

Uzman Karışımlarında Yeni Bir Yaklaşım: Uzman Kararlarının Yeni Bir Geçiş Fonksiyonuyla Birleştirmesi

Faruk BULUT¹

M. Fatih AMASYALI²

A.Coşkun SÖNMEZ³

^{1,2,3}Bilgisayar Mühendisliği Bölümü
Elektrik-Elektronik Fakültesi
Yıldız Teknik Üniversitesi, Davutpaşa, İSTANBUL

Email: f0110303@std.yildiz.edu.tr

mfatih@ce.yildiz.edu.tr

acsonmez@ce.yildiz.edu.tr

Özet

Kolektif öğrenme (Ensemble) metotlarından biri olan uzman karışımları, öğrenme başarısını artırmak için kullanılan yöntemlerden biridir. Bu yöntemde veri seti bölümlere ayrılarak her bir bölüm için ayrı bir uzman eğitilir ve geçiş fonksiyonu ile (gating function) ile uzmanların kararları birleştirilir. Literatürde genelde uzman olarak, yüksek performans ve hızlarından dolayı karar ağaçları tercih edilir. Bu çalışmada veri setini kümeleme (clustering) yöntemiyle alt veri setlerine bölerek her bir alt veri seti için ayrı bir karar ağaçları inşa edilmiştir. Geliştirilen geçiş fonksiyonu ile uzmanların kararları birleştirilmiştir. Bu geçiş fonksiyonu sayesinde, her hangi bir test noktası için tüm uzmanların test noktasına olan uzaklıklarına bağlı olarak verdikleri kararlar ağırlıklandırılmış ve ortak komite kararı bulunmuştur. Yapılan denemelerde tekil karar ağaçlarına ve en yakın komşu algoritmasına göre uzman karışımlarının daha başarılı sonuçlar verdiği gözlemlenmiştir.

1. Giriş

Bir veri kümesinin alt bölgeleri birbirinden farklı yapılar içerebilir. Bu farklılıklar, örnek (samples) sayısından, sınıf etiketi sayısından, verilerin tipinden, seyrek veya yoğun oluşundan kaynaklanıyor olabilir. Bahsedilen bu farklılıklar sınıflandırma başarısını olumlu ya da olumsuz etkileyebilmektedir. Sınıflandırma başarısını artırmak amacıyla tüm veri kümesi için tek bir model belirlemek yerine her bir alt bölgesi için farklı modeller tasarlamak gerekebilir.

Çalışmamızda bu alt bölgeleri ayrı bir veri seti gibi düşünerek bahsedilen bu etkileri en aza indirecek ve genel sınıflandırma başarısını artıracak bir yöntem geliştirilmiştir. Bu amaçla tüm veri seti için tek bir

öğrenici yerine, veri setinin her bir alt bölümü için ayrı ayrı öğrenici inşa edilmiştir. Veri setinin alt bölümlerini belirlemek için kümeleme (clustering) işlemi yapılmıştır.

Uzman Karışımları (Mixture of Experts, UK) algoritması temelde iki soruya cevap arar. Birincisi veri setinin alt bölümlerinin nasıl belirleneceği, diğeri ise test örneklerinin nasıl etiketleneceğidir. Literatürde [1-3] bu sorulara cevap olarak veri setini bölümleme işlemi için Beklenti Artırımı (Expectation Maximization, BA) algoritması ve etiketleme işlemi için BA yöntemine özgü bir geçiş fonksiyonu (Gating Function) gösterilmektedir. Bilindiği üzere BA yöntemi, yumuşak bir kümeleme (soft clustering) yapmaktadır. Test örneğinin etiketlenmesinde ait olduğu alt küme için tasarlanmış uzmanının kararı geçerlidir.

Bu çalışmamızda veri setinin alt bölümleri k-means yöntemiyle katı kümeleme (hard clustering) yapılarak belirlenmiştir. Her bir alt bölüm ayrı bir Karar Ağacı (Decision Tree, KA) ile modellenerek homojen bir kolektif öğrenme yapısı oluşturulmuştur. Bir test örneğinin etiketlenmesi için tüm uzmanların kararlarını birleştiren yeni bir geçiş fonksiyonu (gating function) tasarlanmıştır.

