



# Programlama Eğitiminde Öğrenci Performansının, PLS-SEM ve Makine Öğrenmesi Algoritmalarını Kullanan Karşılaştırmalı Bir Yaklaşımla Tahminlenmesi

Yazılım Mühendisliği Ana Bilim Dalı

Yüksek Lisans

Aykut DURAK

ORCID 0000-0001-7070-9048

Tez Danışmanı: Doç. Dr. Vahide BULUT

Kasım 2023

İzmir Kâtip Çelebi Üniversitesi Fen Bilimleri Enstitüsü öğrencisi **Ayktur DURAK** tarafından hazırlanan **Programlama Eğitiminde Öğrenci Performansının, PLS-SEM ve Makine Öğrenmesi Algoritmalarını Kullanan Karşılaştırmalı Bir Yaklaşımla Tahminlenmesi** başlıklı bu çalışma tarafımızca okunmuş olup, yapılan savunma sınavı sonucunda kapsam ve nitelik açısından başarılı bulunarak jürimiz tarafından YÜKSEK LİSANS olarak kabul edilmiştir.

**ONAYLAYANLAR:**

**Tez Danışmanı:** **Doç. Dr. Vahide BULUT**  
İzmir Kâtip Çelebi Üniversitesi

**Jüri Üyeleri:**

**Doç. Dr. Aytuğ ONAN**  
İzmir Kâtip Çelebi Üniversitesi

**Doç. Dr. Gül ÖZÜDOĞRU**  
Ahi Evran Üniversitesi

**Savunma Tarihi: 01.11.2023**

# Yazarlık Beyanı

Ben, **Aykut DURAK**, başlığı **Programlama Eğitiminde Öğrenci Performansının, PLS-SEM ve Makine Öğrenmesi Algoritmalarını Kullanan Karşılaştırmalı Bir Yaklaşımla Tahminlenmesi** olan bu tezimin ve tezin içinde sunulan bilgilerin şahsıma ait olduğunu beyan ederim. Ayrıca:

- Bu çalışmanın bütünü veya esası bu üniversitede Yüksek Lisans / Doktora derecesi elde etmek üzere çalıştığım süre içinde gerçekleştirilmiştir.
- Daha önce bu tezin herhangi bir kısmı başka bir derece veya yeterlik almak üzere bu üniversiteye veya başka bir kuruma sunulduysa bu açık biçimde ifade edilmiştir.
- Başkalarının yayımlanmış çalışmalarına başvurduğum durumlarda bu çalışmalara açık biçimde atıfta bulundum.
- Başkalarının çalışmalarından alıntıladığımda kaynağı her zaman belirttim. Tezin bu alıntılar dışında kalan kısmı tümüyle benim kendi çalışmamdır.
- Kayda değer yardım aldığım bütün kaynaklara teşekkür ettim.
- Tezde başkalarıyla birlikte gerçekleştirilen çalışmalar varsa onların katkısını ve kendi yaptıklarımı tam olarak açıkladım.

Tarih: 01.11.2023

---

# Programlama Eğitiminde Öğrenci Performansının, PLS-SEM ve Makine Öğrenmesi Algoritmalarını Kullanan Karşılaştırmalı Bir Yaklaşımla Tahminlenmesi

## ÖZ

Bu çalışmada, bilgisayar programlama eğitimi alan öğrencilerin kısmi en küçük kareler yapısal eşitlik modellemesi (PLS-SEM) algoritması kullanarak programlama performansını (düşük, yüksek) etkileyen faktörlerin tahmin edilmesi amaçlanmaktadır. Araştırmanın katılımcılarını programlama eğitimi alan 763 öğrenci oluşturmuştur. Veri setinin analizinde betimsel analizler, makine öğrenmesi algoritmaları ve PLS-SEM analizi kullanılmıştır. Ayrıca düşük ve yüksek programlama performansı gösteren gruplar arasındaki modellerin farklılaşmasını incelemek için çoklu grup yapısal eşitlik modellemesi kullanılmıştır. Verilerin analizinde PLS-SEM yöntemi için SmartPLS 4 programı, makine öğrenmesi algoritmaları sonuçlarının karşılaştırılması için Knime 4.7 programından yararlanılmıştır. Araştırmanın sonuçlarına göre, yüksek düzeyde programlama performansı gösteren grupta, düşük performans gösteren gruba göre modelin açıklama yüzdesi oldukça yüksektir. Araştırmanın bulgularına göre yaş, eğitim düzeyi ve genel akademik başarı ile programlama performans puanları ilişkili bulunmuştur. Ayrıca programlama deneyimi, tutumu ve programlamada yetkilendirmesi ile programlama performansı ilişkilidir. Düşük ve yüksek programlama performans durumlarına göre en yüksek doğruluk sonucuna sahip algoritmanın karar ağaçları [sırasıyla (.966), (.966)] algoritması olduğu sonucuna

ulařılmıştır. Elde edilen bu puanların %90'ın üzerinde olması tahmin performansının yeterli olduđu řeklinde yorumlanabilir.

**Anahtar Sözcükler:** Programlama eğitimi, programlama performansı, bilgi işlemsel kimlik, bilgi işlemsel düşünme perspektifi, programlamada yetkilendirme, programlama kaygısı; PLS-SEM; makine öğrenmesi algoritmaları

# Predicting Student Performance in Programming Education with a Comparative Approach Using PLS-SEM and Machine Learning Algorithms

## Abstract

The aim of the current study was to predict the factors that affect the programming performance (low, high) of students taking computer programming education using the PLS-SEM algorithm. The participants of the study were 763 students who received programming education. Descriptive analysis, machine learning algorithms, and analysis of PLS-SEM were used to analyze the data set. In addition, multi-group structural equation modeling was used to examine the differentiation of models between groups with low and high programming performance. The SmartPLS 4 program was used for the PLS-SEM method in the analysis of the data and the Knime 4.7 program was used to compare the results of the machine learning algorithms. The results of the study show that the percentage of explanation of the model in the group with high programming performance is quite high compared to the group with low performance. The results of the study show that age, education level, general academic performance and programming performance are related. In addition, programming experience, attitude, and programming competence are related to programming performance. The study concluded that the algorithm with the highest accuracy in terms of low and high programming performance is the decision tree algorithm [(0.966),

(.966)]. The fact that these values are above 90% can be interpreted as sufficient estimation performance.

**Keywords:** Programming training, programming performance, computational identity, computational thinking perspective, programming empowerment, programming anxiety; PLS-SEM; machine learning algorithms

Kızım Nurbanu Deniz'e...



# Teşekkür

Yüksek lisans tez çalışmamın gerçekleştirilmesinde değerli görüşlerini ve desteğini esirgemeyen ve her ihtiyaç duyduğumda dönüş sağlayan kıymetli danışmanım Doç. Dr. Vahide BULUT'a;

Jüri üyeleri Doç. Dr. Aytuğ ONAN'a ve Doç. Dr. Gül ÖZÜDOĞRU'ya;

Yüksek lisans eğitimimde her türlü desteği bizlerden esirgemeyen, kendilerinden ders aldığım, alanda önemli kazanımlar elde etmemi sağlayan İzmir Kâtip Çelebi Üniversitesi Yazılım Mühendisliği Anabilim Dalı öğretim üyelerine;

Tezimin veri toplama sürecinde ve tez sürecinin gelişiminde görüşlerini esirgemeyen kıymetli hocam Doç. Dr. Nilüfer ATMAN USLU'ya;

Bu tezin veri toplama süreçlerinin gerçekleşmesine doğrudan ya da dolaylı olarak katkı sağlayan tüm öğretim üyelerine, okul idarecilerine, öğretmenlere ve öğrencilere;

Varlığıyla bana güç veren kızım Nurbanu Deniz DURAK'a ve eşim Hatice YILDIZ DURAK'a teşekkürü bir borç bilirim.

# İçindekiler

<b>Yazarlık Beyanı .....</b>	<b>ii</b>
<b>Öz.....</b>	<b>iii</b>
<b>Abstract.....</b>	<b>v</b>
<b>Teşekkür .....</b>	<b>viii</b>
<b>Şekiller Listesi .....</b>	<b>xii</b>
<b>Tablolar Listesi.....</b>	<b>xiii</b>
<b>Kısaltmalar Listesi .....</b>	<b>xiv</b>
<b>Bölüm 1 .....</b>	<b>1</b>
<b>Giriş .....</b>	<b>1</b>
1.1. Programlamada Yetkilendirme.....	2
1.2. Programlama Kaygısı .....	3
1.3. Bilgi İşlemsel Kimlik .....	4
1.4. Bilgi İşlemsel Düşünme Perspektifi .....	5
1.5. Sınırlıklar.....	5
<b>İlgili Araştırmalar .....</b>	<b>7</b>
<b>Genel Bilgiler .....</b>	<b>10</b>
3.1 Makine Öğrenmesi .....	10
3.1.1. Makine Öğrenmesi Algoritmaları.....	10

3.2. PLS-SEM Algoritması .....	14
<b>Materyal ve Yöntemler .....</b>	<b>16</b>
4.1. Araştırma Modeli, Hipotezler ve Katılımcılar.....	16
4.2. Veri Toplama Araçları.....	19
4.3. Veri Seti.....	21
4.3.1. Verilerin Toplanması .....	21
4.3.2. Veri Öznitelikleri ve Analizler .....	21
<b>Bulgular.....</b>	<b>24</b>
5.1. PLS-SEM.....	24
5.1.1. Ölçüm Modelinin Değerlendirilmesi .....	24
5.1.2. Yapısal Eşitlik Modeli .....	32
5.1.3. Çoklu Grup Analizi.....	35
5.2. Makine Öğrenmesi Algoritmaları.....	38
5.2.1. Modelin Doğruluğu (Accuracy).....	38
5.2.2. Modelin Hassasiyeti (Precision) .....	40
5.2.3. Modelin Duyarlılığı (Recall) .....	41
<b>Tartışmalar .....</b>	<b>42</b>
<b>Sonuç ve Öneriler.....</b>	<b>45</b>
<b>Kaynaklar .....</b>	<b>46</b>
<b>Ekler .....</b>	<b>54</b>
<b>Ek A.....</b>	<b>55</b>

<b>Ek B .....</b>	<b>56</b>
<b>Özgeçmiş .....</b>	<b>59</b>

# Şekiller Listesi

Şekil 4.1: Araştırma modeli .....	16
Şekil 4.2: Veri analizi prosedürü.....	23
Şekil 5.1: Yapısal Eşitlik Modeli .....	32

# Tablolar Listesi

Tablo 4.1: Katılımcı özellikleri .....	18
Tablo 5.1: Faktör yükleri.....	25
Tablo 5.2: Yapı güvenilirliği ve geçerliliği.....	26
Tablo 5.3: Fornell-Larcker kriterleri .....	27
Tablo 5.4: Heterotrait-Monotrait Ratio (HTMT) .....	30
Tablo 5.5. Hipotez testi sonuçları .....	33
Tablo 5.6: Çoklu grup analizi sonuçları .....	36
Tablo 5.7: Çıktı değişkenine göre algoritma doğruluk sonuçları.....	38
Tablo 5.8: Çıktı değişkenlere göre algoritma çapraz doğruluk sonuçları .....	39
Tablo 5.9: Çıktı değişkenlere göre algoritma hassasiyet sonuçları.....	40
Tablo 5.10: Çıktı değişkenlere göre algoritma duyarlılık sonuçları .....	41

# Kısaltmalar Listesi

CA	Cronbach's alpha
CB-SEM	Kovaryans tabanlı yapısal eşitlik modellemesi (covariance-based structural equation modeling)
CR	Bileşik güvenilirlik (composite reliability)
kNN	k-nearest neighbors
LR	Logistic regression
MGA	Çoklu Grup Analizi (Multi Group Analysis)
PLS-SEM	Kısmi en küçük kareler yapısal eşitlik modellemesi (partial least squares structural equation modeling)
RF	Random forest
SVM	Support vector machines
UTAUT2	Birleşik kabul ve teknoloji kullanımı teorisi
YEM	Yapısal eşitlik modellemesi

# Bölüm 1

## Giriş

Bilgisayar eğitimine katılımı desteklemek, bu eğitimi yaygınlaştırmak, günümüz toplumunun sosyal ve ekonomik olarak ilerlemesi için önemlidir. Bilgisayar bilimi eğitiminde, programlamanın önemi her geçen gün artmaktadır. Nitekim programlama öğretimi, öğrencilerin bilgisayar bilimlerindeki kariyer kararlarını, ilgi ve meşguliyetlerini geliştirmek için giderek daha yararlı hale gelmiştir. Öte yandan programlama performansı sadece bilgisayar bilimlerindeki ilerlemelere işaret etmemektedir. Kong ve Lin [1] tarafından yapılan çalışmada, programlamada yüksek performans gösterenlerin, programlamaya yönelik tutumlarının akış deneyimleriyle ve programlama niyetleriyle pozitif ilişkili olduğu gösterilmiştir. Ayrıca, yüksek programlama performansı gösteren öğrencilerin bilgiyi işleme, stratejiler oluşturma ve çözümler yaratma konusunda daha güçlü yeterlikleri olduğu vurgulanmıştır. Wang vd. [2], programlamada yüksek başarı gösterenlerin daha karmaşık bilgi çerçeveleri ve daha güçlü kendi kendine öğrenme becerileri olduğunu vurgulamıştır.

