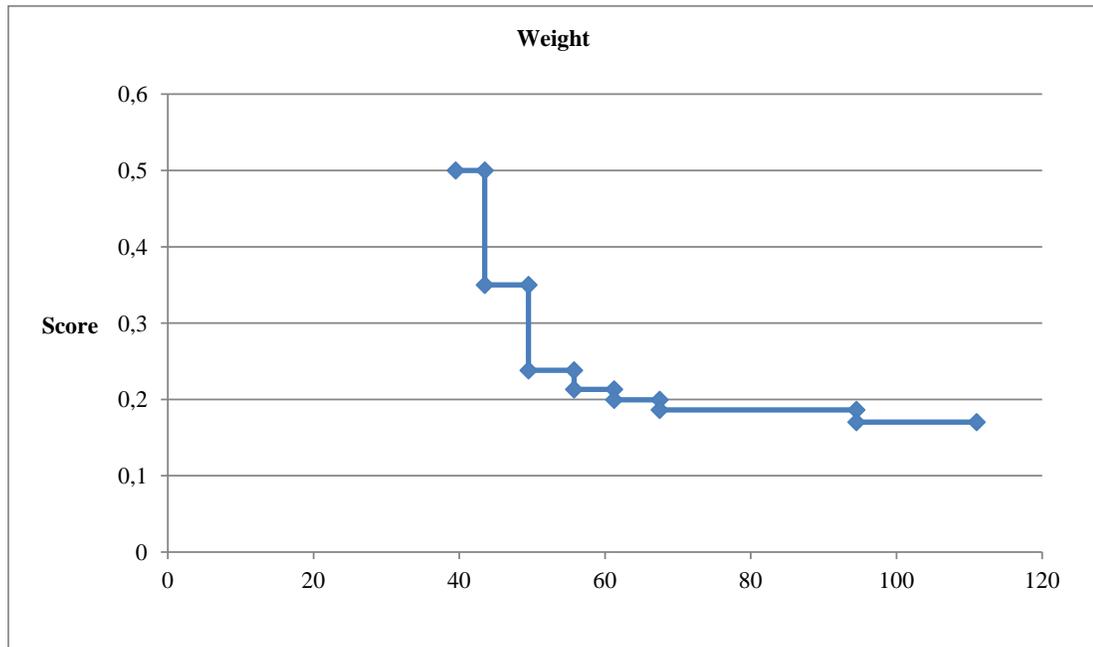


Figure 5.6: Rule for numerical feature, Weight.

Chapter 6

Suggestion of the Best Treatment Protocol

This chapter presents detailed information about suggestion methods that are Nearest Successful Neighbour Based Suggestion (NSNS), k nearest Neighbour Based Suggestion (k NNS) and Decision Tree Based Suggestion (DTS). Also, the detailed information about performance evaluation metrics namely; pessimistic metric (m_p), optimistic metric (m_o), validated optimistic metric (m_{vo}) and validated pessimistic metric (m_{vp}) are given.

6.1 Suggestion Introduction

Suggestion systems aim to give a direction to users in a specific area that can provide reliability in decisions and in parallel with results. These systems can be developed for many different areas like economics and medicine where there exist many choices that can affect the result directly. In such cases, making a decision to select the best choice that provides to get the desired result. In medicine, especially while deciding the best treatment protocol for the patient is a really challenging process. Since, there are many treatment techniques for an illness, selection of the treatment protocol that provides a successful result is

very important. In order to solve this problem, we developed algorithms that are based on machine learning techniques.

Our problem set belongs to patients who have infertility problems. Infertility is the state of the couple who is unable to have a baby. For this problem, there are many ongoing treatment techniques. However, In Vitro Fertilization (IVF) is the major treatment for infertility among the assisted reproductive technologies. This treatment includes decisions about many different types of drugs and dosages that have heavy side effects on the woman. So, the selection of the proper treatment protocol is vital because when a cycle fails, everything must be repeated from the beginning.

This thesis proposes three algorithms that are served as the solution of this problem. Due to the fact that there is no suggestion system in the literature, evaluating the correctness of the algorithms is not clear. So, we developed four performance evaluation metrics namely, pessimistic metric (m_p), optimistic metric (m_o), validated optimistic metric (m_{vo}) and validated pessimistic metric (m_{vp}).

6.2 NSNS: Nearest Successful Neighbour Based Suggestion

Machine learning algorithms have been used in developing models that can be used to make diagnosis in medical domains. On the other hand, another possible application area where machine learning techniques can be utilized is the suggestion of the best treatment for a given patient. Having found similar patients, it is logical for an algorithm to suggest treatments that have been successful among those patients. In domains where contributions of features and possible results of treatment are not crystal clear, physicians sometimes feel the necessity to decide heuristically. The procedure they apply is generally, remembering patients that were similar to the current patient and applying the treatment that has been successful among similar cases. This heuristic is the basis of our algorithm. Our inductive bias for suggestion is that, the nearest successful neighbour's treatment

would be suitable for the queried instance. Therefore, we list all neighbours and find the closest successful neighbour to suggest the treatment. As a distance metric, for numerical features Euclidean metric is used. For categorical features, if the value of the feature for query instance and the test instance is the same, than the distance equals to 0, 1 otherwise. So, we developed an algorithm called NSNS (Nearest Successful Neighbour Based Suggestion).

$$d(q, p) = \sqrt{(q_1 - p_1)^2 + (q_2 - p_2)^2 + \dots + (q_n - p_n)^2} = \sqrt{\sum_{i=1}^n (q_i - p_i)^2} \quad (6.1)$$

There are two main points that affect the result of the treatment in IVF; namely the drug to be applied and the dosage of the drug. Since protocol selection determines the drug and the dosage that are going to be used, these two factors are decided to give a direction to the treatment. The suggestion technique for both of the drug and the dosage is the same. Having found the nearest successful patient, we suggest the drug and the dosage that have been applied to that successful patient. By this way, we aim to increase chance of success for the queried patient.

6.3 *k*NNS: *k* Nearest Neighbour Based Suggestion

Similar to the NSNS algorithm, *k*NNS calculates the distance between queried instance and the training instances. After that, list all instances considering their distance values. Different from the NSNS algorithm, we consider failure cases in this algorithm. Also we suppose that *k* can be different from 1. Considering *k* nearest neighbours, based on their class labels and distance values, we calculate a score for the value of the suggestible feature.

*k*NNS is a *k* Nearest Neighbour based algorithm. The *k*NN algorithm classifies a query instance by a majority vote of its neighbours. That is, the query instance

is assigned to the class most common among its k nearest neighbours as it is shown in Figure 6.1. In the example, the query instance (green circle) should be classified either to the first class of blue squares or to the second class of red triangles. If $k = 3$ (solid line circle) it is assigned to the second class because there are 2 triangles and only 1 square inside the inner circle. If $k = 5$ (dashed line circle) it is assigned to the first class (3 squares vs. 2 triangles inside the outer circle) In this thesis, we want to suggest a treatment protocol to the patient. That is, instead of a class value, the suggestion algorithm has to return a suggestion and a value which represents the score of the suggestion.

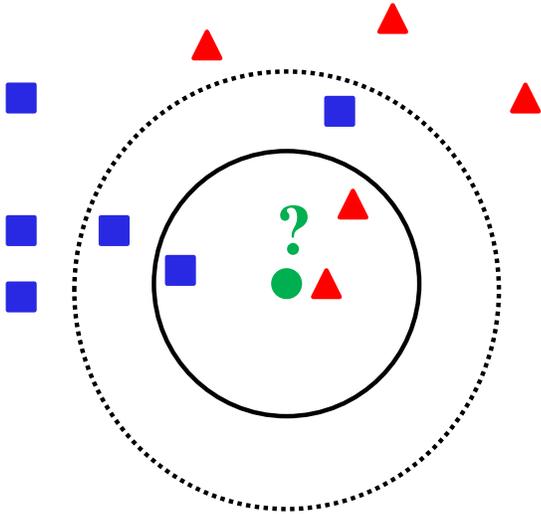


Figure 6.1: Example of k NN classification.

In IVF treatment, doctors generally make their decisions based on their past experiences. When a new patient couple applied to the clinic, the doctors consider the past couples that are the most similar to the new one. It is easily understood that, this method is similar to the k NN algorithm which is very popular in data mining and machine learning domains. Due to the fact that k NN is easy to interpret for doctors, we developed a new algorithm based on it called k NNS in order make suggestions.

In order to find the k nearest neighbours, the distance between the training instances and the query instance must be computed. After computing all distances

between a query instance and the training instances, neighbours are sorted in increasing order based on their distance values and k nearest of them are selected. After that, class labels and applied treatment protocols for neighbours are determined. Treatment protocols that are applied to selected neighbours are identified as possible alternatives. However, we still do not know which one is the best for the queried instance. In order to find the score of the suggestible value, we use the Equation 6.2.

$$S_{sv} = \begin{cases} \frac{1}{e^{d(q,p)}} & \text{if Class Label = Successful} \\ \frac{-1}{e^{d(q,p)}} & \text{if Class Label = Failure} \end{cases} \quad (6.2)$$

In the equation, S_{sv} represents the score of the suggestible value. That means among the alternatives for the given query, alternative with the higher score value is the most proper one. If this alternative is applied to the patient, than the chance of achieving the desired result will be maximized. For example in Figure 6.2, $k = 7$ so we consider the nearest 7 neighbours for suggestion. For the first nearest neighbour, the applied treatment protocol is P1 and the class label is Successful. So, according to the equation 6.2, W_{sv} for P2 equals $\frac{1}{e^{d(q,p_1)}}$. For the second nearest neighbour, the applied treatment protocol is P4 and the class label is Successful. The score value for P4 is computed similarly like P1 and it equals $\frac{1}{e^{d(q,p_2)}}$. For the third nearest neighbour, the applied treatment protocol is P2 and the class label is Failure. So, according to the equation, W_{sv} for P2 equals $\frac{-1}{e^{d(q,p_3)}}$. The algorithm calculates this equation for each neighbour and at the end, we get the sum of the score values of the same treatment protocols. For example, treatment protocol P2 is applied two instances and one of them successes and the other one fails. As a result, the final score value for treatment protocol P2 equals $\frac{-1}{e^{d(q,p_3)}} + \frac{1}{e^{d(q,p_7)}}$. In Table 6.1 and 6.2, example values for neighbour distances and the alternatives with score values are given.

According to the Table 6.2, our algorithm firstly suggest the treatment protocol P4 with score value 0,93 and after that it suggests treatment protocol P1 with score value 0,13. Alternatives with score values that are smaller than 0 are not suggested if there exist an alternative a score value that is greater than 0.

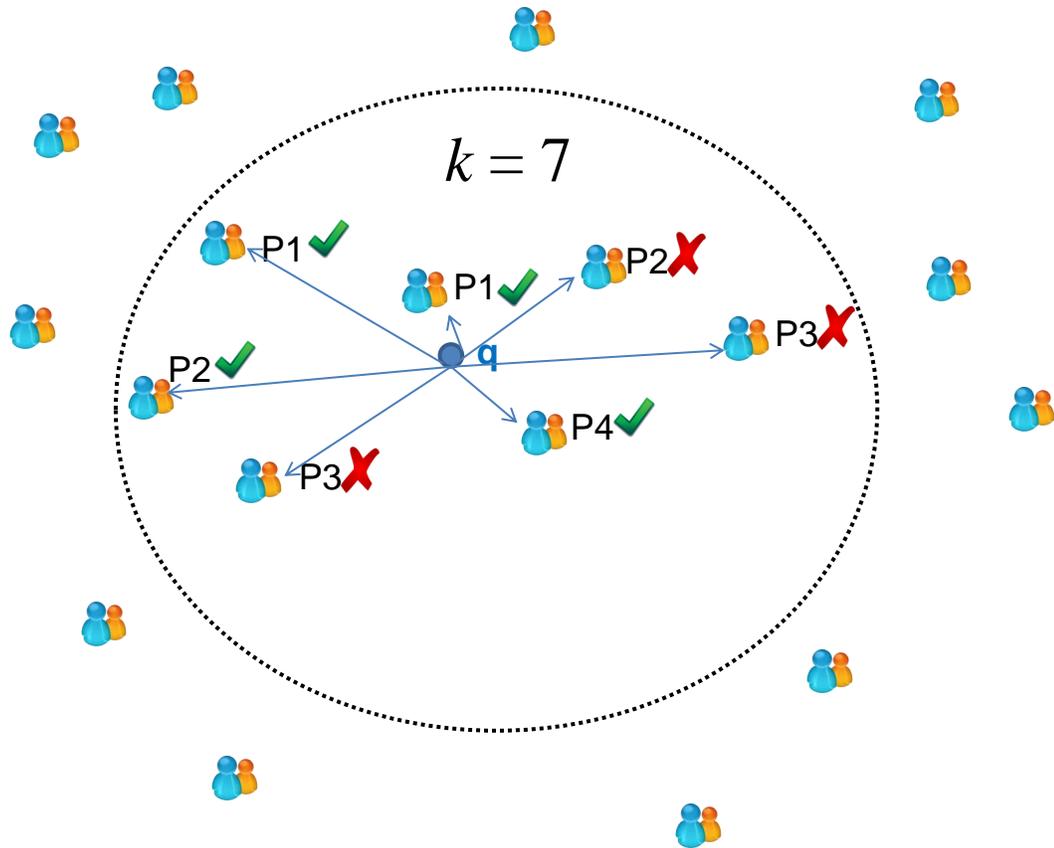


Figure 6.2: Example for to suggest an alternative treatment protocol with score value in IVF treatment.

This output is really valuable for doctors because it says which treatment protocol is most suitable for the patient. Especially the score values indicate the convenience of the treatment protocol for the patient.

6.4 DTS: Decision Tree Based Suggestion

A decision tree is a decision support tool that uses a tree-like model of decisions an possible consequences. It is a one way to represent an algorithm. In data mining and machine learning, decision tree learning uses a decision tree as a predictive model which maps observations about an instance to conclusions about

Table 6.1: Example of the k NNS.

Neighbour	Distance	Applied treatment protocol	Class Label	S_{sv}
N_1	0,05	P1	S	$\frac{1}{e^{0,05}} = 0,95$
N_2	0,07	P4	S	$\frac{1}{e^{0,07}} = 0,93$
N_3	0,13	P2	F	$\frac{-1}{e^{0,13}} = -0,87$
N_4	0,17	P3	F	$\frac{-1}{e^{0,17}} = -0,84$
N_5	0,19	P1	S	$\frac{1}{e^{0,19}} = 0,82$
N_6	0,21	P3	F	$\frac{-1}{e^{0,21}} = -0,81$
N_7	0,22	P2	S	$\frac{1}{e^{0,22}} = 0,80$

Table 6.2: Suggested treatment protocols with score values.

Suggested treatment protocol	S_{sv}
P4	0,93
P1	0,13
P2	-0,07
P3	-1,65

the instances target value. These tree models are also known as classification or regression trees. In these tree structures, each internal node tests an attribute, each branch corresponds to an attribute value and each leaf node represents the class label. Figure 6.3 illustrates an example decision tree for predicting whether a person cheats or not.

In the decision tree, root node and the internal nodes represent an attributes that has specific values. These values represent test conditions and create branches. As it is shown in example, decision tree is constructed from the data gathered in training instances. Once a decision tree is constructed, rules from root node to leaf nodes are determined and each test instance follows one of these rules and reach the target leaf node as it is shown in Figure 6.4.

Decision tree models are used to solve classification problems. They are simple and widely used classification techniques. Furthermore, decision trees can be

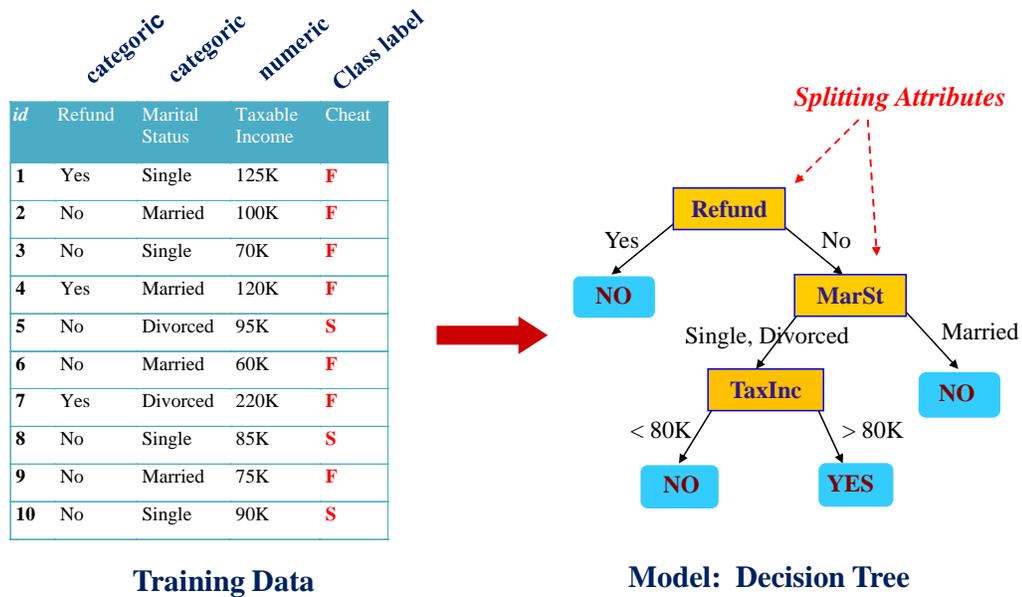


Figure 6.3: Example of training phase in Decision Tree.

converted to a set of rules. Thus, this representation is considered as comprehensible.

