T.C.

GEBZE TEKNİK ÜNİVERSİTESİ

SOSYAL BİLİMLER ENSTİTÜSÜ

YAZILIM PROJELERİ PERSONELİ SEÇİMİNDE YAPAY SİNİR AĞLARI TEKNİĞİNİN KULLANILMASI VE BİR UYGULAMA

Gülay GENÇ EROĞLU

YÜKSEK LİSANS TEZİ

İŞLETME ANABİLİM DALI

GEBZE

2020

T.C.

GEBZE TEKNİK ÜNİVERSİTESİ

SOSYAL BİLİMLER ENSTİTÜSÜ

YAZILIM PROJELERİ PERSONELİ SEÇİMİNDE YAPAY SİNİR AĞLARI TEKNİĞİNİN KULLANILMASI VE BİR UYGULAMA

Gülay GENÇ EROĞLU

YÜKSEK LİSANS TEZİ

İŞLETME ANABİLİM DALI

Tez Danışmanı

Prof. Dr. Bülent SEZEN

GEBZE

2020

GEBZE TEKNİK ÜNİVERSİTESİ	**YÜKSEK LİSANS JÜRİ ONAY FORMU**

GTÜ Sosyal Bilimler Enstitüsü Yönetim Kurulu'nun 01/06/2020 tarih ve 2020/12 sayılı kararıyla oluşturulan jüri tarafından 12/06/2020 tarihinde tez savunma sınavı yapılan GÜLAY GENÇ EROĞLU'nun tez çalışması İŞLETME Anabilim Dalında YÜKSEK LİSANS tezi olarak kabul edilmiştir.

JÜRİ

ÜYE
(TEZ DANIŞMANI) : Prof. Dr. Bülent SEZEN

ÜYE : Doç. Dr. Esin SADIKOĞLU

ÜYE : Doç. Dr. Rüya ŞAMLI

ONAY

Gebze Teknik Üniversitesi Fen Bilimleri Enstitüsü Yönetim Kurulu'nun
01/06/2020 tarih ve 2020/12 sayılı kararı.

ÖZET

Yazılım projeleri, kurumun ihtiyaçlarını karşılamaya veya süreçlerini iyileştirmeye yönelik, belli bir kapsam ve bütçe dâhilinde, başlangıç ve bitiş tarihi belli olan, belirli hedeflere sahip teknoloji yaşam döngüsüdür. Yazılım projelerinin başarısız olmasının en önemli sebeplerinden biri doğru personelin doğru projelere konumlandırılmamasıdır.

Bu çalışmada daha spesifik ve artık olmazsa olmaz bir alan olan yazılım projeleri için çalışacak kişilerin uygunluğunu ölçecek şekilde Makine Öğrenimi teknikleri kullanarak, proje başarısını arttırmak ve başarılı sonuçlar elde etmek için göz önüne alınması gereken parametreler önerilmiştir.

Yazılım projelerinde çalışacak personelin seçimi için çoklu kriterlerin kullanılacağı bir yöntem ile ilerlemek, projelerin başarılı olarak sonuçlanması için gerekli parametrelerden biri olan kaynak probleminin çözümlenmesini kolaylaştıracaktır. Literatürde tanımlı ve projenin başarılı olmasını etkileyen; Kritik Başarı Faktörleri'ne (KBF) dayalı olarak projenin başarısız olmasının analiz edilen kök-nedenleri de dikkate alınarak anket oluşturulmuştur. Veri analizinde kullanılan veriler, anketteki soruların cevabını kapsamaktadır. Anketimiz banka yazılım projelerinde çalışmış 105 katılımcı tarafından doldurulmuştur. Ankette çalışılan projenin özellikleri ve ilgili projede çalışmış personelin yetenekleri toplanmıştır. Projeler benzerliklerine göre gözetimsiz öğrenme tekniği ile sınıflandırılmıştır. Sınıflandırma için k-ortalama (k-means) kümeleme algoritmasından yararlanılmıştır. Sonuçta yeni geliştirilecek proje değerleri sisteme girildiğinde, projenin sınıflandırılmasının doğru olduğu görülmüştür. Yeni projelerde çalışabilecek adayların, bu sınıflandırmaya uyumluluğu makine öğreniminin öğrenme yöntemlerinden biri olan gözetimli öğrenme tekniği ile belirlenmiştir. Adayı yeni geliştirilecek projeye seçebilmek ve sistemi eğitmek için gözetimli öğrenme metotlarından Yapay Sinir Ağları'ndan (YSA) faydalanılmıştır. Bu analiz sonucunda ilgili proje için çalışacak adayların daha önce de aynı kategoride çalışan adaylardan seçildiği, yapılan sağlama testleriyle izlenmiştir.

Anahtar Kelimeler: Yazılım Proje Yönetimi Riskleri, Personel Seçimi, Veri Analizi, Yapay Sinir Ağları, K-ortalama, Makine Öğrenimi

SUMMARY

Software projects are a technology lifecycle with specific objectives, with a start and end date, within a scope and budget, to meet the requirements of the organization or to improve its processes.

In the study, we proposed the parameters that should be considered in order to increase project success and achieve successful results by using machine learning techniques to measure the suitability of people who will work for software projects.

Defined in the literature and affecting the success of the project; based on the Critical Success Factors a questionnaire was created taking into account the root causes of the project's failure. The data used in the data analysis includes the answers to the questions in the survey. The characteristics of the project and the capabilities of the employee who worked in the project were collected. The projects were classified according to their similarities with unsupervised learning technique. The k-means clustering algorithm was used for classification. The applicability of new projects to this classification was determined by supervised learning technique which is one of the learning methods of machine learning. Supervised learning methods and artificial neural networks algorithm were used to select the candidate for the new project and to train the system. As a result of, it was observed that the candidates who will work for the related project were selected from the candidates who worked in the same category before.

Keywords: Software Project Management Risks, Employee Selection, Data Analysis, Artificial Neural Networks, K-Means, Machine Learning

TEŞEKKÜR

Yüksek Lisans tezimi ve makalemi hazırladığım süreçte, değerli bilgilerini benimle paylaşan, desteğini esirgemeyen ve ne zaman danışsam bana kıymetli zamanını ayırıp tecrübesiyle ve güler yüzüyle yanımda olan kıymetli sayın tez danışmanım Prof. Dr. Bülent Sezen'e teşekkürü bir borç biliyor ve şükranlarımı sunuyorum. Yine çalışmamda özellikle bilgisayar mühendisliği mesleği ve istatistik bilimleri alanındaki tecrübesiyle bana ışık olan, pratik öneriler sunan ve rahat bir çalışma ortamı sağlayan sevgili eşim Çağrı EROĞLU'na, çocukluğumdan bu yana sevgi ve saygı kelimelerinin anlamlarını bilecek şekilde yetiştirerek beni bu günlere getiren ve benden hiçbir zaman desteğini esirgemeyen bu hayattaki en büyük şansım olan aileme sonsuz teşekkürler.

İÇİNDEKİLER

ÖZET i

SUMMARY ii

TEŞEKKÜR iii

İÇİNDEKİLER iv

SİMGELER ve KISALTMALAR DİZİNİ vi

ŞEKİLLER DİZİNİ vii

TABLOLAR DİZİNİ viii

1.GİRİŞ 1

2. LİTERATÜR TARAMASI 2

2.1. Personel Seçimi ile İlgili Literatür Taraması 2

2.2. Yazılım, Yazılım Mühendisliği ve Yazılım Proje Yönetimi 5

3. METOTLAR 8

3.1. Anket 8

3.1.1 Proje Değerlendirme Anket Soruları 8

3.1.2 Aday Değerlendirme Anket Soruları 12

3.2. Veri Analizi 19

3.2.1. Anket değerlendirme 19

3.2.2. Makine Öğrenimi 22

KAYNAKLAR 30

EKLER 34

Ek A: Tez Çalışması Kapsamında Yapılan Yayınlar 34

Ek B: Proje Değerlendirme Anket Soruları 35

Ek C: Aday Değerlendirme Anket Soruları 39

Ek D: K-ortalama Kümeleme Algoritması ile Projelerin Gruplanmasını Sağlayan Python Kodu 44

Ek E: YSA ile Projeye Uygun Adayın Seçimini Sağlayan Python Kodu 47

ÖZGEÇMİŞ **49**

SİMGELER ve KISALTMALAR DİZİNİ

Simgeler ve Kısaltmalar		**Açıklamalar**
AKDS	:	Akıllı Karar Destek Sistemini
BT	:	Bilgi Teknolojileri
İK	:	İnsan Kaynakları
PCA	:	Principal Component Analysis (Temel Bileşen Analizi)
SVM	:	Support Vector Machine (Destek Vektör Makinesi)
YSA	:	Yapay Sinir Ağları

ŞEKİLLER DİZİNİ

Şekil No:	**Sayfa**
3.1: Anketi Cevaplayan Katılımcıların Unvana Göre Yüzdelik Dağılımı	20
3.2: Projelerin 2 Boyutlu Uzayda Dağılımı	24
3.3: Öğrenme Grafiği	28

TABLOLAR DİZİNİ

Tablo No:	**Sayfa**
3.1: Croanbach's Alfa Güvenilirlik Analizi Sonuçları	21
3.2: Faktör Analizi Sonuçları	21

1. GİRİŞ

Yazılım projelerinin başarısız olmasının en önemli sebeplerinden biri personel seçiminde gerekli hassasiyetin gösterilmemesidir. Her ne kadar yazılım projesi başarısızlıkları ve nedenleri geniş çapta çalışılmış olsa da yazılım projesinin başarısız sonuçlanmasına neden olan kriterlerden personel seçiminin ne şekilde yapılacağına dair bir çalışma geniş çerçeveli olacak şekilde sunulmamıştır. Özellikle Bilgi Teknolojileri (BT) personeli seçimi insan kaynakları departmanı için hâlâ tam olarak anlaşılmış bir alan değildir. Bundan dolayı yazılım projelerinde çalışacak personelin seçimi için çoklu kriterlerin kullanılacağı bir yöntem ile ilerlemek, projelerin başarılı olarak sonuçlanması için gerekli parametrelerden biri olan kaynak probleminin çözümlenmesini kolaylaştıracaktır.

Bu çalışmanın amacı banka sektöründe geliştirilecek yazılım projelerinin başarılı tamamlanmasını sağlayacak şekilde, Yapay Sinir Ağları (YSA) ve makine öğrenimi teknolojilerini kullanarak, projede çalışacak yazılım uzmanının objektif olarak seçilmesine öneri niteliğindedir. YSA tekniği kullanılarak yapılacak veri analizi sayesinde yeni alımlar için yeni adaylar veya mevcut çalışanlar arasında da proje için uygun adaylar objektif ve objektife yakın bir şekilde seçilebilecektir. Çünkü makine öğreniminde; bir bilgisayar, bir sınıflandırma kararının kendi temsilini öğrenecektir. Bunu da sonuç olarak oluşturduğunda, çıktı insan anlayışı dikkate alınmadan yapılmış olacaktır (Burrell, 2016). Kurumların çalışılacak proje için eşleştirecek kaynağı veri analizi yöntemi ile objektif veya objektife yakın bir şekilde seçmesi işverenlerdeki iş yükünü azaltacak ve şirketlerin hem görünür hem de görünmez maliyetlerden tasarruf etmesini sağlayacaktır. Çalışmada yazılım projelerinin riskleri göz önüne alınarak bankada tamamlanmış veya yeni başlanacak yazılım projeleri baz alınmıştır.

Bu çalışmanın kalan kısmı aşağıdaki gibi yapılandırılmıştır. Bölüm 2'de ilgili çalışmalar açıklanmıştır, Bölüm 3'te, anket ve veri analizi hakkında detaylı bilgiler verilmiştir, Bölüm 4, bu çalışmanın katkılarını ve sonuçlarını ifade etmektedir.

2. LİTERATÜR TARAMASI

2.1. Personel Seçimi ile İlgili Literatür Taraması

Personel seçimi iş hayatında şirketler için en önemli unsurlardan biridir. Doğru personel seçimi, personel verimliliği ve iş sürekliliği açısından önemlidir. Personel seçimi kadar önemli olan bir diğer konu da alınan personelin doğru görevde konumlandırılmasıdır. Özellikle yazılım sektöründe doğru personelin seçilmesi, yazılım projelerinin zamanında ve beklentiyi karşılayacak şekilde bitmesini sağlamaktadır. Personel seçimi problemlerine literatürde bahsedilmiş olup, yazılım projelerine YSA tekniği kullanılarak personel seçimi süreciyle ilgili bir çalışma, spesifik olarak yapılmamıştır. Yazılım projeleri için personel seçiminin doğru, objektif veya objektife yakın bir şekilde yapılması, projelerde doğru adayların çalışarak projelerin beklenen zamanda ve ihtiyaç gereksinimlerine uygun olarak tamamlanmasını sağlayacak; işverenlerdeki iş yükünü azaltarak şirketlerde de hem görünür hem de görünmez maliyetlerden tasarruf edilmesine olanak tanıyacaktır.

Personel seçimiyle ilgili bir çalışmada şirket koşullarını dikkate alarak doğru personel seçiminin, işgücü maliyetlerini optimize etmek ve kurumsal hedeflere ulaşmanın insan kaynaklarının öncelikli hedefi olduğundan bahsedilmiştir (Bedir ve Eren, 2015). İnsan kaynakları yönetiminde personel seçim işlevinin amacı, çalışan yetenekleri ile işin gerekleri arasındaki en iyi ilişkiyi kurmak için; çalışmak üzere kuruma başvurmuş adayların, kişiliklerini, almış oldukları eğitimi, yeteneklerini ve ilgi duydukları konuları inceleyerek, adaylar arasından en uygun yeteneklere sahip olanları göreve çağırmaktır.

Yüksek teknoloji şirketlerinin personel seçimi problemlerine alternatif bir bakış açısı getirilmek istenen bir çalışmada, geleneksel insan kaynakları yönetimi yaklaşımlarına ek olarak, en uygun olan yetenekleri bulmak için etkin personel seçim mekanizması geliştirmeye acil ihtiyaç olduğu vurgulanmıştır (Chien and Chen, 2008). İlgili çalışmada personel seçiminde yararlı kurallar oluşturmak için karar ağacı ve ilişkilendirme kurallarına dayanan bir veri madenciliği çerçevesi geliştirerek boşluğun doldurulması amaçlanmıştır (Chien and Chen, 2008). Ampirik çalışma, yarı iletken endüstrisinde bu yaklaşımın pratik olarak uygulanabilir olduğunu göstermiştir.