Programlama performansını etkileyen değişkenlerin incelenmesi, bilgisayar bilimi kariyer ilgisini ve çeşitli üst düzey bilişsel becerilerini geliştirme açısından önemlidir. Bununla birlikte programlamanın öğrenilmesine yönelik ilgi her geçen gün artmaktadır [3]. Öte yandan programlama, öğrenciler için algılanması ve öğrenilmesi zor bir konudur [4] ve bu algılanan zorluk programlamaya yönelik ilgi, tutum, motivasyon ve öz yeterlikleri olumsuz yönde etkilemektedir [5]. Bu nedenle, öğrencilerin bilgisayar bilimi ile ilgili kendilerini nasıl gördüklerini açıklayan bilgi işlemsel kimlik, bilgi işlemsel düşünme perspektifi, programlama yetkilendirme ve programlama kaygısının, programlama performansı ile ilişkisinin araştırılması önemlidir. Yüksek ve düşük programlama performansı ile ilişkili yapıların açıklanması ve bu faktörlerden hangilerinin en önemli olduğunu belirlenmesi



programlama performansının kimler için desteklenebileceğini anlamak ve bu deneyimleri geliştirmeyi kolaylaştırmak için önemli bir konu olarak karşımıza çıkmaktadır

Bu çalışma, öğrencilerin mevcut durumunu belirlemek, bilgi işlemsel kimlik gelişimini teşvik etmek amacı taşımaktadır. Bu çalışma sonucunda elde edilen veriler, öğrencilerin bilgisayar bilimi kariyerlerine yönelik ilgi yaratmak üzere eğitim müdahalelerinin nasıl tasarlanacağına ışık tutabilir.

Tüm bu gerekçeler bağlamında, bu tez kapsamında, bilgisayar programlama eğitimi alan öğrencilerin, öz bildirime dayalı veri toplama aracına verdikleri yanıtların PLS-SEM ve makine öğrenmesi algoritmaları kullanarak analiz edilmesi yoluyla, programlama performansını (düşük, yüksek) etkileyen faktörlerin tahmin edilmesi amaçlanmaktadır.

## 1.1. Programlamada Yetkilendirme

Yetkilendirme, ilk olarak sosyal bilimlerde kullanılmıştır. Daha sonra eğitim alanında da bu kavram kendine yer bulmuştur. Bireyler, öğrenen ve anlam yaratan kişiler olarak kendilerini yetkin hissettiklerinde, eğitim güçlendirici olmaktadır. Temelde yetkilendirme, insanlara kendi yaşamları ve kendilerini ilgilendiren konular üzerinde kontrol sağlamayı vurgulayan bir yapıdır [6]. Yetkilendirmeyi tek kavram olarak alan araştırmacıların (örn. [7]) yanı sıra bu kavramı birden fazla bileşene ait yapı olarak da değerlendirenler (örn. [8]) bulunmaktadır. Yetkilendirme kavramı genel olarak öz-kontrol, otonomi, kendi kaderini tayin etme, öz-güven gibi kavramlarla ilişkilendirilmektedir. Yetkilendirme kavramına bireylerin bir özelliği olarak bakmak, bu kavramın bireyler tarafından kazanılacağını ve geliştirilebileceğini düşündürmektedir [9].

Thomas ve Velthouse [10] yetkilendirme yapısının dört bileşenden (anlamlılık, etki, kendi kaderini tayin etme/seçim ve yeterlilik) oluştuğunu düşünmektedir. Frymier vd. [11] ise eğitim bağlamında bu bileşenleri anlamlılık, etki ve yetkinlik bağlamında ele almıştır. Programlama eğitimi bağlamında ise Kong vd. [12] bir çerçeve önermiştir. Nitekim Zimmerman'a [13] göre yetkilendirme alana özgü bir süreçtir. Kong vd.[12] öğrencilerin kendi kaderini tayin etme durumundan ziyade yetkilendirmeyi algılanan

yaratıcılığın incelemesi bakışıyla ele almıştır. Bu bakış açısıyla programlama yetkilendirme, öğrencilerin bir programlama görevinin amacını anlamlı olarak kabul ettiklerinde, görevleri tamamlamanın etkili olduğunu düşündüklerinde, programlama görevlerini yaratıcı bir şekilde tamamlayabileceklerine inandıklarında ve bu görevleri tamamlamak için yeterlilik düzeylerini yeterli algıladıklarında ortaya çıkan bir yapıdır. Kong vd. [12] tarafından tanımlanan çerçeveye göre anlamlılık, öğrencinin bir görevin amacına ilişkin algıladığı değerdir. Bu bağlamda bir öğrenme sürecinde, öğrenme görevini anlamlı olarak algılayan öğrencilerin kendilerini güçlü hissetme olasılıklarının yüksek olması beklenir. Etki, bir görevin başarılmasının bireyde yarattığı farkı algılama derecesidir ve görevi tamamlamanın bir etkisi olduğuna inanan bir kişi daha fazla iç motivasyona sahip olacaktır. Programlama ile yapılan katkı öğrencilerin akademik yaşamları üzerinde olumlu etkiler bırakır. Programlama öz yeterliliği, kişinin bir programlama görevini iyi bir şekilde yerine getirmek için gerekli beceri ve yeteneklere sahip olduğuna dair inancıdır [14], [15]. Öğrenciler daha fazla öz yeterliliğe sahip olduklarında, bir görevi yapma yeterliliklerine daha fazla güvenirler. Bir programlama görevini başarıyla tamamlayabileceklerine inanan öğrencilerin kendilerini güçlü hissetme olasılıkları daha yüksektir. Dolayısıyla, bu yetkilendirme bileşenleri görevin başarıyla tamamlanması olasılığını artırabilir. Sonuç olarak, problem çözme, akıl yürütme gibi üst düzey düşünme, çeşitli bilişsel beceriler ve deneyimi birlikte gerektiren programlama performansı [16] ile programlama yetkilendirmenin ilişkili olduğu düşünülmektedir [17].

## 1.2. Programlama Kaygısı

Kaygı, stres yaratan durumların neden olduğu üzüntü ve gerginlik gibi negatif duygu ve gözlenebilen tepkilerdir. Bilgisayar bilimi öğretiminde kaygı, bilgisayara yönelik kaygı kavramı ile ele alınırken, programlama eğitiminde de programlama kaygısı ile kavramsallaştırılır. Literatürde, bilgisayar kaygısı, bilgisayar teknolojisinin kullanıldığı düşünüldüğünde ya da gerçekten bilgisayar kullanıldığında yaşanan endişedir [18]. Kaygı, bireyin teknolojiyi kontrol etme ve yönetme durumunda teknolojinin kullanımına ilişkin bilgi ve beceri eksikliğine sahip olduğu hissine kapılmasıyla ortaya çıkabilir [19], [20].

Programlama kaygısı ise programlama ile ilgili bir kaygı yaşanması durumudur. Programlama kaygısı, öğrencilerin öğrenmede sorun yaşamalarına neden olabilir. Bu nedenle, programlamanın öğretimi sürecinde karşılaşılan zorluklardan birinin programlama kaygısı [21] olduğu söylenebilir. Nitekim Mazzone vd. [22] öğrencilerin kaygıları ile akademik performansları arasında negatif yönde bir ilişki olduğunu, Çelik ve Yeşilyurt [18] ise kaygının öz yeterliliği doğrudan veya dolaylı olarak etkilediğini vurgulamaktadır. Öte yandan, Azarfam ve Jabbari [23] bilgisayar kaygısının, eğitimde teknoloji entegrasyonunu olumsuz etkileyen bir faktör olduğu için dikkate alınması gerektiğini belirtmektedir. Buna ek olarak, kaygının akademik ertelemeye neden olduğu da belirtilmiştir (örn. [24]). Bu çalışmada, programlama kaygısı, programlama performansına yönelik önemli çıkarımlar yapılmasına katkı sağlayabileceğinden araştırmaya dahil edilmiştir.

### 1.3. Bilgi İşlemsel Kimlik

Kimlik, bireyin kendini, insanların birey olarak nasıl algıladıkları ve deneyimledikleriyle ilgilidir [25]. Kimlik farklı disiplinler için önemli bir ilgi konusu olarak ele alınabilir. Wenger [26] ise öğrenmenin "bir kimlik deneyimi" olduğunu belirtir ve öğrenme ile kimliğin özünde birbirine bağlı olduğunu vurgular. Sfard ve Prusak [27] ise öğrencilerin kimlik inşasına yönelik araştırmaların, öğrencilerin eğitim deneyimlerini ve akademik sonuçlarını nasıl algıladıklarına ışık tutabileceğini iddia etmektedir. Disiplinler arası bağlamda kimlik, öğrencilerin ilgili disiplinle ilgili deneyimleri geliştirdikçe şekillenir. Bu bağlamda kimlik yapısının bilgisayar bilimi için uyarlanabileceği görülmektedir [28].

Kong ve Lai [8], kimliği bilgi işlemsel düşünme bağlamında ele almıştır. Bilgi işlemsel kimlik, okuldaki programlama ve bilgi işlemsel düşünme etkinliklerine katılımı ilişkilendirilmiştir. Kong ve Wang [29] bilgi işlemsel kimliği, “programlama meşguliyeti, programlama aidiyeti, programlama gerçekleştirme ve programlama hedef belirleme” boyutları bağlamında ele almıştır. Programlama meşguliyeti, programlama aktivitelerine yönelik aktif öğrenme sürecidir. Programlama aidiyeti, ortak programlama çıkarlarını paylaşan öğrenme grubuna bağlılık hissidir. Programlama gerçekleştirme, öğrencilerin programlama öğreniminde kendilerini gerçekleştirmeleridir. Programlama hedef belirleme ise, öğrencilerin programlama

becerilerini geliştirecek hedefler belirleyerek kariyer seçimlerini ve daha zengin deneyimler yaşamalarını sağlamayı amaçlamaktadır. Bilgi işlemsel kimlik öğrencilerin kendi çözümlerini tasarlama sorumluluğunu taşımasını gerektirdiğinden, yüksek düzeyde bilgi işlemsel kimlikle, daha yüksek programlama performansının ilişkili olabileceği düşünülmüştür.

## 1.4. Bilgi İşlemsel Düşünme Perspektifi

Bilgi işlemsel düşünme, bilgisayar bilimi için temel kavramları çerçevesinde problem çözmeyi, sistem tasarlamayı ve insan davranışını anlamayı içeren bir süreçtir [30]; [31]. Bu kavramın öncüsü olan Wing'e [32] göre bilgi işlemsel düşünme problemi anlama, uygun çözümler üretme, soyutlama ve yansıtma ile ilgili düşünme örüntülerini içerir [31]. Ayrıca bilgi işlemsel düşünme verilerin analizi, veri özetleme ve modelleme gibi becerileri de kapsamaktadır [33]. Kısaca, bilgi işlemsel düşünme karmaşık problemlerin çözümünde bilgisayar bilimi kavramlarının etkin kullanımıyla ilişkilidir [34]. Bilgi işlemsel düşünme perspektifi ise bireylerin kendilerini bilgi işlemsel düşünür (computational thinker) olarak görme biçimlerini ifade eder.

Öte yandan bilgi işlemsel düşünme, programlama derslerinde geliştirilmesi amaçlanan temel bilişsel becerileri desteklemede etkilidir [1]. 21. yüzyılın en önemli becerileri arasında sayılan bilgisayar programlama becerisi, bilgi işlemsel düşünce becerilerini geliştirmek için temel bir strateji olarak ele alınır [35], [36], [32]. Nitekim bilgi işlemsel düşünme farklı üst düzey düşünme becerileri ve bilgi alanlarını içermektedir. Bu bilgi alanları ve becerilerin geliştirilmesinde programlama süreçleri etkin rol oynamaktadır. Sadece bilgisayar bilimlerinde kariyer yapmaya ilgi duyanlar için değil, herkes için önemli beceriler olan bilgi işlemsel düşünme ve programlama becerilerinin gelişimini destekleyebilmek için bu yapıların ilişkisinin incelenmesi önemlidir.

## 1.5. Sınırlıklar

Mevcut çalışmanın bazı sınırlıkları vardır. İlk olarak, bu çalışmada, programlama dersi alan öğrencilerin düşük ve yüksek bilgisayar programlama performansına yönelik yordayıcı değişkenlerin tahmin edilmesine odaklanmıştır. Ancak farklı eğitim düzeylerinde programlama dersinin farklılık içerikleri ve etkililiği farklılık

gösterebilir. Bu çalışma bu konuda sınırlı bilgi içermektedir. Benzer şekilde farklı eğitim kurumlarında programlama performans puanının ölçümleri çeşitlilik gösterebilir. Bu çeşitlilik, yüksek ve düşük gruptaki öğrenciler arası değişkenliğe neden olmuş olabilir.

## Bölüm 2

### İlgili Araştırmalar

Bu bölümde, bilgisayar programlama ve araştırma değişkenleri ile ilgili PLS-SEM ve makine öğrenmesini kullanan çalışmalar incelenmiştir. Çalışmalar en güncelden geçmişe doğru sıralanmıştır.

Sivasakthi ve Padmanabhan [37] tarafından yapılan çalışmada, öğrencilerin programlama performansını tahmin etmek için iki sınıflandırma algoritması (Naive Bayes ve karar ağacı) kullanılarak bir model önerilmiştir. Araştırmanın katılımcılarını bilgisayar bölümü lisans öğrencileri oluşturmaktadır. Bu çalışmada sınıflandırma stratejilerinin temel amacı, öğrencilerin zayıf noktalarını belirlemelerine ve programlama becerilerini geliştirmelerine yardımcı olmaktır. Test sonuçları, Naive Bayes'in %91.02'lik yüksek doğrulukla karar ağacından daha iyi olduğunu göstermiştir.