In medical domains, there are many situations where decision must be made reliably and effectively. Decision making models with automatic learning functions are most appropriate for performing such tasks. Decision trees are reliable and provide high classification accuracy. So, they have been used in many different medical domains for decision making [69], [70], [71]. However, classification is not the only problem that must be determined. Especially in medicine, treatment technique for an illness is very important for prediction of the result. Since it directly affects the patients life, choosing the proper treatment for the patient is vital. Also, in IVF treatment, the importance of the treatment protocol is really high because when a cycle fails, that means a selected treatment protocol is not proper for the patients and achieving the desired result is not possible, patient couple may be obliged to try one more time from the beginning. Effect of the medicines that are used in the IVF treatment are damaging and treatment costs high. So, decision of the treatment protocol is very crucial for patient couple. Since, there is not any research on suggestion for the best treatment protocol, we developed a decision tree based algorithm namely, Decision Tree Based

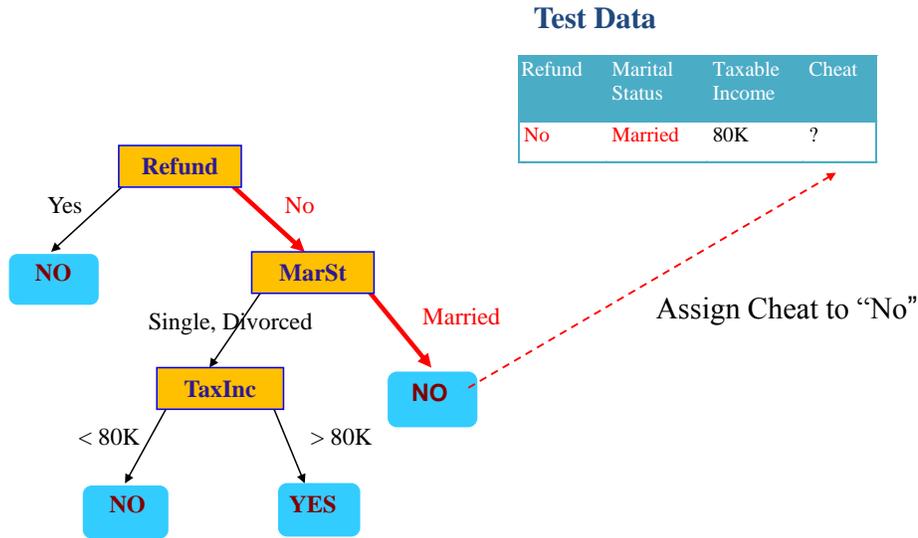


Figure 6.4: Example of testing phase in Decision Tree.

Suggestion (DTS) that aims to suggest a best treatment protocol for a selected patient. Since, our dataset belong to IVF patients so the algorithms aims to give a suggestion that increases the chance of having a baby.

DTS is different from a simple decision tree algorithm. In decision tree classification, there exist one decision tree that is generated from training data. However, in DTS there are more than one decision trees. In IVF dataset, there are two categoric suggestible features that are “Ovulation_Induction_Protocol” and “Ovulation_Induction_Dose_Protocol”. “Ovulation_Induction_Protocol” has 18 different values and “Ovulation_Induction_Dose_Protocol” has four different values. Based on the values of the selected suggestible feature, the training dataset is partitioned as it is shown in Figure 6.5. That means, for each suggestible value there exists a decision tree.

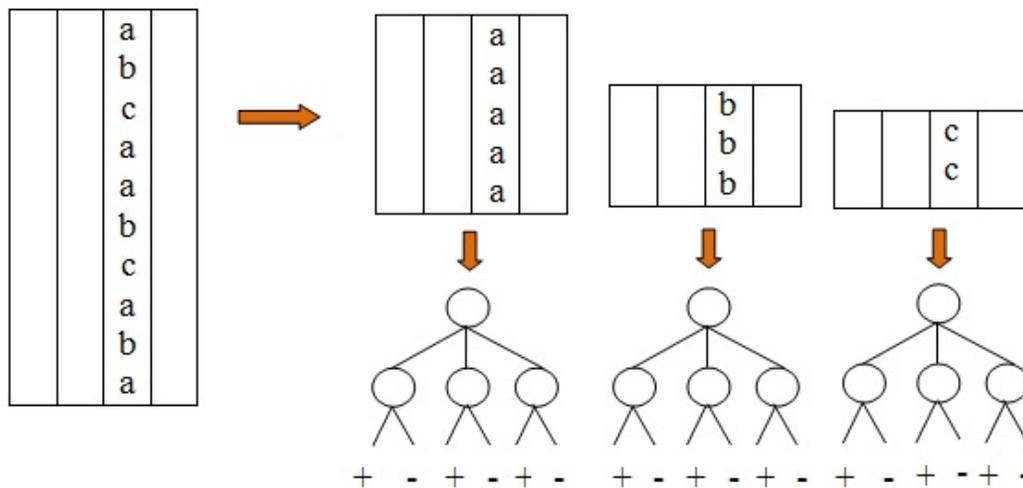


Figure 6.5: Splitting the training files in DTS.

After the construction of the training datasets, we need to convert them into *.arff* file format because we use `weka.classifiers.trees.J48` package to construct model from training instances. When the file conversion is completed, we make a system call from our source code using the command `java -cp weka.jar weka.classifiers.trees.J48 -t trainFileName -d modelName -no-cv -c 1` for each training dataset and Weka constructs a decision tree model. We use `weka.classifiers.trees.J48` as a classifier. Furthermore, “trainFileName” represents the name of the training datasets. We create a model and save it in a file named `modelName`. Model file names are unique. They are composed of the fold number and the value of the suggestible feature. Lastly, we represent the class label as the first feature in the dataset so, in order to provide Weka to understand it, we give the index number of the class label. Otherwise, Weka accepts the last feature as a class label.

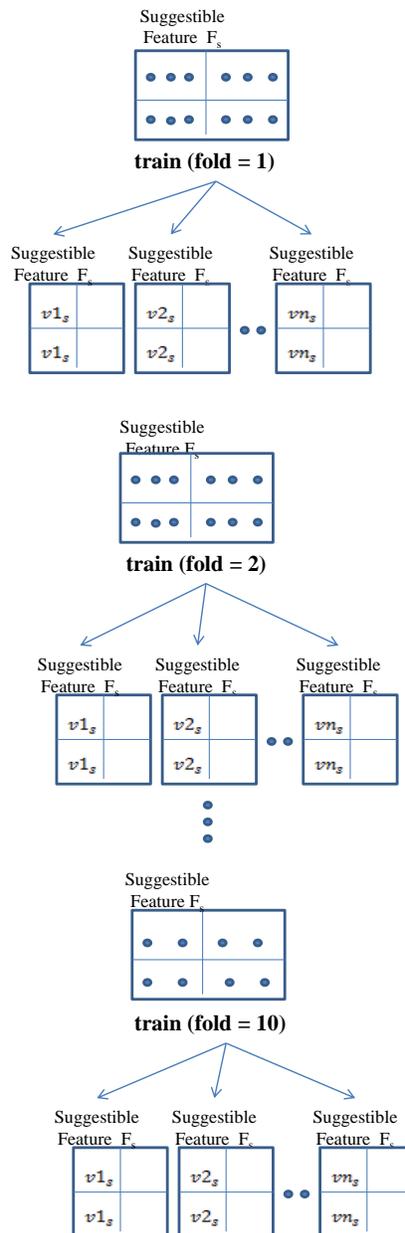


Figure 6.6: Generation of the training datasets for each fold in DTS.

When the model construction is completed, testing instances for each fold entered to the models that represent the values of the suggestible feature. Each test instance is classified as Successful or Failure and assigned a confidence factor

value. For the classification phase, we again use `weka.classifiers.trees.J48` package.

For the performance evaluation, we need three values from the output of the test phase. The first one is the actual class label of the test instance. Second one is the value of the suggestible feature that was previously applied to the test instance. Lastly, the confidence factor of the model is calculated. Confidence factor of the class label ($CF_{label(\mathbf{s},\mathbf{f})}$) represents the total number of instances that are tested as n_{test} and misclassified instances as n_{error} in the format of (n_{test}/n_{error}) and the following equation is the calculation.

$$CF_{label(\mathbf{s},\mathbf{f})} = \frac{n_{test} - n_{error}}{n_{test}} \quad (6.3)$$

As it is same in the k NNS, we give suggestions with score values. In this technique, score values are determined by considering the confidence factor of the decision tree that belongs to each value of a suggestible feature for each fold. Equation 6.4 illustrates the conditions for assigning score values to the values of the suggestible features.

$$score_{sv} = \begin{cases} 1 - CF_{label(\mathbf{s},\mathbf{f})} & \text{if class label} = \mathbf{f} \\ CF_{label(\mathbf{s},\mathbf{f})} & \text{otherwise} \end{cases} \quad (6.4)$$

6.5 Performance Evaluation Metrics

To the best of our knowledge, suggestion is a new topic in the literature and there is no evaluation metric for testing the developed algorithms in this domain. We developed two suggestion algorithms and in order to make their performance evaluations, we described four new metric that are called pessimistic metric (m_p), optimistic metric (m_o), validated optimistic metric (m_{vo}) and validated pessimistic metric (m_{vp}).

In suggestion, a suggestible categoric feature f_s is selected and its values

$V_f = v_1, v_2 \dots, v_k$ can be suggested for the queried instance. In the evaluation phase, three important issues must be considered for the metrics. The first one is the applied (a) value for the suggestible feature. The second one is the suggested (s) value that is suggested by our systems. The last one is the actual class label ($r \in \{\mathbf{s}, \mathbf{f}\}$) of the queried instance. Considering these three issues, we developed four possible evaluation metric cases;

- ($s = a$) & ($r = \mathbf{s}$): Suggested value equals to the applied value and the class label is Successful (n_{as} is the number of instances matching this case).
- ($s \neq a$) & ($r = \mathbf{s}$): Suggested value is not equal to the applied value and the class label is Successful (n_{ds} is the number of instances matching this case).
- ($s = a$) & ($r = \mathbf{f}$): Suggested value equals to the applied value and the class label is Failure (n_{af} is the number of instances matching this case).
- ($s \neq a$) & ($r = \mathbf{f}$): Suggested value is not equal to the applied value and the class label is Failure (n_{df} is the number of instances matching this case).

Performance evaluation metrics are developed based on these four cases and detailed information about them are introduced in the following subsections.

6.5.1 m_p : pessimistic metric

In this metric, we consider only the first case as a good result. In the first case, the class label is successful. Suggestible value that is applied to the query instance and the predicted value that is suggested by our system are the same. So, in this case we are sure that our suggestion is proper for the query instance and this case

is considered as “Good”. Due to the fact that our point of view is pessimistic, all other three cases can be considered as “Bad”. As a result, based on these information the equation for m_p is defined as below:

$$m_p = \frac{n_{as}}{n_{as} + n_{ds} + n_{af} + n_{df}} \quad (6.5)$$

6.5.2 m_o : optimistic metric

In this metric, we consider first, second and fourth cases as a good result. In the first case, we can sure that our suggestion can be suitable for the query instance because it is same as the applied value and the result is successful. In the second case, predicted and applied values are not equal and the class label is successful. If we consider this case optimistically, this does not mean that the suggested value is a bad choice for the query instance. If the suggested value is applied to the query instance, than the result may be successful. So, this case is considered as “Good” like case one. In the third case, we are sure that the suggested value is not proper for the query instance because it is same as the applied value and the class label is failure. Accordingly, this case is considered as “Bad”. In the last case, predicted value is not equal to the suggested one and the class label is failure. We cannot say that, if the suggested value is applied to the patient, than achieving the desired result will be maximized exactly. However, according to the optimistic viewpoint the suggested value can be proper for the query instance and this case also is considered as “Good”. As a result, based on these information the equation for m_o is defined as below:

$$m_o = \frac{n_{as} + n_{ds} + n_{df}}{n_{as} + n_{ds} + n_{af} + n_{df}} \quad (6.6)$$

6.5.3 m_{vo} : validated optimistic metric

In this metric, we consider first case as a good result, third case as a bad result and we ignore second and fourth cases. Since the result is not clear in cases second and third, we ignore them. They do not have any contribution to this metric. So, based on these information the equation for m_{vo} is defined as below:

$$m_{vo} = \frac{n_{as}}{n_{as} + n_{af}} \quad (6.7)$$

6.5.4 m_{vp} : validated pessimistic metric

If we analyse the cases, first one is absolutely true and the third one is absolutely false. For the second case, in the pessimistic view our suggestion is different from the applied value and the result is successful. So, we can say that our system suggests a wrong value and this case can be considered as “Bad”. Since the result is not accurate in case four, this one is ignored in the calculation of this metric. Therefore, based on these information the equation for m_{vp} is defined as below:

$$m_{vp} = \frac{n_{as}}{n_{as} + n_{ds} + n_{af}} \quad (6.8)$$

Table 6.3 illustrates an example of the calculation of the performance evaluation metrics.

Table 6.3: Example of the performance evaluation metrics calculation.

Train No	Applied (a)	Suggested (s)	Class Label (r)	m_p	m_o	m_{vo}	m_{vp}
1	a	a	S	1	1	1	1
2	a	b	S	0	1	-	0
3	a	a	F	0	0	0	0
4	a	b	F	0	1	-	-
5	a	c	S	0	1	-	0
6	a	c	F	0	1	-	-
7	a	a	S	1	1	1	1
8	a	a	F	0	0	0	0
9	a	d	S	0	1	-	0
10	a	a	S	1	1	1	1
Results of the metrics =				3 / 10	8 / 10	3 / 5	3 / 8

Chapter 7

Empirical Evaluation

In this chapter, we present the test results for developed algorithms for both prediction and suggestion. All algorithms are tested using stratified 10-fold cross-validation technique. Stratification guaranties that the ratio of positive instances and negative instances remains through out each fold.

In the next subsections, we give the test results of the prediction and suggestion algorithms. In order to test the ranking algorithms for prediction, the AUC metric is used and to test the classification performance, accuracy is used. Due to the fact that suggestion is a new topic, there is no performance evaluation metric to test them. So, we developed four performance evaluation metrics and the validation of the suggestion algorithms are done by using these metrics.

7.1 Estimation of the Chance of Success

In order to support the theoretical backgrounds of the ranking algorithms with empirical results, we compared all of them. We use two performance evaluation metrics that are AUC and accuracy. The following sections give more detailed information about testing and the evaluation.

7.1.1 Computation of the AUC metric for Prediction

Since the AUC values are used to measure the predictive performance, we use this metric to test our algorithms.

As it is explained in Chapter 4, each ranking algorithm learns a model from a set of training instances. After gathering the model, test instances are given to this model and ranking functions of the algorithms assigns a score value for each test instance. In Figure 7.1, testing the first fold of the dataset is illustrated. Since we use 10-fold cross-validation technique, this step is repeated ten times for each fold as it is shown in Figure 7.2. After completing this step for each fold, all test instances have a score value that are assigned by the ranking algorithms.

The first fold of test

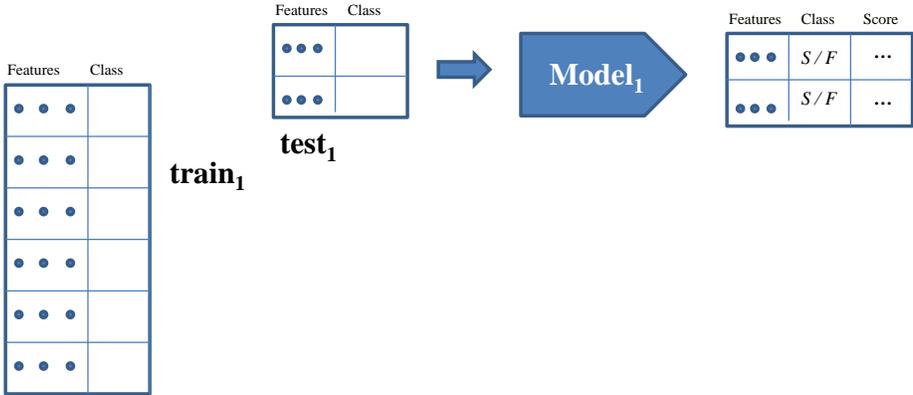


Figure 7.1: First fold for testing instances using AUC.

Now, we have 10-fold cross-validated and scored test instances. In the next step, these instances are integrated and sorted based on their score values and the AUC metric is computed as it is illustrated in Figure 7.3.

A stratified 10-fold cross-validation is employed to calculate AUC values of the ranking algorithms. Figures 7.4, 7.5 and 7.6 represent the test conducted with IVFa, IVFb, and IVFc datasets.

The i_{th} fold of test

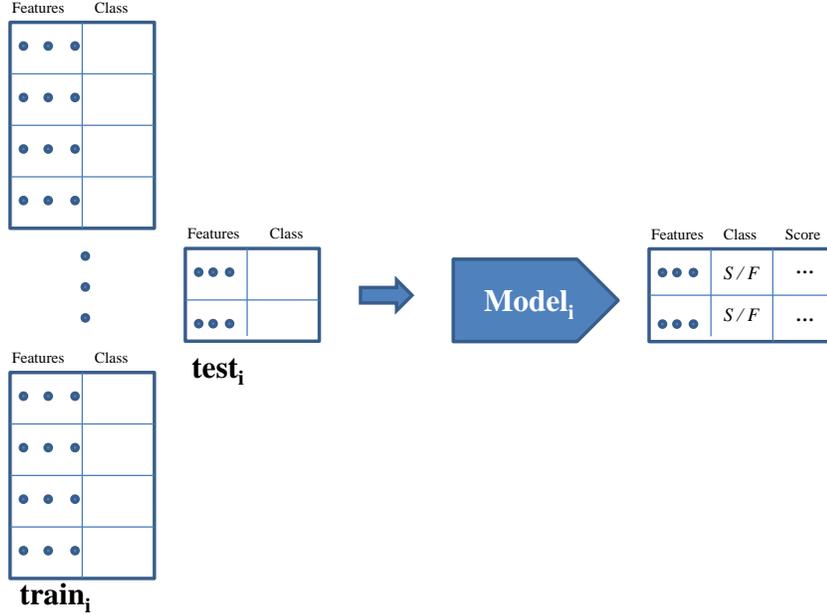


Figure 7.2: i th fold for testing instances using AUC.