Alternatif veri madenciliği teknikleri olarak YSA tekniğinin kullanılarak gelecekteki araştırmalarda çeşitli yaklaşımları karşılaştırmak için incelenmesi yönünde öneride bulunmuştur. Ayrıca, işletmeciler veya yönetim seviyesi işleri gibi diğer işlere ve insan sermayesini geliştirmek için eşleştirilen yetenekleri bulmak için diğer sektörlere de uygulanabileceğini savunmuştur (Chien and Chen, 2008).

İnsan Kaynakları'nda (İK) karar verme sürecini geliştirmek için Akıllı Karar Destek Sistemini (AKDS) ve Veri tabanında Bilgi Keşfi uygulayan Akıllı İnsan Kaynakları Bilgi Sistemi çerçevesiyle İK sorunlarına çözüm getirilmesi önerilmiştir (Masum et al., 2018). Bilgilerin işlenmesi zor olduğu için de sadece bilgi-tabanlı sistem ile formal olmayan bilgiyi yakalamanın zor olduğu vurgulanmıştır. Bilgi Tabanlı Sistem, karmaşık problemleri çözmek için kullanılan ve bilgilerin bulunduğu veri tabanını kullanan bir bilgisayar programıdır. Hibrit teknikler kullanımının İK AKDS için en etkili yöntem olabileceği vurgulanmıştır. Bu tezde daha spesifik ve artık olmazsa olmaz bir alan olan yazılım projeleri için çalışacak kişilerin uygunluğunu ölçecek şekilde, hibrit teknikleri kullanarak proje başarısını arttırmanın ve başarılı sonuçlar elde etmek için göz önüne alınması gereken parametreler de önerilmiştir.

BT personelinin iş değişikliğini azaltmak ve personelin elde tutulmasının projenin başarısı üzerinde önemli bir etkisi olması sebebiyle, personelin motivasyon düzeyleri iyileştirilmelidir (Hall et al., 2008). Çünkü işe alım maliyetinin yanı sıra doğru personelin projeye dâhil edilmemesinin beceri, bilgi ve deneyim kaybına bağlı olarak dolaylı maliyetleri de olacaktır.

BT'nin ve dolayısıyla BT çalışanlarının bir şirketin gelişimi ve büyümesi için çok önemli bir faktör oluşturduğunu çalışmasında vurgulayan Kelemenis ve diğerleri. (2010), aynı zamanda BT personelinin seçiminin firmanın özel hedeflerine, BT araçlarının kullanılabilirliğine ve karar vericilerin bireysel tercihlerine bağlı olmasının karmaşık bir sorun teşkil ettiğine de değinmektedir. Sezgisel kararlardan ziyade analitik yöntemlerle personel seçiminin önemi savunulmuştur. Bu yaklaşım, tezin önemini destekler niteliktedir.

Başarılı yazılım geliştirme ve yönetimi sadece kullanılan teknolojilere, yöntemlere ve süreçlere değil, aynı zamanda ilgili insanların yargılarına ve kararlarına

da bağlıdır (Feldt et al., 2010). Yazılım geliştirmede insan faktörlerinin önemine ve özellikle yazılım mühendisliği süreçlerine katılan insanların kişisel özelliklerine olan ilgiye rağmen, bu faktörler büyük ölçüde gözden kaçmış ya da ampirik çalışmalara dayanmamıştır (Feldt et al., 2010). Yazılım sektöründe, işverenlerin yazılım uzmanı adaylarda ince yeteneklerin önemini daha fazla kabul etmesi gerekmektedir (Ahmed et al., 2015). Bu sebeplerden tez kapsamında analitik düşünme, inisiyatif alma ve problem çözme becerisi gibi parametrelerin adaylar için değerlendirilmesi istenmiştir. Projede çalışacak yazılım uzmanı adayının ilgili projeye uygun seçilmesi, projenin başarılı bir şekilde tamamlanması için oldukça önem arz etmektedir. Bu nedenle tezin literatüre katkı sağlayacak bir araştırma olacağı düşünülmektedir.

Yazılım geliştiricilerini daha iyi bir şekilde yeniden değerlendirmek ve üretkenliklerini iyileştirmek için daha iyi bir şekilde desteklemeyi amaçlayan bir araştırmada yazılım geliştiricilerinin verimliliğini nasıl arttırabileceklerinin cevabı aranmıştır (Meyer et al., 2014). Araştırmanın büyük bir kısmı, üretkenliği, saat başına kaynak kodlarının sayısına dayanarak hesaplama verimliliği gibi belirli tanımları ortaya koymuştur. Çünkü teknolojinin hızla geliştiği dünyada verimli çalışacak ve üretebilecek yazılımcılara ihtiyaç gün geçtikçe daha da artmaktadır. Tez kapsamında önerilen yeni çalışılacak proje için uygun yazılım geliştirme uzmanı eşleştirmek ile sektördeki bu açığın kapatılması da hedeflenmiştir.

Purna Sudhakar (2012) çalışmasında, yazılım geliştirme projeleri için kavramsal başarı faktörleri modelini geliştirerek, başarı faktörlerini sınıflandırmış ve her bir kategorideki faktörleri bularak ürün, takım, proje yönetimi ve iletişim faktörlerini vurgulamayı amaçlamıştır. Ayrıca yazılım projelerinin başarılı olarak tamamlanması gereken faktörleri incelemiş proje başarısının anlamını bulmaya çalışmıştır. Yazılım projelerinin başarı oranını artırarak para, zaman, donanım, yazılım, insan kaynakları gibi birçok bütçe kaleminin azaltılabileceğine dikkat çekmiştir.

Bir bireyin bir yazılım ekibindeki rolüne atanması, bu nedenle, bazı kişilik özelliklerinin belirli görevleri yerine getirmek için daha uygun olabileceğinden, proje başarısı için kritik bir faktör olarak görülmektedir (Cruz et al., 2015; Acunã and Juristo, 2014). Ayrıca; yazılım geliştirme pratiğini etkileyebilecek sonuçlar için daha fazla araştırma yapılması yönünde de vurgu yapılmıştır. Yazılım geliştirme için

duyulan doğru yeteneklerin seçimi için literatürde yeterli model bulunmamaktadır. Bu sebeple bu tezdeki çalışmanın da proje başarısını etkileyen faktörlere göre en uygun adayın bulunmasının literatüre özgün bir çalışma katacağı öngörülmektedir.

Yazılım projelerinin başarılı olarak tamamlanması için, başarısızlık nedenlerinin sağlıklı bir şekilde anlaşılması ve ortadan kaldırılması gerekmektedir. Bir yazılım projesi başarısızlığı, insanların, görevlerin, yöntemlerin ve proje ortamının birbiriyle bağlantılı olduğu çok boyutlu bir sürecin sonucudur (Lehtinen et al., 2014).

2.2. Yazılım, Yazılım Mühendisliği ve Yazılım Proje Yönetimi

Yazılım, günümüzde her alanda hayatımıza giren ve hayatımızı kolaylaştıran bir alandır. Yazılım projesi, kurumun ihtiyaçlarını karşılamaya veya süreçlerini iyileştirmeye yönelik, belirli bir kapsam ve bütçe dâhilinde yapılan, başlangıç ve bitiş tarihi belli olan, belirli hedeflere sahip, birbiriyle karmaşık ilişkili görevler ve faaliyetler barındıran teknoloji yaşam döngüsüdür. Yazılım mühendisliği, dünya üzerinde hâlâ yeni denebilecek bir süreçtir. Bir projenin yazılım projesi olarak ele alınabilmesi için tanımlı bir problem veya yeni bir ticaret alanı olması gerekmektedir. Yazılım projeleri yapılacak işe göre, sıfırdan geliştirilecek bir proje veya mevcut yapı üzerinde geliştirilecek bir proje olabileceği gibi, kurumun teknoloji altyapısını tamamen değiştirecek, değişim projesi de olabilir. Yazılım karmaşıklığı, personelin deneyimlerine ve onların zekâsına bağlıdır. Deneyimli ve sağlam bir teknik arka plan programcısı, en az sayıda hata ile kaliteli yazılım geliştirecektir (Yadav and Yadav, 2017). Bu sebeple yazılım mühendisinin yetenekli ve başarılı olması geliştirilecek büyük, kompleks projeler için çok önemlidir.

Bir yazılım projesinin performansı, hedef, maliyet, zaman ve arzulanan ürün kalitesi seviyesi elde etme yeteneği açısından değerlendirilir ve bu nedenle, bir yazılım projesi, verilen zaman ve maliyet dâhilinde önceden mutabık kalınmış kalite düzeyinde tamamlanmışsa başarılıdır (Agarwal and Rathod, 2006).

Purna Sudhakar (2012), açıkça tanımlanmış hedefleri, üst yönetim desteğini, yetkin proje yöneticisini, yeterli iletişim kanallarını, yetkin ekip üyelerini, yeterli kaynak tahsisini, kontrol mekanizmalarını, müşteriye yanıt verme mekanizmalarını ve

geri bildirim mekanizmalarını, yazılım geliştirme projeleri için kritik başarı faktörleri olarak belirlemiştir. Ayrıca bilgi sistemi başarısı için en önemli beş faktörün; çıktıların doğruluğu, çıktı güvenilirliği, çıkışın zamanlaması, gereksinimlerin gerçekleştirilmesi ve sistemlerde kullanıcı güveni olduğundan bahseder. Tez kapsamında yapılan ankette projenin yönetim önceliği, kaynak sayısı, takımla çalışma becerisi şeklinde parametrelere yer verilmiştir.

Yazılım Projelerinin Başarısız Olma Sebepleri

Yazılım projeleri; kapsamın net olmaması, kaynak eksikliği, karmaşıklığın ve bağımlılığın artması sebeplerinden zamanında bitmemektedir ve bu sebeplerle uzayan proje süresinden dolayı bütçe yetmemeye başlar. Projenin bu sebeplere rağmen başarılı olarak tamamlanması için hem değişikliklere hızlıca adapte olunması gerekir hem de karmaşıklıkla başa çıkılabilmesi gerekmektedir (Alakuş vd., 2017; Savolainen et al., 2012). Bu başarılamadığı zaman yazılım projeleri de maalesef başarısızlıkla sonuçlanmaktadır. Projelerin başarılı bir şekilde tamamlanması için, bir proje planının olması, projede yazılım yapacak personelin öz yeterliliğinin beklenen seviyede olması, yeterli sayıda yazılım geliştirme uzmanının olması ve proje çıktısının müşteriyi memnun edecek düzeyde olması gerekmektedir. Dolayısıyla projenin tahmin edilen tamamlanma tarihi değiştirilemiyorsa, projede çalışan özellikle yazılım geliştirme uzmanlarının sayısı, uzmanın yetenekleri dikkate alınarak arttırılmalı ve projenin başarısızlıkla sonuçlanmasını engelleyecek bir bakış açısıyla ilerlenmesi sağlanmalıdır. Başka bir araştırmaya göre de proje başarısızlığı "ya iptal edilmiş ya da hedeflerini karşılamamış" bir projeyi ifade etmektedir (Procaccino et al., 2002; Agarwal ve Rathod, 2006). Bu parametrelerin en önemli başarı kriteri olarak kabul edildiğini ve işlevselliğin, projenin başarısını tanımlamada yazılım kalitesi üzerinden derecelendirildiği belirtilmiştir. Yazılım sistemlerinin başarı ya da başarısızlığına katkıda bulunan faktörlerin başlıcaları şu şekildedir: (1) Proje gereksinimleri, (2) proje yönetimi, (3) müşteriler ve kullanıcılar, (4) tahmini ve zamanlama, (5) proje yöneticisi, (6) yazılım geliştirme süreci ve (7) geliştirme personelinin öz yeterliliği (Lehtinen et al., 2014; Linberg (1999); Procaccino et al., 2002). Başarının, uygulayıcıların proje ile ilgili deneyim kazanmasıyla ilgili olduğu ileri sürülmüştür. Yazılım geliştirme uzmanının başarısı veya başarısızlığı, yazılım geliştirme uzmanının süreçteki kritik rolüne ve proje paydaşları arasındaki benzersiz görüşüne bağlı olarak; yazılım

geliştirme süreci için önemlidir. Yazılım geliştirme uzmanları hem yaptıkları hem de kimlerle etkileştikleri açısından yazılımın tasarımında ve yapımında ön sıradadırlar (Procaccino and Verner, 2002). Literatür taraması kapsamında da görüldüğü üzere geliştirme personelinin ilgili proje için uygun olup olmaması direkt olarak projeyi etkileyen unsurlardan biridir. Bu da tezin faydalı bir çalışma olduğunun en önemli göstergesidir. Ayrıca projelerin başarılı olma kriterlerini bilen ve anlayan bir yazılımcı ile çalışmak her zaman avantajlıdır. Bu sebeple; öz yetenekleri yeterli seviyede olan, müşterinin ihtiyaçlarını anlayan ve iletişimi güçlü olan bir yazılımcı ile projeyi yürütmek projenin başarılı olarak sonuçlandırabilme ihtimalini arttırmaktadır.

Geliştiricilerin bakış açısına göre başarılı bir proje/proje yöneticisi, kendileri için yeterli zaman ayıran müşteriler/kullanıcılar/proje sponsoru ile iyi tanımlanmış bir proje kapsamına bağlıdır. Yönetimin bakış açısının daha çok müşteri ve kullanıcılara odaklı olması, yazılım geliştirme uzmanlarının motivasyonlarını düşürmektedir. Bu sebeple BT yöneticilerinin yazılım geliştirme uzmanlarının motivasyon kaybını önleyecek şekilde mutlu olmalarını sağlayacak bir yönetim bilincinde olması gerekmektedir (Procaccino et al., 2002). Geliştirme uzmanlarının ve üstlendikleri görevlerin yönetimi ne kadar iyi olursa geliştirme uzmanlarının motivasyonu o kadar yüksek olur. Geliştirme uzmanının motivasyonu, başarılı yazılım geliştirmenin genel kabul görmüş yönetim ve organizasyonel bakışının başarısı için kritik öneme sahiptir. Teknolojik olarak başarılı olduğu kadar iletişim becerisinin güçlü ve ekiple uyumlu bir çalışma becerisine sahip olması da adaylarda aranan bir diğer yetenektir.

Tez kapsamında ankette yazılım projesinin başarılı tamamlanmasına vesile olacak faktörlere yer verilmiştir. Ayrıca bundan sonraki bölümde veri analizi için kullanılan metotlar hakkında bilgi verilmiştir.

3. METOTLAR

3.1. Anket

Literatürde tanımlı ve projenin başarılı olmasını etkileyen; Kritik Başarı Faktörleri'ne dayalı olarak projenin başarısız olmasının analiz edilen kök-nedenleri de dikkate alınarak anket oluşturulmuştur. Veri analizinde kullanılan veriler ankette sorulan soruların cevabını kapsamaktadır. Anket banka sektöründe çalışan 105 banka yazılım projelerinde çalışmış kişiler tarafından doldurulmuştur. Ankette yer alan veya almayan parametreler sebepleri ve referanslarıyla birlikte tek tek açıklanmıştır. Ankette yer alan sorulara **Ek B: Proje Değerlendirme Anket Soruları ve Aday Değerlendirme Anket Soruları** başlıklarıyla **EKLER** ana başlığı altında yer verilmiştir.