Su vd. [38] tarafından yapılan çalışmada, programlama uygulamalarında öğrencilerin öğrenme çıktılarını tahmin etmek amaçlanmıştır. Öğrencilerin öğrenme özellikleri ile programlama performansı arasındaki ilişkiyi anlamak için makine öğrenmesi teknolojileri kullanılmıştır. Programlama uygulamaları altı hafta olarak, “giriş/çıkış, temel veri türleri, diziler, diziler, yapılar, dinamik programlama, özyineleme ve sanal işlevleri” konularını içerecek şekilde yapılandırılmıştır. Etkinlik sonunda, sertifika almak için her soru puanlanmıştır. Bu çalışmada, tahmin için destek vektör makineleri (DVM), lojistik regresyon (LR), rastgele orman (RO) ve en yakın komşu (kNN) algoritmaları kullanılmıştır. kNN algoritması en iyi performansı, destek vektör makineleri ve rastgele orman algoritmaları ikinci performansı ve lojistik regresyon algoritması ise en kötü performansı göstermiştir.

Anand vd. [39] tarafından gerçekleştirilen çalışmada, programlama dersinin performansını açıklamak üzere öğrencilerin önkoşul derslerdeki performanslarına, ek koşullara ve ders çalışma durumlarına bakılmıştır. Öğrencileri sınıflandırmak için özyinelemeli kümeleme tekniği adı verilen makine öğrenmesi yaklaşımını kullanılmıştır. Alt gruplarda bulunan öğrencilerin her birine, gruplarına göre otomatik eğitim seti ve notlar sağlanmıştır. Bir süre sonra yeni bir değerlendirme yapılmış ve öğrenciler güncel performanslarına göre kümelenmiştir. Alt gruptaki öğrencilerin çoğu üst gruba geçecek şekilde bu işlem üç kez tekrarlanmıştır. Özyinelemeli kümeleme tekniği uygulanmadan önce ve sonra sonuçlar her gruptaki öğrenci sayıları ile karşılaştırılmıştır. Sonuçlar, bu yaklaşımın düşük performans gösteren öğrencilerin değerlendirmelerini sağlamak ve tahmin etmek için etkili bir yol sağladığını ve böylece öğrencinin daha etkin yönlendirildiğini göstermiştir.

Marjan vd. [40] tarafından yapılan çalışmada, yükseköğretim öğrencilerinin bilgisayar programlama becerilerini ölçmeye ve geliştirmeye yönelik bir sistem önerilmiştir. Önerilen sistem, (i) sınıflandırma süreci ve (ii) öğrenme süreci için iki anahtar modül içermektedir. Sınıflandırma modülü, bir öğrencinin mevcut durumunu tahmin etmektedir. Öğrenme süreci modülü, öğrencinin öğrenme kalitesini artırmak için önerilerin ve geri bildirimlerin oluşturulmasına yardımcı olmaktadır. Sınıflandırma modülü için, görevlerle ilgili gerçek bir veri seti hazırlanmıştır. Bu veri seti altı makine öğrenmesi algoritmaları (DVM, karar ağacı, yapay sinir ağları, RO, kNN ve naive bayes sınıflandırıcısı) kullanılarak, performans ölçüm metriklerini ve uyum iyiliği incelenmiştir. Deneysel sonuçlar, RO ve DVM'nin diğer modellere göre daha doğru tahmin sonuçları ürettiğini göstermektedir. Ek olarak, yüksek sınıflandırma doğruluğu elde etmeye yönelik kritik özellikleri belirlemek için kritik faktör analizi gerçekleştirilmiştir. Son olarak, öğrenme süreci modülünde, öğrencilerin programlama becerilerini geliştirmelerine yardımcı olacak bir iyileştirme mekanizması tasarlanmıştır.

Bilegjargal ve Hsueh [41] tarafından yapılan çalışmada, çevrimiçi programlama yarışmalarındaki ortamları değerlendirmek için bir çevrimiçi jüri sistemi geliştirilmiştir. Bu sistem, öğrencilerin programlama becerilerini geliştirmek ve onlara programlama uygulamalarında yardımcı olmak için uygulanmıştır. Uygulama sonrası bu sistemin çevrimiçi programlama derslerinde öğrenciler tarafından kabulü

araştırılmıştır. Bu çalışmada, birleşik kabul ve teknoloji kullanımı teorisi (UTAUT2) modeli ve PLS-SEM algoritması uygulanmıştır. 187 lisans öğrencisiyle gerçekleştirilen bu çalışmanın sonuçları, en çok hedonik motivasyon, öz yeterlilik ve sosyal etkinin öğrencilerin jüri sistemini kullanma niyeti üzerinde önemli olduğunu göstermiştir.

Sivasakthi [42] tarafından yapılan çalışmada, tahmine dayalı bir veri madenciliği modeli kullanılmıştır. Bu model ile bilgisayar uygulamaları dersi birinci sınıf lisans öğrencilerinin programlamaya giriş performanslarının tahmin edilmesi amaçlanmıştır. Toplanan veriler, öğrencilerin demografik özelliklerini, giriş programlama notunu içermektedir. WEKA kullanılarak multilayer perception, naive bayes, SMO, J48 ve REPTree gibi çeşitli sınıflandırma algoritmaları uygulanmıştır. Sonuç olarak, en iyi performans gösteren sınıflandırma algoritmasını bulmak için beş sınıflandırıcının hepsinin karşılaştırması yapılmıştır. Bu beş sınıflandırıcı için bir bilgi akış modeli çizilmiştir.

ElGamal [43] tarafından yapılan çalışmada, programlama derslerinde öğrenci performansını tahmin etmek için bir eğitsel veri madenciliği modeli sunulması amaçlanmıştır. Söz konusu çalışma, öğrencilerin matematik dersi deneyimleri, programlama yeteneği, problem çözme becerileri, cinsiyet, lise matematik notu, yaşadığı yer, bilgisayar programlama deneyimi ve e-öğrenme kullanımı gibi faktörleri içermektedir. Önerilen model üç aşamadan oluşmaktadır; veri ön işleme, öznelik seçimi ve kural çıkarma algoritmasıdır. Çıkarılan kurallar, lise matematik puanı/notu ve programlama yeteneği değişkenlerinin önemini doğrulamıştır. Eğitimcilerin, üniversitelerdeki bilgisayar bilimleri dersi performansını desteklemek için bu değişkenleri dikkate alması önerilmektedir.



# Bölüm 3

## Genel Bilgiler

### 3.1 Makine Öğrenmesi

Makine öğrenmesi, yapay zekanın bir alanıdır. Bu bölümde, mevcut çalışmada kullanılan çeşitli makine öğrenmesi algoritmalarının açıklaması yapılmıştır.

#### 3.1.1. Makine Öğrenmesi Algoritmaları

Bu çalışmada, araştırma modeli için kullanılan kestirim ve sınıflandırma teknikleri şu şekildedir:

- Karar ağaçları
- En yakın k-Komşu
- Naive bayes sınıflandırıcı
- Destek vektör makineleri
- Lojistik regresyon

**Karar ağaçları:** Bu algoritma, verileri belirli bir parametreye göre sürekli olarak bölerek sınıflandırma ve regresyon problemlerini çözmek için kullanılan bir denetimli makine öğrenmesi yaklaşımıdır [44]. Bu teknik eş zamanlı sonuçlar elde edilmesine olanak tanıdığı için diğer tahminleme tekniklerinden daha çok tercih edilmektedir. Sınıflandırma ağacında karar değişkeni kategoriktir. Regresyon ağacında ise karar değişkeni sürekli dir. Karar ağacının avantajları, regresyon ve sınıflandırma problemlerinde kullanılabilmesi, kategorik ve nicel değerleri işleme kolaylığı,

niteliklerdeki eksik değerleri en olası değerle doldurabilmesidir. Bu algoritma, veri hakkında gerekli özetler türetebilmek ve sınıflama doğruluğunu incelemek için tercih edilmektedir. CART, ID3 ve rastgele orman yöntemleri en sık kullanılan tekniklerdendir. Karar ağaçları, büyük veri kümelerinde çok iyi performans gösterirken, küçük veri kümeleri için önerilmemektedir. Karar ağaçlarında bölme için en popüler kriter, gini impurity için "gini", information gain için "entropi" değeridir. Analizlerde Knime programında Gini kriteri seçilmiştir ve kullanılan denklem Denklem 3.1'de [45] sunulmuştur.

$$\text{Entropy: } H(x) = - \sum_{i=1}^n p(x_i) \log_2 p(x_i) \quad (3.1)$$

$$\text{Gini}(E) = 1 - \sum_{i=1}^c p_i^2$$

**En yakın k-Komşu (kNN):** Filtreleme tekniği olarak kullanılan bu teknikte, veri setlerinde yer alan kullanıcılar farklı özellikleri yönünden benzerliklerine göre sınıflandırılmaktadırlar. kNN hem sınıflandırma hem de regresyon için kullanılabilir. Kullanıcılar sınıflandırıldıktan sonra yeni gelen kullanıcı k adet yakınlığı olan kullanıcı ile komşu olacak biçimde konumlandırılmaktadır. Kullanıcıların arasındaki mesafeler kosinüs, pearson benzerliği ve Öklid uzaklığı gibi ölçüm yöntemleri ile hesaplanabilmektedir. Bu ölçümler sonucunda yeni kullanıcının k adet komşuları ile benzer özellikler gösterdiğine ulaşılır. En çok benzerliklerin olduğu n kadar kullanıcı ile öneriler top-n şeklinde verilir. kNN verileri üzerinde, veri noktalarının benzerlik ölçülerine göre yapılan sınıflandırma, her noktanın k en yakın komşusunun salt çoğunluk oyuyla hesaplanır [46]. Bu yöntemin bazı avantajları vardır. Bu yöntem gürültülü (noisy) eğitim verilerine karşı oldukça dayanıklıdır ve doğruluğu veri kalitesine bağlıdır. Bansal vd. [47]'ye göre kNN özellikle veriler belirsiz olduğunda uygulama için birincil tercih olabilir. kNN algoritması, denetimli makine öğrenmesi tekniği altında konumlandırılmıştır ve kullanımı en kolay algoritmalarından biridir. Hem sınıflandırma hem de regresyon için uygun olmasına rağmen, ağırlıklı olarak sınıflandırma için kullanılır. Herhangi bir eksik değeri atamak ve verileri yeniden örnekleme için kullanışlıdır. Dezavantajları ise kNN'de, dikkate alınacak en uygun komşu sayısını seçmektir. Genel bir iç model oluşturmaya odaklanmaz. Bu çalışmada

düşük ve yüksek programlama performansı olmak üzere iki vaka vardır. Düşük ve yüksek performans olarak belirtilen iki veri noktası atanmıştır. kNN'de 'k', veri noktasının komşu sayısını ifade eder. k için uygun bir değere karar vermek en önemli süreçtir ve bu işleme parametre ayarı denir. 1 veya 2 gibi çok düşük bir k değeri gürültülü sonuçlara yol açar. Knime’da k değeri 3 olarak belirlenmiştir. kNN için formüle dayalı hesaplama Denklem 3.2 ile yapılabilir [47]:

$$d(x, y) = \sqrt{\sum_{i=1}^n (x_i - y_i)^2} \quad (3.2)$$

**Destek vektör makineleri:** Bu algoritma hem sınıflandırma hem de regresyon problemleri için kullanılabilir. DVM, eğitim veri setindeki duruma göre nesnelere doğru bir şekilde sınıflandırmayı amaçlar. Bu algoritma ile hem yarı yapılandırılmış hem de yapılandırılmış veriler işlenebilir. Yüksek boyutlu verilerle çalışabilir. DVM sınıflandırıcısının temel amacı, sınıfları ayırmak için en uygun hiperdüzlemi bulmaktır. Bu şekilde iki sınıfı maksimum bir marjla ayıran bir hiper düzlem belirlenir. Yüksek boyutlu uzaylarda etkilidir. Farklı matematiksel işlemlere göre farklı davranabilir. Bununla birlikte, veri seti fazla gürültü içerdiğinde, DVM iyi performans göstermez [48]. Bu algoritmalar oldukça gelişmiştir ve birçok alanda kullanılabilir. DVM kanser teşhisi, kredi kartlarında dolandırıcılık tespiti, yüz tespiti ve metin sınıflandırması gibi alanlarda uygulanabilir. Dolayısıyla, gözlem sayısı ve öznelik sayısı yüksek olduğunda DVM denenebilir [49]. DVM’nin diğer avantajları, yarı yapılandırılmış ve yapılandırılmış verileri işleyebilme, yüksek boyutlu verilerle ölçeklenebilme şeklindedir [48].