As the red line implies, RIMARC used for ranking instances, is more successful than $\text{RI}k\text{NN}$, SVM^{light} and $\text{RI}w\text{NN}$ with 10-fold cross-validation with stratification.

The AUC of RIMARC for IVFa dataset is 0,708. For other algorithms, as they are shown in Table 7.1., the AUC values are as follows: SVM^{light} is 0.578, $\text{RI}k\text{NN}$ is 0.580 for $k = 100$ and $\text{RI}w\text{NN}$ is 0.597 for $k = 100$ as the best case.

The AUC of RIMARC for IVFb dataset is 0,689. For other algorithms, as they are shown in Table 7.1., the AUC values are as follows: SVM^{light} is 0.589, $\text{RI}k\text{NN}$ is 0.610 for $k = 99$ and $\text{RI}w\text{NN}$ is 0.626 for $k = 51$ as the best case.

The AUC of RIMARC for IVFc dataset is 0,986. For other algorithms, as they are shown in Table 7.1., the AUC values are as follows: SVM^{light} is 0.814, $\text{RI}k\text{NN}$ is 0.742 for $k = 89$ and $\text{RI}w\text{NN}$ is 0.752 for $k = 99$ as the best case.

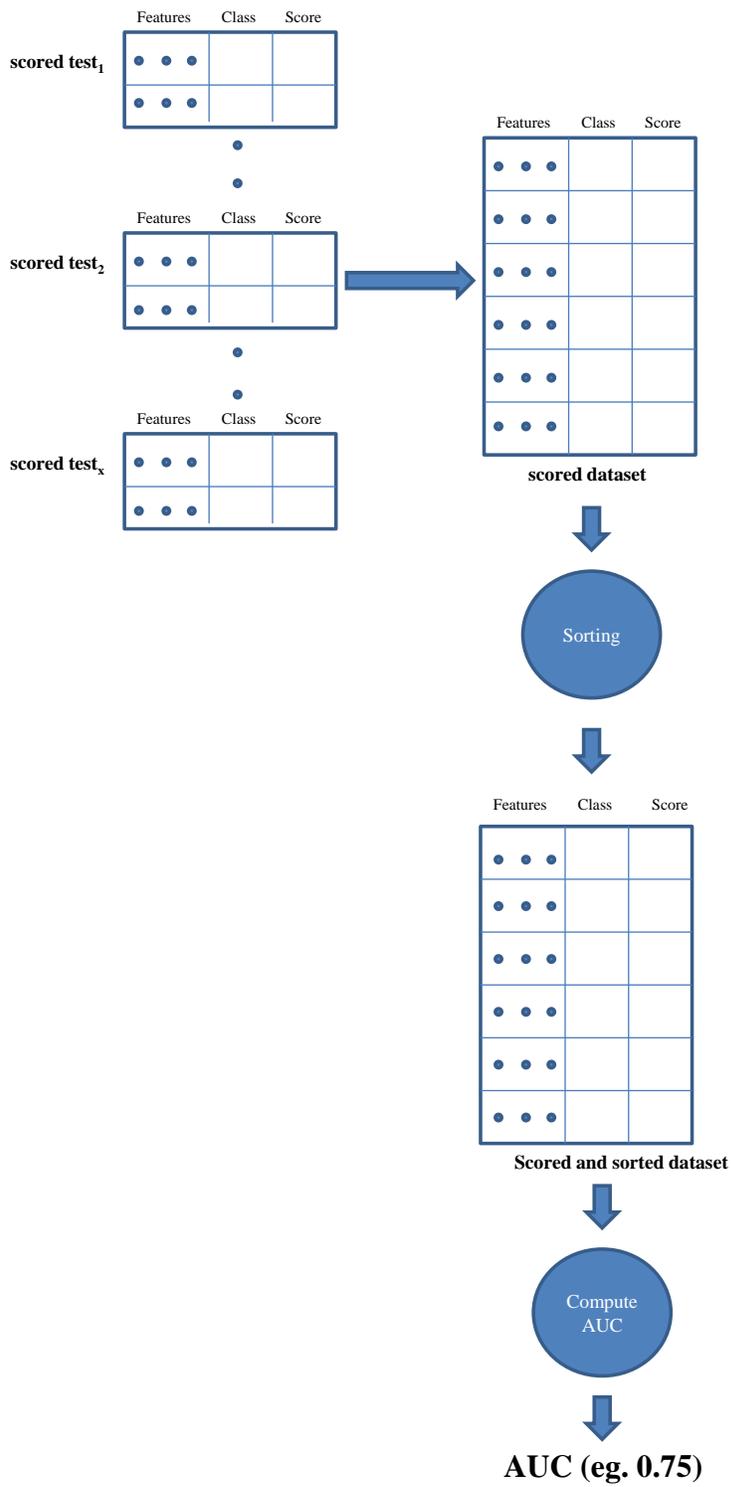


Figure 7.3: Computation of the AUC metric.

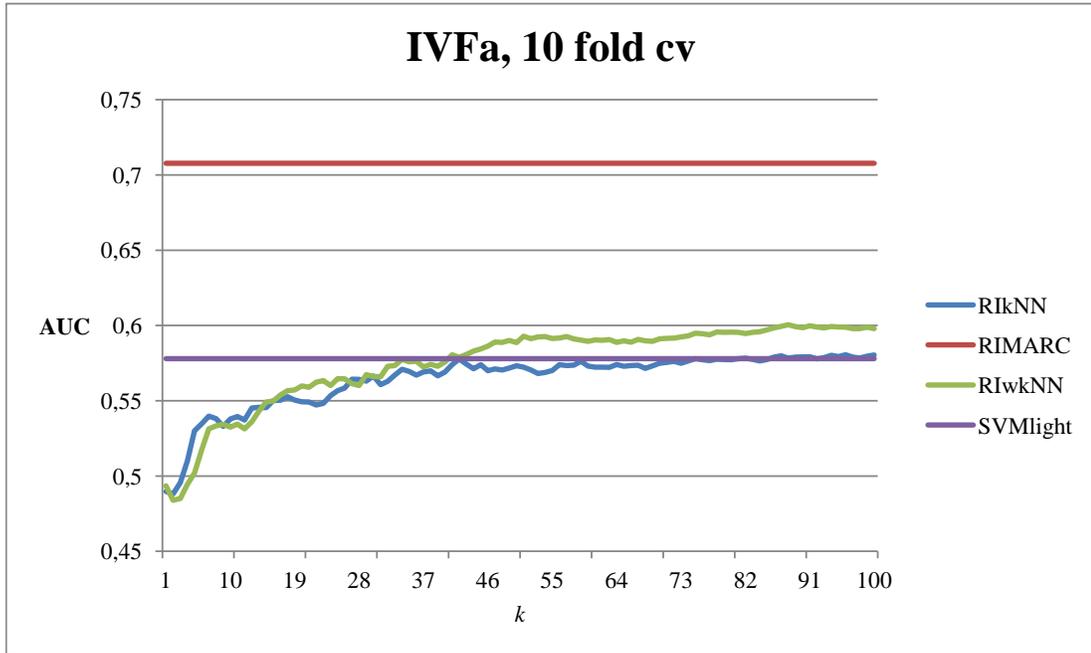


Figure 7.4: Experimental result for dataset IVFa based on AUC.

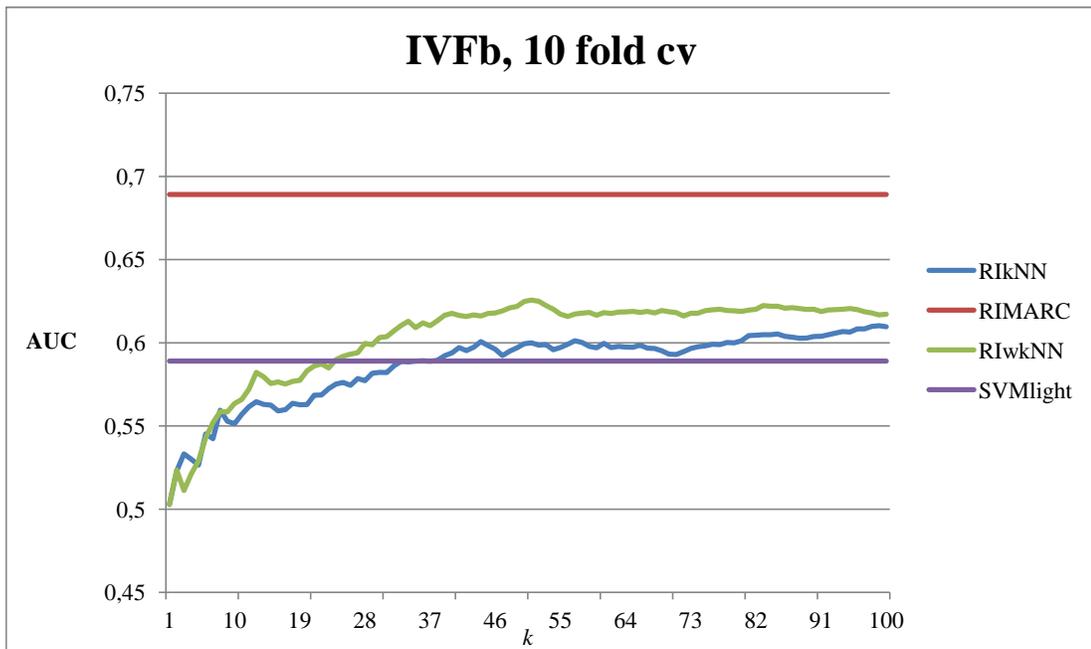


Figure 7.5: Experimental result for dataset IVFb based on AUC.

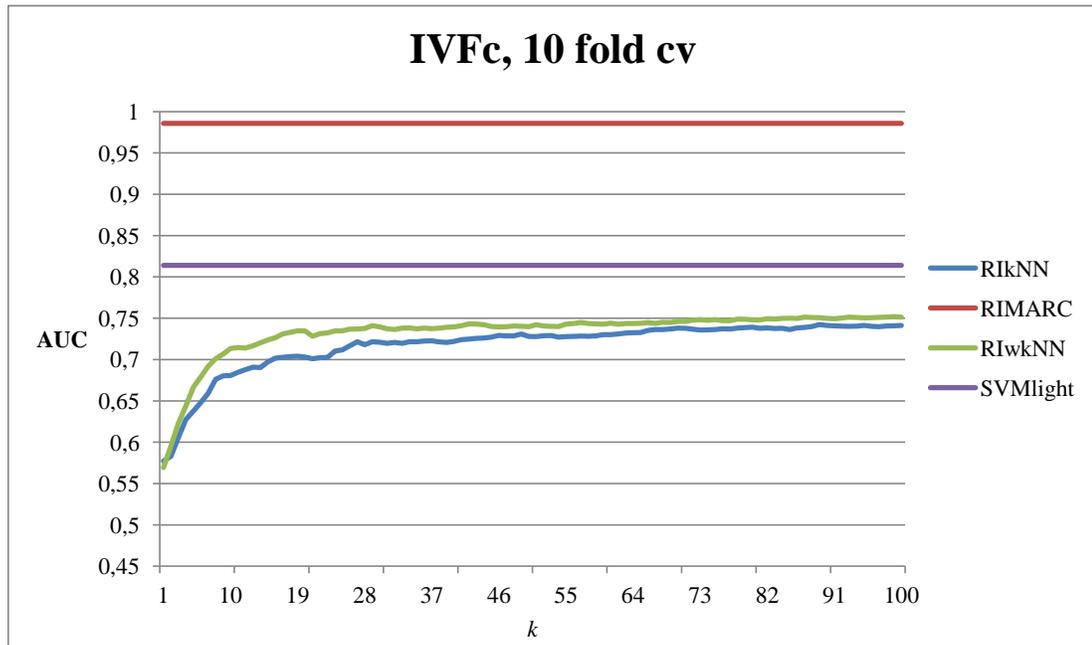


Figure 7.6: Experimental result for dataset IVFc based on AUC.

Table 7.1: AUC values for ranking algorithms for datasets IVFa, IVFb and IVFc.

Dataset	Ranking Algorithm	AUC
IVFa	RIMARC	0,708
	RIwkNN	0,597
	RIkNN	0,580
	SVM ^{light}	0,578
IVFb	RIMARC	0,689
	RIwkNN	0,626
	RIkNN	0,610
	SVM ^{light}	0,589
IVFc	RIMARC	0,986
	SVM ^{light}	0,814
	RIwkNN	0,752
	RIkNN	0,742

According to the prediction performances, RIMARC outperforms other ranking algorithms in all test datasets. In addition to that, it calculates feature

weights and rules that are in a human readable form and easy to interpret.

Based on this information, the main characteristics of RIMARC can be summarized as follows: It achieves high AUC values. It is a non-parametric algorithm and it does not require tuning of parameters in order to achieve best performance. It is robust to missing feature values and the ranking function is in a human readable form that can be easily understood by domain experts, listing the effects of features based on feature weights and how their particular values affect the ranking.

7.1.2 Computation of the Accuracy for Classification

Developed ranking algorithms were modified as a classification algorithms to predict the class label of the queried instance as Successful or Failure.

The working principle of the classification algorithms that are determined in this thesis is as follows: First of all, these algorithms learn a model from a training dataset similar as the ranking algorithms. After that, using this model, score values are assigned to each training instance and scored training instances are sorted as it is illustrated in Figure 7.7. In ranking algorithms, score values are assigned to test instances and AUC metric is computed. However, for accuracy calculation, both training and testing instances are scored.

Since our dataset belongs to IVF patients, we try to give an easily comprehensible result. From the patients point of view, if a doctor says that the chance of success is %x to the patient, the patient understands that x people over 100 people who have infertility problem have a baby after the IVF treatment. This kind of result makes sense for the patient. So, when a test instance comes, we find its location among training instances based on their score values. After that, we select $k = 100$ nearest neighbours using score values and find instances that are successful (have a class label \mathbf{s}). As it is shown in Figure 7.8, n_s represents the number of instances that are classified as \mathbf{s} among k instances. Computing the Equation 7.1 that equals to accuracy computation, we find the chance of success

of the test instance.

$$\text{chance of success} = \frac{n_s}{k} \quad (7.1)$$

In order to classify the test instance, Equation 7.2 is used. If n_s/k is greater than or equal to 0,5, our algorithms classify these test instances as **s**, otherwise **f**.

$$\text{predicted class} = \begin{cases} \mathbf{s} & \frac{n_s}{k} \geq 0,5 \\ \mathbf{f} & \text{else} \end{cases} \quad (7.2)$$

The comparison of these three algorithms are shown in Figure 7.9, 7.10 and 7.11 for datasets IVFa, IVFb and IVFc.

The accuracy of $RIkNN$ and $RIwkNN$ for IVFa dataset is 0.794 for $k = 100$ as the best case. For other algorithms, as they are shown in Table 7.2, the accuracy values are as follows: RIMARC is 0,792 and SVM^{light} is 0.669.

The accuracy of $RIwkNN$ is for IVFb dataset is 0.795 for $k = 34$ as the best case. For other algorithms, as they are shown in Table 7.2, the accuracy values are as follows: $RIwkNN$ is 0.794 for $k = 100$ as the best case, RIMARC is 0,782 and SVM^{light} is 0.687.

The accuracy of RIMARC for IVFc dataset is 0,964. For other algorithms, as they are shown in Table 7.2, the accuracy values are as follows: SVM^{light} is 0.808, $RIwkNN$ is 0.786 for $k = 73$ and $RIkNN$ is 0.784 for $k = 71$ as the best case.

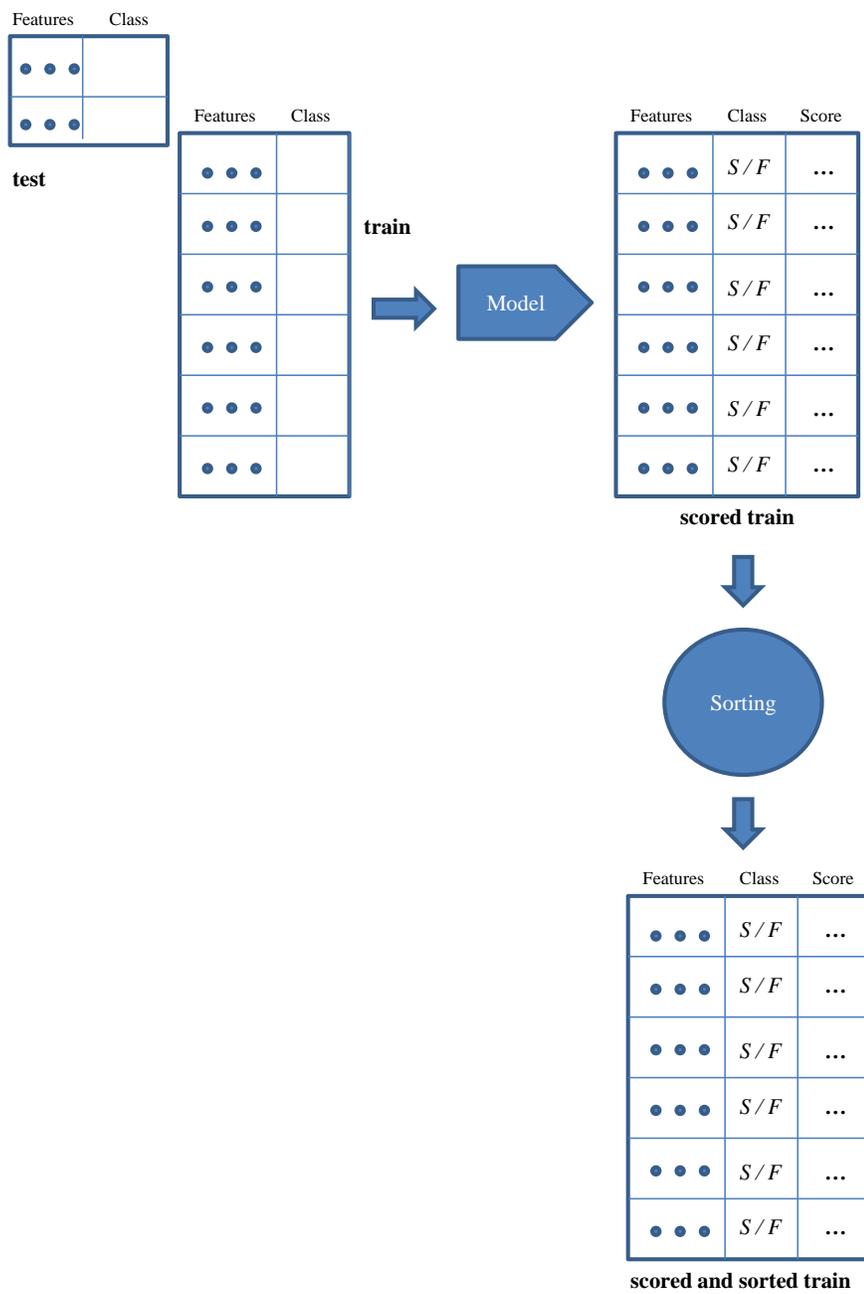


Figure 7.7: Creating sorted and scored training dataset.