3.1.1 Proje Değerlendirme Anket Soruları

Unvanınız: Anketi cevaplayan katılımcıların unvan bilgisini de alarak, profillerine tez kapsamında yer verilmek istenmiştir. Bundan sonraki akademik çalışmalara katkı sağlamak amaçlanmıştır. Anketi dolduran katılımcıların Proje Yöneticisi, BT İş Analisti, BT Analiz Müdürü, Yazılım Test Uzmanı, Yazılım Geliştirme Uzmanı, Yazılım Geliştirme Müdürü ve Diğer şeklinde unvanlardan birini seçmesi istenmiştir.

Proje Süresi: Yazılım projesinin tamamlanması için planlanan zaman anlamına gelmektedir. Proje süresi projenin gereksinimlerine, projede çalışacak kişi sayısına, proje paydaşlarına ve projede çalışacak kişilerin proje deneyimlerine istinaden belirlenir (Yazgan ve Sönmez, 2015). Projenin ne zaman tamamlanacağının bilinmesi, projede çalışacak paydaşlar için bir hedeftir. Uzun soluklu projelerde başarılı ve verimli çalışmak için, projede çalışacak yazılım geliştiricilerinin yüksek motivasyona sahip, iletişim becerilerinin güçlü ve takım çalışmasına uygun olması gerekmektedir. Proje süresi, geliştirilecek projenin süresine bağlı olarak çalışabilecek adayları seçmek için önemli bir kriterdir.

Proje Karmaşıklığı: Yazılım projesi başarısızlığının sebebi, genellikle proje karmaşıklığı ile ilişkilidir (Lehtinen et al., 2014). Proje karmaşıklığı arttıkça, projenin yönetilmesi daha zor hale gelmektedir. Çünkü projede karmaşık olan sadece ürün ve ihtiyaçlar değil, aynı zamanda yeni donanım, protokol ve projede çalışacak kaynakların yönetimi de ayrı bir yetenek istemektedir. Bu sebeple proje karmaşıklığı kriteri, geliştirilecek proje için önem arz etmektedir. Karmaşık projede çalışan bir aday, yeni geliştirilecek bir yazılım projesi için uygun bir aday olacaktır.

Projede Çalışan Yazılım Uzmanı Sayısı: Yazılım projelerinin başarılı olarak tamamlanmasında en önemli kriterlerden biri de ilgili projede çalışacak yazılım uzmanı sayısıdır. Kaynakların yeterliliğine dair algılar, üstlendikleri projelerin içsel değeri hakkındaki inançlara yol açarak insanları psikolojik olarak etkiler (Linberg, 1999). Yazılım projelerinde ilgili proje planı oluşturulurken, gereksinimlere ve kapsama göre projede çalışacak kişi sayısı belirlenir. Projenin büyüklüğüne göre, projede çalışacak proje yöneticisi, analist, yazılımcı ve test uzmanları belirlenir. Bu soru yeni projede değerlendirilecek adayın, çalıştığı projenin büyüklüğü hakkında bilgi verecektir.

Projede Dış Kaynak (Outsource) Eleman Çalışıyor Mu?: Bazı projeler büyüklüklerine, bankada olmayan teknolojik gereksinimlerine göre dışarıdan danışmanlık alabilir. İlgili adayın bu danışmanlarla çalışıp çalışmadığı bu anket sayesinde öğrenilerek, geliştirilecek yeni projeye uygun bir aday olup olmadığının anlaşılması adına ayrı bir öneme sahiptir. Tedarikçinin temel amaçlarından birinin gelecekte ek işler elde etmek olduğunu ve bu nedenle mevcut faaliyetlerin tedarikçinin bakış açısından başarısının ana kriterlerinden birinin müşteriyle yapılacak gelecekteki anlaşmaların görünümü olduğunu gösterdiğini ifade etmiştir (Savolainen et al., 2012). Ancak bu parametre faktör ve güvenilirlik analizine göre ankette yer almaması gereken bir parametre olduğundan anketten çıkarılmıştır.

Projede Çalışan Toplam Kişi Sayısı: Yazılım projelerinde ilgili proje planı oluşturulurken, gereksinimlere ve kapsama göre projede çalışacak kişi sayısı belirlenir. Projenin büyüklüğüne göre, projede çalışacak proje yöneticisi, analist, yazılımcı ve test uzmanları belirlenir (Yazgan ve Sönmez, 2015). Proje sürecinde kapsamın değişmesine, projenin önceliğine göre çalışacak kişi sayısı azaltılabilir veya

arttırılabilir. Projenin ek yazılım personeli veya ek kaynaklar almayacağı konusunda bir algı olduğunda, yazılım geliştiricilerinin projenin amaçlarına tam olarak bağlı kalmayacak şekilde motive olabileceğinden bahseder. Bu da proje sürecini olumsuz yönde etkiler.

Proje Kapsamı İyi Tanımlanmış Mıydı? Proje kapsamı, geliştirilecek projenin çerçevesini, çıktıdan beklentileri net bir şekilde ortaya koyar. Proje kapsamında dahil olacak ve olmayacak konuları belirler. Proje kapsamı iyi tanımlanmışsa, yönetim ve geliştiricilerin projeyi başarı olarak görmeleri daha olasıdır (Procaccino et al., 2002; Agarwal and Rathod, 2006). Kapsam sürüncemesinin yaşanmaması, ilgili yazılım projesinin başarılı olarak tamamlanmasını sağlayacak en önemli ilk faktör olarak varsayılabilir.

Projenin Organizasyondaki Önceliği: Projenin organizasyondaki öncelik algısı projenin başarılı bir çıktısının oluşmasını önemli ölçüde etkilemektedir. Önceliği ne kadar yüksekse proje ekibinin projeye verdiği önem ve destek o noktada artmaktadır. Linberg (1999) de çalışmasında yüksek öncelikli projelere verilen desteğin ve iş birliğinin daha fazla olduğunu belirtmiştir. Yeni geliştirilecek proje yüksek öncelikli ise, önceliği yüksek projelerde çalışmış adayların değerlendirilmesi uygun olacaktır. Çünkü önceliği yüksek projenin bir an önce hayata geçirilmesi istenir. Bu sebeple bu deneyimde çalışan kaynakların yeni geliştirilecek proje için tercih edilmesi uygun olacaktır.

Projede Riskler Göz Ardı Edilmemiştir: Yeni BT uygulamalarının başarılı bir şekilde uygulanmasını kolaylaştırmak için, proje riski sorunları ele alınmalı ve çoklu paydaşların beklentileri uygun şekilde yönetilmelidir. Kullanıcıların ve geliştiricilerin proje riski ile ilgili farklı algıları barındırabileceği ölçüde, çatışma alanları ortaya çıkabilir. Kullanıcılar ve geliştiriciler proje risk faktörleri konusunda farklı algılara sahiplerse, çatışmaların ortaya çıkma potansiyeli artmış olabilir (Keil et al., 2002). Bu sebeple her iki paydaş için de risklerle ilgili bir strateji geliştirmek projenin geleceği için önem kazanmaktadır. Whittaker (1999) çalışmasında risk yönetiminin yetersiz ve proje planının zayıf olma durumunun başarısızlığa sürüklediğinden bahsetmiştir. Risk yönetim sürecindeki ilk adım, riskin kendisini tanımlamaktır. Bazı riskler proje boyunca sabit kalabilir ve riski gerçekten etkisi oluştuğu anda

gözlemlenebilir. Bu risklerin bazıları henüz yazılım geliştirme çalışmalarına bile başlamadan belli olabilir, bazıları ise geliştirme veya test sırasında belli olabilir. Bu sebeple risklerin yönetilme şekline göre ilgili projelerde çalışan uzmanlar, yeni geliştirilecek proje türüne göre karşılaşılacak riskleri yönetme açısından fayda sağlayacaktır.

Proje, Proje Planına Uygun Bir Şekilde Tamamlanmıştır: Proje planı, yürütülecek proje için hedef çizelgesidir. Planlama eksikliği büyük bir olasılıkla projeyi başarısızlığa sürükleyecektir (Dvir et al., 2003). Planlama aynı zamanda proje çıktısının fonksiyonel özelliklerine odaklanmaktadır. Aslında hedeflenen ürünün çıktısı projenin planlama evresinde resmedilir. Ayrıca proje planının hazırlanma evresinde, projede çalışacak kaynak sayısı, kullanılması gereken teknolojiler, proje kapsamı, bütçe, proje süresi belli olacağından, proje planlama evresi proje yönetimi açısından önemli bir evredir. Dahası proje yönetim seviyesinde, teknik çalışmanın etkin bir şekilde ilerlemesine izin vermek için yapılması gereken faaliyetlerin ve süreçlerin planlanması üzerinde durulmaktadır. Proje yönetimi mesleki standartları kılavuzlarında planlama projedeki belirsizliği azalttığından, proje başarısının olasılığını artırdığı varsayılır. Eğer tarih gerçekçi değilse proje başarısız olur. Yapay son tarihler vermek proje riskini arttırır (Schmidt et al., 2001). Bu sebeple test zamanı ve eğitim süresini azaltmadan projenin tamamlanması için gerçekçi bir proje planı tanımlanmalıdır.

Proje Gereksinimleri Tamamen Anlaşılmıştır: Gereksinimler projelerin kapsamının belirlenmesini sağlayan, projenin en önemli yapı taşlarından biridir. Projenin tamamlanma tarihi uzadıkça projenin gereksinimleri değişecektir. Yeni sistemin gerekliliklerini tam olarak tanımlayabilmek için gereksinimlerin iyice anlaşılması gerekmektedir. Çünkü gereksinimlerin yanlış anlaşılması; yeni sistemin gerekliliklerini tam olarak tanımlamamakta, sonuç olarak projeyi tamamlamak için gerekli olan gerçek çalışma çabasını, beceri setlerini ve teknolojiyi anlamamaya neden olmaktadır (Schmidt et al., 2001). Ayrıca alan bilgisi eksikliği eksik gereksinim tanımına yol açar. Bu sebeple proje geliştirmede çalışacak kaynakların yeteneklerinin aslında projenin başarılı bir şekilde tanımlanması için birebir etkisi olduğu da bu faktörle destekleyici niteliktedir.

Kullanıcıların Projeye Katılım Seviyesi: Proje hedeflerinin ve çıktılarının gereksinimlerini doğru bir şekilde tanımlamak için, müşteri veya son kullanıcı müdahalesinin projenin başından sonuna kadar olması gerekir (Dvir et al., 2003). Gereksinimlerin ve şartnamelerin iyice anlaşılıp geliştirilmesi, son kullanıcı ile sıkı iş birliğine bağlıdır. Bu yüzden kullanıcı ile projenin her aşamasında yakın çalışmak önemlidir. Literatürde kullanıcıların projeye katılım seviyesi yüksek olan projelerin başarı oranının daha yüksek olduğu savunulmuştur (Dvir et al., 2003).

Projenin Bütçe Seviyesi: Proje başarısı, bütçesini, teslimatını ve iş hedeflerini karşılayan bir proje olarak tanımlanmaktadır (Linberg, 1999). Projenin bütçe seviyesi, projenin kapsamından, teknolojiye yapılacak yatırıma, kaynak sayısına, projenin büyüklüğüne ve projeye verilecek öneme kadar bize birçok ipucu verir (Linberg, 1999; Whittaker, 1999; Schmidt et al., 2001). Bu sebeple adayların daha önce çalıştığı projelerin bütçe seviyesini öğrendiğimiz veriyle, yeni geliştirilecek projelere doğru adayın eşleştirilmesi açısından önem arz etmektedir. Ankette bu kriter "düşük bütçe", "orta bütçe" ve "yüksek bütçe" kriterleriyle kullanılmıştır.

Gereksinimlerin Organizasyonun Teknolojisine Uygunluğu: Gereksinimlerin gerçekçi olmadığına dair bir algı olduğunda, yazılım geliştiricileri cesaretini kaybedebilir ve projenin amaçlarına tam olarak bağlı kalmayabilir. Gerçekçi olmayan gereksinimlerin tehlikeleri, gerçekçi olmayan mükemmeliyetçiliğin tehlikelerine benzer (Linberg, 1999). Maslow (1971), gerçekçi olmayan mükemmeliyetçiliğin tehlikelerini şöyle tanımlar: "Buradaki yaygın bir sıralama, hayal kırıklığına yol açan kaçınılmaz başarısızlığa yol açan, tüm ideallere ve tüm normatif umutlara ve çabalara karşı ilgisizlik, cesaret kırıcı ya da aktif düşmanlığa yol açan mükemmeliyetçilik ya da gerçekçi olmayan beklentilerdir." Bu sebeple, yazılım projesinin başarılı bir şekilde tamamlanması isteniyorsa, gereksinimlerin de organizasyonun teknolojisine uygun olması önem arz etmektedir.

3.1.2 Aday Değerlendirme Anket Soruları

Yaşınız: İnsanların yaşları arttığında, yani deneyim kazandığında dünyaya daha uyumlu yaşamaya başlamaktadır (Feldt et al., 2010). Yaş aldıkça mühendislerin

acemilikleri azalmakta, profesyonellikleri artmaktadır. Daha küçük yaştaki personeller ise birden çok işte çalışmaya isteklidir (Feldt et al., 2010). Projeye göre projede çalışması istenen yaş aralığı değişiklik gösterir. Bu sebeple yaş projede çalışacak uygun adayın seçimi için önemli bir kriterdir.

Cinsiyetiniz: Kadınların, mesleki statülerini arttırmanın önemli bir başarı faktörü olduğu konusunda erkeklere oranla daha çok hemfikir olduğu anlaşılmış ve buna sebebiyet veren durumun erkek egemen bir sektörde çalışmanın getirdiği kaygının sebep olabileceği ihtimali üzerinde durulmuştur (Verner ve Procaccino, 2002). Yazılım sektöründe çalışan yazılım uzmanı personelinin büyük çoğunluğu erkekten oluşmaktadır. Bunun birkaç sebebi bulunmaktadır. Erkeklerin kadınlardan daha iyi kod yazma inanışı birinci sebeptir. Projenin kritikliğine göre yeni evli kadınların hamile kalma olasılığı daha yüksek olduğundan projede tercih sebebi olmamaları yazılım sektöründe çalışan kadın oranını azaltmaktadır. BT mesleğinde cinsiyet konusunun incelendiği çalışmada kadınlar, yöneticiler tarafından teknik olarak eşit derecede vasıflı kişiler olarak algılanırken, "stresli / baskılı" ortamlarda daha düşük iş tatmini bildirmişlerdir (Gallivan, 2004). Ama aynı şekilde teknik beceri konusunda erkekler ile kadınların arasında bir fark görülmediği de belirtilmiştir (Gallivan, 2004; Verner et al., 2002). Anket analizinde bu faktör varyasyon oluşturması sebebiyle dikkate alınmamıştır. Çünkü iki cinsiyet arasında herhangi bir üstünlük yoktur.