**Naive bayes:** Bu algoritma basit bir yapıdadır ve koşullu olasılığa dayalıdır. Bu teknik farklı olgular arasında olasılıksal ilişkinin ortaya konması amacı için sınıf olasılıklarına bakılmasını gerektirir. Bu teknik için Singular Value Decomposition ve Matrix factorization gibi algoritmalar kullanılmaktadır. Naive bayes’in uygulaması kolaydır ve daha az eğitim verisi ile çalışır. İkili ve çok sınıflı sınıflandırma problemleri için kullanılabilir. Özellikle birçok değişken içeren modeller için hesaplama açısından yoğun bir tablo oluşturur. Naive bayes, öneri sistemi gibi uygulamalarda kullanılabilir. Bu algoritma belge veya metin sınıflandırması, spam filtreleme gibi birçok gerçek dünya probleminde hem ikili hem de çoklu sınıf

kategorileri için kullanılabilir. Gürültülü verileri etkili bir şekilde sınıflandırmak ve sağlam bir tahmin modeli oluşturmak için etkilidir [46]. Bununla birlikte bazı dezavantajlara sahiptir. İncelenen özelliklerin bağımsızlığına ilişkin varsayımları nedeniyle performansı etkileyebilir. Kullanılan denklem Denklem 3.3'de sunulmuştur.

$$P(A|B) = \frac{P(B|A)*P(A)}{P(B)} \quad (3.3)$$

Denklemden geçen  $P(A|B)$  B olayı gerçekleştiğinde A olayının gerçekleşme olasılığıdır.  $P(A)$ , A olayının gerçekleşme olasılığı,  $P(B)$  ise B olayının gerçekleşme olasılığını ifade eder.  $P(B|A)$ , A olayı gerçekleştiğinde B olayının gerçekleşme olasılığının göstergesidir.

**Lojistik regresyon:** Bu algoritma bir sınıflandırma problemini çözmek için kullanılır. Bu algoritma girdi değişkenlerinin değerlerine bağlı olarak bir olayın gerçekleşme/gerçekleşmeme olasılığını (0 ve 1 cinsinden) verdiği için binom sonucunu üretir. Lojistik regresyonun çok terimli, sıralı sonuçları da olabilir. Dolayısıyla lojistik regresyon, kategorik olan hedef değişkenin tahminiyle ilgilenir. Bu algoritma düşük ve yüksek performans sınıflarını belirlemede hangi girdi özelliklerinin en yararlı olduğunu öğrenmek için etkili bir sınıflandırıcıdır. Lojistik regresyonda bir kesme noktası elde etmek için özelleştirilmiş performans ölçütlerinin belirtilmesi gerekir. Bu bağlamda bu çalışmada programlama performans puanının düşük ve yüksek düzeyleri için 50 puan (0-100 arası skalada) kesme noktası olarak belirlenmiştir. Nitekim eğitim kurumlarının programlama dersinden başarılı sayılabilmek için belirlediği değer bu şekildedir. Lojistik regresyon uygulama basitliği, hesaplama verimliliği, eğitim açısından verimlilik, düzenleme kolaylığı gibi avantajlar sağlar. Ayrıca lojistik regresyon, verilerdeki çoklu bağlantıdan etkilenmez. Doğrusal regresyon ise sürekli değişken değerlerinin tahminini yapar [49]. Ek olarak, Sarker [46]'e göre bu algoritma yüksek boyutlu veri kümelerinde uyumlu çalışabilir. Veri kümesi doğrusal olarak ayrılabilirliğinde iyi performans gösterir. Lojistik regresyon bazı dezavantajları barındırmaktadır. Karar mekanizması doğrusal olduğundan, doğrusal olmayan problemlerin çözülememesi ve tüm bağımsız değişkenlerin tanımlanmadıkça iyi çalışmaması bu dezavantajlardan bazılarıdır. Lojistik regresyon, tipik olarak olasılıkları tahmin etmek için Denklem 3.4'de sigmoid fonksiyonu olarak da adlandırılan fonksiyonu kullanır [46].

$$g(z) = \frac{1}{1+\exp(-z)} \quad (3.4)$$

Lojistik regresyonun, sınıflandırma yapmak kullandığı sigmoid fonksiyonu, verileri 0 ve 1 arasına sıkıştırmak için kullanılan fonksiyondur. Denklem bu fonksiyonun ifadesidir.

**Rastgele orman algoritması:** Bu algoritma, birden fazla karar ağacı üzerinden, her bir karar ağacını farklı bir gözlem örneği üzerinde eğitir. Bu şekilde çeşitli modeller üreterek sınıflandırma oluşturmayı sağlamaktadır. Bu algoritma çeşitli uygulama problemlerinde, makine öğrenmesi ve veri bilimi alanında kullanılan bir topluluk sınıflandırma tekniğidir [46]. Bu yöntem, farklı veri seti alt örneklemelerinde paralel olarak birkaç karar ağacı sınıflandırıcısına uyan "paralel gruplandırma" kullanır. Böylece karar ağaçlarının dezavantajları olan, yüksek varyasyonu azaltır, aşırı uydurma problemini en aza indirir ve tahmin doğruluğunun kontrolünü arttırır. Bu nedenle, birden çok karar ağacına sahip rastgele orman bu noktada avantaj sağlayarak daha doğru sonuçlar üretir [50]. Hem sınıflandırma hem de regresyon problemlerine uyarlanabilir. Ayrıca kategorik ve sürekli değerler için uygundur. Rasgele orman algoritması, ölçeklenebilirliği, gürültüye karşı dayanıklılığı, yorumlaması ve görselleştirmesinin kolay olması nedeniyle avantajlar sağlar. Dezavantaj olarak birden fazla karar ağacına sahip olduğu için tahmin üretmede nispeten yavaş olarak değerlendirilebilir.

## 3.2. PLS-SEM Algoritması

Yapısal eşitlik modellemesi (YEM), varsayımsal ilişkileri test etmek ve faktörler arasındaki ilişkilerin gücünü değerlendirmek için kullanılan bir yöntemdir [51]. Bu yöntem çeşitli değişken ve yapı içeren bir model yoluyla yordayıcı değişkenlerin sonuç değişkeni üzerindeki etkilerini incelemelerini sağlar. Dolayısıyla bu yöntem çoklu değişkenler arasındaki karmaşık teorik ilişkileri test etmek için çok faydalı bir teknik olarak değerlendirilebilir. Temelde iki yapısal eşitlik modellemesi yöntemi önerilmektedir: kovaryans tabanlı yapısal eşitlik modellemesi (covariance-based structural equation modeling) (CB-SEM) ve PLS-SEM.

Kovaryans tabanlı yapısal eşitlik modellemesi sık kullanılan bir tekniktir. Bu yöntem çeşitli değişken ve yapı içeren bir model yoluyla yordayıcı değişkenlerin sonuç değişkeni üzerindeki etkilerini incelemelerini sağlar. Dolayısıyla bu yöntem çoklu değişkenler arasındaki karmaşık teorik ilişkileri test etmek için çok faydalı bir teknik olarak değerlendirilebilir.

Kovaryans tabanlı yapısal eşitlik modellemesi sık kullanılan bir teknik olsa da son dönemlerde PLS-SEM yönteminin popülerliği artmıştır [52]. Bu popülerlikte PLS-SEM'in veri dağılımına ilişkin daha az varsayımda bulunmasının ve normal olmayan verileri sağlam bir şekilde açıklamasının [53] rolü vardır. Ayrıca PLS-SEM, gizli yapılar ve göstergeler arasındaki basit ya da karmaşık modeller için uygundur ve küçük örneklem boyutlarında da çalışabilir [54]. CB-SEM'de ise, çok değişkenli normallik ihlalinin açıklamak önemlidir [55]. Bu yöntem, normallik, birçok katı varsayım ve büyük örneklem boyutları gerektirir.

PLS-SEM, temel bileşenler analizini PLS yol modellemesi olarak bilinen OLS regresyonları ile birleştirerek kısmi model yapıları ile tahminler [53]. Amaç, bir tahmin yaklaşımıyla teorik bir çerçeveyi test etmek, yerleşik teorileri ilerleterek karmaşık ilişkileri anlamaktır.

PLS algoritması, yol katsayılarını, yapıların öngörme doğruluğunu (gizli veya açık), model uyumunu ve açıklanan varyansı tahmin eder. Ayrıca, her yolun istatistiksel önemini belirlemek için bir bootstrapping prosedürü kullanır. Ek olarak, öngörme düzeyini tahmin etmek ve her yapının güvenilirliğini ve geçerliliğini değerlendirmek için blindfolding ve PLSpredict prosedürlerini kullanılır.

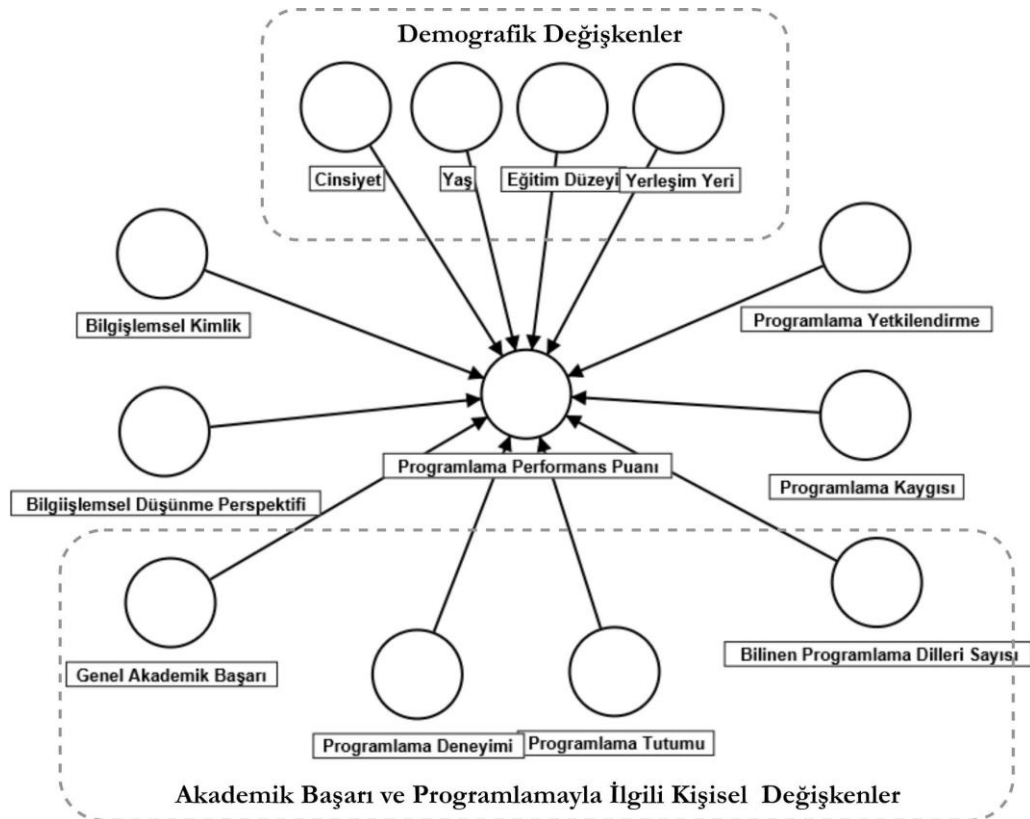
Bir PLS-SEM yaklaşımı, "a, b'yi artırır" / "daha yüksek bir a, daha yüksek bir b'ye yol açar" gibi normatif ifadeleri test eder ve nedensel ilişkiye yönelik bir sonuç için ilgili anahtar değişkenleri belirlemek olanaklı olmaktadır [51].

## Bölüm 4

# Materyal ve Yöntemler

### 4.1. Araştırma Modeli, Hipotezler ve Katılımcılar

Bu araştırmada, 12- 24 yaşları arasında programlama eğitimi alan öğrencilerin öz bildirimine dayalı veri toplama aracına verdiği yanıtların PLS-SEM ve makine öğrenmesi algoritmaları ile analiz edilmesine odaklanılmaktadır. Bu çalışmada, düşük ve yüksek programlama performansını etkileyen yapıları incelemek için bir model önerilmektedir. Şekil 1’de, önerilen araştırma modeli sunulmuştur.



Şekil 4.1: Araştırma modeli

Hipotezler:

H1a. Programlama performans puanı ile cinsiyet arasında anlamlı bir ilişki vardır.

H1b. Programlama performans puanı ile yaş arasında anlamlı bir ilişki vardır.

H1c. Programlama performans puanı ile eğitim düzeyi arasında anlamlı bir ilişki vardır.

H1d. Programlama performans puanı ile yaşanan yerleşim yeri arasında anlamlı bir ilişki vardır.

H2. Programlama performans puanı ile bilgi işlemsel kimlik arasında anlamlı bir ilişki vardır.

H3. Programlama performans puanı ile bilgi işlemsel düşünme perspektifi arasında anlamlı bir ilişki vardır.

H4. Programlama performans puanı ile genel akademik başarı puanı arasında anlamlı bir ilişki vardır.

H5. Programlama performans puanı ile programlama deneyim süresi arasında anlamlı bir ilişki vardır.

H6. Programlama performans puanı ile programlama tutumu arasında anlamlı bir ilişki vardır.

H7. Programlama performans puanı ile bilinen programlama dilleri sayısı arasında anlamlı bir ilişki vardır.

H8. Programlama performans puanı ile programlama kaygısı arasında anlamlı bir ilişki vardır.

H9. Programlama performans puanı ile programlama yetkilendirme arasında anlamlı bir ilişki vardır.

Bu çalışmanın katılımcılarını, lise ve üniversitede öğrenim gören ve en az bir bilgisayar programlama dersi almış olan öğrenciler oluşturmaktadır. Sosyal medya ve hazır mesaj sistemleri kullanılarak programlama dersi alan öğrencilere ulaşılmıştır.