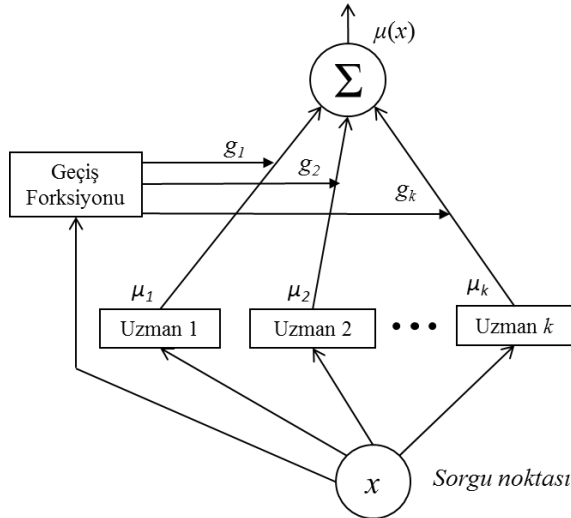
Çalışmamız beş bölümden oluşmaktadır. İkinci bölümde UK'nın çalışma prensibine; üçüncü bölümde uygulama ve deneysel sonuçlara; dördüncü bölümde üzerinde çalıştığımız yöntemin karmaşıklık analizine; beşinci bölümde ise yorumlara yer verilmiştir.

2. UK İçin Yeni Bir Geçiş Fonksiyonu

Kolektif öğrenme (Ensemble Learning) yöntemlerinden biri olan UK için son 20 yıl içerisinde teorik çalışmalar yapılmış, farklı yaklaşımlar

sunulmuştur [2]. UK'da amaç, veri setini böl-yönet mantığıyla daha az karmaşık bir yapıya indirgeyerek sınıflandırma performansını artırmaktır. UK ile veri seti en uygun alt kümelerle bölünür ve her bir alt küme için bir uzman tasarlanarak o bölgede sınıflandırma başarısı artırılmaya çalışılır. Her hangi bir test noktasının sınıflandırılması normalde ait olduğu alt bölgenin uzmanı ile yapılmaktadır. Fakat çalışmamızda test noktasının sınıflandırılması tüm bölgelere ait uzmanların kararlarının birleştirilmesi ile yapılmıştır. Bu işlemi geçiş fonksiyonu (*gate function*) adındaki bir mekanizma yapmaktadır [1,3]. Şekil-1'de görülen UK'nın çalışma mantığı şu basamaklardan oluşur:

1. Veri seti K adet alt kümeye bölünür.
2. Her bir alt küme için bir Uzman (*Expert*) eğitilir.
3. Gelen x test örneğinin etiketlenmesi için tüm uzmanlara danışılır ve kararları (μ_i) alınır.
4. Geçiş fonksiyonu mekanizması ile kararlar ağırlıklandırılır ($g_i * \mu_i$) ve test noktasının her bir sınıfa ait olma olasılığı belirlenir.
5. En yüksek oranın ait olduğu sınıf ile test noktası etiketlenir (*majority voting*).



Şekil 1. UK'nın çalışma prensibi

Çalışmamızda farklı bir uzman tasarımı ve buna bağlı olarak bir geçiş fonksiyonu geliştirilmiştir. Şekil-1'de görüldüğü gibi x test noktasının etiketlenmesinde her bir uzmanın verdiği karar geçiş fonksiyonu tarafından ağırlıklandırılır. Geçiş fonksiyonun tasarlanmasında uzakta bulunan uzmanların etkisi az; yakında olanların etkisi fazla olması gerektiği düşüncesinden yola çıkılarak *Shepard* metodundan [4] yararlanılmıştır.

Bir test örneğinin sadece bir uzmana danışılması durumunda sınıflandırma hatalarının oluşması muhtemeldir. Bu test örneği alt kümelerin karar sınırlarına yakınsa hangi uzmanın kararına danışılacağı problem oluşturmaktadır. Bu çalışmada bu soruna çözüm aranmış ve bahsedildiği üzere yeni bir geçiş fonksiyonu tasarlanmıştır.

Formül 1'de w_i , i . uzmanın hesaplamadaki ağırlığıdır. E_j , j . uzmanın öğrendiği alt kümenin merkez noktasıdır. Merkez noktası küme içindeki eğitim örneklerinin aritmetik ortalaması ile bulunmaktadır ve ilgili uzmanı temsil etmektedir.

$$w_i = \frac{1}{d_i^2} = \frac{1}{d_i(x, E_j)^2} \quad (1)$$

$$d(x, E_j) = \sqrt{\sum_{i=1}^A (a_i(x) - a_i(E_j))^2} \quad (2)$$

x test noktası ve E_j arasındaki uzaklık Öklid uzaklık formülü (2) ile bulunmaktadır. A, uzayın boyut (özellik) sayısıdır ve herhangi bir noktanın özellikleri vektör olarak şu şekilde tanımlanmaktadır: $\langle a_1(x), a_2(x), a_3(x), \dots, a_A(x) \rangle$. Burada $a_r(x)$, x 'in r . özelliğini göstermektedir.