Eğitim Seviyeniz: Daha yüksek bir eğitim düzeyine sahip olan yazılım uzmanları, bütçe dâhilinde bitirilen bir projeye daha fazla önem vermektedirler (Verner and Procaccino, 2002). Ancak eğitim düzeyi yüksek olanın yazılım geliştirme projelerinde daha başarılı olduğunu vurgulayan bir çalışmaya rastlanmamıştır.

Projede Kullandığı(nız) Diller: Ankette değerlendirilen veya anketi değerlendiren adayın projede kullandığı dil bilgisini öğrenmek amacıyla sorulan araştırma sorusudur. Projenin karmaşıklık seviyesi ve büyüklüğü, yazılım uzmanın o projede kullandığı dil(lerin) bilgisi bize adayın programlama becerisi ve deneyimi hakkında ipucu verir. Deneyimli ve sağlam bir teknik arka plan programcısı, en az sayıda hata ile kaliteli yazılım geliştirir (Yadav ve Yadav, 2017).

Projede Kullanılan/kullanılacak Diller: Bilgisayarlarda yazılım projesi geliştirebilmek adına, elektronik cihazın anlayacağı şekilde program geliştirmek gerekmektedir. Programlama dili bu amaçlara hizmet etmek için hazırlanmış özel kelime, semboller, komutlar, kütüphaneler ve yazım kuraları bütünüdür. Her dilin yazılım kuralı farklıdır (Ersoy vd., 2011). Yazılım projesinde kullanılan diller yazılım projesinin gereksinimlerine göre farklılık gösterir. Projede kullanılmış veya kullanılacak dilleri öğrenmek için ankette yer alan bu sorumuz adayın da projede kullanılacak dili bilip bilmediğini kıyaslamaya imkân tanıyacaktır. Ankette yer alan dillere sektörde yaygın kullanımına göre karar verilmiştir. JAVA, C#, C, C++, PL/SQL, MySQL, JavaScript, PHP ve diğer (Aruoba and Fernández-Villaverde, 2015; Tilkov and Vinoski, 2010; Prechelt, 2000; Walia and Gill, 2014; Boicea, 2012). Aruoba ve Fernández-Villaverde (2015) çalışmasında Mac'te ve bir Windows bilgisayarda kodların yürütme sürelerini ekonomi sektöründeki hesaplara göre incelemiş ve her dilin ilgili çalışmaya bağlı güçlü ve zayıf yönlerini kısaca yorumlamış ve geliştirilecek algoritmaya göre bu değerlerin değişebileceğini de vurgulamıştır. Bu sebeple projede kullanılacak dil önemlidir.

Aktif Olarak Kullanıdığı(nız) Diller: Adayın çalışacağı projede kullanılacak dilleri bilip bilmediğini öğrenmek ve programlama kabiliyeti hakkında veri oluşturmak amacıyla ankette yer verilen bir sorudur. Adayın iyi bir yazılım geliştirebilmesi için, talepleri gerçekleştirebilecek algoritmaları kurgulama yeteneğinin yüksek olması gerekmektedir (Ersoy vd., 2011). Yazılım geliştirmede önemli olan, problemlerin sebebinin anlaşılarak, hızlıca çözüme kavuşturma yeteneğine sahip olmaktır. Adayın projede kullanılacak dili biliyor olması, projeye adaptasyonu hızlandıracak ve deneyimi sayesinde geliştirme süresini kısaltacaktır.

Yazılım Geliştirme Araçları Kullanımı: Yazılım geliştirici için kalite araçları önemlidir. Bu araçlar arasında yazılım editörleri, derleyiciler, hata ayıklayıcılar, bağlayıcılar, kod çözümleyicileri, yapılandırma kontrol sistemleri, problem raporlama sistemleri, e-posta vb. bulunur. Yazılım geliştirme araçlarının kalitesi ve gelişmiş olanlarının kullanımı zamanı daha verimli kullanılmasına olanak tanır ve hatalı kod yazılmasını önlemeye yardımcı olur (Linberg, 1999). Özellikleri gelişmiş yazılım araçlarının kullanımı kod kaybını engeller, ayrıca yazılım geliştiricinin kodu başkasına aktarımı sırasında kolaylık sağlar.

İlgili Metodolojileri Kullanım Yeteneği(m) Var: Projelerin başarılı tamamlanabilmesi için adaya bildiği ve(ya) çalışılan projede kullanılan metodoloji bilgileri sorulmuştur. Bu modellerle analiz, tasarım, yazılım ve test süreçlerinin ne şekilde ele alınacağı, yönetileceği belirlenmiştir. Projelere göre yönetim modeli değişiklik gösterir. Adayın projede kullanılacak metodolojiyi biliyor olması, adayı avantajlı kılar. Ancak yapılan ankette, katılımcıların çalıştığı projelerde Agile veya Şelale metodolojileri üzerinden proje yönetimi gerçekleştirdiği görülmüştür. Şelale ve Agile metodolojileri dışındaki metodolojiler "0" varyans oluşturduğu için bu parametre anketten çıkarılmıştır. Aşağıda ankette yer alan yazılım metodolojileri yer almaktadır.

Şelale Modeli (Waterfall Model): Yazılım mühendisliği alanında kullanılan en eski ve temel model olan Şelale modeli, proje yönetim standardı olarak kabul görmektedir. Şelale modelinde fazlardaki faaliyetler sona erdikten sonra geriye dönüş yapılamaz (Gencer ve Kayacan, 2017). Şelale yöntemi, her aşamada üretilen çok özel aktiviteler ve teslim edilenler ile ardışık bir dizi adımı takip etmeye dayanan bir yazılım geliştirme modelidir (Farrell, 2007).

Çevik Modeli (Agile Model): Günümüzde yazılım projelerinde artık daha çok tercih edilen yöntemdir. İşler iş değerine göre önceliklendirilir. Çevik yöntemler, iteratif teknikler kullanarak projenin küçük bölümlerini geliştirerek risklerin en aza indirilmesine odaklanır. Her yineleme en fazla dört hafta kadar sürmelidir. İterasyonun kendisi, planlama, gereksinimler, tasarım, geliştirme ve test dahil olmak üzere projenin tüm bileşenlerini kapsamalıdır. Her proje yinelemesi, proje gereksinimlerinin belirli bir bölümünü uygular ve yineleme tamamlandığında ürün gerekirse üretim ortamına aktarılacak bir durumdadır (Farrell, 2007).

Uç Programlama (Extreme Programming - XP): Müşterinin sahada çalışma yöntemini benimseyen ve takım çalışmasını savunan, bu yönden Çevik felsefesine de benzeyen bir metodolojidir. Basitlik, cesaret, geri dönüş ve iletişim gibi temellere dayanarak projenin yönetimi sağlanır. XP projeleri dört aşamada çalışır: planlama, tasarım, kodlama ve test etme. Planlama, kullanıcı hikâyelerini tasarlamayı, proje programını oluşturmayı ve her bir döngü için yinelemeyi planlamayı içerir. Günlük toplantıların ana iletişim yöntemi olarak önemi vurgulanmaktadır. Tasarım aşaması,

sadeliğin tasarlanmasını, sistemin metaforu ile çalışmayı ve tasarımın sürekli olarak yeniden düzenlenmesini içerir. Kodlama fazı, kodlama standartlarını belirlemeyi, önce ünite testlerini kodlamayı, çift kullanarak kodlamayı temel alır (Farrell, 2007).

Spiral Modeli: Yinelemeli geliştirme, sık yinelemelerde izlenen bir dizi tasarım kodu testi döngüsünü içerir. Bu iterasyonlar yazılımı tasarlamak ve geliştirmek için kullanılır ve tüm proje gereklilikleri tam olarak yerine getirilinceye kadar proje gereksinimlerini aşamalı olarak düzeltmek için kullanılır (Farrell, 2007). Yazılım geliştirme safhasında risk analizi önemlidir ve sürekli inceleme halindedir. Proje kapsamı projeye başlanacağı zaman net değilse ilgili metodun kullanılması uygundur**.**

İteratif Şelale Modeli (Iterative Waterfall Model): Şelale yönteminin modifiye edilmiş halidir (Farrell, 2007). Sonraki safhaya geçmeden hatayı düzeltme avantajına sahip modeldir. Uygulaması kolaydır ve kapsamı net olan projeler için uygundur, ancak büyük ve kritik projeler için uygun değildir.

***Artırımsal Geliştirme Modeli*:** Projeyi modül modül geliştirme yöntemini benimseyen metodolojidir. Modüller tamamlandıkça birbirine eklenir. Her ek artış, operasyonel işlevselliği veya önceki sürümlerde mevcut olmayan yetenekleri ekler. Artan yazılım geliştirme, doğada iteratiftir (Benediktsson ve Dalcher, 2003).

***Prototipleme Süreç Modeli*:** Bir projeyi küçük segmentlere ayırır ve bu segmentlerin her biri sırasında ürünün küçük prototiplerini oluşturur. Prototipler müşteri gereksinimlerini karşılayana kadar incelenir, analiz edilir ve işlenir (Farrell, 2007). Prototip süreç modelinin avantajı, müşterinin ihtiyaçlarını karşılayacak şekilde alternatif prototipler önermesi ve kapsamın süreç içinde değişmesine yardımcı olmasını desteklemesidir. Ancak araştırma sırasında dokümantasyon oluşturulamaması bu modelin dezavantajlarından biridir. Diğer bir dezavantajı da prototip oluşturmak için işlevsel olmayan geliştirme de yapılmış olabileceğidir. Bu da yöntemin maliyetini arttırmaktadır.

***V Modeli*:** İsminden de anlaşılacağı gibi V şeklindeki bir modeldir. Her aşamanın testi vardır. Test dokümantasyon süreci ile başlar. Sistem gereksinimi ve sistem testi birbirleriyle birlikte çalışır, yazılım gereksinimleri, yazılım testi,

entegrasyon testi ile yazılım tasarımı ve birim testi yazılım uygulaması ile birlikte çalışır. Geliştirme sırasında ihtiyaçları değişen projeler için uygun olsa da küçük projeler için uygun değildir (Chandra, 2015).

Uzun soluklu motivasyonla çalışır(ım): Uygulayıcının motivasyonu, başarılı yazılım geliştirmenin genel kabul görmüş yönetim ve organizasyonel bakışının başarısı için kritik öneme sahiptir (Procaccino ve Verner, 2002). Motivasyon projede üretimi ve kaliteyi arttırır. Özellikle yazılım geliştirme personelinin motivasyonunun yüksek olması, ürünün kaliteli olarak tamamlanmasında büyük katkı sağlar (Linberg, 1999). Çalışanlar uzun saatler çalıştığında ödüllendirilmez ise, motivasyonları azalmakta ve başarısızlık riski artmaktadır (Verner et al., 2008).

Yeni teknolojileri kolaylıkla öğrenir(im): Yazılım sektörü her geçen gün gelişen ve yenilenen teknolojilerle değişen bir alandır (Ahmed et al., 2015). Yeni geliştirilecek proje için daha önce kullanılmamış bir uygulama, programlama dili vs. kullanılacaksa, yazılım uzmanının ilgili konuyu hızlıca öğrenip projede uygulaması beklenir.

Takımla uyumlu çalışma becerisi: Takım ruhu yazılım projelerinin başarısında önemli bir etkiye sahiptir (Egorova et al., 2010). Çünkü takımla uyumla çalışma becerisine sahip biri projeye adaptasyonu yüksek olan ve sorunları takımla birlikte daha hızlı bir sürede çözüme kavuşturma potansiyeline de sahip olacaktır. Ayrıca birey uzmanlığının yetersiz kaldığı zamanlarda başkasından yardım isteme gibi sosyal etkileşimde bulunabilmelidir. Bu sebeple özellikle büyük projelere seçilecek adayın takımla uyumlu çalışma becerisine sahip olması gerekmektedir.

Bireysel çalışma becerisi: Bir yazılım mühendisinin performansına kişiliğin etkisi, hangi kişilik özelliklerinin veya türlerinin yazılım mühendisliği ile ilgili çeşitli iş görevleri için ideal olduğunu anlamak için yapılan bir araştırmada; kişiliğin bir yazılım projesinin çıktılarını teknoloji, süreç veya araçlardan daha fazla etkileyebileceğini düşündüğünden bahsetmiştir (Cruz et al., 2015). Bu sebeple bir insan bireysel olarak çalışma becerisine sahipse bu kişiliği hakkında olumlu ipuçları verecektir. Etkin bir şekilde kendi etkinlik gündemini organize edebiliyorsa, önceliklerini belirleyerek, kendi zamanını olabildiğince verimli kullanabiliyorsa

bireysel çalışma becerisi yüksektir. Ayrıca yazılım geliştirme uzmanlarının çoğu, görevlerini tamamladıklarında, planlanan hedeflerine ulaştıklarında veya hedefleri üzerinde ilerleme kaydettiklerinde iş günlerinin verimli olduğunu belirtmişlerdir (Meyer et al., 2014).

Stres Toleransı: Stres, yazılım projelerinin olmazsa olmaz duygusudur. Çalışmanın zamanında tamamlanması stresi kimi zaman motive edicidir, kimi zamansa çalışmayı verimsiz kılar. Ancak bazı yazılım projeleri, proje süresinin kısa olması, yetersiz kaynak, teknoloji gibi durumlarından dolayı stres altında çalışacak personele ihtiyaç duyar. Stres toleransı yüksek olanlar, zaman baskısı altında ve anlaşmazlık, muhalefet ve sıkıntı karşısında etkili bir şekilde hareket eder (Acuña and Juristo, 2004).

Krizi yönetme becerisi: Çatışma yazılım projelerinde her süreçte yaşanabilecek bir durumdur. İş birimiyle BT arasında, iş birimlerinin veya BT'nin kendi çalışanları arasında, proje yöneticisi ile paydaşlar arasında herhangi bir sürece karar verirken veya o sürecin yapılma şekline karar verirken yaşanabilir. Projede önemli olan çatışmanın ne kadar başarılı yönetebildiğidir. Linberg (1999) çalışmasında çatışmanın kişisel bir sebepten dolayı yaşandığında mümkün olduğunca yöneticilerin dâhil olmaması gerektiğinden bahsetmiştir. Ama aynı zamanda çatışmanın zaman kaybı yarattığından, çalışma ortamında bir çatışma olduğunu anladığında çatışmaya sebep olan konuları anlayıp, çatışmayı önlemeye yönelik adımlar attığından da bahsetmiştir. Bu sebeple yeni projelerde çalışacak kişinin çatışmayı yönetebilecek bir şekilde çalışabilmesi önem arz etmektedir.