Veriler 2023 yılı akademik yılı içerisinde, Google online formlar aracılığıyla toplanmıştır ve öz bildirimine dayalıdır. Katılımcılar, bir onam formu doldurarak, çalışmaya katılmaya gönüllü olduğunu bildirmiştir. Verilerin toplanmasında uygun örnekleme yaklaşımı kullanılmıştır. Bu yaklaşım, uygun özellikte amaca uygun kolay erişilebilir katılımcılara kısa sürede ulaştırırsa da, katılımcıların seçiminde öznellik ve veri yanlılığı ile ilgili bazı sorunları gündeme getirmektedir. Bu sınırlılığını aşmak üzere katılımcıların sayısının olabildiğince artırılması, verilerin anonim ve farklı bölgelerden toplanmasına dikkat edilmiştir. Dağıtılan 1000 anketten toplam 763 öğrenci (%76,3 dönüş oranıyla) anketinin tüm maddeleri eksiksiz doldurulmuştur.

Tablo 4.1: Katılımcı özellikleri

Özellikler	Frekans	Yüzde
<b>Cinsiyet</b>		
Kadın	462	60.6
Erkek	301	39.4
<b>Yaş</b>	$\bar{x} = 17.90, SS=3.67$	
<b>Yaşanılan yer</b>		
Kırsal	77	10.1
Kentsel	686	89.9
<b>Eğitim düzeyi</b>		
Lise	442	57.9
Üniversite	321	42.1
Genel akademik başarı ortalamaları	$\bar{x} = 80.67, SS=11.56$	
Programlama dersi başarı ortalamaları	$\bar{x} = 67.53, SS=19.89$	
Programlama tutumu	$\bar{x} = 2.96, SS=1.42$	
Programlama deneyimi	$\bar{x} = 1.18, SS=1.72$	
Bilinen programlama dilleri sayısı	Ranj 1-7 $\bar{x} = 1.1, SS=1.4$	

Katılımcıların %60.6'sı kadın, %39.4'ü erkektir. Katılımcıların yaş ortalamaları 17.90'dır. Ayrıca, katılımcıların %57.9'u lise, %42.1'i üniversite öğrencisidir. Tüm sınıf düzeylerinden benzer oranlarda katılım gerçekleşmiştir. Katılımcıların %10.1'i kırsal bölgelerde, %89.9'u il ya da ilçe merkezlerinde yaşamaktadır. Çalışma kapsamında, 64 ilden katılımcıya ulaşılmıştır.

Katılımcıların genel akademik başarı ortalamaları 100 üzerinden 80.67'dir. Programlama dersi başarı ortalamaları 100 üzerinden 67.53'tür. Programlamaya yönelik tutum puanları ise 5 üzerinden 2.96'dır (orta düzey). Bilinen programlama dili sayısı 1 ile 7 arasında değişmektedir. Ancak katılımcıların çoğunluğu 1 programlama dili bildiğini belirtmiştir. Katılımcıların programlama deneyimi ortalama 1.18 yıldır.

## 4.2. Veri Toplama Araçları

Bu tez çalışmasında kişisel bilgi formu ve dört ölçek kullanılarak veriler toplanmıştır. Veri toplama araçları programlama dersi alan öğrencilere online ortamda uygulanmıştır.

**Kişisel bilgi formu:** Bu formda 9 soru yer almaktadır. Bunlar cinsiyet, yaş, sınıf düzeyi, vb. ile ilgili bilgileri toplamak amacıyla oluşturulmuştur. Ayrıca akademik başarı, öğrencilerin genel akademik not ortalamaları alınarak, programlama dersi başarısı bilişim teknolojileri dersi akademik başarı notu alınarak oluşturulmuştur.

**Bilgi işlemsel kimlik ölçeği:** Bu ölçeğin orijinali Kong ve Wang [29] tarafından geliştirilmiş ve Türkçeye uyarlaması Yıldız-Durak [17] tarafından yapılmıştır. Bu ölçek 16 maddeden oluşmaktadır. Ölçek 5'li Likert tipindedir. Bu ölçekte faktörler programlama katılımı, programlama ilişkisi, programlama gerçekleştirme ve programlama hedef belirleme şeklindedir. Örnek maddeler şu şekildedir:

- Programlama görevlerinin içeriği, programlama öğrenmeye ilgi duymama sağlıyor.
- Programlamanın eğlenceli olduğunu düşünüyorum.
- Programlama görevlerine gerçekten ilgi duyuyorum.

**Programlama yetkilendirme ölçeği:** Bu ölçek Kong vd. [12] tarafından oluşturmuş ve Kong ve Lai [8] tarafından 3 yapı bağlamında tekrar incelenmiştir. Bu çalışmada Kong ve Lai [8] tarafından ele alınan 3 yapı (programlama yetkilendirmenin anlamlılığı, etkisi ve özyeterliliği) kullanılmıştır. Bu ölçeğin Türkçeye uyarlaması Yıldız-Durak [17] tarafından yapılmıştır. Ölçek 5’li Likert yapıdadır. Örnek maddeler şu şekildedir:

- Programlama işime yarar.
- Programlama benim için önemli.
- Programlama, hedeflerime ulaşmama yardımcı olacak.

**Bilgi işlemsel düşünme perspektifi ölçeği:** Bu ölçek, Brennan ve Resnick [56] tarafından önerilen bilişim teknolojileri çerçevesine dayalı olarak Kong ve Wang [29] tarafından geliştirilmiştir. Türkçeye uyarlaması Yıldız-Durak [17] tarafından yapılmıştır. Ölçek 5’li Likert tipindedir. Bu ölçekte alt boyutlar “bağlantı kurma becerisi, ifade etme becerisi ve sorgulama becerisi” şeklindedir. Örnek maddeler şu şekildedir:

- Programlama yoluyla yeni fikirleri ifade etmekten heyecan duyuyorum.
- Kendimi programlama yoluyla ifade edebildiğim için mutlu hissediyorum.
- Programlama tutkum var çünkü programlama ile yeni şeyler oluşturabiliyorum.

**Programlama kaygısı ölçeği:** Bu ölçek Yildirim ve Özdener [57] tarafından geliştirilmiştir. Ölçek 11 madde ve 5’li Likert tipindedir. Ölçekte akran kaygısı, programlama beceri kaygısı şeklinde iki boyut vardır. Örnek maddeler şu şekildedir:

- Daha önce programlama dersi alan arkadaşlarımın seviyesine yetişemeyeceğime inanmak beni kaygılandırır.
- Benim yazamadığım bir kodu çoğu sınıf arkadaşım yazabilmesi beni kaygılandırır.

- Program yazarken çözüm için gerekli olan basamakları (algoritmayı) doğru oluşturabileceğim konusunda şüphelerim var.

## 4.3. Veri Seti

### 4.3.1. Verilerin Toplanması

Veriler, geçerliği ve güvenilirliği sağlanmış ölçeklerdeki Likert tipi maddeler içeren elektronik bir form ile toplanmıştır. Veri toplama araçları için ölçek sahiplerinden ölçek kullanım izni alınmıştır. Veri toplama sürecinde Google online formlar aracılığıyla toplanan veriler elektronik tablolama programına aktarılmıştır. Veri toplama aracında, öğrencilerin isim, kimlik ya da okulları gibi kişisel ve özel bilgileri alınmamıştır. Veri toplama süreci öncesi Etik Kurul onay belgesi alınmıştır (İzmir Katip Çelebi Üniversitesi Fen Bilimleri Etik Kurulu toplantı no: 2023-06, karar no: 02 sayılı karar). Hazırlanan formun elektronik bağlantı adresi, sosyal medya ortamları ve hazır mesaj uygulamaları aracılığıyla paylaşılmıştır. Öğrencilerin araştırmaya tamamen gönüllülük esası ile katılması sağlanmıştır.

### 4.3.2. Veri Öznitelikleri ve Analizler

Veri seti öznitelikleri 11 olarak; örnek sayısı ise 763'tür. Veri setindeki öznitelikler Tablo 4.2'de sunulmuştur.

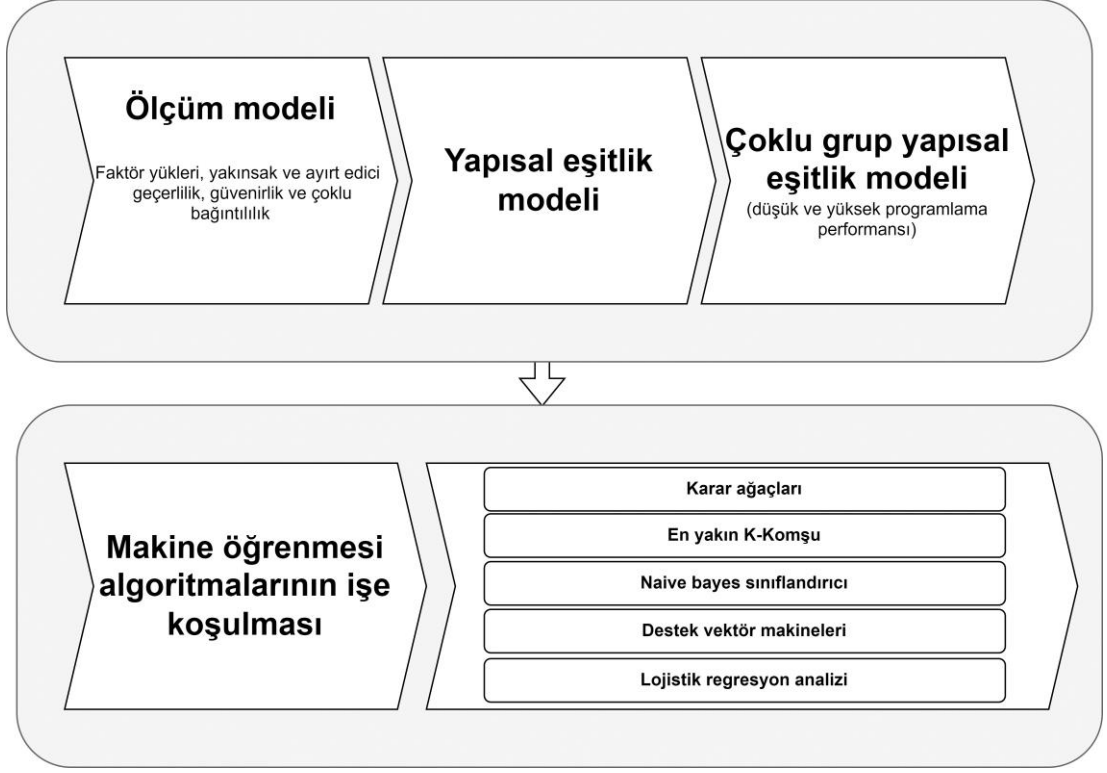
Tablo 4.2: Öznitelikler ve açıklamalar

<b>Öznitelikler</b>	<b>Açıklamalar</b>
yas	Öğrencinin yaşı
cinsiyet	Öğrencinin cinsiyeti
egitimduzey	Öğrencinin eğitim düzeyi
yasanilanyer	Öğrencinin yaşadığı yer
progdeneyim	Programlama deneyimi (yıl olarak)
progdilsay	Bilinen programlama dili sayısı
tutum	Programlamaya yönelik tutum
<b>BK</b>	<b>Bilgi işlemsel kimlik</b>
BKPE	Programlama meşguliyeti

BKPB	Programlama bağlantısı
BKPG	Programlama gerçekleştirme
BKPH	Programlama hedefi belirleme
<b>BD</b>	<b>Bilgi işlemsel düşünme perspektifi</b>
BDIY	Bağlanma yeteneği
BDB	İfade etme yeteneği
BDS	Sorgulama yeteneği
<b>PE</b>	<b>Programlama yetkilendirme</b>
PEA	Anlamlılık
PEE	etki
PEO	Öz yeterlilik
<b>K</b>	<b>Programlama kaygısı</b>
KCA	Akran kaygısı
KSC	Programlama beceri kaygısı

Bu çalışmada, varsayılan araştırma modelini test etmek için iki farklı teknik kullanılmaktadır. Katılımcılardan elde edilen veri tabanında, PLS-SEM ve makine öğrenmesi algoritmaları kullanılarak öğrencilerin programlama performansının tahminlenmesine çalışılmıştır. İlk teknik olan PLS-SEM yaklaşımı için, SmartPLS 4 yazılımı aracılığıyla PLS-SEM yapılmıştır. İkinci teknik, bu çalışma modelindeki bağımlı yapıları tahmin etmek için Knime 4.7 aracılığıyla makine öğrenmesi algoritmaları kullanılmıştır. Bu kapsamda, veri dosyası, PLS-SEM, naive bayes, destek vektör makineleri, karar ağacı, rastgele orman, yapay sinir ağı, lojistik regresyon sınıflandırıcı ile incelenmiştir.

Sınıflandırma algoritmaları kullanılmadan önce verilerin ön inceleme aşaması gerçekleştirilmiştir. Bu aşamada katılımcıların ölçeklerden aldıkları puanlar z puanına çevrilmiştir. Daha sonra cinsiyet, yaşanan yerleşim yeri ve eğitim durumu gibi değişkenler kategorik değişkenler olduğundan dolayı dummy değişkene dönüştürülmüştür. Karar ağaçları, kNN, destek vektör makineleri, naive bayes sınıflandırıcılar ve lojistik regresyon kullanılarak modeller oluşturulmuştur (Bkz. Şekil 4.2.).



Şekil 4.2: Veri analizi prosedürü

# Bölüm 5

## Bulgular

### 5.1. PLS-SEM

#### 5.1.1. Ölçüm Modelinin Değerlendirilmesi

Ölçme modelinde yakınsak ve ayırt edici geçerlilik (convergent ve discriminant validity) test edilmiştir. Ayrıca güvenilirlik ve çoklu bağıntılılık (collinearity) katsayılarının uygunluğu incelenmiştir. Elde edilen bulgular Tablo 5.1, Tablo 5.2 ve Tablo 5.3'te sunulmuştur.