Tom Mitchell tarafından önerilen *Distance-weighted-kNN* sınıflandırıcının [4], [6] çalışma prensibi buraya uyarlanmış ve geçiş fonksiyonu olarak (3) numaralı formül geliştirilmiştir. g_i , i . uzmanın x test noktası için hesaplamadaki ağırlığıdır.

$$\text{Geçiş Fonksiyonu} = g_i = \frac{w_i}{\sum_{j=1}^K w_j} \quad (3)$$

$$G = \sum_{i=1}^k g_i = 1 \quad (4)$$

K'nın alt küme ve aynı zamanda uzman sayısını gösterdiği 4 numaralı formülde her bir uzman için atanan yüzdelik ağırlıklar toplamı yani G , 1'e eşittir. Görüldüğü üzere x örneğinin sınıfı belirlenirken uzaklığa bağlı olarak her bir uzmanın (Karar ağaçlarının) verdikleri kararlar ağırlıklandırılmıştır.

x 'in ait olabileceği sınıfların yüzdelik olarak en büyüğü ile etiketleme yapılabilmesi için sınıflandırma fonksiyonu formülümüz (5) ve bu formülün genişletilmiş hali (6) şu şekildedir:

$$\mu(x) = \underset{c \in C}{\operatorname{argmax}} (g_i \cdot \mu_i^c(x)) \quad (5)$$

$$\mu(x) = \underset{c \in C}{\operatorname{argmax}} \left(\frac{\sum_i^K w_i \cdot \mu_i^c(x)}{\sum_j^K w_j} \right) \quad (6)$$

$\mu: R \rightarrow C$ olan μ fonksiyonu tüm sınıf etiketleri için ayrı ayrı hesaplama yapıp maksimum argümanı sınıf etiketi olarak x 'e atamaktadır. Burada C , veri setindeki tüm sonlu sınıf etiketleri kümesidir ve şu şekilde tanımlanabilir: $\{c_1, c_2, c_3, \dots, c_s\}$. $\mu_i^c(x)$ fonksiyonu ise, i . uzmana göre x 'in c sınıfından olma olasılığını hesaplamaktadır.

3. Pratik Uygulama ve Deneysel Sonuçlar

Uygulama alanı olarak 36 adet UCI Benchmark veri seti [5] seçilmiştir. Tüm veri setlerine ait değerler normalize edilmiş, kayıp değerler yer değiştirilmiş, nominal değerler ikili sayısal değerlere dönüştürülerek kullanıma hazır hale getirilmiştir. Tüm algoritmalar MATLAB ortamında, her bir veri seti üzerinde 5x2 çapraz geçirme ile test edilmiştir.

Yapılan uygulamalarda veri setleri k -means kümeleme yöntemiyle (iterasyon sayısı=100) k değerleri sırasıyla 2 ile 50 arasında alınarak kümeleme işlemi yapılmıştır. Veri setlerini daha yüksek sayıda kümeler ayırmak hem hesaplama süresini artırmakta hem de seyrek veri setlerinde çok az elemanlı alt kümeler oluşturmaktadır. Bu durumda k -means'teki k parametresi artırıldığında (veri setindeki eleman sayısına yaklaşıldığında) *distance-weighted-kNN* sınıflandırmasına benzer bir yapının ortaya çıktığı gözlemlenmiştir. Fazla örnekleri bulunan veri setlerini daha fazla sayıda alt kümelere; az sayıda örnekleri bulunan veri setlerinin de az sayıda kümeler bölmenin sınıflandırma başarısını artırdığı gözlemlenmiştir.

3.1. Temel Uzman seçimi

Bu çalışmada temel uzman seçimini, karar ağaçları ile yapılmıştır. k -means ile oluşturulan k tane veri setinin her biri ayrı bir veri seti gibi düşünülüp k tane KA inşa edilmiştir.