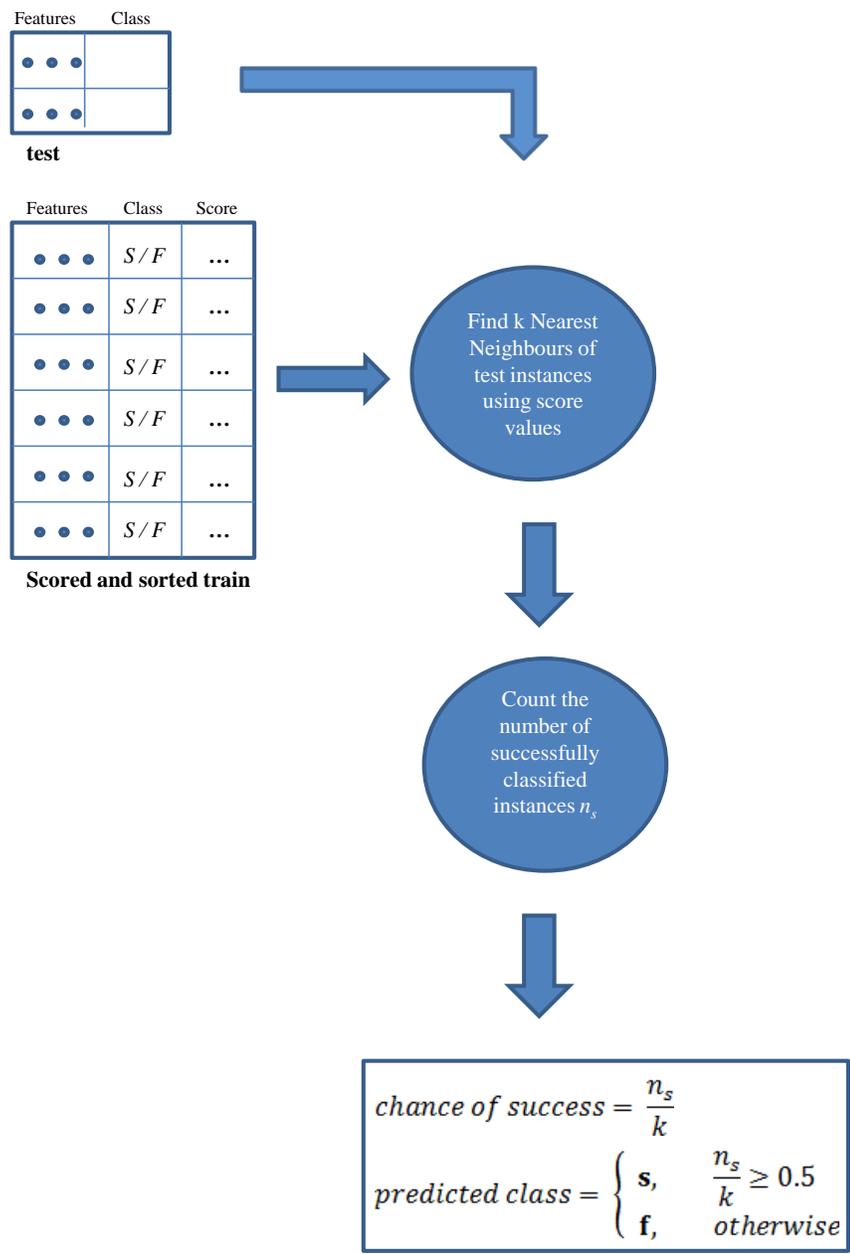


Figure 7.8: Classification of the test instances.

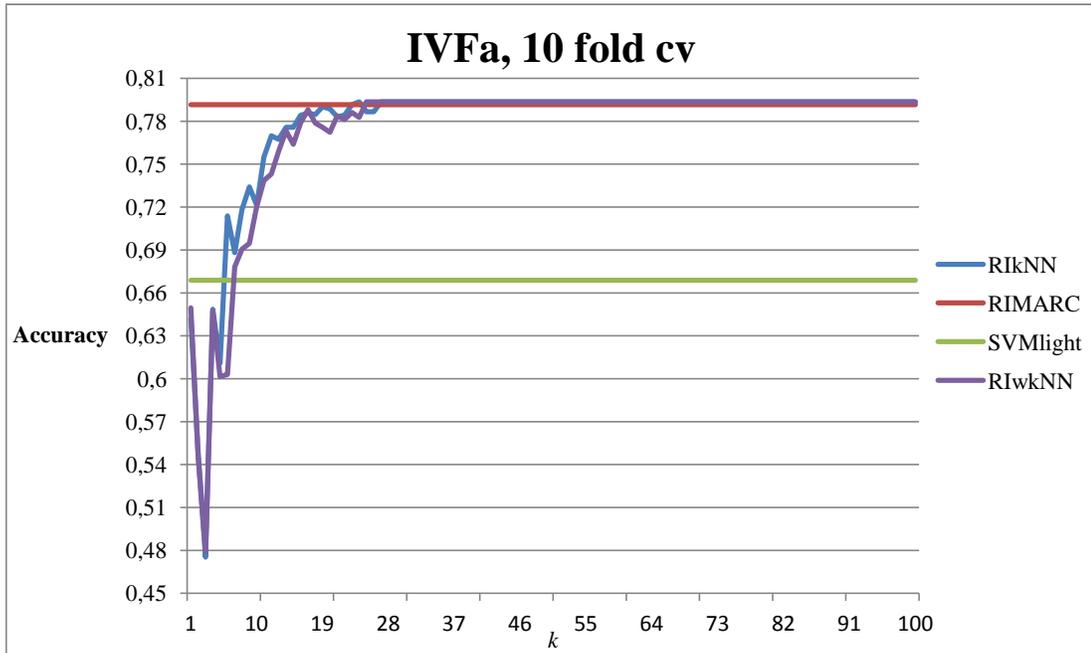


Figure 7.9: Experimental result for dataset IVFa based on accuracy.

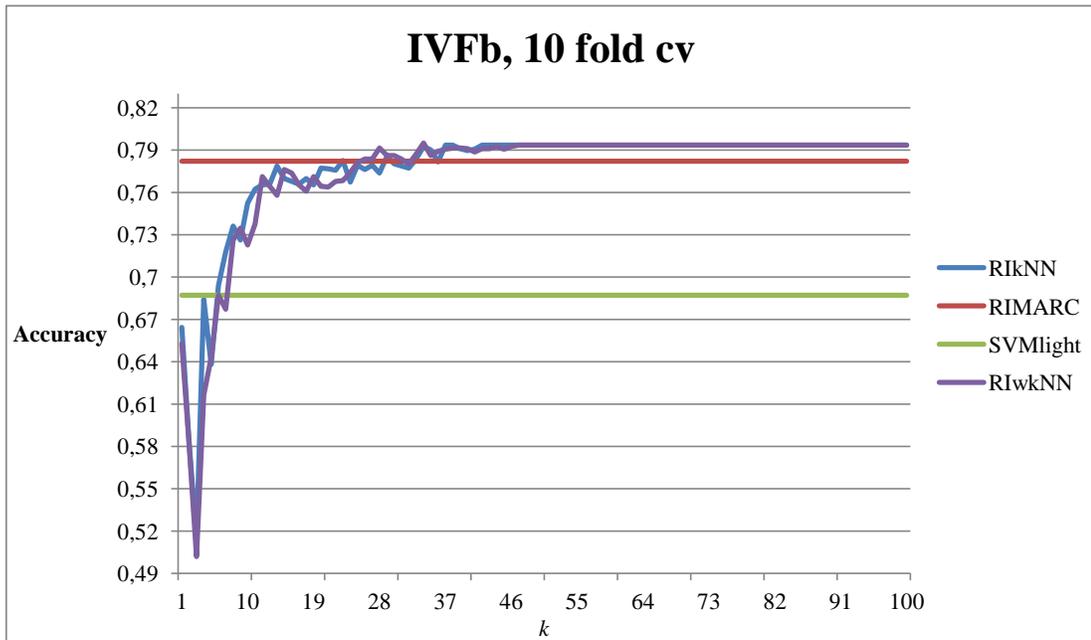


Figure 7.10: Experimental result for dataset IVFb based on accuracy.

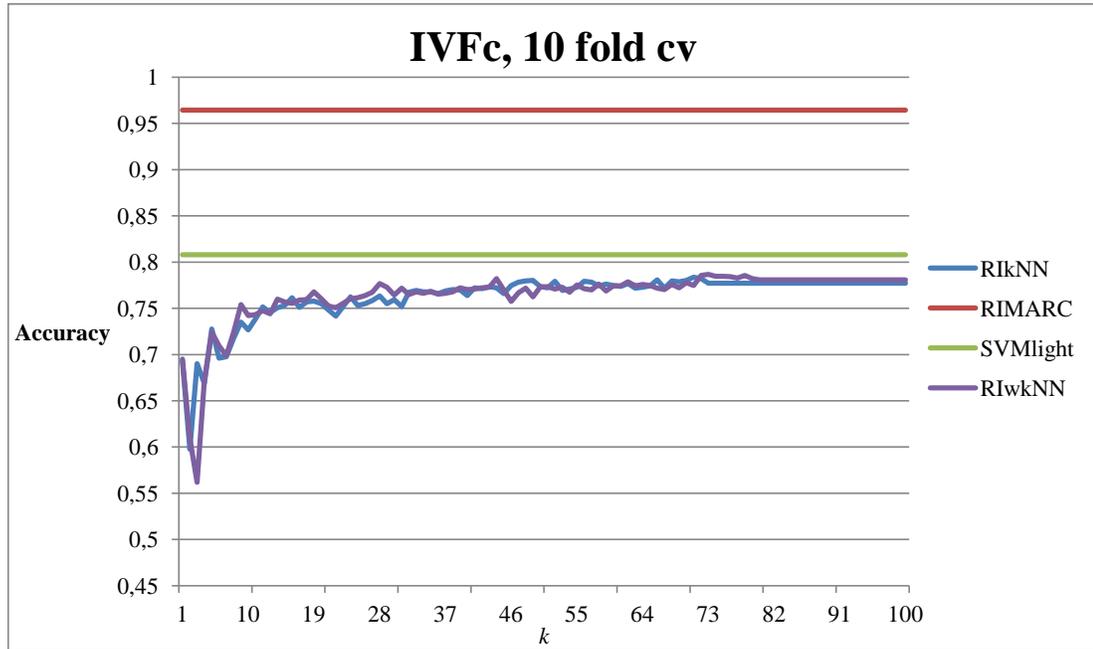


Figure 7.11: Experimental result for dataset IVFc based on accuracy.

Table 7.2: Accuracy values for ranking algorithm for datasets IVFa, IVFb and IVFc

Dataset	Classification Algorithm	Accuracy
IVFa	RIkNN	0,794
	RIwkNN	0,794
	RIMARC	0,792
	SVM ^{light}	0,669
IVFb	RIwkNN	0,795
	RIkNN	0,794
	RIMARC	0,782
	SVM ^{light}	0,687
IVFc	RIMARC	0,964
	SVM ^{light}	0,808
	RIwkNN	0,786
	RIkNN	0,784

According to the experimental results for classification, the performance of

the RIMARC algorithm and $RIkNN$ is nearly the same for datasets IVFa and IVFb. Since $RIkNN$ is a parametric algorithm, achieving high accuracy values can be possible by using different k values. On the other hand, RIMARC is a non parametric method and it is not a classification algorithm. For IVFa and IVFb datasets, the success of $RIkNN$ does not make sense because if we classify all instances as Failure than we get 80% accuracy. In IVFc, RIMARC outperforms other algorithms and the accuracy rate is higher than the default one that is 80%. For all datasets, SVM^{light} has the weakest performance in classification.

7.2 Suggestion of the Best Treatment Protocol

In the literature, there is not any suggestion system so evaluating the correctness of the developed algorithms is open to the discussion. In this thesis, we proposed four performance evaluation metrics to test the correctness of the algorithms.

A stratified 10-fold cross-validation is employed to calculate performance evaluation metrics. We calculated pessimistic metric, optimistic metric, validated optimistic metric and validated pessimistic metric based on performance evaluation cases that are described in Chapter 6 for the algorithms NSNS, $kNNS$ and DTS. In IVF dataset, we have two suggestible features that are “Ovulation_Induction_Protocol” and “Ovulation_Induction_Dose_Protocol”. The results are illustrated in Figures between 7.12 and 7.19.

In the calculation of m_p , we consider the instances with cases where the applied and suggested treatment protocol is equal and the result of the treatment is Successful among 2020 instances. For the first suggestible feature that is “Ovulation_Induction_Protocol”, in NSNS 169 number of instances matches this case. In $kNNS$, 3 number of instances are matching this case for $95 < k \leq 100$ as the worst case and 78 number of instances are matching this case for $k = 2$ as the best case. According to the DTS, 8 number of instances match this case among 2020 instances. Lastly, in wNSNS, 171 number of instances match the case. For the second suggestible feature that is “Ovulation_Induction_Dose_Protocol”, in

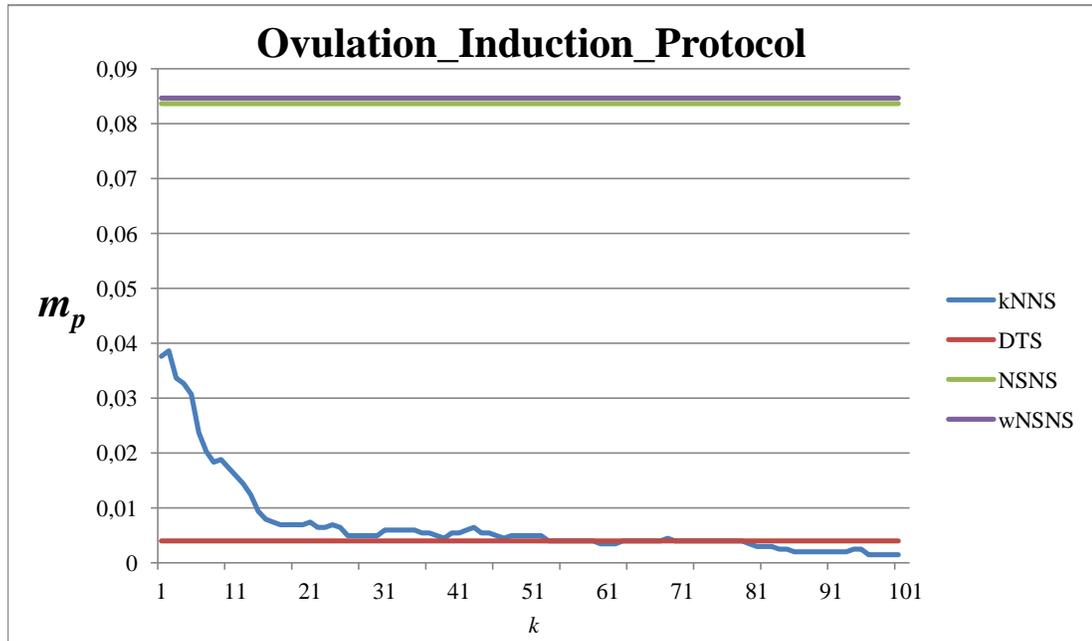


Figure 7.12: Experimental result for “Ovulation_Induction_Protocol” based on pessimistic metric (m_p).

NSNS 178 number of instances matches this case. In k NNS, 51 number of instances are matching this case for $k = 2$ as the worst case and 112 number of instances are matching this case for $k = 5$ as the best case. According to the DTS, 151 number of instances match this case among 2020 instances. Lastly, in wNSNS, 186 number of instances match the case.