Tablo 5.1: Faktör yükleri

<b>Yapılar</b>	<b>Boyutlar</b>	<b>Faktör yükleri</b>
Bilgi İşlemsel Düşünme Perspektifi	BDB	0.921
	BDIY	0.943
	BDS	0.964
Bilgi İşlemsel Kimlik	BKPB	0.890
	BKPE	0.925
	BKPG	0.905
	BKPH	0.910
Programlama Kaygısı	KCA	0.940
	KSC	0.940
Programlama Yetkilendirme	PEA	0.891
	PEE	0.907
	PEO	0.897

Tablo 5.1'e göre, yakınsama geçerliği için faktör yüklerinin 0.70'den büyük olması [58] olması gerekir. Tüm faktörlerin 0.70'den büyük olduğu tespit edilmiştir. Tablo 5.2'deki değerler incelendiğinde yapı güvenilirliği ve geçerliliği ile ilgili değerlerin literatürde beklenen eşik değerlerinin üstünde olduğu değerlendirilmiştir. Ortalama açıklanan varyans (OAV) değerlerinin 0.50'nin üzerinde olduğu, güvenilirlik katsayılarının için Cronbach alpha (CA) ve bileşik güvenilirlik (Composite reliability (CR)) değerlerinin 0.70'in üzerinde olduğu görülmektedir.



Tablo 5.2: Yapı güvenilirliği ve geçerliliği

		CA	CR (rho_a)	CR (rho_c)	OAV
Bilgi İşlemsel		0.940	1.028	0.960	0.889
Düşünme Perspektifi					
Bilgi İşlemsel		0.930	0.963	0.949	0.824
Kimlik					
Programlama Kaygısı		0.869	0.869	0.938	0.884
Programlama Yetkilendirme		0.884	0.927	0.926	0.807

Ölçme modelinin ayırt edici geçerliği için Fornell ve Larcker kriteri ve HTMT oranları kullanılmıştır. Fornell ve Larcker kriterleri, modeldeki yapılar arasındaki korelasyonlar ile AVE değerlerinin karekökünün ile karşılaştırılmasıdır. Tablo 5.3'e göre bu geçerliğin sağlandığı görülmektedir.

Tablo 5.3: Fornell-Larcker kriterleri

	1	2	3	4	5	6	7	8	9	10	11	12	13
1. Bilgi İşlemsel Düşünme Perspektifi	0.943												
2. Bilgi İşlemsel Kimlik	0.884	0.908											
3. Bilinen Programlama Dilleri Sayısı	0.348	0.371	1.000										
4. Cinsiyet	0.121	0.143	0.126	1.000									
5. Eğitim Düzeyi	0.217	0.197	0.184	-0.085	1.000								
6. Genel Akademik Başarı	0.082	0.093	0.081	0.000	-0.462	1.000							
7. Programlama Deneyimi	0.296	0.308	0.724	0.137	0.059	0.162	1.000						
8. Programlama Kaygısı	0.274	0.217	-0.061	-0.047	0.022	-0.021	-0.090	0.940					

9. Programlama Performans Puanı	0.194	0.192	0.406	0.087	0.088	0.190	0.483	-0.048	1.000				
10. Programlama Tutumu	0.585	0.609	0.416	0.205	0.197	0.072	0.385	0.012	0.370	1.000			
11. Programlama Yetkilendirme	0.894	0.861	0.347	0.121	0.227	0.035	0.303	0.229	0.165	0.569	0.898		
12. Yaş	0.155	0.142	0.079	-0.081	0.754	-0.432	-0.069	0.057	-0.122	0.075	0.152	1.000	
13. Yerleşim Yeri	0.046	0.080	0.089	0.083	-0.102	0.106	0.129	-0.068	0.019	0.065	0.027	-0.069	1.000

Tablo 5.4'e gre, lme modelinde HTMT deęerleri ayırt edici geerlik ile ilgili veriler sunulmaktadır. HTMT deęerinin 0.90'un altında olması nerilmek [59] ve bu alıřmadaki tm deęerler bu eřik deęerinin altındadır.

Tablo 5.4: Heterotrait-Monotrait Ratio (HTMT)

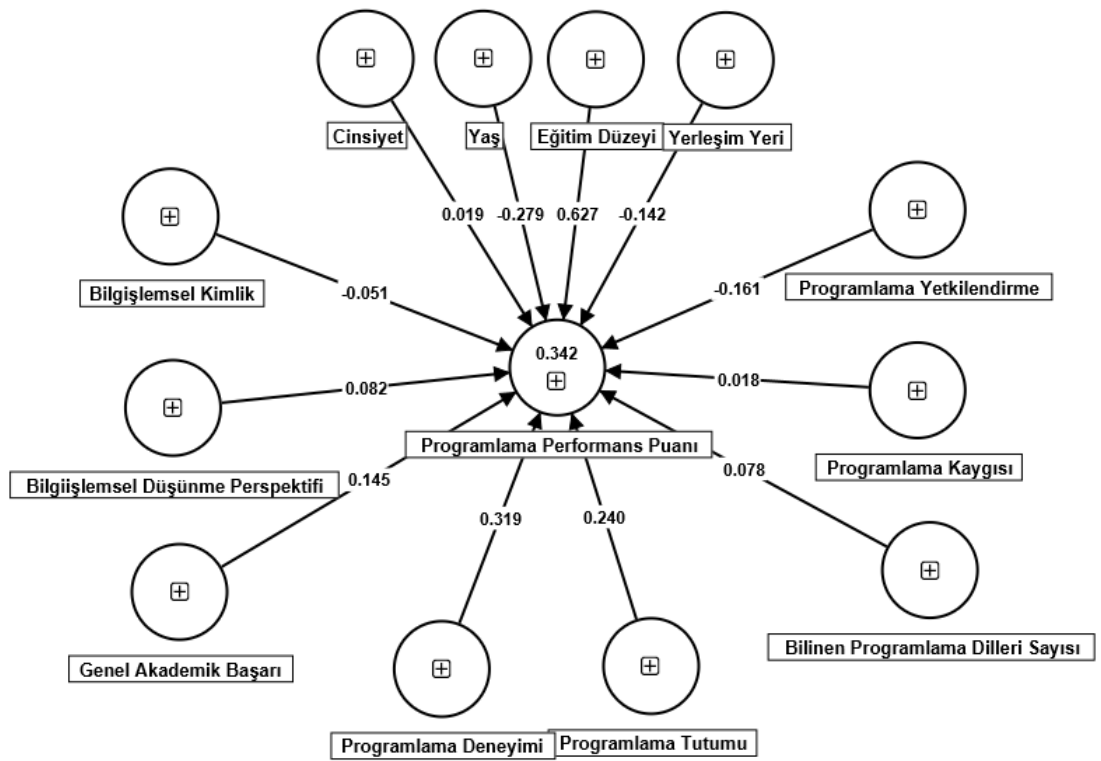
	1	2	3	4	5	6	7	8	9	10	11	12	13
1. Bilgi İşlemsel - Düşünme Perspektifi													
2. Bilgi İşlemsel Kimlik	0.951												
3. Bilinen Programlama Dilleri Sayısı	0.349	0.382											
4. Cinsiyet	0.118	0.145	0.126										
5. Eğitim Düzeyi	0.232	0.205	0.184	0.085									
6. Genel Akademik Başarı	0.070	0.090	0.081	0.000	0.462								

7. Programlama Deneyimi	0.295	0.313	0.724	0.137	0.059	0.162						
8. Programlama Kaygısı	0.309	0.250	0.065	0.050	0.023	0.023	0.097					
9. Programlama Performans Puanı	0.187	0.191	0.406	0.087	0.088	0.190	0.483	0.052				
10. Programlama Tutumu	0.601	0.631	0.416	0.205	0.197	0.072	0.385	0.055	0.370			
11. Programlama Yetkilendirme	0.90	0.958	0.363	0.116	0.246	0.032	0.309	0.273	0.169	0.607		
12. Yaş	0.168	0.151	0.079	0.081	0.754	0.432	0.069	0.062	0.122	0.075	0.168	
13. Yerleşim Yeri	0.043	0.080	0.089	0.083	0.102	0.106	0.129	0.073	0.019	0.065	0.029	0.069

Sonuç olarak, test edilen ölçme modelinin geçerlik ve güvenilirliğinin alanyazında önerilen eşik değerler bağlamında sağlandığı görülmüştür. Dolayısıyla sunulan kanıtlar ölçme modelinin geçerli olduğu göstermiştir.

### 5.1.2. Yapısal Eşitlik Modeli

Ölçme modelinin geçerliği sağlandıktan sonra, araştırmada önerilen yapısal model, PLS-SEM algoritması ile test edilmiştir. Bootstrapping “5000 subsample” olarak belirlenmiştir. Yapısal modele ilişkin bulgular Şekil 5.1’de gösterilmiştir.



Şekil 5.1: Yapısal Eşitlik Modeli

Tablo 5.5. Hipotez testi sonuçları

<b>Yol</b>	<b>B</b>	<b>Standart sapma</b>	<b>t</b>	<b>p</b>	<b>Kabul/ret</b>
H1a Cinsiyet → Programlama Performans Puanı	0.019	0.068	0.283	0.777	Ret
H1b Yaş → Programlama Performans Puanı	-0.279	0.058	4.824	0.000	Kabul
H1c Eğitim Düzeyi → Programlama Performans Puanı	0.627	0.110	5.713	0.000	Kabul
H1d Yerleşim Yeri → Programlama Performans Puanı	-0.142	0.091	1.554	0.120	Ret
H2 Bilgi İşlemsel Kimlik → Programlama Performans Puanı	-0.051	0.072	0.711	0.477	Ret
H3 Bilgi İşlemsel Düşünme Perspektifi → Programlama Performans Puanı	0.082	0.079	1.038	0.299	Ret
H4 Genel Akademik Başarı → Programlama Performans Puanı	0.145	0.036	4.059	0.000	Kabul
H5 Programlama Deneyimi → Programlama Performans Puanı	0.319	0.056	5.732	0.000	Kabul



---

H6	Programlama Tutumu → Programlama Performans Puanı	0.240	0.038	6.334	0.000	Kabul
H7	Bilinen Programlama Dilleri Sayısı → Programlama Performans Puanı	0.078	0.053	1.477	0.140	Ret
H8	Programlama Kaygısı → Programlama Performans Puanı	0.018	0.035	0.513	0.608	Ret
H9	Programlama Yetkilendirme → Programlama Performans Puanı	-0.161	0.074	2.194	0.028	Kabul

---

Şekil 5.1 ve Tablo 5.5’de görüldüğü üzere, programlama performans puanı ile cinsiyet ve yerleşim yeri arasında ilişki olduğuna dair hipotezler reddedilmiştir (H1a, H1d). Programlama performans puanı ile yaş ve eğitim düzeyi arasında ilişki olduğuna dair hipotezler desteklenmiştir (H1b, H1c).

Programlama performans puanı ile bilgi işlemsel kimlik ve bilgi işlemsel düşünme perspektifi arasında ilişki yoktur (H2, H3). Programlama performans puanı ile genel akademik başarı, programlama deneyimi ve programlama tutumu arasında anlamlı ilişki vardır (H4, H5, H6). Bilinen programlama dilleri sayısı ve programlama kaygısı, programlama performans puanını anlamlı şekilde etkilememiştir (H7, H8). Programlama yetkilendirme ile programlama performans puanı arasındaki yol katsayısı istatistiksel olarak anlamlı bulunmuştur (H9). Ek olarak açıklanan varyans yüzdeleri programlama performans puanı için %34.2 olarak hesaplanmıştır.

### 5.1.3. Çoklu Grup Analizi

Yapısal modeller ile ilgili bulgularda alt örneklem grupları (düşük programlama performansı, yüksek programlama performansı) arasında farklılıklarının istatistiksel olarak anlamlı olup olmadığını test etmek için çoklu grup analizi (multi group analysis) yapılmıştır. Çoklu grup analiz sonuçları Tablo 5.6’da sunulmuştur.

Tablo 5.6: Çoklu grup analizi sonuçları

	Düşük performans		programlama	Yüksek performans		programlama
	Yol katsayısı	t	P	Yol katsayısı	t	p
Cinsiyet → Programlama Performans Puanı	-0.045	0.370	0.711	-0.027	0.351	0.726
Yaş → Programlama Performans Puanı	0.055	0.651	0.515	-0.085	1.111	0.267
Eğitim Düzeyi → Programlama Performans Puanı	-0.688	3.276	0.001	-0.940	5.629	0.000
Yerleşim Yeri → Programlama Performans Puanı	-0.141	1.697	0.090	-0.244	1.743	0.081
Bilgi İşlemsel Kimlik → Programlama Performans Puanı	-0.145	0.797	0.425	0.091	0.937	0.349
Bilgi İşlemsel Düşünme Perspektifi → Programlama Performans Puanı	0.069	0.576	0.564	-0.068	0.688	0.492
Genel Akademik Başarı → Programlama Performans Puanı	0.066	1.026	0.305	0.173	3.058	0.002
Programlama Deneyimi → Programlama Performans Puanı	-0.074	1.340	0.180	0.009	0.171	0.864

---

Programlama Tutumu → Programlama Performans Puanı	-0.032	0.435	0.664	0.134	2.501	0.012
Bilinen Programlama Dilleri Sayısı → Programlama Performans Puanı	-0.008	0.147	0.883	-0.001	0.022	0.982
Programlama Kaygısı → Programlama Performans Puanı	-0.112	1.632	0.103	-0.092	2.218	0.027
Programlama Yetkilendirme → Programlama Performans Puanı	0.152	1.185	0.236	-0.053	0.639	0.523

---

Tablo 5.6 incelendiğinde, düşük ve yüksek programlama performansı gösteren alt örneklemi için programlama performansı puanı ve eğitim düzeyi arasındaki ilişkinin anlamlı olduğu bulunmuştur.