Analitik düşünme becerisi: Bir yazılımcının kod yazarken algoritmayı doğru yorumlaması, veri yapılarını tanımlaması ve geliştirme yapılan programlama dilinin sözdizimi ve özelliklerini ayrıntılı bir şekilde anlamasını gerektirir ve bu da analitik ve problem çözme becerilerini gerektirir (Ahmed et al., 2015). Bu sebeple yazılım uzmanında bu becerinin yüksek olması beklenir.

İnisiyatif alma becerisi: Yazılım geliştirme sürecinde projenin analizi çok iyi bir şekilde yapılmış olmasına rağmen, bazen düşünülemeyen senaryolar için alınacak aksiyonlar için yazılımcının karar vermesi, inisiyatif alması beklenir. Yazılım

geliştirme bir takım çalışmasıdır, ancak yazılımcının bazı durumlarda, bağımsız çalışma yeteneği olması gerekir. Sonuç olarak yazılımcının ne yapılması gerektiği söylenmeden, beklemek yerine inisiyatif alınması beklenir (Ahmed et al., 2015). Bu sebeple de bazı projelerde inisiyatif alma becerisi yüksek adayların çalıştırılması projenin tıkanmadan ilerlemesini sağlar.

Problem çözme becerisi: Yazılım, problem çözme yetenekleri, bilişsel yönler ve sosyal etkileşim gibi insan faaliyetlerinin bir yan ürünüdür (Ahmed et al., 2015). Yazılım geliştirme süreci de temel olarak problem çözmeyi gerektirir. Çünkü yazılım geliştirme süreci boyunca, kodlar yazılırken başarılı senaryolar kadar, hatalı senaryoların da kullanıcıyı yönlendirecek şekilde geliştirilmesi gerekir. Ancak bazen altyapının düzgün tasarlanmaması, performans, döngü seçimi, sonlandırma kriterleri, programın daha küçük fonksiyonlara bölünmesinin hatalı olması gibi sebeplerden dolayı yazılımda problemler oluşabilmekte ve bu problemlerin hızlıca tespit edilmesi ve çözülmesi beklenmektedir.

Değişime ve öğrenmeye açık olma becerisi: Yazılım sektöründe kullanıcı istekleri sürekli değişiklik talep eder (Ahmed et al., 2015). Bu sebeple yazılım sektöründe çalışan özellikle yazılım uzmanlarının ihtiyaçların değişmesi sonucunda bu değişime ve öğrenmeye açık olması beklenir.

Deneyim (Kaç yıldır yazılım yapıyor(sunuz)): Verner ve arkadaşları (2002) çalışmasında daha deneyimli olan yazılım uzmanlarının mesleki gelişimlerine daha çok önem verdiğinden bahsetmiştir. Deneyim süresi arttıkça proje yönetimi ve verimli çalışma yeteneği arttığı için büyük projeler için deneyimli yazılım uzmanı tercih sebebi olabilmektedir.

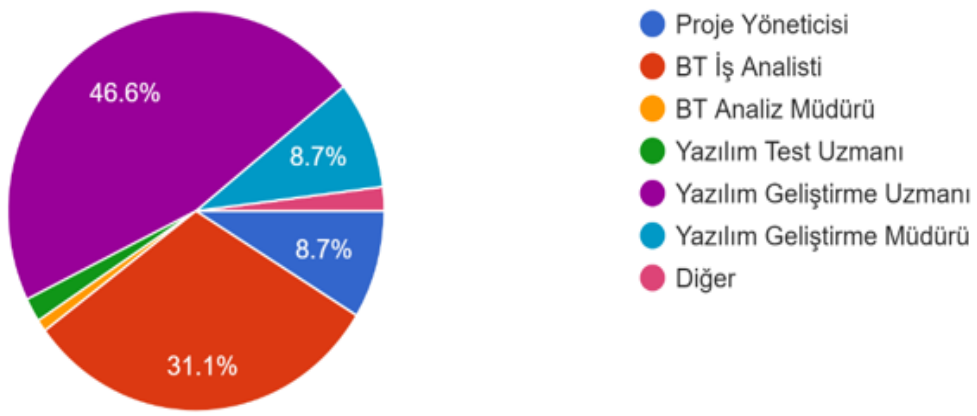
3.2. Veri Analizi

3.2.1. Anket değerlendirme

BT projelerinin aktif ve dinamik bir şekilde geliştirildiği banka sektörü için adaylar değerlendirilmiştir. Ankete katılanlar, çalışmanın keşif niteliğine göre bir uygunluk örneğini temsil etmişlerdir. Anketin güvenilirliğini ölçmek için SPSS Cronbach Alpha’s ve madde faktör analizi için de Temel Bileşen Analizi (Principal

Component Analysis (PCA)) seçeneği kullanılmıştır. Literatür taraması sırasında hem projelerin başarısını etkileyen faktörler hem de insan kaynaklarının adayları seçme yöntemleri incelenmiştir. Sorular çalışmada yer alan literatür taraması sırasında ve bazı katılımcılarla birlikte geliştirilmiştir. İlk cevaplar öncelikle bir bankada çalışan küçük bir grupla e-posta üzerinden ve(ya) birebir görüşmelerle alınmıştır. Cevaplar sırasında alınan geri bildirim ve bu küçük örneklem üzerindeki SPSS analizi ve aday seçimi sırasında kullanılan makine öğrenimi algoritmaları sırasında bazı sorular geliştirilmiş olup, bazı soruların cevap alma şekli değiştirilmiştir. Anket banka yazılım projelerinde çalışmış 105 katılımcı tarafından doldurulmuştur.

Anketi cevaplayan katılımcıların unvana göre yüzdelik dağılımı aşağıdaki gibidir. Ankette her unvan, çalıştığı projeyi; bu projede çalışan ve başarılı bulduğu adayı değerlendirmiştir. Aynı proje hem proje yöneticisi hem de projede çalışan yazılım geliştirme uzmanı tarafından değerlendirilmiş olabilir. Farklı projeler için aynı yazılım geliştirme uzmanı da değerlendirilmiş olabilir. Bir adayın birden fazla unvan tarafından değerlendirilme ihtimali olduğu için objektif sonuca yaklaşılmıştır. Unvan sorusu tamamen bilgi amaçlı alınmış olup, analize dahil edilmemiştir.



Şekil 3.1: Anketi Cevaplayan Katılımcıların Unvana Göre Yüzdelik Dağılımı

Proje başarısını etkileyen faktörlere ek olarak, adayın demografik, deneyim ve becerilerine yönelik toplamda 36 soru sorulmuştur. Anket üçlü Likert ölçeği tekniği kullanılarak veya proje/aday bilgisini anlayacak sorulardan oluşmuştur: Likert ölçeği

ile cevaplandırılan sorular katılımcının veya projenin önceliği, teknoloji uyumluluğu göz önünde bulundurulan sorulardır. Bilgi soruları da teknik bilgi, beceri ve deneyimine bağlı sorulardır.

Anket çalışması sonuçlarının güvenilirliğini analiz etmek amacıyla literatürde en çok kullanılan Croanbach's Alfa'dan faydalanılmıştır. SPSS programı ile parametreler güvenilirlik analizine dahil edilmiştir. Tablo 3.1'de Croanbach's Alfa Güvenilirlik Analizi Sonuçları'na yer verilmiştir. Croanbach's Alfa değerimiz 0,7'nin üzerinde, bu değer de anketimizin kabul edilebilir düzeyde güvenilir olduğunu göstermiştir.

Tablo 3.1: Croanbach's Alfa Güvenilirlik Analizi Sonuçları

Cronbach's Alfa	Stanart parametrelere dayalı Cronbach Alfa (Cronbach's Alpha Based On Standardized Items)	Parametre Sayısı (N of Items)
0,707	0,711	34

Tablo 3.2: Faktör Analizi Sonuçları

Kaiser-Meyer-Olkin Örnekleme Yeterliliğinin Ölçüsü (KMO)(Kaiser-Meyer-Olkin Measure of Sampling Adequacy)		0,609
Bartlett'in Küresellik Testi- (Bartlett's Test of Sphericity)	Yaklaşık Ki-Kare	1260,239
	df	561
	Sig.	,000

Anketin geçerliliğini ölçmek için faktör analizinden faydalanılmıştır. Faktör analizi için de SPSS programı kullanılmıştır; Tablo 3.2'de Faktör Analizi Sonuçları'na yer verilmiştir. Faktör analizinde parametrelerin birbirleriyle ilişkileri incelenmiştir. Sıfır varyasyon oluşturduğu için projede kullanılan diller ve adayın bildiği dillerin alındığı iki ayrı sorudan adayın bildiği diller sorusu anket analizine tabi olmuştur. Cinsiyet parametresi de faktör analizi için sıfır varyasyon oluşturduğundan anket kümesinden çıkarılmıştır.

Parametreler ve arasındaki görünmez ilişkinin geçerlilik analizi ile ölçümü ile makine öğrenimi için kullanılacak veri seti son halini almıştır. Tez kapsamında ana amaç; yazılım projelerine seçilecek adayların belirleyeceği bir otonom sistemi

hazırlamak olduğu için, bunun için oluşturulan modelde kullanılacak tüm öznitelikler temel faktör analizi ile değerlendirilmiştir. Buna ek olarak, k-ortalama kümeleme algoritması içerisinde de "Temel Bileşen Analizi (PCA)" yapılmıştır. Temel Bileşen Analizi k-ortalama kümeleme algoritmasının daha performanslı çalışmasını sağlamaktadır. Bu analiz ile ilişkili öznitelikler birleşmektedir. Aday seçimi için kullanılan YSA yönteminde oluşturulan modelde birbirleri ile korelasyon içeren öznitelikler matematiksel olarak elenir. Yani matematiksel olarak öznitelik seçimi yapılmaktadır. Bu sebeple SPSS üzerinde yapılan Faktör Analizi veri seti grubu için sağlama yöntemi olarak kullanılmıştır.

Örneklem sayısı 105 olduğundan faktör analizi sonucunun Kaiser-Meyer-Olkin Örnekleme Yeterliliğinin Ölçüsü (KMO) 0,5 üzerinde; Bartlett'in Küresellik Testi anlamlılık (significance (sig.)) değerinin de 0,5'in altında olması gerekmektedir. Tablo 3.2'de Faktör Analizi Sonuçları tablosundan görüleceği üzere bu iki kriterde sağlanmış durumdadır.

3.2.2. Makine Öğrenimi

Makine öğrenimi, Yapay Zekânın bir alt dalıdır. Bilgisayarların "öğrenme"sini sağlayacak algoritmaların ve çalışmaların yürütüldüğü, sürekli gelişen bir alandır. Günümüzün en hızlı büyüyen teknik alanlarından biridir. Sağlık, trafik, güvenlik, bilgisayar oyunları, elektronik ev eşyaları gibi hayatın birçok alanında makine öğrenmesi çalışmalarından faydalanılmaktadır. Karar verme, belirsiz veri içerisinden anlamlı veri oluşturma ve süreçleri iyileştirme gibi birçok faydalı çalışmaya olanak veren makine öğrenmesi istatistik dalında da çok sık başvurulan bir çalışma alanıdır (Jordan and Mitchell, 2015). Makine öğrenmesinde amaç kullanılacak alana uygun algoritma ile ilgili nesneye yapacak işi öğretmek ve işi makinenin kendini eğiterek yapmasını sağlamaktır. Bunun için de öğrenmenin sağlanacağı veri seti hazırlanması gerekir. Bu çalışmada projenin değerlendirildiği anket çıktıları k-ortalama kümeleme algoritması, adayın değerlendirildiği anket çıktıları makine öğrenimi için veri seti olarak hazırlanmıştır. Belirli bir makine öğrenme algoritması bir "sınıflandırıcı" ve bir "öğrenen" olarak genellikle iki paralel işlem veya iki ayrı algoritma içerir. Sınıflandırıcılar girdi alır (bir dizi özellik olarak adlandırılır) ve bir çıktı üretir (bir "kategori") (Burrell, 2016).

Makine öğrenmesi öğrenme yöntemine göre üç gruba ayrılır:

- Gözetimli Öğrenme (Supervised Learning),
- Gözetimsiz Öğrenme (Unsupervised Learning)
- Takviyeli Öğrenme (Reinforcement Learning)

Tez kapsamında Gözetimli ve Gözetimsiz öğrenme yöntemleri kullanılmıştır.

3.2.2.1. Gözetimsiz Öğrenme (Unsupervised Learning)

Gözetimsiz öğrenme, gruplanmamış, üzerinde çalışılmamış veriler arasındaki bağıntının ortaya çıkarılma işlemidir. Gözetimli öğrenmeden farkı, veri setinin etiketleme işlemi olmaksızın sisteme işlenmesidir (Jordan and Mitchell, 2015). İlgili veri seti arasındaki bağlantı tamamen gözetimsiz öğrenme tekniği ile sağlanır. Tez kapsamında gözetimsiz öğrenme çalışmalarından en iyi bilinen k-ortalama kümeleme yönteminden yararlanılmıştır (Dinçer, 2006). Gözetimsiz öğrenmede hedef, grupları keşfetmektir. K- ortalama kümeleme algoritmasıyla projeler gruplanmıştır.

3.2.2.1.1. K- ortalama Kümeleme

K-ortalama kümeleme algoritması Hartigan ve Wong (1979) tarafından tarif edilmiştir. K-ortalama kümelemede, küme sayısı dışarıdan verilir. Amaç bir veri topluluğunu verilerin benzerliğine göre bir k sayıdaki veri grubuna bölmektir (Dhanachandra et al., 2015). Küme sayısına göre birbirine benzer veriler aynı kümeye, birbirinden uzak veriler farklı kümeye yerleştirilir. Tez kapsamında Python programlama diliyle ilgili algoritma gerçekleştirilmiştir. Proje Değerlendirme Anket Sorularından alınan cevaplar k-ortalama kümeleme algoritmasına girdi olarak verilmiştir. Algoritma içerisinde projelerin gruplanması için girdi olarak verilen değer 4 olarak belirlenmiştir. 102 veri seti örneklem olarak seçilmiştir. Proje değerlendirmesi için kullanılan parametre adeti 31 olarak oluşmuştur. Bunun nedeni bazı proje değerlendirme sorularına çoklu seçim yapılabilmesidir.