UK uygulamalarında KA kullanımı mantığı ilk olarak Jordan ve Jacobs tarafından anlatılmıştır [7]. Bugüne dek UK yöntemleri üzerinde pek çok çalışma [2] yapılmış ve farklılıkları gösterilmiştir. UK'nın günümüzde olduğu gibi *Radial Basis* fonksiyonlarıyla kullanılmasını Lei Xu önermiştir [8]. KA'ların, kompakt ve sınıflandırmayı kolaylaştırıcı özelliğinden ötürü *Radial Basis* fonksiyon ağırlarında TB-RBF

(*Decision Tree based Radial Basis Function*) adıyla kullanıldığı bilinmektedir [9]. Hızlı bir şekilde eğitilebilmeleri, karar verme hızlarının yüksek oluşu ve beyaz kutu (*white box*) özellikleri ile çalışma sistemlerinin kolay analiz edilebilmesi gibi niteliklerinden ötürü KA'lar diğer öğrencilere göre daha avantajlıdır. Kolektif öğrenme metodlarında temel öğrenici olarak genelde KA'lar tercih edildiği de bir gerçektir [9-10].

3.2. Ağırlıklandırma Yöntemi

Uzmanların kararları (1) numaralı formül ile uzaklığa bağlı olarak yapılmaktadır. Buradaki ağırlıklandırma işlemi için kullanılan farklı yöntemler vardır: $1/d$, $1/d^2$ ve $1/d^3$. $1/d^2$ 'li yöntem, $1/d$ 'ye göre yakındaki öğelerin etkisini çok fazla; uzaktakinin etkisini çok az almaktadır. Geçiş fonksiyonu için kullanılan $1/d^2$ 'li formül, yapılan deneysel uygulamalarda diğerinden daha iyi sonuçlar verdiği için tercih edilmiştir.

3.3. Sonuçlar

İki sınıflandırma mekanizması tarafından elde edilen sonuçların istatistiksel anlamlılığını belirleyebilmek için T-Test yöntemi kullanılmıştır. T-Test sonuçları üç farklı değer içermektedir: *win* (başarılı), *loss* (başarısız) ve *tie* (eşit). Bu sayede geliştirilen yöntemin başarılı olup olmadığı tespit edilebilmektedir. Tablo-1'de 36 UCI veri seti ile yapılan uygulamaların sonuçları görülmektedir. Geliştirdiğimiz UK yöntemi 3 ayrı sınıflandırıcı ile kıyaslanmıştır. Bunlar 1NN (*En Yakındaki Komşu* algoritması), en iyi sonucu veren k -NN (k – *En yakın Komşuluk* algoritması) ve KA sınıflandırıcılarıdır. T-Test sonucuna göre elde edilen *win*, *tie* ve *loss* sayıları Tablo-1'de görülmektedir. Görüldüğü üzere UK'nın KA ile kıyaslanmasında 11 adet veri kümesinde performans artışı (*win*) sağlanmış, 23'ünde değişme olmamış, sadece 2 tanesinde başarısız olunmuştur. Bu da üzerinde çalışılan yöntemin başarılı sonuçlar elde ettiğini göstermektedir.

Tablo 1. Sonuçlar

T-Test Sonucu	UK'yı 1NN ile Karşılaştırma	UK'yı En iyi kNN ile Karşılaştırma	UK'yı KA ile Karşılaştırma
<i>Win</i>	21	11	11
<i>Tie</i>	11	18	23
<i>Loss</i>	4	7	2

Tüm veri setleri için oluşturulmuş KA sınıflandırıcısı ile bu uygulamada yaptığımız UK sınıflandırıcısının

doğruluk oranları Tablo-2’de görülmektedir. Bu tablo ile tekil bir öğrenci olan KA’nın sınıflandırma performansı ile UK’nın performansı T-Test sonuçlarıyla kıyaslanabilmektedir.