In the calculation of m_o , we consider the instances with all cases except where the applied and suggested treatment protocol is equal and the result of the treatment is Failure among 2020 instances. Since it is an optimistic view, the result of other three cases can be Successful. For the first suggestible feature that is “Ovulation_Induction_Protocol”, in NSNS 1350 number of instances matches this case. In k NNS, 1657 number of instances are matching this case for $k = 1$ as the worst case and 1998 number of instances are matching this case for $97 < k \leq 100$ as the best case. According to the DTS, 2006 number of instances match this case among 2020 instances. Lastly, in wNSNS, 1364 number of instances match the case. For the second suggestible feature that is “Ovulation_Induction_Dose_Protocol”, in NSNS 1253 number of instances matches this case. In k NNS, 1545 number of

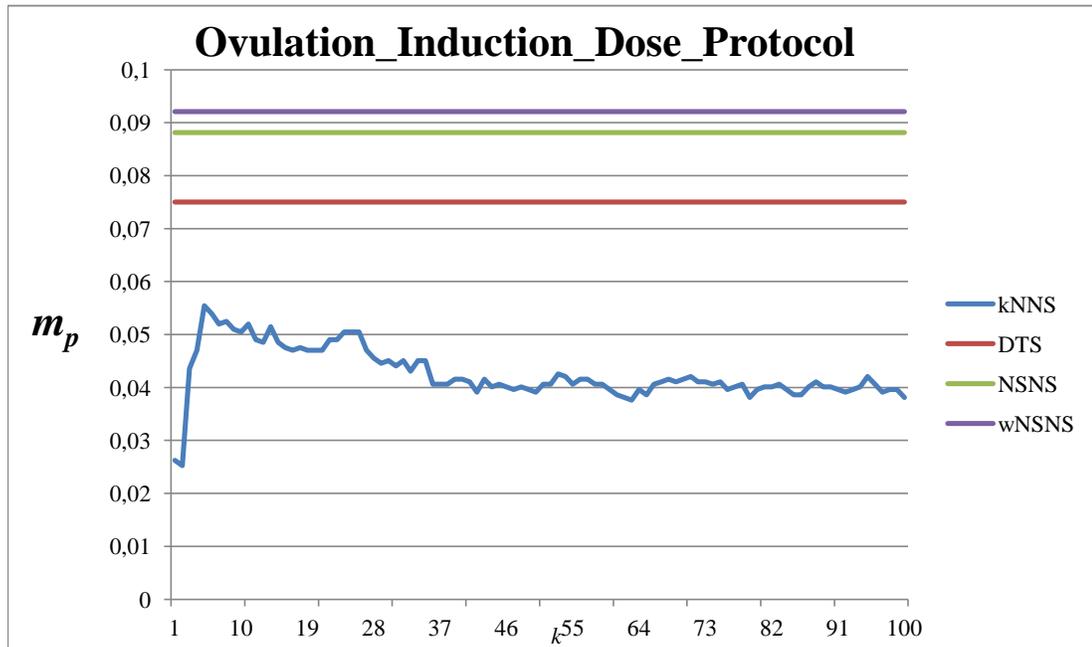


Figure 7.13: Experimental result for “Ovulation_Induction_Dose_Protocol” based on pessimistic metric (m_p).

instances are matching this case for $k = 5$ as the worst case and 1755 number of instances are matching this case for $k = 2$ as the best case. According to the DTS, 1614 number of instances match this case among 2020 instances. Lastly, in wNSNS, 1182 number of instances match the case.

In the calculation of m_{vo} , we consider the instances with the case where the applied and suggested treatment protocol is equal and the result of the treatment is Successful. Since it is a validated optimistic view, we have to consider the case whose result is known, accurate and successful. Cases that have uncertain results are ignored. For the first suggestible feature that is “Ovulation_Induction_Protocol”, in NSNS 169 number of instances matches this case among 839 instances. In k NNS, 3 number of instances are matching this case among 26 instances for $95 < k < 99$ as the worst case and 62 number of instances are matching this case for $k = 5$ among 265 instances as the best case. According to the DTS, 8 number of instances match this case among 22 instances. Lastly, in wNSNS, 171 number of instances match the case among 827 instances. For the second suggestible feature that is “Ovulation_Induction_Dose_Protocol”, in

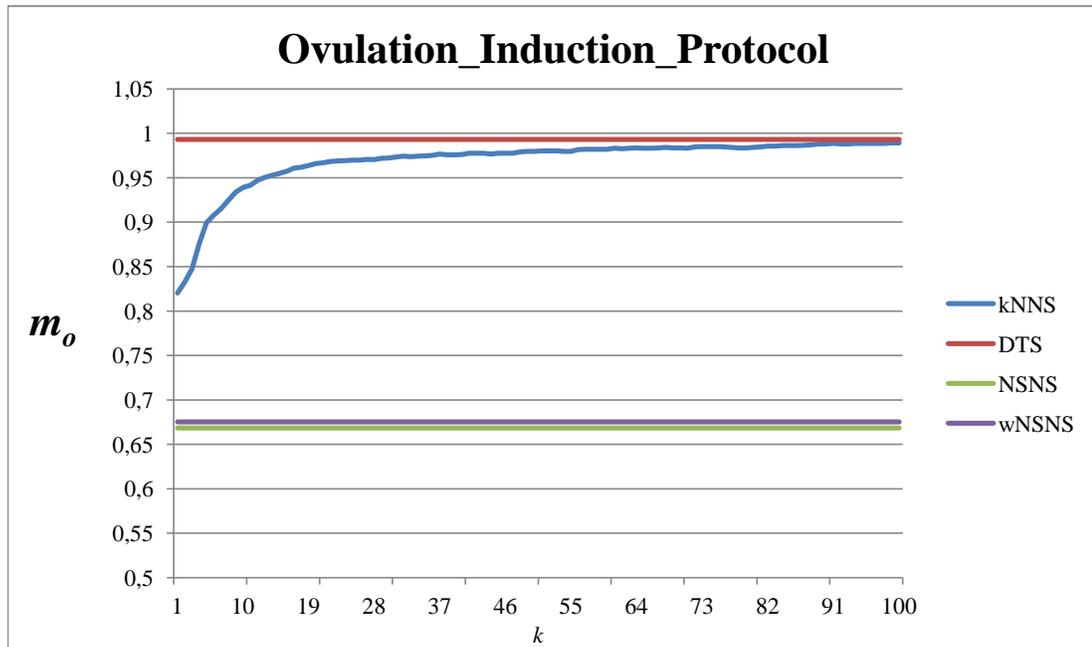


Figure 7.14: Experimental result for “Ovulation_Induction_Protocol” based on optimistic metric (m_o).

NSNS 178 number of instances matches this case among 945 instances. In k NNS, 53 number of instances are matching this case among 331 instances for $k = 1$ as the worst case and 102 number of instances are matching this case for $24 < k < 27$ among 426 instances as the best case. According to the DTS, 151 number of instances match this case among 557 instances. Lastly, in wNSNS, 186 number of instances match the case among 1024 instances.

In the calculation of m_{vp} , we consider the instances with the case where the applied and suggested treatment protocol is equal and the result of the treatment is Successful. Since it is a validated pessimistic view, we have to consider the case whose result is known. Cases that have uncertain results are ignored.

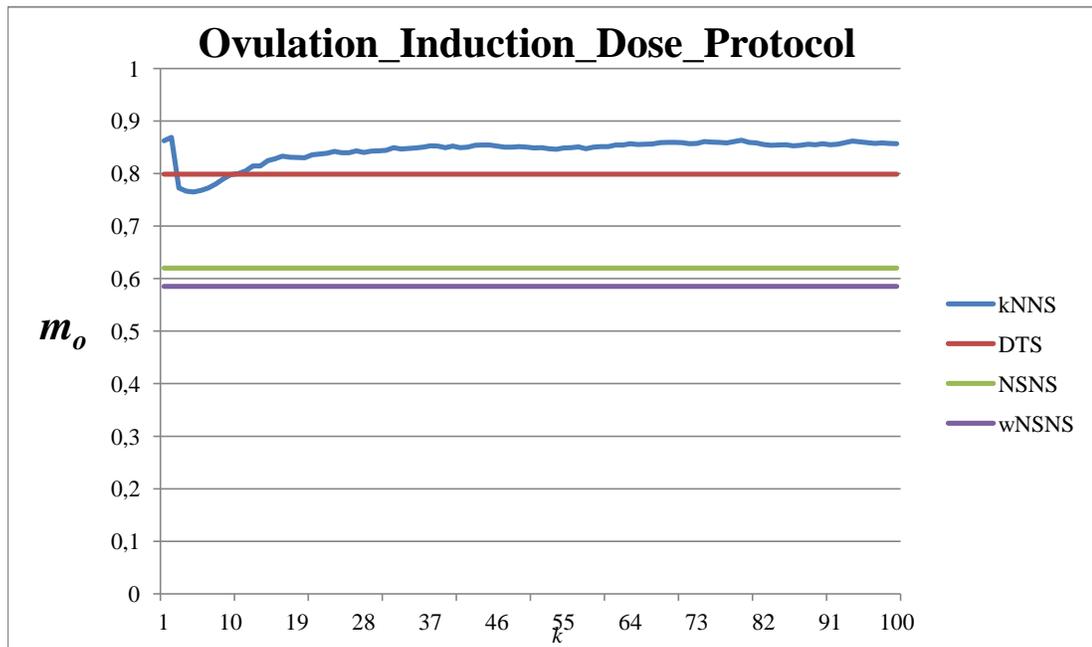


Figure 7.15: Experimental result for “Ovulation_Induction_Dose_Protocol” based on optimistic metric (m_o).

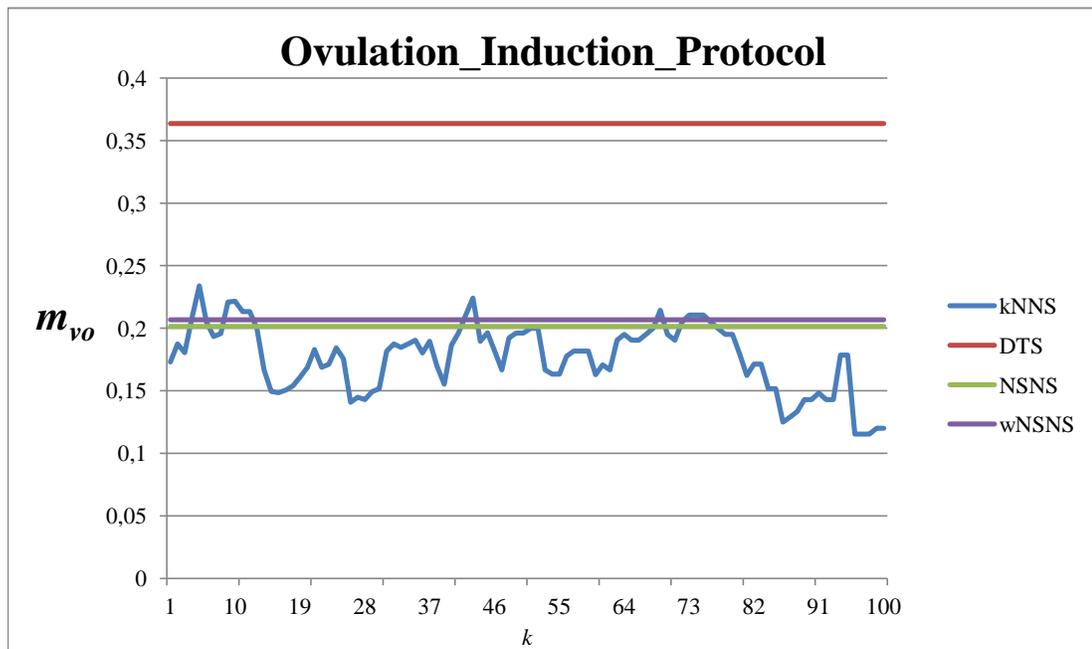


Figure 7.16: Experimental result for “Ovulation_Induction_Protocol” based on validated optimistic metric (m_{vo}).

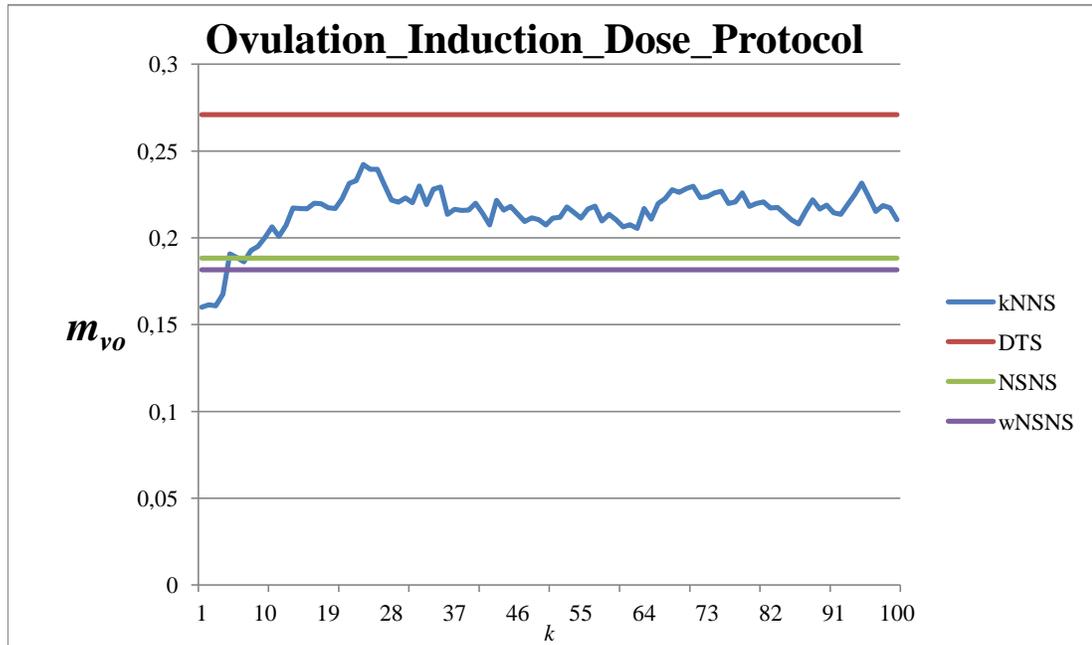


Figure 7.17: Experimental result for “Ovulation_Induction_Dose_Protocol” based on validated optimistic metric (m_{vo}).

For the first suggestible feature that is “Ovulation_Induction_Protocol”, in NSNS 169 number of instances matches this case among 1087 instances. In k NNS, 3 number of instances are matching this case among 440 instances for $95 < k < 99$ as the worst case and 78 number of instances are matching this case for $k = 2$ among 755 instances as the best case. According to the DTS, 8 number of instances match this case among 431 instances. Lastly, in wNSNS, 171 number of instances match the case among 1073 instances. For the second suggestible feature that is “Ovulation_Induction_Dose_Protocol”, in NSNS 178 number of instances matches this case among 1184 instances. In k NNS, 51 number of instances are matching this case among 682 instances for $k = 2$ as the worst case and 102 number of instances are matching this case for $k = 24$ among 736 instances as the best case. According to the DTS, 151 number of instances match this case among 823 instances. Lastly, in wNSNS, 186 number of instances match the case among 1255 instances.

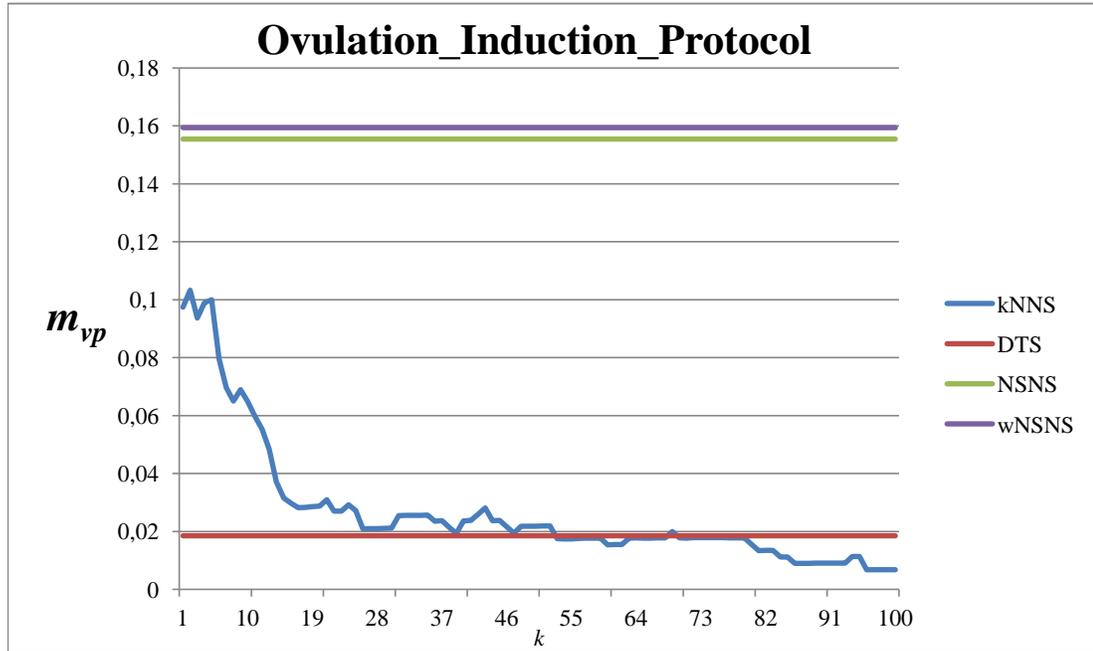


Figure 7.18: Experimental result for “Ovulation_Induction_Protocol” based on validated pessimistic metric (m_{vp}).

Based on the matching cases and considered number of instances for each performance evaluation case, metrics are computed and the results are given in Table 7.3 and 7.4.