Ek D: K-ortalama Kümeleme Algoritması ile Projelerin Gruplanmasını Sağlayan Python Kodu başlığında da detaylı olarak incelenebilecek gruplama için kullanılan değerler aşağıdaki gibidir.

n_clusters = 4

sample_size = 102

n_features = 31

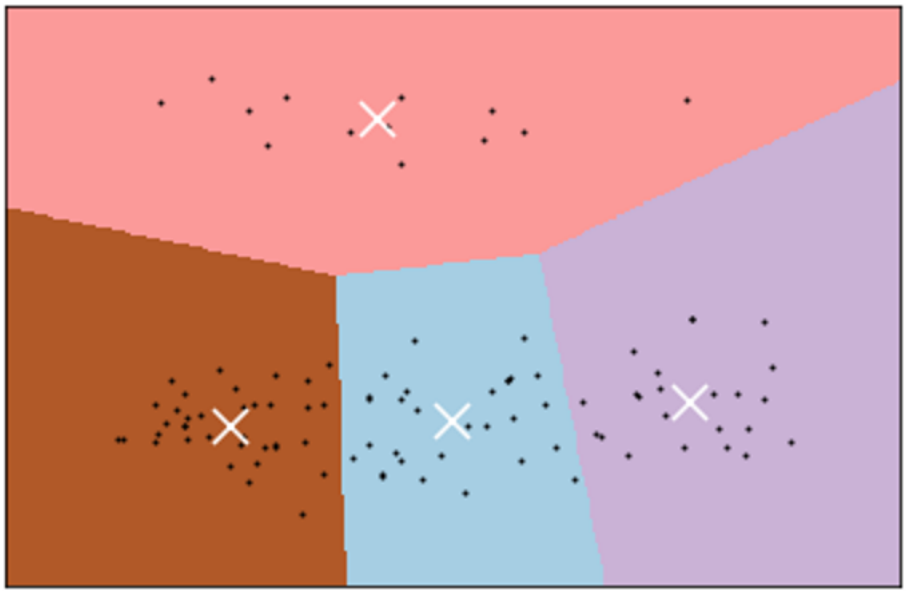
Değerlendirilen projelerden oluşan veri seti aşağıdaki formatta girilmiştir. Örnek olarak tezde 3 tanesine yer verilmiştir:

2,2,3,2,2,2,1,3,3,3,3,2,1,0,0,0,1,0,0,0,0,0,0,1,0,0,0,0,0,0,3

1,1,1,2,1,3,3,2,3,2,3,1,0,1,0,0,0,0,0,0,0,0,0,0,0,0,1,0,0,0,3

4,3,3,1,3,1,1,2,3,3,3,3,1,0,1,0,0,1,0,0,0,0,1,0,0,0,0,0,0,0,3

Veri setinde programa girdi teşkil eden projeler, k-ortalama gözetimsiz öğrenme tekniğiyle gruplandırılmıştır. Şekil 3.2'de projelerin 2 boyutlu uzaydaki dağılımı verilmiştir.



Şekil 3.2: Projelerin 2 Boyutlu Uzayda Dağılımı

Şekil 3.2'de görüldüğü üzere öznitelikler olarak birbirine yakınlık içeren projeler iki boyutlu uzayda birbirine çok yakın olacak şekilde bir yoğunluk oluşturmuştur. K-ortalama kümeleme algoritmasının 4 grup çıktı elde edecek şekilde çalıştırılması neticesinde gruplardan üçünün yoğun olan bölgede oluştuğu gözlenmiştir. Matematiksel ifadeyle 2 boyutlu uzayda yakın olan noktalar unsurların benzerliklerini ifade etmektedir. 4. ve diğerlerinden ayrı olan gruptaki projeler hem diğer 3 gruptaki projeden hem de birbirlerinden öznitelikler kapsamında farklı olduğu gözlemlenmiştir. K-ortalama kümeleme algoritması 4 grupla sınırlandırdığı için bu ayrık yoğunluk ayrı bir proje grubu olarak temsil edilmiştir.

Ayrıca gruplamalardan çıkan sonuç neticesinde veri setimizdeki adayın hangi gruba uygun olduğu gözlemlenmiştir. **Aday Değerlendirme Anket Soruları**'nın çıktılarından oluşan veri setine her bir adayın ait olduğu proje grubu eklenmiştir yani adayların öznitelik seti başarılı olduğu projeyle etiketlenmiştir. Bu etiketler ile uygun adayın seçimi için gözetimli öğrenme algoritmalarından YSA'dan faydalanılacak altyapı oluşturulmuştur.

3.2.2.2. Gözetimli Öğrenme (Supervised Learning)

Adayları belirlerken kullanılan metottur. Makine öğrenme yöntemleri arasında en çok kullanılan öğrenme yöntemi gözetimli öğrenme yöntemidir (Jordan and Mitchell, 2015). Gözetimli öğrenme çıktıları girdilerle eşleştirecek şekilde çalışır. Gözetimli öğrenmede eşleştirilmiş verilerle diğer veri setinin eşleştirilmesi amaçlanır. Gözetimli makine öğrenme algoritmaları şunlardır: Doğrusal Regresyon, Lojistik Regresyon, Naïve Bayes Sınıflandırıcı, Destek Vektör Makinesi (Support Vector Machine-SVM); İkinci Dereceden Sınıflayıcılar, K-En Yakın Komşu Kestirimi, Yükseltme, Karar Ağacı, Rastgele Orman (Randon Forest - RF); YSA, Bayes Ağları vb. (Osisanwo et al., 2017).

Genellikle gözetimli öğrenme algoritmalarından SVM'nin en hassas ve doğru algoritma olduğu görülmektedir (Osisanwo et al., 2017). Ancak SVM ikili sonuçlar üreten veri setlerinde kullanıma uygundur. Örneğin: Rengi "Beyaz" veya "Beyaz Değil" gibi. Lineer Regresyon ise daha çok benzer özellik değerlerine sahip ögeleri gruplara ayırmayı hedeflemektedir. Belge sınıflandırmasında olduğu gibi, genellikle

her öğenin bir belgedeki bir kelimenin sayım sayısı olduğu durumlarda çok iyi çalışır. Lojistik regresyon ile tahmin, ikilik bir sonuç doğurur. Daha az doğru sonuçlar üretir (Osisanwo et al., 2017). Bayes ağları, istatistiksel öğrenme algoritmalarının en iyi bilinen temsilcisidir (Osisanwo et al., 2017). Karar ağaçları veya sinir ağlarına kıyasla en ilginç özelliği, belirli bir sorun hakkında önceki bilgileri kesinlikle dikkate alma olasılığıdır. Tez kapsamında çoklu sonuçlar üreten veri seti olduğu için Yapay Sinir Ağları'ndan yararlanılmıştır.

3.2.2.2.1. Yapay Sinir Ağları

YSA, insan beyninin sinir ağlarını taklit eden ve bilgiyi işleme şeklini simule etmek için tasarlanmış biyolojik sinir ağlarından ilham alan bir hesaplama tekniğidir. YSA'lar, verideki kalıpları ve ilişkileri tespit ederek bilgilerini toplarlar, deneyim yoluyla öğrenirler veya eğitilirler. Bir YSA, sinir yapısını oluşturan ve tabakalar halinde düzenlenmiş, katmanlar (ağırlıklar) ile bağlantılı yüzlerce tek birimlerden, yapay nöronlardan veya işleme elemanlarından oluşur. Sinirsel hesaplamaların gücü bir ağdaki nöronları bağlamaktan gelir. Her işleme elemanı, ağırlıklı girişlere, gizli katmana (beyindeki birbirine bağlı nöronlar gibi) ve bir çıkışa sahiptir. Ağırlıklar ayarlanabilir parametrelerdir ve bu anlamda bir sinir ağı parametreli bir sistemdir. Her giriş, gizli bir katmana bağlanır, oradan da sinir ağı tasarımındaki bir çıkışa bağlanır. Eğitim sırasında, üniteler arası bağlantılar, tahminlerdeki hata en aza indirilene ve ağ belirtilen doğruluk seviyesine ulaşana kadar optimize edilir. Ağ bir kez eğitilip test edildikten sonra, çıkışı öngörmek için yeni giriş bilgileri verilerek doğruluğu kontrol edilir (Agatonovic-Kustrin, 2000).

105 adayın çalıştığı proje grupları k-ortalama sınıflandırıcısından yararlanarak belirlenmiştir. Adayların parametreleri çalıştıkları proje gruplarıyla birlikte sisteme girdi olarak işlenmiştir. Adayı yeni geliştirilecek projeye seçebilmek ve sistemi eğitmek için Yapay Sinir Ağları'ndan yararlanılmıştır. K-ortalama kümeleme algoritmasından çıkan gruplarla ilgili adayların eşleştirilmesi sonucu oluşan veri seti YSA algoritmasında girdi olarak kullanılmıştır. İlgili veri setinde ilk 46 değer adayın özelliklerini, son değerde çalıştığı proje grubunu temsil eder.

Değerlendirilen adaylardan ve k-ortalama kümeleme algoritmasının çıktısıyla oluşan veri seti aşağıdaki formatta girilmiştir. Örnek olarak tezde 3 tanesine yer verilmiştir:

1,0,1,2,0,1,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,3,3,1,1,0,0,0,0,0,0,
3,3,3,3,3,3,3,2,3,3,1,1,3

2,1,0,3,1,0,1,0,0,1,0,0,0,1,1,0,0,0,0,0,0,0,0,0,0,3,3,1,0,0,0,1,0,0,
0,3,3,3,3,2,2,3,3,3,3,3,2,1

3,1,0,2,1,0,0,0,1,0,0,0,0,0,1,1,0,0,1,0,0,0,0,0,0,3,3,1,1,0,0,1,0,0,0,
3,3,3,3,2,2,3,2,2,2,3,3,1

Girilen bu veri seti aşağıdaki hiper-parametre değerlerine göre makine öğrenmesini gerçekleştirmiştir.

epochs=48, batch_size=16, validation_split=0.1

epochs: Eğitim tur sayısı

batch_size: Her bir eğitim turunda kullanılacak veri seti sayısı

validation_split: Her bir eğitim turunda eğitim için kullanılan veri setinin doğruluğunun test edileceği örneklem sayısını belirleyen oran.

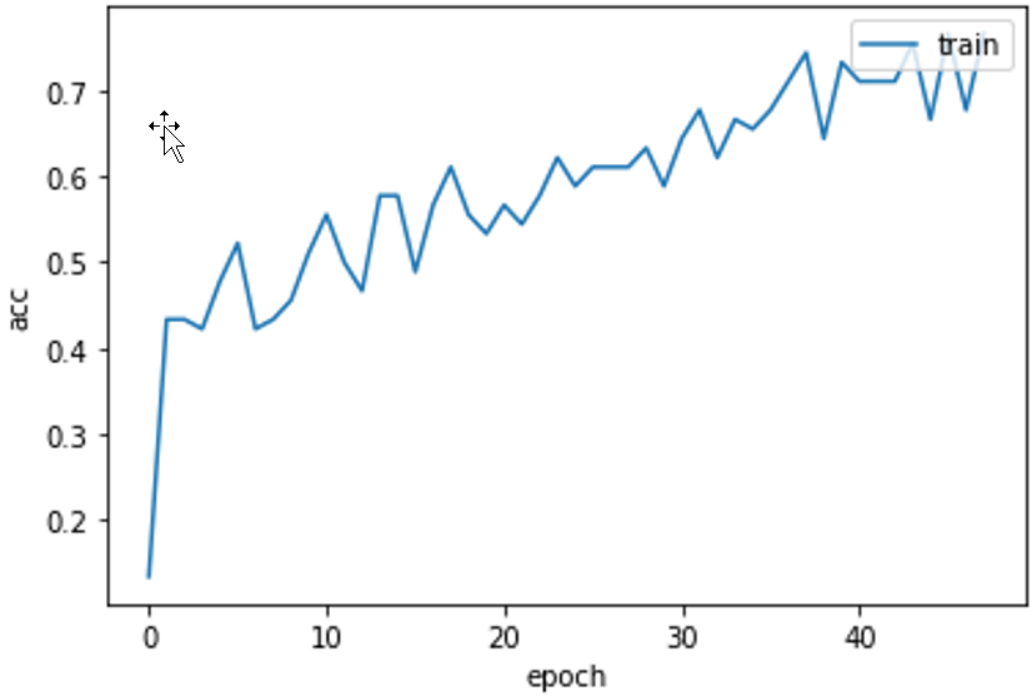
İlgili hiper-parametre değerleri tahmini değerler olarak girilmiştir. Çünkü çalışmadaki amaç makinenin tekrara düşmeden öğrenmesini sağlamaktır. **Ek E: YSA ile Projeye Uygun Adayın Seçimini Sağlayan Python Kodu** başlığında görüldüğü üzere yeni aday değerleri girilmiştir. Öğrenen makinenin adayı doğru gruba atadığı görülmüştür. Girilen yeni adayın değeri aşağıdaki gibidir:

```
guess=
np.array((2.0,1.0,0.0,3.0,0.0,1.0,0.0,0.0,1.0,0.0,0.0,0.0,0.0,1.0,0.0,0.0,0.0,0.0,0.0,0.0,1.0,0.0,
0.0,0.0,1.0,0.0,3.0,3.0,1.0,1.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,3.0,3.0,3.0,3.0,3.0,3.0,3.0,2.0,3.0
,3.0,3.0,3.0)).reshape(1,46)
print("Predict Result: ")
print(model.predict_classes(guess))
```

İlgili adayı önerdiği proje grubunun 3 olduğu gözlemlenmiştir.

```
Predict Result:
[3]
```

Verilen hiper-parametrelere göre öğrenme grafiği aşağıdaki gibidir.



Şekil 3.3: Öğrenme Grafiği

Şekil 3.3: Öğrenme Grafiği'nde belirtilen ***"acc"*** kavramı üretilen yapay sinir ağı modelinin yaptığı tahminlerin gerçek değerle olan karşılaştırmalarını ifade eder. Daha detaylı anlatılacak olursa; bu değer belirli bir kümede yapılan tahminlerin doğruluk oranıdır. Buradaki belirli küme kavramı literatürde ***"validation split"*** olarak geçen her öğrenme yinelemesinde veri kümesinin kaçta kaçının modelin eğitilmesinde ve kaçta kaçının modelin doğrulamasını ifade eder. X ekseninde belirtilen ***"epoch"*** değeri modelin eğitilmesi için tekrarlanan yineleme sayısını ifade eder. ***"Epoch"*** değeri arttıkça geri yayılım algoritması (adam optimizer) tekrarlanacağından her nöronun katsayı değerleri modele daha uygun tahmin yaptıracak hale gelir. Grafikten anlaşılacağı üzere model eğitimi tekrarlandıkça (epoch) tahminler gittikçe daha doğruya yaklaşmaktadır. Burada önemli olan husus isabet değerinin (acc) hiçbir zaman 1 olmamasıdır. Bunu elde etmek önemlidir. Çünkü isabet değeri 1 olduğunda model veri setini ezberlemiştir. Veri setinin dışında bir öznitelik grubu sorgulandığında doğru tahmin etme ihtimali çok düşüktür. Tez kapsamında yapılan hiper-parametre ayarlama (tuning) çalışmasında isabet değerinin **0,9** değerine ulaşması için çaba gösterilmiştir. Bu sebeple hiper-parametre değerleri ***epochs 48, batch_size 16, validation_split 0,1*** olarak modelimize uygulanmıştır.