Tablol 2. UCI Test Sonuçları

Veri seti Adı	KA Doğruluk Oranı	UK		KA ile UK’yı karşılaştırma	
		k	Doğruluk Oranı	Yüzdelik fark	T-Test Sonucu
abalone	0.2122	50	0.2231	5.11	win
anneal	0.9883	2	0.9879	-0.05	tie
audiology	0.8615	2	0.8166	-5.22	loss
autos	0.6614	2	0.6426	-2.84	tie
balance-scale	0.7811	30	0.8614	10.28	win
breast-cancer	0.6692	20	0.7028	5.02	win
breast-w	0.9465	4	0.9536	0.76	tie
col10	0.7575	3	0.7559	-0.21	tie
colic	0.8092	2	0.8212	1.48	tie
credit-a	0.8197	4	0.8533	4.10	win
credit-g	0.6870	20	0.7140	3.93	win
d159	0.9698	2	0.9610	-0.90	tie
diabetes	0.7036	10	0.7375	4.81	win
glass	0.6527	5	0.6615	1.35	tie
heart-statlog	0.7415	35	0.8015	8.09	win
hepatitis	0.7923	4	0.8103	2.28	tie
hypothyroid	0.9943	2	0.9932	-0.11	tie
ionosphere	0.8547	35	0.8991	5.20	win
iris	0.9333	2	0.9400	0.71	tie
kr-vs-kp	0.9901	2	0.9838	-0.63	tie
labor	0.8596	2	0.8526	-0.82	tie
letter	0.8234	2	0.8203	-0.38	tie
lymph	0.7901	3	0.7775	-1.60	tie
mushroom	1.0000	2	0.9997	-0.03	tie
primary-tumor	0.4265	2	0.4351	2.02	tie
ringnorm	0.8842	10	0.9375	6.03	win
segment	0.9436	2	0.9474	0.39	tie
sick	0.9824	2	0.9846	0.22	tie
sonar	0.7000	5	0.7231	3.30	tie
soybean	0.8824	2	0.8809	-0.17	tie
splice	0.9254	2	0.9066	-2.03	tie
vehicle	0.6849	5	0.6835	-0.21	tie
vote	0.9361	3	0.9549	2.01	tie
vowel	0.6873	9	0.7143	3.94	win
waveform	0.7394	35	0.8051	8.89	win
zoo	0.9762	2	0.9214	-5.61	loss

İlk sütun tekil KA sınıflandırıcısının veri kümeleri üzerinde verdiği doğruluk değerini; ikinci ve üçüncü sütun ise UK sınıflandırıcısının değerlerini göstermektedir. İkinci sütunda UK için oluşturulan alt küme sayısını yani k -means yöntemindeki k parametresini; sonraki sütun UK’nın k değerine bağlı olarak verdiği doğruluk değerini göstermektedir. Son iki sütunda UK ile KA karşılaştırıldığında yüzdelik artış-azalış ile T-Test sonucu verilmektedir. Yaptığımız UK uygulamasında doğruluk oranının bazı veri setlerinde KA doğruluk oranına göre %10.28’e kadar artış sağladığı gözlemlenmiştir.

4. Yöntemin Karmaşıklık Analizi

Uzman olarak inşa edilen bir KA’nın büyük O notasyonuna göre zaman karmaşıklığı şu formül ile açıklanmaktadır[11]:

$$O(A * n * \log n) \quad (7)$$

n örnek sayısını, A ise özellik sayısını göstermektedir.

k -means kümeleme yönteminin karmaşıklığı ise şu formül ile analiz edilmektedir [12]:

$$O(n * k * I * A) \quad (8)$$

k küme sayısını, I iterasyon sayısını simgelemektedir.

Geliştirdiğimiz UK yönteminde k tane KA oluşturmanın maliyeti ise şu şekildedir:

$$O\left(k * A * \frac{n}{k} * \log \frac{n}{k}\right) = O\left(A * n * \log \frac{n}{k}\right) \quad (9)$$

Bir veri için yalın bir KA inşa etmek ile k tane KA inşa etmek arasındaki zaman karmaşıklığı oranını şu şekilde bulunabilir:

$$O\left(\frac{A * n * \log n}{A * n * \log \frac{n}{k}}\right) = O\left(\frac{\log n}{\log \frac{n}{k}}\right) \quad (10)$$

k parametresi her zaman için küçük bir tamsayı olarak alındığından dolayı önemsenmeyebilir. Bu durumda k tane KA oluşturmak zaman karmaşıklığı açısından maliyetli bir durum olmadığı sonucuna varılabilir.

I ve A sayılarının her bir uzman için aynı olduğu düşünülüp sabitlenirse, UK yönteminin karmaşıklığı şu şekle dönüşür:

$$O\left(A * n * \log \frac{n}{k}\right) + O(n * k) + O(k) \quad (11)$$

Yukarıdaki zaman karmaşıklığı analizinde ilk iki karmaşıklık k tane KA ve k tane küme oluşturmayı göstermektedir. En sonda bulunan $O(k)$ karmaşıklığı

ise geiş fonksiyonuna aittir ve k küçük bir tamsayı olduđu için önemsenmeyecek etki düzeyindedir.

Yapılan incelemelerde BA kümeleme yöntemi içeren uzman karışımının karmaşıklığı [2-3] ile k -means yöntemli çalışmamızın karmaşıklığının birbirine benzer olduđu görülmüştür.