Table 7.3: Results of performance evaluation metrics for suggestible feature “Ovulation_Induction_Protocol”

Suggestion Algorithm	m_p	m_o	m_{vo}	m_{vp}
kNNBS	0,039	0,989	0,234	0,103
DTBS	0,004	0,993	0,364	0,019
NSBS	0,084	0,668	0,201	0,155
wNSBS	0,085	0,675	0,207	0,159

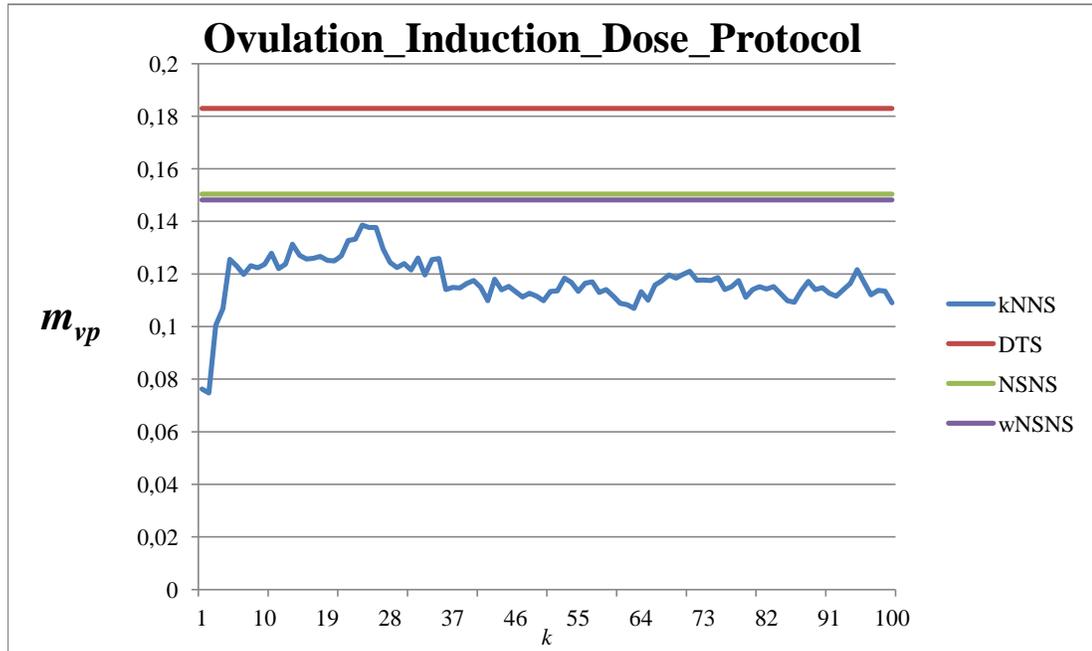


Figure 7.19: Experimental result for “Ovulation_Induction_Dose_Protocol” based on validated pessimistic metric (m_{vp}).

Table 7.4: Results of performance evaluation metrics for suggestible feature “Ovulation_Induction_Dose_Protocol”

Suggestion Algorithm	m_p	m_o	m_{vo}	m_{vp}
k NNBS	0,055	0,869	0,242	0,139
DTBS	0,075	0,799	0,271	0,183
NSBS	0,088	0,620	0,188	0,150
wNSBS	0,092	0,585	0,182	0,148

According to the performance evaluation metrics, DTS outperforms other suggestion algorithms almost in each metric.

Chapter 8

Risk Analysis and Suggestion for Treatment (RAST)

In this chapter, a decision support system namely, RAST is given. RAST is a web based system and it is currently used in Etlik Züübeyde Hanım Woman's Health and Teaching Hospital.

8.1 RAST Introduction

RAST is a decision support system that is composed of a DBMS (MySQL) and a web-server (IIS). It is implemented in the PHP programming language. Information about patients is stored as records in a table in the database. The dataset for experimental results includes 2020 patient records. The users continuously enter patient records to the system and currently we have 2124 patient records. Each patient record is composed of 163 attributes. The attributes can be grouped into the following categories:

- Personal information: Name, Address, Phone, file_no, Year...
- Clinical parameters: Female_Age, Cycle_No, Weight, PCOS, D3_FSH,

Method_Sperm_Retrieval, Sperm_Count, Sperm_Motility, Male_FSH. . .

- Treatment related parameters: Ovulation_Induction_Protocol, Supressed_FSH, Ovulation_Induction_Dose, Follicle_Count_15_17mm, OPU_Procedure. . .
- Embryo transfer parameters: Assisted_Hatching, ET_Progesteron, Embryo_Transfer_Procedure, Embryo_Transfer_Type, Day_Embryo_Transfer, Freezing_Embryo_Procedure. . .
- Result: BHCG results, IVF_Outcome, Clinical_Pregnancy_Outcome, Ongoing_Pregnancy_Outcome. . .

Each attribute has important characteristics, such as data type, distinctive, predictive, required, querable, predicted, display color and weight. Attribute data types are numeric, ordinal, categoric and string. Some examples for each data type are given below:

- Numeric attributes: Female_Age, Cycle_No, Result_BHCG. . .
- Ordinal attributes: Ovulation_Induction_Dose_Initial, HCG_Dose. . .
- Categoric attributes: Embryo_Transfer_Procedure, OPU_Procedure, Ovulation_Induction_Protocol. . .
- String attributes: Name, Address, Notes. . .

Once the type of a variable is set by the administrator, they cannot be modified later. If a variable is set as Distinctive, the value of this attribute is shown in the search results to find the searched record easily. Predictive attributes are necessary for similarity computations and only categorical, numerical and ordinal variables can be set as Predictive. Predicted feature is used as the class feature in machine learning algorithms. For a Querable feature, the best value for this feature for the optimum result can be queried by the suggestion system. The value of the Required feature must be a non-missing value for querying the feature in higher positions by the suggestion system. Some examples of the Distinctive, Predictive, Predicted, Querable and Required attributes are given below:

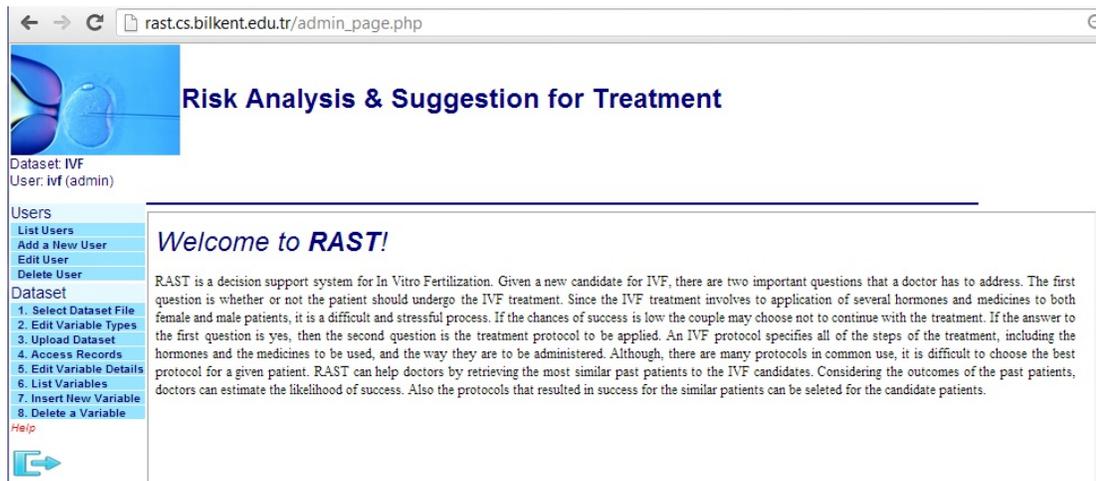


Figure 8.1: Administrator interface to RAST.

- Distinctive: Female_Name, Age, File_No, Cycle_No. . .
- Predictive: IVF_Outcome, Result_BHCG, Ovulation_Induction_Protocol. . .
- Queriable: Ovulation_Induction_Protocol, Ovulation_Induction_Dose_Protocol. . .
- Required: Female_Age, Cycle_No. . .
- Predicted: Result. . .

Color attribute is used to group attributes that have similar medical characteristics. The characteristics, except the type, of the variables can be modified by the administrator, see Figure 8.1 and Figure 8.2. Even after the dataset is created, the administrator can insert new variables if needed. The position of a variable in the data entry page is important. Since it is easier for humans to follow a flow from the top of the page to the bottom, variables whose values are determined earlier in time should be on top of the page, while the ones related with the results should be at the bottom. The position of a variable on the page is set by the attribute called PosGUI. This also allows variables that are common to be placed close to each other on the data entry page.

For IVF treatment, doctors generally make their decisions based on their past experiences. When a new patient couple applied to the clinic, the doctors consider

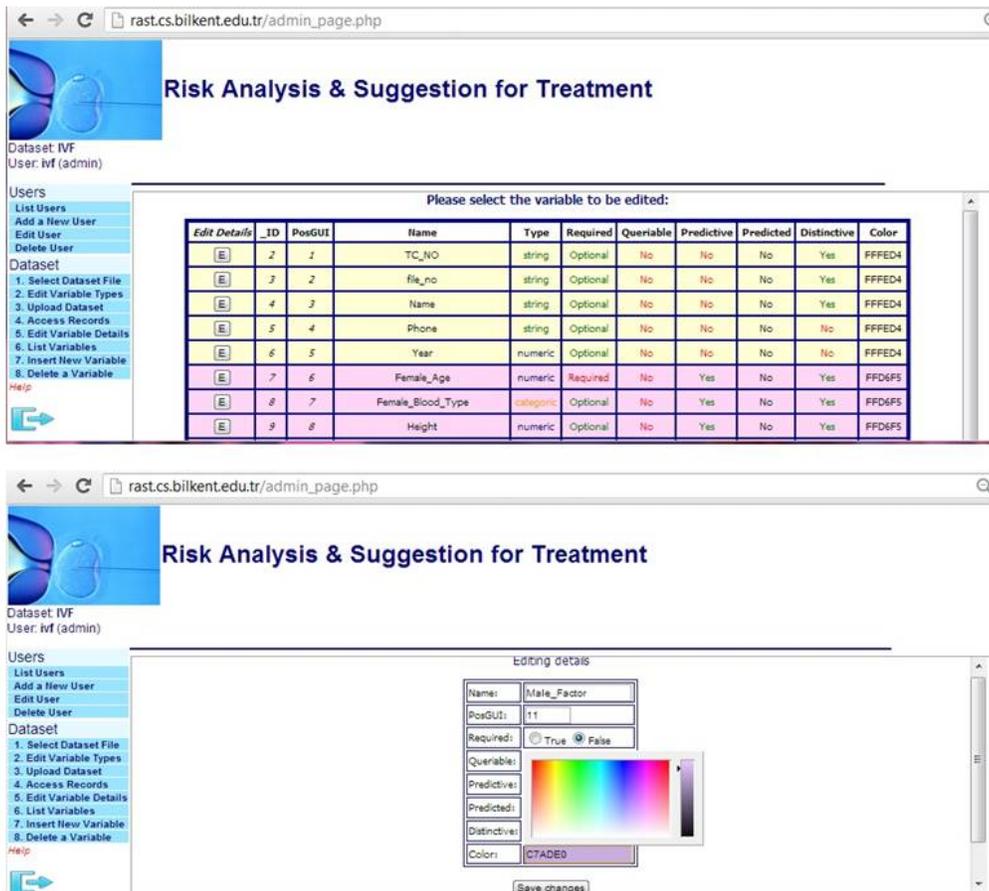


Figure 8.2: Editing variable details.

the past couples that are the most similar to the new one. However, remembering all patient records is a very difficult task for humans. In order to help doctors in chance estimation and deciding on the best treatment protocol for IVF, the RAST system was developed.

8.2 Ensuring the Data Correctness

Ensuring the correctness of the data is essential for a decision support systems that makes inferences from the past experiences. RAST requires that the values of categorical or ordinal variables to be selected from a list of valid values. However, such a list is not possible for numerical variables. We noticed that the source of

incorrect data entry is due to wrong assumption in the unit of the measurement for numerical variables; e.g., entering 1.65, assuming meters, for the height, while cm is the expected. RAST tries to guarantee the correct entry of data using boundary values that are `minAccept`, `maxAccept`, `minExpect`, `maxExpect` for attributes. Lower values than `minAccept` and higher values than `maxAccept` are not allowed to be inserted into the system. If a numeric value is between `minExpect` and `maxExpect` range, it means that data is expected to be true and can be inserted into the system. However, if a numeric value is between `minAccept`-`minExpect` and `maxExpect`-`maxAccept`, system asks for the user approval.

8.3 User Interface

User interfaces that are designed for clinical decision support systems can introduce new forms of error. Interface problems have the potential to contribute to adverse medical events. In order to overcome problems, user interfaces should be designed properly [72], [73].

Actions that any user with admin privileges can be grouped under two categories, namely user operations and dataset operations. In the user operations, admin can list, edit, delete and insert new users into the system. In the database operations, there exist two options for entering the data to the database. In the first option, a tab separated text file can be loaded into the system by using the upload interface under the “Select Dataset File” action. The first line of the file has to consist of tab separated variable names. When the “Select Dataset File” is clicked, only the variable names are loaded into the system and their types are set to unknown as default. After that, “Edit Variable Types” page is opened automatically. The system accepts four different variable types that are numeric, categoric, ordinal and string. While setting variable types, for each numeric value, admin has to set the `minAccept`, `minExpect`, `maxExpect` and `maxAccept` values. For each categoric and ordinal variable, new tables are created in the database and each new value for these variables is inserted into their corresponding tables. Having registered the data types of the variables, the data containing the records

of past patients, can be loaded by through the “Load Dataset” action. After loading the dataset, the admin can edit variable characteristics such as position, color etc., under “Edit Variable Details” action.

As the second option for data entrance, admin enters variable characteristics by inserting them manually under the “Insert New Variable” action. Through the “Delete a Variable” action, all variables in the system are listed and admin can choose a variable to be deleted.

Having the dataset loaded into the database, admin can list all or desired patient records using the “Access Records” action. Search operation is based on the values of the Distinctive variables. Search values for categoric and ordinal variables are selected from a drop-down list. Search values for the string variable can start with the “%” character to find similar records. The first characters of a numerical variable can be relational operators, such as “=” or “>”. If none of the distinctive variables is given a search key, all patient records are listed. An example result of this search operation is showed in Figure 8.3.

In addition to record searching, new patient information can be inserted into the system manually. When “Add” button is clicked, all variables are listed with no information and admin can fill them and save the record. Moreover, using “Download these matching records” button, matching patient information is downloaded by tab separated values into intended location on computers.

Dataset: IVF
User: ivf (admin)

Users
List Users
Add a New User
Edit User
Delete User

Dataset
1. Select Dataset File
2. Edit Variable Types
3. Upload Dataset
4. Access Records
5. Edit Variable Details
6. List Variables
7. Insert New Variable
8. Delete a Variable

Help

Female_Age: <22
Female_Blood_Type: A Rh+
Height:
Weight: B Rh+
Endometriosis: A Rh+
Cycle_No: O Rh+
embryocryo: AB Rh+
D3_FSH: O Rh-
Male_Blood_Type: A Rh-
TESE_Outcome: B Rh-
OPU_Procedure: AB Rh-
OPU_Date:
Date_BHCG:
IVF_Outcome:
Result:
(% symbol can be used as a wild card)
Search

↓ Matching results

Dataset: IVF
User: ivf (admin)

Users
List Users
Add a New User
Edit User
Delete User

Dataset
1. Select Dataset File
2. Edit Variable Types
3. Upload Dataset
4. Access Records
5. Edit Variable Details
6. List Variables
7. Insert New Variable
8. Delete a Variable

Help

Matching Records

Actions	ID	TC_NO	file no	Name	Female Age	Female Blood Type	Height	Weight	Endometriosis	Cycle No
E D N P S	594		558		19	A Rh+	167	55	yok	1
E D N P S	1593		165		20	A Rh+	158	72	yok	2
E D N P S	1594		165		21	A Rh+	158	72	yok	3
E D N P S	1753		62		21	A Rh+	157	60	yok	1
E D N P S	1865		1336		20	A Rh+	156	60	yok	1

Figure 8.3: Searching for past cases and list of matching records.

As it is seen in Figure 8.3, there are five buttons under the “Actions”. These buttons are “Edit”, “Delete”, “Neighbour”, “Predict” and “Suggest”. Using

“Edit” button, admin can change patient information. If “Edit” button is clicked then all variables and their information for selected patient is listed and necessary changes can be done. Also, using “Delete” button, all information related to selected patient is deleted from the system. Other action button is “Neighbour”. Using this button, similar records to the current patient are listed based on their similarity values in decreasing order as it is shown in Figure 8.4. Other action button is “Predict” that is used for estimating the chance of success of the IVF treatment for the selected patient. Figure 8.5 illustrates the chance of the patient whose ID is 1593. Chance value is calculated using the RIMARC algorithm. As it is shown that, the chance of this patient equals to 68%. Classification versions of our ranking algorithms say that if the chance is greater than or equal to 0.5, then we classify this instance as Successful. So, according to this, the RIMARC algorithm classifies this patient as Successful. When we compare this result to the actual one, we see that these two values are equal which can be considered as the proof of the correctness of our algorithm. Last action button is “Suggest” that is used for suggesting the best treatment protocol of the selected patient. In Figure 8.6, suggested values are listed for the first suggestible value that is “Ovulation_Induction_Protocol” for the selected patient whose ID is 1593. Results are determined using NSNS and k NNS. NSNS suggests “OC + long luteal”. For k NNS, k value equals to 100. According to the results, there are many values that are not applied among 100 nearest neighbours. So, the score value of them equals to 0. Remaining values have negative score values that means the rate of the Failure records among those records are higher than the rate of the Successful records. If we compare the applied, suggested values and the class label of this selected patient, we see that the suggested value from NSNS is equal to the applied one and the class label is Successful. It again can be seen as a proof of the correctness of the suggestion algorithm.