Genel akademik başarı, programlama tutumu ve programlama kaygısı, yüksek düzeyde performansı gösteren alt örneklem için programlama performansı puanı ile ilişkilirken, düşük düzeyde programlama performansı gösteren grupta anlamlı ilişki bulunamamıştır. Ek olarak düşük ve yüksek performans gösteren alt örneklem için açıklanan varyans yüzdeleri değişmektedir. Düşük alt grupta açıklanan varyanslar %10.7, yüksek grup için %48.6 olarak hesaplanmıştır.

## 5.2. Makine Öğrenmesi Algoritmaları

Modelin performansının belirlenmesi amacı ile doğruluk, hassasiyet, duyarlılık ve daha az veri ile doğruluk hesaplamalarında daha doğru sonuçlar elde edilmesini sağlayan çapraz doğruluk oranları hesaplanmıştır. Hesaplamalar çıktı değişkeni için oluşturulan modelin doğruluk sonuçlarının ortalaması şeklinde raporlanmıştır.

### 5.2.1. Modelin Doğruluğu (Accuracy)

Modelin doğruluğu (accuracy) için bulgular Tablo 5.7 ve Tablo 5.8’de sunulmuştur.

Tablo 5.7: Çıktı değişkenine göre algoritma doğruluk sonuçları

<b>Algoritma / Çıktı değişkenler</b>	<b>Düşük programlama performansı</b>	<b>Yüksek programlama performansı</b>
<b>Karar ağaçları</b>	0.966	0.966
<b>kNN</b>	0.764	0.764
<b>Destek vektör makineleri</b>	0.796	0.796
<b>Naive bayes</b>	0.690	0.690

<b>Lojistik regresyon</b>	0.790	0.790
<b>Rastgele orman</b>	0.790	0.790

Tablo 5.7 incelendiğinde düşük programlama performans durumlarına göre en yüksek doğruluk sonucuna sahip algoritmanın karar ağaçları (.966) algoritması olduğu görülmektedir. Yüksek programlama performans durumlarına göre hesaplanan doğruluk sonuçlarında yine karar ağaçlarının (.966) en doğru sonucu verdiği bulgusuna ulaşılmıştır.

Tablo 5.8: Çıktı değişkenlere göre algoritma çapraz doğruluk sonuçları

<b>Algoritma / Çıktı değişkenleri</b>	<b>Düşük programlama performansı</b>	<b>Yüksek programlama performansı</b>
<b>Karar ağaçları</b>	0.74	0.74
<b>kNN</b>	0.74	0.74
<b>Destek vektör makineleri</b>	0.790	0.790
<b>Naive bayes</b>	0.524	0.524
<b>Lojistik regresyon</b>	0.793	0.793
<b>Rastgele orman</b>	0.831	0.831

Tablo 5.8 incelendiğinde düşük programlama performans durumlarına göre en yüksek çapraz doğruluk sonucuna sahip algoritmanın rastgele orman (0.831) algoritması olduğu görülmektedir. Yüksek programlama performans durumlarına göre hesaplanan çapraz doğruluk sonuçlarında yine rastgele orman (.966) en doğru sonucu verdiği bulgusuna ulaşılmıştır.

## 5.2.2. Modelin Hassasiyeti (Precision)

Modelin hassasiyeti (precision) için bulgular Tablo 5.9’te sunulmuştur.

Tablo 5.9: Çıktı değişkenlere göre algoritma hassasiyet sonuçları

<b>Algoritma / Çıktı değişkenler</b>	<b>Düşük programlama performansı</b>	<b>Yüksek programlama performansı</b>
<b>Karar ağaçları</b>	0.957	0.975
<b>kNN</b>	0.748	0.779
<b>Destek vektör makineleri</b>	0.811	0.784
<b>Naive bayes</b>	0.771	0.617
<b>Lojistik regresyon</b>	0.789	0.791
<b>Rastgele orman</b>	0.786	0.794

Tablo 5.9 incelendiğinde düşük programlama performans durumlarına göre en yüksek hassasiyet sonucuna sahip algoritmanın karar ağaçları (0.957) algoritması olduğu görülmektedir. Yüksek programlama performans durumlarına göre yapılan hassasiyet hesaplamalarında en yüksek hassasiyete sahip algoritmanın yine karar ağaçları (0.975) olduğu sonucuna ulaşılmıştır.

### 5.2.3. Modelin Duyarlılığı (Recall)

Modelin duyarlılığı (recall) için bulgular Tablo 5.10'da sunulmuştur.

Tablo 5.10: Çıktı değişkenlere göre algoritma duyarlılık sonuçları

<b>Algoritma / Çıktı değişkenler</b>	<b>Düşük programlama performansı</b>	<b>Yüksek programlama performansı</b>
<b>Karar ağaçları</b>	0.972	0.960
<b>kNN</b>	0.748	0.779
<b>Destek vektör makineleri</b>	0.744	0.843
<b>Naive bayes</b>	0.646	0.747
<b>Lojistik regresyon</b>	0.763	0.815
<b>Rastgele orman</b>	0.757	0.82

Tablo 5.10 incelendiğinde düşük programlama performans durumlarına göre en yüksek duyarlılık sonucuna sahip algoritmanın karar ağaçları algoritması (0.972) olduğu görülmektedir. Ayrıca yüksek programlama performans durumlarına göre kullanılan algoritmanın duyarlılık sonuçları incelendiğinde en yüksek duyarlılığa sahip algoritmanın karar ağaçları (0.960) olduğu bulgusuna ulaşılmıştır.



## Bölüm 6

### Tartışmalar

Bu çalışmada, bilgisayar programlama eğitimi alan öğrencilerin PLS-SEM algoritması kullanarak programlama performansını etkileyen faktörlerin tahmin edilmesi ve düşük, yüksek programlama performansına göre etkilerinin farklılaşmalarının incelendiği dokuz hipotez test edilmiştir.

Çalışmanın bulgularına göre katılımcıların cinsiyetleri ve yaşadıkları yerleşim yerleri ile programlama performans puanları arasında anlamlı bir ilişki tespit edilememiştir (H1a, H1d). Bunu destekler biçimde çeşitli çalışmalarda cinsiyet ile programlama performansı arasında anlamlı fark bulunamamıştır [35]. Öte yandan literatürde cinsiyet ve yaşanan bölgenin kırsal ya da şehir olması ile ilgili literatürde çelişkili bulgular mevcuttur. Tsai vd.[60] tarafından yapılan çalışmada düşük ve orta deneyimli öğrenciler için algoritma ve hata ayıklama gibi programlama görevlerinde cinsiyet açısından farklılıkların bulunduğunu belirtmiştir. Atman Uslu [61] ise cinsiyet farklılığına değinirken, öğrencilerin bilgi işlemsel kimlik gibi profillerinin de dikkate alınması gerektiğini belirtir. Bu bağlamda cinsiyetin etkisinin programlama aktiviteleri özelinde ve öğrenci kimlikleri bağlamında derinlemesine incelenmesinde fayda olduğu söylenebilir.

Araştırmanın bulgularında göre yaş ile programlama performans puanları negatif yönlü anlamlı bir ilişki gösterirken (H1b), eğitim düzeyi ile pozitif bir ilişki (H1c) göstermiştir. Literatürde yaş ve eğitim düzeyi ile programla performansı ile ilişkili değişkenler açısından birbiri ile tutarlı olmayan sonuçlara rastlanmaktadır. Bu durumun sebepleri olarak örneklemelerin programlama deneyimi ve düzey olarak farklılıklar taşıması gösterilebilir. Yıldız Durak vd. [33]'e göre öğrencilerin aldıkları dersler bağlamında ve diğer gelişmeler ile edindikleri deneyimler, programlamaya

yönelik tutum ve programlama performanslarının sınıf ve yaş düzeyi ile pozitif/negatif yönlü bir ilişkiye neden olduğunu belirtmektedir.

Bilgi işlemsel kimlik ve bilgi işlemsel düşünme perspektifi ile programlama performans puanları arasındaki ilişkinin anlamlı olmadığı görülmüştür (H2, H3). Programlama ve bilgi işlemsel düşünme değişkenleri, bilgisayar bilimleri için temel kavramları kullanır. Bu bağlamda bir ilişki olmaması dikkat çekici ve literatürle örtüşmeyen bir bulgudur. Bu durumun sebebi öğrencilerin profil yapıları ve kimlik tiplerinin farklılaşması olabilir.

Genel akademik başarı puanı ile programlama performansı arasında pozitif yönlü anlamlı bir ilişki vardır (H4). Programlama, problem çözme, soyutlama, algoritmik düşünme, yaratıcı düşünme gibi becerileri geliştirmeyi içerir [62]. Dolayısıyla fen, matematik, sosyal bilimler gibi derslerin başarısından oluşan genel akademik başarının bu becerilere bağlı olduğu düşünüldüğü için akademik başarısı yüksek öğrencilerde programlama performans puanlarının yüksek olması beklenen bir sonuçtur. Özellikle bu çalışmada yüksek programlama performansına sahip öğrencilerin, programlama performansı ile genel akademik başarılarının pozitif ilişkili olması literatürde bahsedilen durumu desteklemektedir.

Programlama deneyimi ve tutumu ile programlama performansı arasında pozitif yönlü anlamlı bir ilişki vardır (H5, H6). Tsai vd. [60] tarafından yapılan çalışmada bilgisayar programlama deneyimi ile bilgisayar programlama öz yeterliliği arasındaki pozitif ilişki bulunmuştur. Aşkar ve Davenport [4] ve Yıldız Durak vd. [33] tarafından yapılan çalışmada programlama özyeterliliği, tutum, ilgi, programlamaya verilen önem bireyin bir görevi başarmak adına ayırdığı zamanı, strateji seçimini, görev planlama ve kontrol gibi çaba ve inanç seviyesini, zorluklarla başa çıkmadaki direncini ve nihai olarak performansını etkilediğini belirtir. Bu bağlamda, öğrencilerin programlama deneyimi ve tutumunun, programlama performansı ile ilişkili olması beklenen bir bulgudur. Ek olarak bilinen programlama dili sayısı ve programlama kaygısı ile programlama performansı arasında bir ilişki için ileri sürülen hipotez reddedilmiştir (H7, H8). Programlamada yetkilendirme ile programlama performansı arasında ise negatif yönlü anlamlı bir ilişki vardır (H9). Genel akademik başarı, programlama tutumu ve programlama kaygısı, yüksek düzeyde performans gösteren alt örneklem için programlama performansı puanı ile ilişkilirken, düşük düzeyde programlama

performansı gösteren grupta ilişki yoktur. Sonuç olarak düşük ve yüksek performans gösteren alt örneklemeler için modeller farklılaşmaktadır.

Düşük ve yüksek programlama performans durumlarına göre en yüksek doğruluk sonucuna sahip algoritmanın karar ağaçları [sırasıyla (.966), (.966)] algoritması olduğu sonucuna ulaşılmıştır. En yüksek çapraz doğruluk sonucuna sahip algoritmanın ise rastgele orman [sırasıyla (.831), (.831)] algoritması olduğu görülmektedir. Düşük programlama performans durumlarına göre en yüksek hassasiyet sonucuna sahip algoritmanın karar ağaçları (.957) algoritması olduğu görülmektedir. Yüksek programlama performans durumlarına göre yapılan hassasiyet hesaplamalarında en yüksek hassasiyete sahip algoritmanın yine karar ağaçları (.975) olduğu sonucuna ulaşılmıştır. Düşük ve yüksek programlama performans durumları ile ilgili veriler incelendiğinde karar ağaçları yönteminin en yüksek duyarlılık oranına sırasıyla (0.972), (0.960) ulaştığı belirlenmiştir. Bergin vd. [63] tarafından yapılan çalışmada programlama başarısını etkileyen faktörleri belirlemek için altı makine öğrenimi algoritmasının araştırılmıştır. Söz konusu çalışmada Naive Bayes'in en yüksek tahmin doğruluğuna sahip olduğu bulunmuştur.

## Bölüm 7

# Sonuç ve Öneriler

Bu tez çalışması teorik bağlamda literatüre çeşitli katkılar sağlamıştır. Bu çalışmanın literatüre en önemli katkısı düşük ve yüksek programlama performansı durumunda çeşitli değişkenlerin etkisini açıklayan bir model ortaya koymasındır.

Bununla birlikte yüksek düzeyde programlama performansı gösteren alt örnekleme modelin açıklama yüzdesi %50'ye yaklaşırken, düşük düzeyde programlama performansı gösteren alt örnekleme modelin açıklama yüzdesi %10'larda kalmıştır. Buna göre programlama eğitiminin desteklenmesinde düşük ve yüksek performans gösteren öğrencilere yönelik yaklaşımların farklılaştırılmasının önemli katkıları olabilir. Bu tür bir araştırma, programlama eğitiminde başarısızlık yaşayan öğrenciler için ilgili müdahalelerin tasarlanmasına yardımcı olabilir.