5. Değerlendirme ve İleri Çalışmalar

Bir veri setinin bazı bölgelerinin diğerlerinden farklı özelliklere sahip olması genel sınıflandırma performansını düşürebilir. Bu nedenle bu bölgeleri diğerlerinden ayırmak amacıyla önce kümeleme yapıldı, sonra her bir bölge için ayrı bir uzman eğitildi, ardından bu uzmanlar bir geiş fonksiyonuyla birleştirilerek uzman karışımı bir sistem geliştirildi. Bu çalışmamızda tüm veriye tek bir model belirlemek yerine, veriyi alt bölgelere ayırıp her biri için farklı modeller belirlemenin sınıflandırma performansını genel olarak artırdığı görüldü.

Uzman oluşturabilmek için veri seti denetimsiz öğrenme ile kümeler bölünmeye çalışılmaktadır. Bu ise hesaplama süresini tekil öğrencilere göre artırmaktadır. k -means ile yapılan kümeleme işleminde her defasında başka küme grupları oluştuđu için sınıflandırma başarısı kümelerin yerleşimine, eleman sayısına ve yapısına bağlı olarak az da olsa değişiklik göstermektedir. Yapılan T-Test analizlerinde hemen hemen aynı sayıda *win*, *tie*, *loss* çıktıları elde edildiği için bu değişiklik önemsenmeyecek düzeydedir.

Ayrıca k -means yöntemi iki boyutlu veri kümelerinde dairesel; üç boyutlu ortamda da küresel tarzda kümeleme yaptığı için alt kümelerde bulunan örnekler o kümenin merkez noktası etrafına saçılmış bir şekilde bulunmaktadır. Kısaca k -means metodunda oluşturulan kümeler iç içe veya sarmal yapıda olmamaktadır. Bu özellik, UK'da uzaklığa bağlı olarak ağırlıklandırma ile sınıflandırma yapan geiş fonksiyonu için çok uygun olmaktadır ve yüksek başarı elde edilmesini sağlamaktadır.

İleri aşamalarda, uzman olarak KA'nın haricinde farklı sınıflandırıcılar ve farklı kümeleme yöntemleri de kullanılarak performans karşılaştırmaları yapılabilir. Ayrıca BA ile yapılan UK çalışmaları ile üzerinde çalıştığımız yöntem değişik açılardan kıyaslanabilir.

6. Kaynaklar

- [1] Alpaydın E, Yapay Öğrenme Kitabı, Uzman Karışimleri, Boğaziçi Üniversitesi Yayınevi, (2010), s. 373-376, 253-257
- [2] Yuksel, S. E. Twenty Years of Mixture of Experts. IEEE Transactions On Neural Networks And Learning Systems, (2012). s. 1177-1193.
- [3] Dempster, A.P., Laird, N.M., Rubin, D.B. "Maximum Likelihood from Incomplete Data via the EM Algorithm". Journal of the Royal Statistical Society, Series B 39, (1977). s.1-38.
- [4] Shepard, D. A two-dimensional interpolation function for irregularly spaced data. Proceedings of the 23rd National Conference of the ACM, (1968). s. 517-523.
- [5] Bache, K. & Lichman, M. UCI Machine Learning Repository <http://archive.ics.uci.edu/mlIrvine>, CA: University of California, School of Information and Computer Science. (2013).
- [6] Mitchell, T. Machine Learning, Title:8.2.1 Distance-Weighted-kNN. USA: McGraw-Hill, (1997). s. 233-236, s. 156-159.
- [7] Michael I. Jordan, Robert A. Jacobs, "Hierarchical Mixture of Experts and the EM Algorithm Journal Neural Computation archive, Vol.6, Issue 2, (1994)., s.181-214
- [8] Lei Xu, RBF nets, Mixture Experts, and Bayesian Ying-Yang Learning, Elsevier Neurocomputing 19 (1998) s.223-257.
- [9] M. Kubat, Decision Trees Can Initialize Radial-Basis Function Networks, IEEE Transaction On Neural Networks, VOL.9, NO.5, (1998), s. 813-821
- [10] Zhi-Hua Zhou, *Ensemble Methods: Foundations and Algorithms*, CRC Press, (2012). s. 270-272
- [11] Ian H. Witten, Eibe Frank, *Data Mining: Practical Machine Learning Tools and Techniques*, 2nd Edition, Elsevier, (2005). s. 196-198
- [12] Myatt, G.J, *Making Sence of Data: A Practical Guide to Exploratory Data Analysis and Data Mining*, Wiley, (2007). s. 120-129.