4. SONUÇ VE DEĞERLENDİRMELER

Günden güne rekabetin arttığı yazılım sektöründe projelerin en hızlı ve kaliteli bir şekilde tamamlanması gerekmektedir. Bu hususta yapılan iş ve bu işe en uygun personellerin seçilmesi esastır. Personel seçiminde en doğru sonuca ulaşmak için personellerin objektif veya objektife yakın bir şekilde değerlendirilmesi gerekmektedir. Fakat personel seçimi zorunlu olarak subjektif değerlendirmeyi de gerektirmektedir. Ayrıca büyük organizasyonlara sahip bankalarda, adayın çalıştığı projeden farklı proje ekiplerinin ilgili adayın yeteneğiyle ilgili bilgisi bulunmayabilmektedir. Tez kapsamında önerilen otonom sisteminin bu ihtiyaç için kullanılabilmesinin organizasyonların harcayacağı maliyet ve zaman eforunu minimize edeceği öngörülmüştür. Çünkü adayın yetenekleri ve deneyimi bir veri tabanında yer almış olacak ve otonom sistemiyle ilgili proje için doğru aday(lar) eşleştirilebilecektir.

Ek olarak veri tabanında bulunan bu veriler güvenilirliği doğrulanmış anketlerle hazırlandığından dolayı objektif değerlendirmeye daha yakındır. Organizasyonda bu şekilde veri toplanması ve otonom bir sistemde işlenerek karar verilmesi neticesinde hem organizasyon içi adaylar için bir farkındalık oluşacak, hem de organizasyona katılacak adaylar daha doğru bir şekilde adreslenebilecektir.

Tez kapsamında önerilen sistemde, sisteme kaydedilmiş projeler benzerliklerine göre 4 kategoriye ayrılmıştır. Yeni geliştirilecek proje değerleri sisteme girildiğinde, projenin sınıflandırılmasının bu kategorilerden benzerliğe göre diğerlerine en yakın gruba düştüğü görülmüştür. Sistemde yeni girilen bir adayın çalışacağı projenin daha önce de aynı kategoride çalışan adaylar için oluşan sonuca benzer tahminlemede bulunduğu, yapılan sağlama testleriyle izlenmiştir. Ancak yeni mezun yazılım uzmanlarının çalıştığı projeler olamayabileceğinden tez çalışması bu adaylar için uygun değildir. Yeni mezun yazılım geliştirme adayları için literatüre katkı sağlayacak şekilde yeni çalışmalar yapılabilir.

Çalışma, farklı türlerde anket soruları düzenlenerek, inşaat, sigorta, telekomünikasyon, endüstriyel çalışma, biyo-teknoloji, mühendislik, araştırma ve ilaç endüstrileri gibi farklı sektörlerin yazılım geliştirme projelerinde de yapılabilir.

KAYNAKLAR

Acuña, S. T., & Juristo, N. (2004). Assigning people to roles in software projects. Software: Practice and Experience, 34(7), 675-696.

Agarwal, N., & Rathod, U. (2006). Defining 'success' for software projects: An exploratory revelation. International journal of project management, 24(4), 358-370.

Agatonovic-Kustrin, S., & Beresford, R. (2000). Basic concepts of artificial neural network (ANN) modeling and its application in pharmaceutical research. Journal of pharmaceutical and biomedical analysis, 22(5), 717-727

Ahmed, F., Capretz, L. F., Bouktif, S., & Campbell, P. (2015). Soft skills and software development: A reflection from the software industry. arXiv preprint arXiv:1507.06873.

Alakuş, T. B., Daş, R., & Türkoğlu, İ. (2017, September). Yazılım geliştirme süreçlerinin analizi: Zorluklar, tasarım prensipleri ve tekniksel yaklaşımlar. In 2017 International Artificial Intelligence and Data Processing Symposium (IDAP) (pp. 1-10). IEEE.

Aruoba, S. B., & Fernández-Villaverde, J. (2015). A comparison of programming languages in macroeconomics. Journal of Economic Dynamics and Control, 58, 265-273.

Bedir, N., & Eren, T. (2015). AHP-PROMETHEE yöntemleri entegrasyonu ile personel seçim problemi: perakende sektöründe bir uygulama. Sosyal Bilimler Araştırma Dergisi, 4(4), 46-58.

Benediktsson, O., & Dalcher, D. (2003). Effort estimation in incremental software development. IEE Proceedings-Software, 150(6), 351-357.

Boicea, A., Radulescu, F., & Agapin, L. I. (2012, September). MongoDB vs Oracle--database comparison. In 2012 third international conference on emerging intelligent data and web technologies (pp. 330-335). IEEE.

Burrell, J. (2016). How the machine 'thinks': Understanding opacity in machine learning algorithms. Big Data & Society, 3(1), 2053951715622512.

Chandra, V. (2015). Comparison between various software development methodologies. International Journal of Computer Applications, 131(9), 7-10.

Chien, C. F., & Chen, L. F. (2008). Data mining to improve personnel selection and enhance human capital: A case study in high-technology industry. Expert Systems with applications, 34(1), 280-290.

Cruz, S., da Silva, F. Q., & Capretz, L. F. (2015). Forty years of research on personality in software engineering: A mapping study. Computers in Human Behavior, 46, 94-113.

Dhanachandra, N., Manglem, K., & Chanu, Y. J. (2015). Image segmentation using K-means clustering algorithm and subtractive clustering algorithm. Procedia Computer Science, 54, 764-771.

(Dinçer, Ş. E. (2006). Veri madenciliğinde K-means algoritması ve tıp alanında uygulanması (Master's thesis, Kocaeli Universitesi, Fen Bilimleri Enstitusu).

Dvir, D., Raz, T., & Shenhar, A. J. (2003). An empirical analysis of the relationship between project planning and project success. International journal of project management, 21(2), 89-95.

Egorova, E., Torchiano, M., & Morisio, M. (2010). Actual vs. perceived effect of software engineering practices in the Italian industry. Journal of Systems and Software, 83(10), 1907-1916.

Ersoy, H., Madran, R. O., & Gülbahar, Y. (2011). Programlama dilleri öğretimine bir model önerisi: robot programlama. Akademik Bilişim 2011 Konferansı.

Farrell, A. (2008). Selecting a software development methodology based on organizational characteristics (Doctoral dissertation, Athabasca University).

Feldt, R., Angelis, L., Torkar, R., & Samuelsson, M. (2010). Links between the personalities, views and attitudes of software engineers. Information and Software Technology, 52(6), 611-624.

Gallivan, M. J. (2004). Examining IT professionals' adaptation to technological change: The influence of gender and personal attributes. ACM SIGMIS Database: the DATABASE for Advances in Information Systems, 35(3), 28-49.

Gencer, C., & Kayacan, A. (2017). Yazılım Proje Yönetimi: Şelale Modeli ve Çevik Yöntemlerin Karşılaştırılması. Bilişim Teknolojileri Dergisi, 10(3), 335-352.

Hall, T., Beecham, S., Verner, J., & Wilson, D. (2008, April). The impact of staff turnover on software projects: the importance of understanding what makes software practitioners tick. In Proceedings of the 2008 ACM SIGMIS CPR conference on Computer personnel doctoral consortium and research (pp. 30-39).

Hartigan, J. A., & Wong, M. A. (1979). Algorithm AS 136: A k-means clustering algorithm. Journal of the Royal Statistical Society. Series C (Applied Statistics), 28(1), 100-108.

Jordan, M. I., & Mitchell, T. M. (2015). Machine learning: Trends, perspectives, and prospects. Science, 349(6245), 255-260.

Keil, M., Tiwana, A., & Bush, A. (2002). Reconciling user and project manager perceptions of IT project risk: a Delphi study 1. Information Systems Journal, 12(2), 103-119.

Kelemenis, A., & Askounis, D. (2010). A new TOPSIS-based multi-criteria approach to personnel selection. Expert systems with applications, 37(7), 4999-5008.

Lehtinen, T. O., Mäntylä, M. V., Vanhanen, J., Itkonen, J., & Lassenius, C. (2014). Perceived causes of software project failures–An analysis of their relationships. Information and Software Technology, 56(6), 623-643.

Linberg, K. R. (1999). Software developer perceptions about software project failure: a case study. Journal of Systems and Software, 49(2-3), 177-192.

Maslow, A.H., 1971. The Farther Reaches of Human Nature, Viking Press, New York, p. 217.

Masum, A. K. M., Beh, L. S., Azad, M. A. K., & Hoque, K. (2018). Intelligent human resource information system (i-HRIS): a holistic decision support framework for HR excellence. Int. Arab J. Inf. Technol., 15(1), 121-130.

Meyer, A. N., Fritz, T., Murphy, G. C., & Zimmermann, T. (2014, November). Software developers' perceptions of productivity. In Proceedings of the 22nd ACM SIGSOFT International Symposium on Foundations of Software Engineering (pp. 19-29). ACM.

Osisanwo, F. Y., Akinsola, J. E. T., Awodele, O., Hinmikaiye, J. O., Olakanmi, O., & Akinjobi, J. (2017). Supervised machine learning algorithms: classification and comparison. International Journal of Computer Trends and Technology (IJCTT), 48(3), 128-138.

Prechelt, L. (2000). An empirical comparison of seven programming languages. Computer, 33(10), 23-29.

Procaccino, J. D., & Verner, J. M. (2002). Software practitioner's perception of project success: a pilot study. International Journal of Computers. The Internet and Management, 10(1), 20-30.

Procaccino, J. D., Verner, J. M., Overmyer, S. P., & Darter, M. E. (2002). Case study: factors for early prediction of software development success. Information and software technology, 44(1), 53-62.

Purna Sudhakar, G. (2012). A model of critical success factors for software projects. Journal of Enterprise Information Management, 25(6), 537-558.

Savolainen, P., Ahonen, J. J., & Richardson, I. (2012). Software development project success and failure from the supplier's perspective: A systematic literature review. International Journal of Project Management, 30(4), 458-469.

Schmidt, R., Lyytinen, K., Keil, M., & Cule, P. (2001). Identifying software project risks: An international Delphi study. Journal of management information systems, 17(4), 5-36.

Tilkov, S., & Vinoski, S. (2010). Node. js: Using JavaScript to build high-performance network programs. IEEE Internet Computing, 14(6), 80-83.

Verner, J., Sampson, J., & Cerpa, N. (2008, June). What factors lead to software project failure? In 2008 Second International Conference on Research Challenges in Information Science (pp. 71-80). IEEE.

Walia, E. S., & Gill, E. S. K. (2014). A framework for web based student record management system using PHP. International Journal of Computer Science and Mobile Computing, 3(8), 24-33.

Whittaker, B. (1999). What went wrong? Unsuccessful information technology projects. Information Management & Computer Security, 7(1), 23-30.

Yadav, H. B., & Yadav, D. K. (2017). Early software reliability analysis using reliability relevant software metrics. International Journal of System Assurance Engineering and Management, 8(4), 2097-2108.

Yazgan, H. R., & Sönmez, P. (2015). Yazılım Geliştirme Projelerinde Risk Yönetimi: Bir Banka ATM Projesi Örneği/Risk Management of Software Development Project: An Example of A Bank ATM Project. Ege Akademik Bakış, 15(1), 111.

EKLER

Ek A: Tez Çalışması Kapsamında Yapılan Yayınlar

Genç Eroğlu, G., & Sezen, B. (2020). “Yazılım projeleri personeli seçiminde yapay sinir ağları tekniğinin kullanılması ve bir uygulama”. IX. Umteb International Congress On Vocatıonal & Technıcal Sciences, 64-324, Gaziantep, Turkey, 7-8 February.

Ek B: Proje Değerlendirme Anket Soruları

Başarılı olarak tamamlanan ve sizin de içinde yer aldığınız bir yazılım projesini veya yeni başlanacak bir yazılım projesini değerlendiriniz.

1. Unvanınız *

- ○ Proje Yöneticisi
- ○ BT İş Analisti
- ○ BT Analiz Müdürü
- ○ Yazılım Test Uzmanı
- ○ Yazılım Geliştirme Uzmanı
- ○ Yazılım Geliştirme Müdürü
- ○ Diğer

2. Proje süresini seçiniz. *

- ○ 0 ay -6 ay
- ○ 7 ay - 12 ay
- ○ 13 ay - 24 ay
- ○ 25 ay - 36 ay

3. Proje karmaşıklığını seçiniz. *

1- Az Karmaşık, 2 - Normal, 3 - Çok Karmaşık

	1	2	3	
Az Karmaşık	○	○	○	Çok Karmaşık

4. Projede çalışan yazılım uzmanı sayısı kaçtır? *

- ○ 2 - 4
- ○ 5 - 10
- ○ 11 - 20
- ○ 21 - 50
- ○ 50+

5. Projede outsource(dış kaynak) eleman(lar) çalıştı mı? *

- ○ Evet
- ○ Hayır

6. Projede çalışan toplam kişi sayısı kaçtır? *

- ○ 5 - 10
- ○ 11 - 20
- ○ 20 - 50
- ○ 50+

7. Proje kapsamı iyi tanımlanmış mı(ydı)? *

1- Katılmıyorum, 2- Ne katılıyorum ne katılmıyorum, 3- Katılıyorum

	1	2	3	
Katılmıyorum	○	○	○	Katılıyorum

8. Proje gereksinimleri tamamen anlaşılmıştır *

1- Katılmıyorum, 2- Ne katılıyorum ne katılmıyorum, 3- Katılıyorum

	1	2	3	
Katılmıyorum	○	○	○	Katılıyorum

9. Kullanıcıların projeye katılım seviyesi: *

1 Beklenenin Altında Katılım sağlandığı, 2 Beklenen Katılımın Sağlandığı ve 3 numaralı seçenek ise Beklenenin Üstünde Katılım sağlandığını yansıtır

	1	2	3	
Beklenenin Altında	○	○	○	Beklenenin Üstünde

10. Projede riskler göz ardı edilmemiştir. *

1- Katılmıyorum, 2- Ne katılıyorum ne katılmıyorum, 3- Katılıyorum

	1	2	3	
Katılmıyorum	○	○	○	Katılıyorum

11. Projenin organizasyondaki önceliği nedir? *

1- Düşük, 2 - Normal, 3 - Yüksek

	1	2	3	
Düşük	○	○	○	Yüksek

12. Proje, proje planına uygun bir şekilde tamamlanmıştır. *

1- Katılmıyorum, 2- Ne katılıyorum ne katılmıyorum, 3- Katılıyorum

	1	2	3	
Katılmıyorum	○	○	○	Katılıyorum

13. Projenin bütçe seviyesi nedir? *

1- Düşük Bütçe, 2 - Orta Bütçe, 3 - Yüksek Bütçe

	1	2	3	
Düşük Bütçe	○	○	○	Yüksek Bütçe

14. Projede kullanılan/kullanılacak dilleri seçiniz. *

- [ ] JAVA
- [ ] C#
- [ ] C
- [ ] C++
- [ ] PL/SQL
- [ ] MySQL
- [ ] JavaScript
- [ ] PHP
- [ ] Diğer

15. Proje hangi metodoloji ile yönetilmiştir? *

- ○ Agile
- ○ Şelale (Waterfall)
- ○ Extreme Programming XP (Uç Programlama)
- ○ Spiral Method
- ○ Iteratif Yenileme
- ○ Artırımsal Geliştirme
- ○ Prototipleme Süreç Modeli
- ○ V Modeli

16. Gereksinimler organizasyonun teknolojisine uygundur? *

1- Katılmıyorum, 2- Ne katılıyorum ne katılmıyorum, 3- Katılıyorum

	1	2	3	
Katılmıyorum	○	○	○	Katılıyorum

Ek C: Aday Değerlendirme Anket Soruları

Değerlendirdiğiniz aday, değerlendirdiğiniz projede yer almış yazılım geliştirme uzmanı olmalıdır.