← → ↻ rast.cs.bilkent.edu.tr/admin_page.php



Risk Analysis & Suggestion for Treatment

Dataset: IVF
User: ivf (admin)

Users

- List Users
- Add a New User
- Edit User
- Delete User

Dataset

- Select Dataset File
- Edit Variable Types
- Upload Dataset
- Access Records
- Edit Variable Details
- List Variables
- Insert New Variable
- Delete a Variable

Help



The most similar record [Next](#)

Similarity = 58.56%

Feature	Current	Past Records
ID		
TC_NO		
file_no		
Name		
Phone		
Year		
Female_Age		
Female_Blood_Type		
Height		
Weight		
BMI		
Male_Factor		
Tubal_Factor		
...		
Cycle_Cancellation		
Date_BHCG		
Result_BHCG		
End_thick_HCG		
IVF_Outcome		
Clinical_Pregnancy_Outcome		
Number_Gestational_Sac		
Ongoing_Pregnancy_Outcome		
Notes		
Coculture		
Result		

Figure 8.4: Searching for similar records to the selected patient.

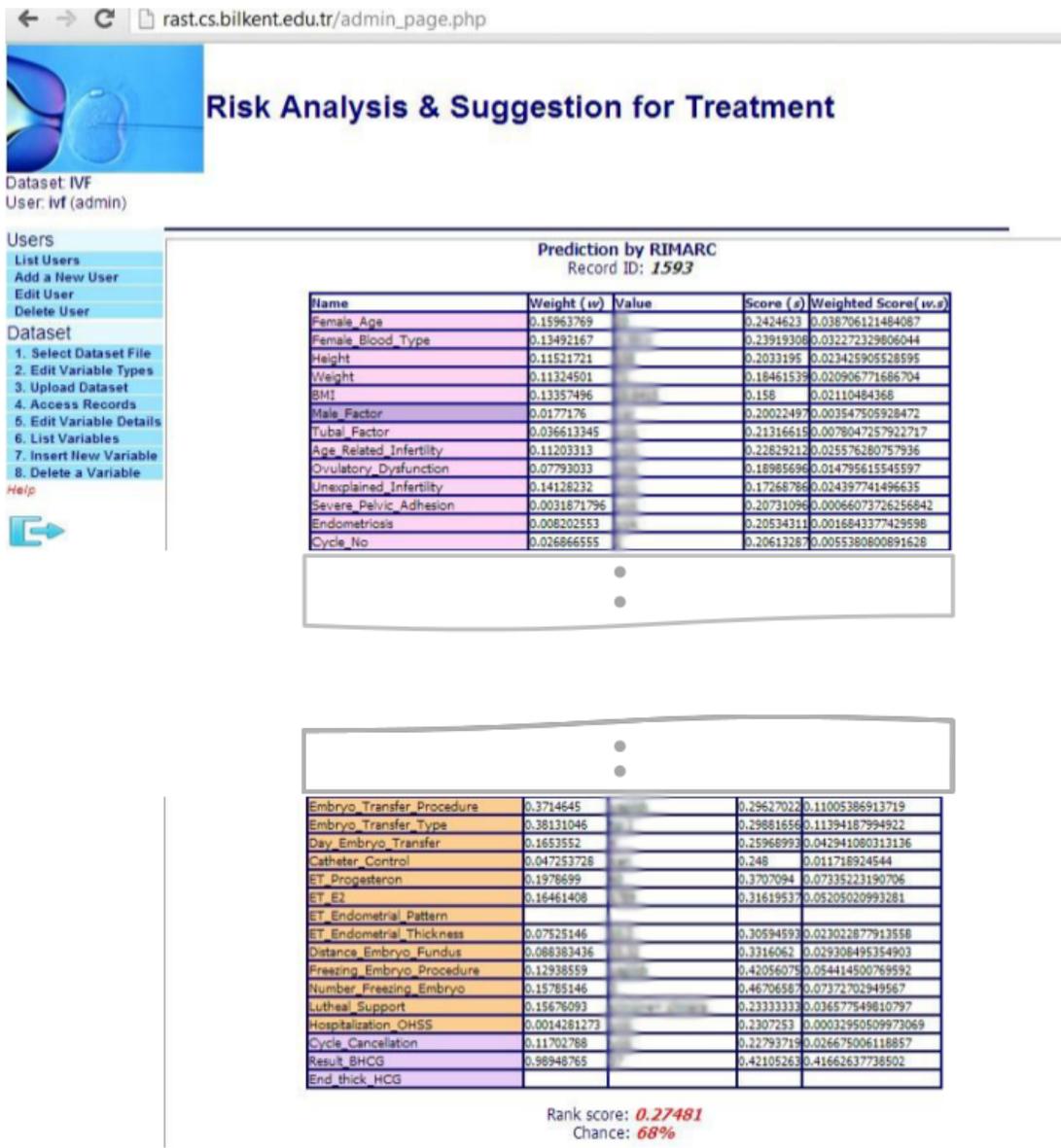


Figure 8.5: Chance estimation for the selected patient.



Figure 8.6: Suggestion for “Ovulation_Induction_Protocol” for the selected patient.

For any user who has staff privileges is allowed to list attributes and access patient records similar as admin. Also, staff can search records using “Access Records” button similar as admin. Moreover, for prediction, the RIMARC algorithm was integrated into the system and using “Learn” button, some analysis about the dataset is made as it is shown in Figure 8.7. Since, RIMARC creates weights and rules, using “Show Weights” and “Show Rules” buttons, produced feature weights and rules are shown as illustrated in Figure 8.8 and Figure 8.9.

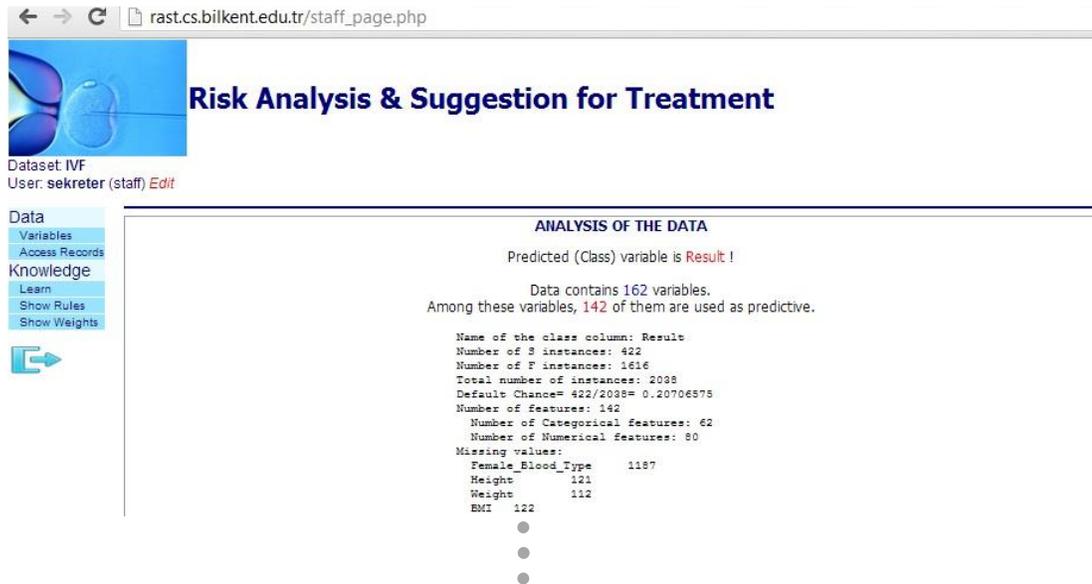


Figure 8.7: Data analysis by the RIMARC algorithm.

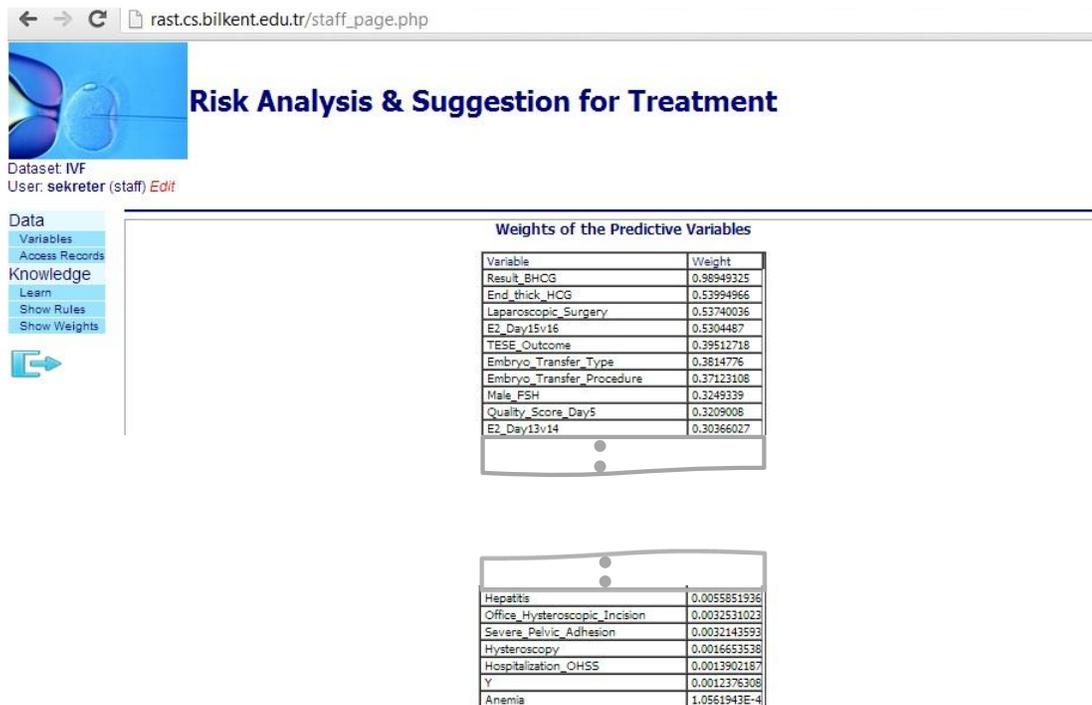


Figure 8.8: Feature weights that are produced by the RIMARC algorithm.

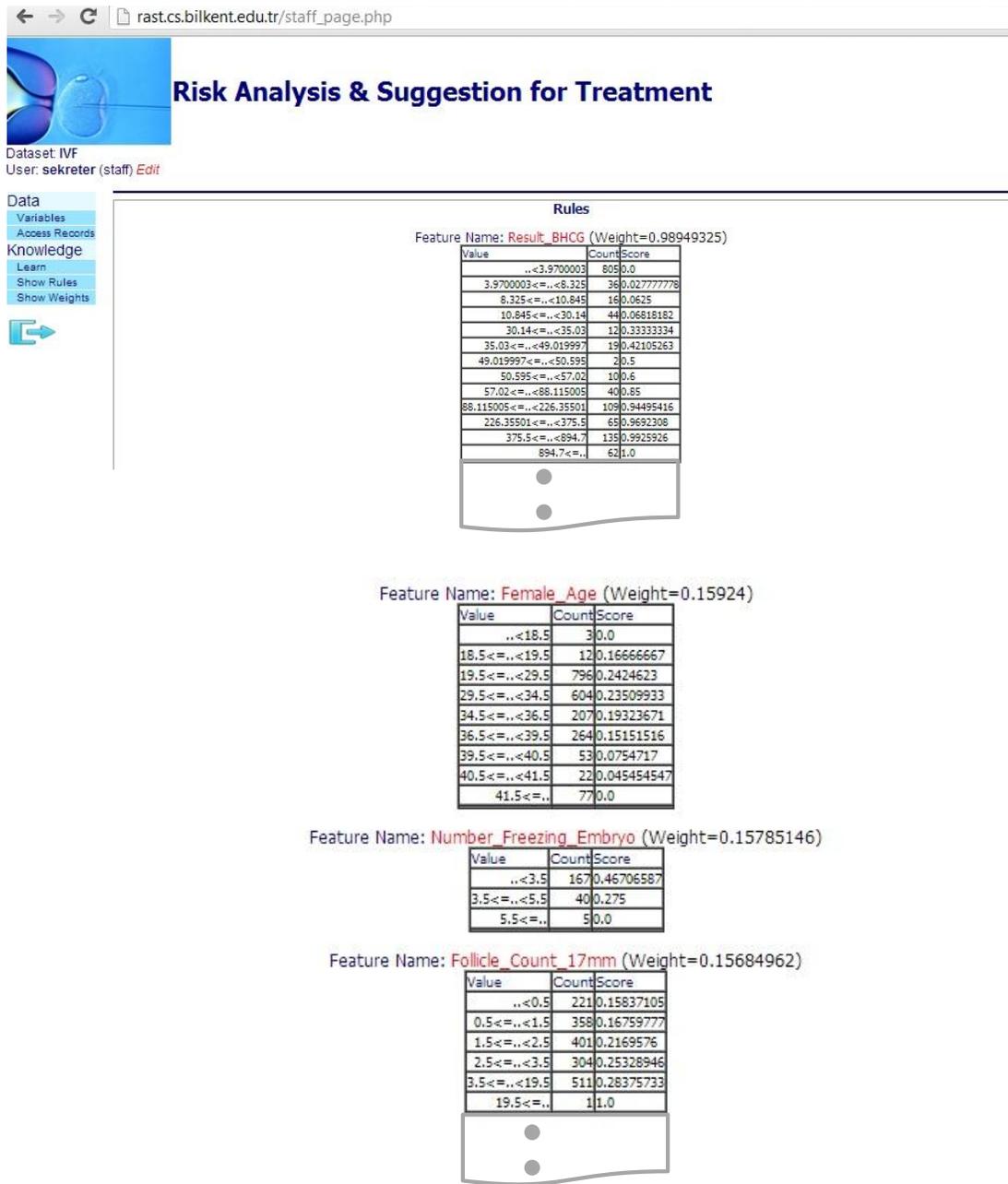


Figure 8.9: Rules that are produced by the RIMARC algorithm.

Last user for the system is guest. Guest is allowed to list variables, patients records, feature weights and rules. Also, guest can download patient records.

Chapter 9

Conclusion and Future Work

In Vitro Fertilization is a common infertility treatment method which female oocytes are inseminated by sperm under laboratory conditions. Given a new candidate for IVF, the first important issue is whether or not go with the IVF treatment. The decision is made jointly by the doctor and the couple. Since the IVF treatment involves an application of several hormones and medicines to both female and male patients, it is a difficult and stressful process. If the chance of success is low, the couple may choose not to continue with the treatment. One way to increase the success rate of the treatment is to build predictive models which take into account results that are derived from different stages from the IVF treatment. Another problem that needs to be solved is the choice of the proper treatment protocol for the given patient.

In this thesis, we gave a discussion about ranking algorithms in order to predict the chance of success in IVF. Then we showed how these ranking algorithms can be modelled as a binary classification problem in machine learning in order to classify instances as Successful or Failure as a result of the IVF treatment. After that, we focused on the suggestion algorithms that can maximize achieving the desired result of the IVF treatment. At last, we developed a decision support system in order to serve these prediction and suggestion algorithms to doctors.

First of all, we worked on the viability prediction models. Our basis algorithm

is RIMARC that is a simple, non-parametric learning algorithm. RIMARC ranks instances by assigning them real values. In order to test the performance of RIMARC, we were interested in constructing different models based on data mining and machine learning techniques in particular support vector machines and k nearest neighbour algorithm.

We present three different methods that are SVM^{light} , $RIkNN$ and RIMARC that, given the clinical parameters of the couple, estimate the chance of success of IVF treatment. Also we implemented the weighted version of the $RIkNN$ that is $RIwkNN$. Given a new patient couple, these methods make estimates of success by considering the results of treatments applied to the past patients. The results indicate that each method performs quite well and they can be used as decision support systems in IVF treatment. However, according to the experimental results it is clearly shown that RIMARC outperforms other two methods in prediction. As a classification algorithm, RIMARC again outperforms other methods in terms of Accuracy on the average.

The second issue is the choice of the best treatment protocol for the patient. An IVF protocol specifies all of the steps of the treatment, including the hormones and the medicines to be used, and the way they are to be administered. In IVF dataset, there are two suggestible values that are “Ovulation_Induction_Protocol” and “Ovulation_Induction_Dose_Protocol”. Although, there are many protocols in common use, it is difficult to choose the best protocol for a given patient. In order to provide this, we developed three different methods that are NSNS, $kNNS$ and DTS. Also we implemented the weighted version of the NSNS that is $wNSNS$.

NSNS is a k Nearest Neighbour based algorithm. The algorithm suggests the treatment protocol that was applied to the nearest and the successful instance. So, we suppose that k equals to 1 and we only consider the successful instance.

$kNNS$ is a k Nearest Neighbour based algorithm. The algorithms considers the class labels and the value of the suggestible feature of the k nearest training instances. It generates a series of alternatives with score values. Score values indicates the importance of the value of the suggestible feature. A suggestible

feature value with highest score means that, if this value is applied to the patient than achieving the desired result will be maximized.

DTS is a decision tree based algorithm. In this algorithm, training dataset are split into smaller training datasets based on the values of the suggestible feature. After splitting, each training dataset includes patient records that only belongs to one value of the suggestible feature. From these training datasets, the algorithm learns a model and classify all test instances.

To the best of our knowledge, in the literature there in not any suggestion system. So, evaluating the performance of the newly developed algorithms is an open issue. In order to overcome this problem, we developed four performance evaluation metrics in order to test the correctness of the suggestion algorithms. These metrics are pessimistic metric (m_p), optimistic metric (m_o), validated optimistic metric (m_{op}) and validated pessimistic metric (m_{vp}). According to the performance evaluation metrics, DTS outperforms other suggestion algorithms in overall evaluation.

In order to bring our algorithms into use, we developed a web based decision support system called RAST (Risk Analysis & Suggestion for Treatment). Now, RAST is in use for IVF dataset however, any other datasets in different domains that needs to make prediction or suggestion can be added to the system. In addition to prediction and suggestion, RAST ensures about data correctness by defining limitations. Doctors can observe how the process will continue and they can compare patients and judge about them. RAST has been used in Etlik Züübeyde Hanım Woman's Health and Teaching Hospital since 5 months.

As a future work, for the ranking and classification algorithms RIMARC can be compared with other methods that aim to maximize AUC and accuracy values. For the suggestion, new algorithms can be developed and performance evaluation metrics can be extended. In addition to IVF dataset, all of these algorithms can be applied to other datasets in medical or different domains and the results can be compared. The interface of the RAST can be improved and remaining suggestion algorithms can be integrated into the system.

To conclude, first of all ranking algorithms are proposed in this thesis for prediction and classification. RIMARC is the most effective predictive model among $RIkNN$, $RIwkNN$ and SVM^{light} . It is easily understandable by domain experts and it will be useful for machine learning community because it is also modelled as a classification method. Furthermore, a new problem is defined namely, suggestion and three algorithms are developed. Since, it is a new research area in the literature, we have to develop performance evaluation metrics called m_p , m_o , m_{vo} and m_{vp} in order to test the correctness of the algorithms. We developed four performance evaluation metrics and validate our algorithms. According to the results, DTS is the most effective suggestive model among NSNS, wNSNS and $kNNS$. The most important parts that contribute to this thesis are the suggestion algorithms and the performance evaluation metrics. Lastly, we developed a decision support system to guide doctors during the IVF treatment. RAST provides data correctness. Also, it gives direction to doctors during the treatment by the help of developed prediction and suggestion algorithms.

Bibliography

- [1] ASRM, “Assited reproductive technologies: A guide for patients,” 1209 Montgomery Highway Birmingham, Alabama 35216-2809, 2011.
- [2] H. A. Güvenir and M. Kurtcephe, “Ranking instances by maximizing the area under roc curve,” *IEEE Transactions on Knowledge and Data Engineering*, vol. 99, 2012.
- [3] V. N. Vapnik, *The nature of statistical learning theory*. No. 0-387-94559-8, New York, NY, USA: Springer-Verlag New York, Inc, 1995.
- [4] S. Agarwal and D. Roth, “Learnability of bipartite ranking functions,” in *Learning Theory* (P. Auer and R. Meir, eds.), vol. 3559 of *Lecture Notes in Computer Science*, pp. 16–31, Springer Berlin Heidelberg, 2005.
- [5] S. Clmenon, G. Lugosi, and N. Vayatis, “Ranking and scoring using empirical risk minimization,” in *Learning Theory* (P. Auer and R. Meir, eds.), vol. 3559 of *Lecture Notes in Computer Science*, pp. 1–15, Springer Berlin Heidelberg, 2005.
- [6] Y. Freund, R. Iyer, R. E. Schapire, and Y. Singer, “An efficient boosting algorithm for combining preferences,” *J. Mach. Learn. Res.*, vol. 4, pp. 933–969, January 2003.
- [7] W. Fan, M. Gordon, and P. Pathak, “Discovery of context-specific ranking functions for effective information retrieval using genetic programming,” *IEEE Transactions on Knowledge and Data Engineering*, vol. 16, no. 4, pp. 523–527, 2004.

- [8] X. Chang, Q. Zheng, and P. Lin, “Cost-sensitive supported vector learning to rank imbalanced data setcost-sensitive supported vector learning to rank imbalanced data set,” in *Cost-Sensitive Supported Vector Learning to Rank Imbalanced Data Set*, vol. 5755 of *Lecture Notes in Computer Science*, pp. 305–314, Springer Berlin Heidelberg, 2009.
- [9] R. Conroy, K. Pyörälä, and A. Fitzgerald, “Estimation of ten-year risk of fatal cardiovascular disease in europe: the score project,” *European Heart Journal*, vol. 11, p. 9871003, 2003.
- [10] R. D’Agostino, S. Ramachandran, and J. Pencina., “General cardiovascular risk profile for use in primary care: the framingham heart study,” *Circulation*, vol. 17, pp. 743753,, 2008.
- [11] F. Provost, T. Fawcett, and R. Kohavi, “The case against accuracy estimation for comparing induction algorithms,” in *In Proceedings of the Fifteenth International Conference on Machine Learning*, pp. 445–453, Morgan Kaufmann, 1997.
- [12] K. Dowd and D. Blake, “After var: The theory, estimation, and insurance applications of quantile-based risk measures,” 2005.
- [13] B. Bradley and M. Taqqu, *Financial risk and heavy tails*, ch. 2, pp. 35–103. Rotterdam: Elsevier, 2003.
- [14] T. Fawcett and F. Provost, “Adaptive fraud detection,” *Data Mining and Knowledge Discovery*, vol. 1, pp. 291–316, 1997.
- [15] W. Krmer, “Wojtek j. krzanowski and david j. hand: Roc curves for continuous data,” *Statistical Papers*, vol. 52, no. 4, pp. 979–980, 2011.
- [16] D. Green and J. Swets, *Signal Detection Theory and Psychophysics*. John Wiley & Sons, 1996.
- [17] M. Zweig and G. Campbell, “Receiver-operating characteristic (roc) plots: A fundamental evaluation tool in clinical medicine,” *Clinical Chemistry*, vol. 39, no. 8, pp. 561–577, 1993.

- [18] M. Pepe, *The Statistical Evaluation of Medical Tests for Classification and Prediction*. Oxford University Press, 2003.
- [19] K. A. Spackman, “Signal detection theory: valuable tools for evaluating inductive learning,” in *Proceedings of the sixth international workshop on Machine learning*, (San Francisco, CA, USA), pp. 160–163, Morgan Kaufmann Publishers Inc., 1989.
- [20] F. Provost and T. Fawcett, “Analysis and visualization of classifier performance: Comparison under imprecise class and cost distributions,” in *In Proceedings of the Third International Conference on Knowledge Discovery and Data Mining*, pp. 43–48, AAAI Press, 1997.
- [21] J. Huang and C. Ling, “Using auc and accuracy in evaluating learning algorithms,” *Knowledge and Data Engineering, IEEE Transactions on*, vol. 17, no. 3, pp. 299–310, 2005.
- [22] A. P. Bradley, “The use of the area under the roc curve in the evaluation of machine learning algorithms,” *Pattern Recognition*, vol. 30, pp. 1145–1159, July 1997.
- [23] T. Fawcett, “An introduction to roc analysis,” *Pattern Recognition Letters*, vol. 27, pp. 861–874, June 2006.
- [24] J. Hanley and B. McNeil, “The meaning and use of the area under a receiver operating characteristic (roc) curve,” *Radiology*, vol. 143, pp. 29 – 36, 1982.
- [25] S. Agarwal, T. Graepel, R. Herbrich, S. Har-Peled, and D. Roth, “Generalization bounds for the area under the roc curve,” *Journal of Machine Learning Research*, vol. 6, pp. 393 – 425, December 2005.
- [26] H. Z. Charles X. Ling, Jin Huang, “Auc: A better measure than accuracy in comparing learning algorithms,” in *Advances in Artificial Intelligence*, vol. 2671 of *Lecture Notes in Computer Science*, pp. 329–341, Springer Berlin Heidelberg, 2003.

- [27] S. Rosset, “Model selection via the auc,” in *Proceedings of the twenty-first international conference on Machine learning*, vol. 69 of *ICML '04*, pp. 89–96, ACM, 2004.
- [28] B. M. Namee, P. Cunningham, S. Byrne, and O. I. Corrigan, “The problem of bias in training data in regression problems in medical decision support,” *Artificial Intelligence in Medicine*, vol. 24, pp. 51–70, January 2002.
- [29] D. M. Tax and R. P. Duin, “Linear model combining by optimizing the area under the roc curve,” in *Proceedings of the 18th International Conference on Pattern Recognition*, vol. 4 of *ICPR '06*, (Washington, DC, USA), pp. 119–122, IEEE Computer Society.
- [30] A. Rakotomamonjy, “Optimizing area under roc curve with svms,” in *Workshop on ROC Analysis in Artificial Intelligence*, pp. 71–80, 2004.
- [31] C. Cortes and M. Mohri, “Auc optimization vs. error rate minimization,” in *Advances in Neural Information Processing Systems*.
- [32] L. Yan, R. Dodier, M. C. Mozer, and R. Wolniewicz, “Optimizing classifier performance via the wilcoxon-mann-whitney statistic,” in *Proceedings of the Twentieth International Conference on Machine Learning*, pp. 848–855, 2003.
- [33] A. Herschtal and B. Raskutti, “Optimising area under the roc curve using gradient descent,” in *Proceedings of the twenty-first international conference on Machine learning*, *ICML '04*, (New York, NY, USA), pp. 49–56, ACM, 2004.
- [34] M. C. Mozer, R. H. Dodier, M. D. Colagrosso, C. G. Salcedo, and R. H. Wolniewicz, “Prodding the roc curve: Constrained optimization of classifier performance,” in *Advances in Neural Information Processing Systems*, pp. 1409–1415, 2002.
- [35] K. Ataman, W. N. Street, and Y. Zhang, “Learning to rank by maximizing auc with linear programming,” in *IN IEEE International Joint Conference on Neural Networks (IJCNN)*, pp. 123–129, 2006.

- [36] U. Brefeld and T. Scheffer, “Auc maximizing support vector learning,” in *ICML workshop on ROC Analysis in Machine Learning*, 2005.
- [37] K.-A. Toh, J. Kim, and S. Lee, “Maximizing area under roc curve for biometric scores fusion,” *Pattern Recognition*, vol. 41, no. 11, pp. 3373 – 3392, 2008.
- [38] C. Ferri, P. A. Flach, and J. Hernández-Orallo, “Learning decision trees using the area under the roc curve,” in *Proceedings of the Nineteenth International Conference on Machine Learning*, pp. 139–146, 2002.
- [39] H. Boström, “Maximizing the area under the roc curve using incremental reduced error pruning,” in *Pruning, Proceedings of the International Conference on Machine Learning 2005 Workshop on ROC Analysis in Machine Learning*, ACM Press, 2005.
- [40] T. Fawcett, “Using rule sets to maximize roc performance,” in *Proceedings of the IEEE International Conference on Data Mining (ICDM)*, pp. 131–138, IEEE Computer Society, 2001.
- [41] R. C. Prati and P. A. Flach, “Roccer: A roc convex hull rule learning algorithm,” in *Proceedings of the ECML/PKDD Workshop on Advances in Inductive Rule Learning*, pp. 144–153, 2004.
- [42] C. Marrocco, R. Duin, and F. Tortorella, “Maximizing the area under the roc curve by pairwise feature combination,” *Pattern Recognition*, vol. 41, no. 6, pp. 1961 – 1974, 2008.
- [43] M. Sebag, J. Azé, and N. Lucas, “Roc-based evolutionary learning: Application to medical data mining,” in *Artificial Evolution*, vol. 2936 of *Lecture Notes in Computer Science*, pp. 384–396, Springer Berlin Heidelberg, 2004.
- [44] C. Marrocco, M. Molinara, and F. Tortorella, “Exploiting auc for optimal linear combinations of dichotomizers,” *Pattern Recognition Letters*, vol. 27, no. 8, pp. 900 – 907, 2006.

- [45] T. Joachims, “A support vector method for multivariate performance measures,” in *Proceedings of the 22nd international conference on Machine learning*, ICML '05, (New York, NY, USA), pp. 377–384, ACM, 2005.
- [46] C. X. Ling and H. Zhang, “Toward bayesian classifiers with accurate probabilities,” in *Advances in Knowledge Discovery and Data Mining*, vol. 2336 of *Lecture Notes in Computer Science*, pp. 123–134, Springer Berlin Heidelberg, 2002.
- [47] T. Calders and S. Jaroszewicz, “Efficient auc optimization for classification,” in *Knowledge Discovery in Databases: PKDD 2007*, vol. 4702 of *Lecture Notes in Computer Science*, pp. 42–53, Springer Berlin Heidelberg, 2007.
- [48] G. Han and C. Zhao, “Auc maximization linear classifier based on active learning and its application,” *Neurocomputing*, vol. 73, pp. 1272–1280, March 2010.
- [49] I. Jurisica, J. Mylopoulos, J. Glasgow, H. Shapiro, and R. F. Casper, “Case-based reasoning in ivf: prediction and knowledge mining,” *Artificial Intelligence in Medicine*, vol. 12, no. 1, pp. 1–24, 1998.
- [50] S. J. Kaufmann, J. L. Eastauh, S. Snowden, S. W.Smye, and V. Sharma, “The application of neural networks in predicting the outcome of in-vitro fertilization,” *Human Reproduction*, pp. 1454 – 1457, 1997.
- [51] R. Saith, A. Srinivasan, D. Michie, , and I. Sargent, “Relationships between the developmental potential of human in-vitro fertilization embryos and features describing the embryo, oocyte and follicle,” *Human Reproduction Update*, pp. 121 – 134, 1998.
- [52] J. R. Trimarchi, J. Goodside, L. Passmore, T. Silberstein, L. Hamel, and L. Gonzalez, “Comparing data mining and logistic regression for predicting ivf outcome,” *Fertil. Steril*, vol. 80, 2003.
- [53] A. Uyar, A. Bener, N. ray, and M. Baheci, “Predicting implantation outcome from imbalanced ivf dataset,” in *The World Congress on Engineering and Computer Science*, 2009.

- [54] A. Uyar, A. Bener, H. Ciray, and M. Bahceci, “Roc based evaluation and comparison of classifiers for ivf implantation prediction,” in *Electronic Healthcare*, vol. 27 of *Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering*, pp. 108–111, Springer Berlin Heidelberg, 2010.
- [55] A. Uyar, H. Ciray, A. Bener, and M. Bahceci, “3p: Personalized pregnancy prediction in ivf treatment process,” in *Electronic Healthcare*, vol. 0001 of *Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering*, pp. 58–65, Springer Berlin Heidelberg, 2009.
- [56] D. A. Morales, E. Bengoetxea, P. L. naga, M. García, Y. Franco, M. Fresnada, and M. Merino, “Bayesian classification for the selection of in vitro human embryos using morphological and clinical data,” *Computer Methods and Programs in Biomedicine*, vol. 90, no. 2, pp. 104 – 116, 2008.
- [57] G. Corani, C. Magli, A. Giusti, L. Gianaroli, and L. Gambardella, “A bayesian network model for predicting the outcome of in vitro fertilization,” in *Sixth European Workshop on Probabilistic Graphical Models*, 2012.
- [58] E. S. Berner, “Clinical decision support systems: State of the art,” tech. rep., Department of Health Services Administration University of Alabama at Birmingham, Rockville, Maryland, June.
- [59] C. Y. Hsu, L. C. Huang, T. M. Chen, L. F. Chen, , and J. C. Chao, “A web-based decision support system for dietary analysis and recommendations,” *Telemedicine and e-Health*, vol. 17, pp. 68 – 75, March 2011.
- [60] R. Chaudhry, S. M. Tullidge-Scheitel, D. A. Parks, K. B. Angstman, L. K. Decker, and R. J. Stroebel, “Use of a web-based clinical decision support system to improve abdominal aortic aneurysm screening in a primary care practice,” *Journal of Evaluation in Clinical Practice*, vol. 18, pp. 666 – 670, June 2012.

- [61] M. L. Graber and A. Mathew, “Performance of a web-based clinical diagnosis support system for internists,” *Journal of General Internal Medicine*, vol. 23, no. 1, pp. 37 – 40, 2008.
- [62] N. Soullier, J. Bouyer, J. Pouly, J. Guibert, and E. de La Rochebrochard, “Estimating the success of an in vitro fertilization programme using multiple imputation,” *Human Reproduction*.
- [63] M. Kurtcephe and H. A. Güvenir, “A discretization method based on maximizing the area under receiver operating characteristic curve,” *International Journal of Pattern Recognition and Artificial Intelligence*, vol. 27, no. 01, p. 1350002, 2013.
- [64] T. M. Mitchell, *Machine Learning*. No. 0070428077, 9780070428072, New York, NY, USA: McGraw-Hill, Inc., 1997.
- [65] T. Cover and P. Hart, “Nearest neighbor pattern classification,” *Information Theory, IEEE Transactions on Information Theory*, vol. 13, no. 1, pp. 21–27, 1967.
- [66] T. Denoeux, “A k-nearest neighbor classification rule based on dempster-shafer theory,” *Systems, Man and Cybernetics, IEEE Transactions on Systems, Man, and Cybernetics*, vol. 25, no. 5, pp. 804–813, 1995.
- [67] T. Hastie and R. Tibshirani, “Discriminant adaptive nearest neighbor classification,” *Pattern Analysis and Machine Intelligence, IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 18, no. 6, pp. 607–616, 1996.
- [68] J. H. Friedman, “Flexible metric nearest neighbor classification,” tech. rep., Department of Statistics and Stanford Linear Accelerator Center, Stanford University, 1994.
- [69] V. Podgorelec, P. Kokol, B. Stiglic, and I. Rozman, “Decision trees: An overview and their use in medicine,” *Journal of Medical Systems*, vol. 26, no. 5, pp. 445–463, 2002.

- [70] A. Azar and S. El-Metwally, “Decision tree classifiers for automated medical diagnosis,” *Neural Computing and Applications*, pp. 1–17, 2012.
- [71] I. Kononenko, “Machine learning for medical diagnosis: history, state of the art and perspective,” *Artificial Intelligence in Medicine*, vol. 23, no. 1, pp. 89 – 109, 2001.
- [72] T. A. Graham, A. W. Kushniruk, M. J. Bullard, B. R. Holroyd, D. P. Meurer, and B. H. Rowe, “How usability of a web-based clinical decision support system has the potential to contribute to adverse medical events,” in *AMIA Annual Symposium Proceedings*, vol. 6, pp. 257 – 261, November 2008.
- [73] S. Tsumoto, “Web based medical decision support system: application of internet to telemedicine,” in *Applications and the Internet Workshops*, pp. 288–293, 2003.