1. Yaşı(nız): *

- 18 - 24
- 25 - 30
- 31 - 40
- 40+

2. Cinsiyeti(niz): *

- Kadın
- Erkek

3. Eğitim seviyesi: *

- Lisans
- Yüksek Lisans
- Doktora
- Diğer

4. Projede Kullandığı(nız) dilleri seçiniz. *

- [ ] JAVA
- [ ] C#
- [ ] C
- [ ] C++
- [ ] PL/SQL
- [ ] MSSQL
- [ ] JavaScript
- [ ] PHP
- [ ] R, PHYTON, SPSS vs. istatiksel yazılım programı
- [ ] Diğer

5. Aktif olarak kullanıdığı(nız) programlama dillerini seçiniz. *

- [ ] JAVA
- [ ] C#
- [ ] C
- [ ] C++
- [ ] PL/SQL
- [ ] MSSQL
- [ ] JavaScript
- [ ] PHP
- [ ] R, PHYTON, SPSS vs. istatiksel yazılım programı
- [ ] Diğer

6. Proje planına uygun olarak çalıştı(m). *

1- Katılmıyorum, 2- Ne katılıyorum ne katılmıyorum, 3- Katılıyorum

	1	2	3	
Katılmıyorum	○	○	○	Katılıyorum

7. Yazılım geliştirme araçlarını başarıyla kullandı(m). *

1- Katılmıyorum, 2- Ne katılıyorum ne katılmıyorum, 3- Katılıyorum

	1	2	3	
Katılmıyorum	○	○	○	Katılıyorum

8. İlgili metodolojileri kullanım yeteneği(m) var. *

- [ ] Agile
- [ ] Waterfall
- [ ] Extreme Programming
- [ ] Spiral Method
- [ ] Iteratif Yenileme
- [ ] Artırımsal Geliştirme
- [ ] Prototipleme Süreç Modeli
- [ ] V Modeli

9. Uzun soluklu motivasyonla çalışır(ım). *

1-Katılmıyorum, 2- Ne Katılıyorum, Ne Katılmıyorum, 3-Katılıyorum

	1	2	3	
Katılmıyorum	○	○	○	Katılıyorum

10. Fazla mesaide çalışır(ım). *

1-İhtiyaç halinde de olsa fazla mesaiye kalmam., 2- İhtiyaç halinde de olsa fazla mesaiye sıcak bakmıyor(um). Ama kalır(ım), 3-İhtiyaç halinde fazla mesaiye kalır(ım)

	1	2	3	
Katılmıyorum	○	○	○	Katılıyorum

11. Takımla uyumlu çalışma becerisini değerlendiriniz. *

1 Beklenenin Altında Performans sergilediğini, 2 Beklenen Performansı sergilediğini ve 3 numaralı seçenek ise Beklenenin Üstünde Performans sergilediğini yansıtır

	1	2	3	
Beklenenin Altında	○	○	○	Beklenenin Üstünde

12. Bireysel çalışma becerisini değerlendiriniz. *

1 Beklenenin Altında Performans sergilediğini, 2 Beklenen Performansı sergilediğini ve 3 numaralı seçenek ise Beklenenin Üstünde Performans sergilediğini yansıtır

	1	2	3	
Beklenenin Altında	○	○	○	Beklenenin Üstünde

13. Stres Toleransını değerlendiriniz *

1 Beklenenin Altında Performans sergilediğini, 2 Beklenen Performansı sergilediğini ve 3 numaralı seçenek ise Beklenenin Üstünde Performans sergilediğini yansıtır.

	1	2	3	
Beklenenin Altında	○	○	○	Beklenenin Üstünde

14. Krizi yönetme becerisini değerlendiriniz. *

1 Beklenenin Altında Performans sergilediğini, 2 Beklenen Performansı sergilediğini ve 3 numaralı seçenek ise Beklenenin Üstünde Performans sergilediğini yansıtır

	1	2	3	
Beklenenin Altında	○	○	○	Beklenenin Üstünde

15. Analitik düşünme becerisini değerlendiriniz. *

1 Beklenenin Altında Performans sergilediğini, 2 Beklenen Performansı sergilediğini ve 3 numaralı seçenek ise Beklenenin Üstünde Performans sergilediğini yansıtır

	1	2	3	
Beklenenin Altında	○	○	○	Beklenenin Üstünde

16. İnisiyatif alma becerisini değerlendiriniz. *

1 Beklenenin Altında Performans sergilediğini, 2 Beklenen Performansı sergilediğini ve 3 numaralı seçenek ise Beklenenin Üstünde Performans sergilediğini yansıtır

	1	2	3	
Beklenenin Altında	○	○	○	Beklenenin Üstünde

17. Problem çözme becerisini değerlendiriniz. *

1 Beklenenin Altında Performans sergilediğini, 2 Beklenen Performansı sergilediğini ve 3 numaralı seçenek ise Beklenenin Üstünde Performans sergilediğini yansıtır

	1	2	3	
Beklenenin Altında	○	○	○	Beklenenin Üstünde

18. Değişime ve öğrenmeye açık olma becerisini değerlendiriniz. *

1 Beklenenin Altında Performans sergilediğini, 2 Beklenen Performansı sergilediğini ve 3 numaralı seçenek ise Beklenenin Üstünde Performans sergilediğini yansıtır

	1	2	3	
Beklenenin Altında	○	○	○	Beklenenin Üstünde

19. Kaç yıldır yazılım yapıyor(sunuz)? *

- ○ 0 - 2 yıl
- ○ 3 - 5 yıl
- ○ 6 -10 yıl
- ○ 10+ yıl

20. Bugüne kadar üzerinde çalıştığı(nız) proje sayısı kaçtır? *

- ○ 0 - 2
- ○ 3 - 5
- ○ 6 -10
- ○ 10+

Ek D: K-ortalama Kümeleme Algoritması ile Projelerin Gruplanmasını Sağlayan Python Kodu

```
# -*- coding: utf-8 -*-
from time import time
import numpy as np
import matplotlib.pyplot as plt

from sklearn import metrics
from sklearn.cluster import KMeans
from sklearn.datasets import load_digits
from sklearn.decomposition import PCA
from sklearn.preprocessing import scale

from sklearn.preprocessing import Imputer
import pandas as pd


veri = pd.read_csv("projects_v1.csv", sep=',')
#no need to dothis because no missing value existing with raw dataset but i added
some!..
veri.replace('?', -12345, inplace=True)
#no need to remove any row or column but i  added a meaningless row!
#veriyeni = veri.drop('10000', axis=1)
#veriyeni = veri.drop(1, axis=0)

imp = Imputer(missing_values=-12345, strategy="mean",axis=0)
#clears all columns and be ready to be processed!
veriyeni = imp.fit_transform(veri)

inputt = veriyeni[:,0:30]

n_clusters = 4
sample_size = 103
n_features = 31

labels = range(1, 102)

print("clusters: %d, \t n_samples %d, \t n_features %d"
      % (n_clusters, sample_size, n_features))


print(82 * '_')
print('init\t\ttime\tinertia\thomo\tcompl\tv-meas\tARI\tAMI\tsilhouette')


def bench_k_means(estimator, name, data):
    t0 = time()
    estimator.fit(data)
```

```
    print('%-9s\t%.2fs\t%i\t%.3f\t%.3f\t%.3f\t%.3f\t%.3f\t%.3f'
          % (name, (time() - t0), estimator.inertia_,
             metrics.homogeneity_score(labels, estimator.labels_),
             metrics.completeness_score(labels, estimator.labels_),
             metrics.v_measure_score(labels, estimator.labels_),
             metrics.adjusted_rand_score(labels, estimator.labels_),
             metrics.adjusted_mutual_info_score(labels,  estimator.labels_),
             metrics.silhouette_score(data, estimator.labels_,
                                      metric='euclidean',
                                      sample_size=sample_size)))

bench_k_means(KMeans(init='k-means++', n_clusters=n_clusters, n_init=4),
              name="k-means++", data=inputt)

bench_k_means(KMeans(init='random', n_clusters=n_clusters, n_init=4),
              name="random", data=inputt)

# in this case the seeding of the centers is deterministic, hence we run the
# kmeans algorithm only once with n_init=1
pca = PCA(n_components=n_clusters).fit(inputt)
bench_k_means(KMeans(init=pca.components_, n_clusters=n_clusters, n_init=1),
              name="PCA-based",
              data=inputt)
print(82 * '_')

#
##########################################################################
#########
# Visualize the results on PCA-reduced data

reduced_data = PCA(n_components=2).fit_transform(inputt)
kmeans = KMeans(init='k-means++', n_clusters=n_clusters, n_init=4)
kmeans.fit(reduced_data)

# Step size of the mesh. Decrease to increase the quality of the VQ.
h = .02     # point in the mesh [x_min, x_max]x[y_min, y_max].

# Plot the decision boundary. For that, we will assign a color to each
x_min, x_max = reduced_data[:, 0].min() - 1, reduced_data[:, 0].max() + 1
y_min, y_max = reduced_data[:, 1].min() - 1, reduced_data[:, 1].max() + 1
xx, yy = np.meshgrid(np.arange(x_min, x_max, h), np.arange(y_min, y_max, h))

# Obtain labels for each point in mesh. Use last trained model.
Z = kmeans.predict(np.c_[xx.ravel(), yy.ravel()])

# Put the result into a color plot
Z = Z.reshape(xx.shape)
plt.figure(1)
plt.clf()
plt.imshow(Z, interpolation='nearest',
```

```
          extent=(xx.min(), xx.max(), yy.min(), yy.max()),
          cmap=plt.cm.Paired,
          aspect='auto', origin='lower')

plt.plot(reduced_data[:, 0], reduced_data[:, 1], 'k.', markersize=2)
# Plot the centroids as a white X
centroids = kmeans.cluster_centers_
plt.scatter(centroids[:, 0], centroids[:, 1],
           marker='x', s=169, linewidths=3,
           color='w', zorder=10)
plt.title('K-means clustering on the digits dataset (PCA-reduced data)\n'
         'Centroids are marked with white cross')
plt.xlim(x_min, x_max)
plt.ylim(y_min, y_max)
plt.xticks(())
plt.yticks(())
plt.show()

lastkmeans = KMeans(init='k-means++', n_clusters=n_clusters, n_init=4)
lastkmeans.fit(inputt)

for val in inputt:
   print("The cluster for {} is {}".format(val, lastkmeans.predict([val,val])[1]))

print("Raw data")

for val in inputt:
   print("{},{}".format(val, lastkmeans.predict([val,val])[1]))
```

Ek E: YSA ile Projeye Uygun Adayın Seçimini Sağlayan Python Kodu

```
# -*- coding: utf-8 -*-

from keras.models import Sequential

from keras.layers import Dense, Activation

from matplotlib import pyplot

from sklearn.preprocessing import Imputer

import numpy as np

import pandas as pd

veri = pd.read_csv("developers_v2.csv", sep=',')

veri.replace('?', -12345, inplace=True)

developerFeatures = veri

imp = Imputer(missing_values=-12345, strategy="mean",axis=0)

#clears all columns and be ready to be processed!

developerFeatures = imp.fit_transform(developerFeatures)

inputt = developerFeatures[:,0:46]

outputt = developerFeatures[:,46]

model = Sequential();

model.add(Dense(64, input_dim=46))

model.add(Activation('relu'))

model.add(Dense(64))

model.add(Activation('relu'))
```

```
model.add(Dense(32))

model.add(Activation('softmax'))

model.compile(optimizer='adam', loss='sparse_categorical_crossentropy',
metrics=['accuracy'])

model_fit = model.fit(inputt, outputt, epochs=48, batch_size=16,
validation_split=0.1)

guess =
np.array([2.0,1.0,0.0,3.0,0.0,1.0,0.0,0.0,1.0,0.0,0.0,0.0,0.0,1.0,0.0,0.0,0.0,0.0,0.0,0.0,1.0,0.0,
0.0,0.0,1.0,0.0,3.0,3.0,1.0,1.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,3.0,3.0,3.0,3.0,3.0,3.0,3.0,2.0,3.0
,3.0,3.0,3.0]).reshape(1,46)

print("Predict Result: ")

print(model.predict_classes(guess))

# plot metrics

pyplot.plot(model_fit.history['accuracy'])

pyplot.show()

pyplot.plot(model_fit.history['accuracy'])

#pyplot.plot(model_fit.history['val_accuracy'])

pyplot.title('model accuracy')

pyplot.ylabel('accuracy')

pyplot.xlabel('epoch')

pyplot.legend(['train'], loc='upper right')

pyplot.show()
```

ÖZGEÇMİŞ

Gülay GENÇ EROĞLU, 1988 Antakya doğumludur. İstanbul Üniversitesi Bilgisayar Mühendisliği bölümü mezunu olan GENÇ EROĞLU, Gebze Teknik Üniversitesi'nde İşletme dalında yüksek lisans yapmıştır. Çalışma hayatına 2010 yılında başlayan GENÇ EROĞLU, sırasıyla Ericsson, Yapı ve Kredi Bankası bilgi teknolojileri alanlarında yazılım ve analist olarak görev almıştır ve 2016 yılında Türkiye İhracat Kredi Bankası'na katılmıştır. Türk Eximbank'ta hâlen Proje Yöneticisi/Analist olarak görev yapmaktadır.