Düşük ve yüksek programlama performans durumlarına göre en yüksek doğruluk sonucuna sahip algoritmanın karar ağaçları algoritması olduğu sonucuna ulaşılmıştır. En yüksek çapraz doğruluk sonucuna sahip algoritmanın ise rastgele orman algoritması olduğu görülmektedir.

Mevcut sonuçlar, programlama performansını tahmin etme konusunda araştırmada kullanılan değişkenlerin yararlı olduğunu doğrulamaktadır. Bu, programlama performans tahminini öğrenmek ve başarıyı geliştirmek için iyi bir başlangıç noktası sağlar. Çevrimiçi programlama sistemlerinde öğrenci özelliklerinin, derin öğrenme kullanılarak tahmin edilmesi performans için faydalı bulunmuştur [64]. Bu bağlamda gelecekteki çalışmalar, gelişmiş derin öğrenme algoritmaları ile bu konuyu daha verimli bir şekilde keşfedebilir.

# Kaynaklar

- [1] Kong S. C. and Lin T., High achievers' attitudes, flow experience, programming intentions and perceived teacher support in primary school: A moderated mediation analysis, *Comput Educ* 2022, 190, doi: 10.1016/j.compedu.2022.104598.
- [2] Wang H.Y., Huang I., and Hwang G.-J., Comparison of the effects of project-based computer programming activities between mathematics-gifted students and average students, *Journal of Computers in Education* 2016, 3(1): 33–45, doi: 10.1007/s40692-015-0047-9.
- [3] Kafai Y. B. Burke Q. Computational Participation: Teaching Kids to Create and Connect Through Code, in *Emerging Research, Practice, and Policy on Computational Thinking*, Cham: Springer International Publishing 2017, 393–405. doi: 10.1007/978-3-319-52691-1\_24.
- [4] Askar P. Davenport D. An Investigation of Factors Related to Self-Efficacy for Java Programming Among Engineering Students 2009, *Online Submission*, 8(1).
- [5] Yildiz Durak H., Saritepeci M. Durak A. Modeling of Relationship of Personal and Affective Variables with Computational Thinking and Programming, *Technology, Knowledge and Learning* 2023, doi: 10.1007/s10758-021-09565-8.
- [6] Czuba C. E. *Empowerment: What Is It?* 1999.
- [7] Sprague J. Hayes J., Self-Determination and Empowerment: A Feminist Standpoint Analysis of Talk about Disability, *Am J Community Psychol* 2000, 28(5): 671–695, doi: 10.1023/A:1005197704441.

- [8] Kong S. Lai M. Computational identity and programming empowerment of students in computational thinking development, *British Journal of Educational Technology* 2022, 53(3): 668–686, doi: 10.1111/bjet.13175.
- [9] Neath J. Schriener K. Power to People with Disabilities: Empowerment issues in employment programming, *Disabil Soc* 1998, 13(2): 217–228, doi: 10.1080/09687599826795.
- [10] Thomas K. W. Velthouse B. A. Cognitive Elements of Empowerment: An ‘Interpretive’ Model of Intrinsic Task Motivation, *Academy of Management Review* 1990, 15(4): 666–681, doi: 10.5465/amr.1990.4310926.
- [11] Frymier A. B., Shulman G. M., Houser, M. The development of a learner empowerment measure, *Commun Educ* 1996, 45(3): 181–199, doi: 10.1080/03634529609379048.
- [12] Kong S.C., Chiu, M. M. Lai, M. A study of primary school students’ interest, collaboration attitude, and programming empowerment in computational thinking education, *Comput Educ* 2018, 127–189, doi: 10.1016/j.compedu.2018.08.026.
- [13] Zimmerman M. A., Psychological empowerment: Issues and illustrations,” *Am J Community Psychol* 1995, 23 (5): 581–599, doi: 10.1007/BF02506983.
- [14] Yildiz Durak H. Digital story design activities used for teaching programming effect on learning of programming concepts, programming self-efficacy, and participation and analysis of student experiences, *J Comput Assist Learn* 2018, 34 (6). doi: 10.1111/jcal.12281.
- [15] Yildiz Durak H. Flipped learning readiness in teaching programming in middle schools: Modelling its relation to various variables,” *J Comput Assist Learn* 2018, 34 (6): 939–959, doi: 10.1111/jcal.12302.
- [16] Yildiz Durak H., Saritepeci, M. Durak A. Modeling of Relationship of Personal and Affective Variables with Computational Thinking and Programming,” *Technology, Knowledge and Learning* 2023, doi: 10.1007/s10758-021-09565-8.

- [17] Yıldız Durak H. The model for middle school students' computational identity," *Educ Inf Technol* 2023, 1–29.
- [18] Çelik V. Yeşilyurt E. Attitudes to technology, perceived computer self-efficacy and computer anxiety as predictors of computer supported education, *Comput Educ* 2013, 60(1): 148–158, doi: 10.1016/j.compedu.2012.06.008.
- [19] Yalçınalp S. Cabı E. Eğitim Teknolojileri Kullanımı Kaygı Ölçeği (ETKKÖ): Geçerlik ve Güvenirlik Çalışması, *İlköğretim Online* 2015, 14(3), doi: 10.17051/io.2015.50515.
- [20] Yıldız Durak H. Flipped classroom model applications in computing courses: Peer-assisted groups, collaborative group and individual learning," *Computer Applications in Engineering Education* 2022, 30(3): 803–820. doi: 10.1002/cae.22487.
- [21] Jenkins T. On the difficulty of learning to program, in *Proceedings of the 3rd Annual Conference of the LTSN Centre for Information and Computer Sciences*, Citeseer, 2002, 53–58.
- [22] Mazzone L., Ducci F., Scoto M. C., Passaniti E., D'Arrigo V.G, Vitiello B. The role of anxiety symptoms in school performance in a community sample of children and adolescents, *BMC Public Health* 2007, 7(1): 347, doi: 10.1186/1471-2458-7-347.
- [23] Azarfam AAY. Jabbari Y. Dealing with teachers' technophobia in classroom," *Advances in Asian Social Science* 2012, 2(2): 452–455, 2012.
- [24] Yıldız Durak H. Modeling of variables related to problematic internet usage and problematic social media usage in adolescents, *Current Psychology* 2020, 39(4):1375–1387, doi: 10.1007/s12144-018-9840-8.
- [25] Peachey A. Childs M. Reinventing ourselves: Contemporary concepts of identity in virtual worlds. *Springer Science & Business Media* 2011.
- [26] Wenger E. *Communities of practice: Learning, meaning, and identity*. Cambridge university press, 1999.

- [27] Sfard A. Prusak A. Telling identities: In search of an analytic tool for investigating learning as a culturally shaped activity, *Educational researcher* 2005, 34(4):14–22.
- [28] Mahadeo J., Hazari Z., Potvin G. Developing a Computing Identity Framework, *ACM Transactions on Computing Education* 2020, 20(1):1–14, doi: 10.1145/3365571.
- [29] Kong S.C. Wang Y.Q. Formation of computational identity through computational thinking perspectives development in programming learning: A mediation analysis among primary school students, *Comput Human Behav* 2020, 106:106230, doi: 10.1016/j.chb.2019.106230.
- [30] Durak H. Y. Saritepeci M. Analysis of the relation between computational thinking skills and various variables with the structural equation model,” *Comput Educ* 2018, 116, doi: 10.1016/j.compedu.2017.09.004.
- [31] Korkmaz Ö., Çakir R., Özden M.Y. A validity and reliability study of the computational thinking scales (CTS), *Comput Human Behav* 2017, 72: 558–569, doi: 10.1016/j.chb.2017.01.005.
- [32] Wing J.M. Computational thinking and thinking about computing, *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences* 2008, 366 (1881): 3717–3725, doi: 10.1098/rsta.2008.0118.
- [33] Yildiz Durak H., Saritepeci M., Durak A. Modeling of Relationship of Personal and Affective Variables with Computational Thinking and Programming,” *Technology, Knowledge and Learning* 2023, 28 (1), doi: 10.1007/s10758-021-09565-8.
- [34] Hsu T.C., Chang S.C., Hung Y.T. How to learn and how to teach computational thinking: Suggestions based on a review of the literature, *Comput Educ* 2018, 126: 296–310, doi: 10.1016/j.compedu.2018.07.004.



- [35] Lau W. W. F. Yuen A. H. K. Modelling programming performance: Beyond the influence of learner characteristics, *Comput Educ* 2011, 57(1): 1202–1213, doi: 10.1016/j.compedu.2011.01.002.
- [36] Saritepeci M. Durak H. Analyzing the effect of block and robotic coding activities on computational thinking in programming education, *Educational research and practice* 2017, 490: 501, 2017.
- [37] Sivasakthi M. Padmanabhan K.R.A. Prediction of Students Programming Performance Using Naïve Bayesian and Decision Tree 2023, 97–106. doi: 10.1007/978-981-19-3590-9\_8.
- [38] Su Y.S., Lin Y.D., Liu T.Q. Applying machine learning technologies to explore students' learning features and performance prediction, *Front Neurosci* 2022, 16, doi: 10.3389/fnins.2022.1018005.
- [39] Anand V. K., Rahiman S. K. A., Ben George E., Huda A. S. Recursive clustering technique for students' performance evaluation in programming courses 2018, *Majan International Conference (MIC), IEEE*, 1–5. doi: 10.1109/MINTC.2018.8363153.
- [40] Marjan M. A., Uddin M. P., Ibn Afjal M. An Educational Data Mining System For Predicting And Enhancing Tertiary Students' Programming Skill, *Comput J* 2023, 66(5): 1083–1101, doi: 10.1093/comjnl/bxab214.
- [41] Bilegjargal D. Hsueh N.L. Understanding Students' Acceptance of Online Judge System in Programming Courses: A Structural Equation Modeling Approach, *IEEE Access* 2021, 9: 152606–152615, doi: 10.1109/ACCESS.2021.3126896.
- [42] Sivasakthi M. Classification and prediction based data mining algorithms to predict students' introductory programming performance, *International Conference on Inventive Computing and Informatics (ICICI), IEEE* 2017, 346–350. doi: 10.1109/ICICI.2017.8365371.
- [43] Elgamal A.F. An Educational Data Mining Model for Predicting Student Performance in Programming Course 2013.

- [44] Ray S. A quick review of machine learning algorithms, 2019 International conference on machine learning, big data, cloud and parallel computing (COMITCon), IEEE 2019, 35–39.
- [45] Pedregosa et al. F. Scikit-learn: Machine learning in Python,” the Journal of machine Learning research 2011, 12: 2825–2830.
- [46] Sarker I.H. Machine learning: Algorithms, real-world applications and research directions, SN Comput Sci 2021, 2(3): 160, 2021.
- [47] Bansal M., Goyal A., Choudhary A. A comparative analysis of K-nearest neighbor, genetic, support vector machine, decision tree, and long short term memory algorithms in machine learning, Decision Analytics Journal 2022, 3, 100071, 2022.
- [48] Kaparathi S. Bumblauskas D. Designing predictive maintenance systems using decision tree-based machine learning techniques,” International Journal of Quality & Reliability Management 2020, 37(4): 659–686, 2020.
- [49] Ray S. A quick review of machine learning algorithms, 2019 International conference on machine learning, big data, cloud and parallel computing (COMITCon), IEEE 2019, 35–39.
- [50] Sarker, I.H. Kayes A.S.M, Watters P. Effectiveness analysis of machine learning classification models for predicting personalized context-aware smartphone usage, J Big Data 2019, 6(1): 1–28, 2019.
- [51] Sukhov A., Friman M., Olsson L.E. Unlocking potential: An integrated approach using PLS-SEM, NCA, and fsQCA for informed decision making, Journal of Retailing and Consumer Services 2023, 74: 103424, Sep. 2023, doi: 10.1016/j.jretconser.2023.103424.
- [52] Hair J. F., Risher J.J., Sarstedt M., Ringle C.M. When to use and how to report the results of PLS-SEM,” European Business Review 2019, 31(1):2–24, doi: 10.1108/EBR-11-2018-0203.

- [53] Hair J.F., Ringle C. M. Sarstedt M. PLS-SEM: Indeed a Silver Bullet,” *Journal of Marketing Theory and Practice* 2011, 19(2): 139–152, doi: 10.2753/MTP1069-6679190202.
- [54] Hair, J. F. Risher J. J., Sarstedt M., Ringle C. M. When to use and how to report the results of PLS-SEM,” *European Business Review* 2019, 31(1): 2–24, doi: 10.1108/EBR-11-2018-0203.
- [55] Kline R. B., *Principles and practice of structural equation modeling*. Guilford publications, 2023.
- [56] Brennan K. Resnick M. *New frameworks for studying and assessing the development of computational thinking* 2012.
- [57] Yıldırım O.G. Özdener N. Development and Validation of the Programming Anxiety Scale, *International Journal of Computer Science Education in Schools* 2022, 5(3): 17–34, doi: 10.21585/ijcses.v5i3.140.
- [58] Hair, J. F. Hult G. T. M., Ringle C. M., Sarstedt M., Thiele K. O. Mirror, mirror on the wall: a comparative evaluation of composite-based structural equation modeling methods, *J Acad Mark Sci* 2017, 45: 616–632.
- [59] Henseler J., Ringle C. M., Sarstedt M. A new criterion for assessing discriminant validity in variance-based structural equation modeling, *J Acad Mark Sci* 2015, 43: 115–135.
- [60] Tsai M.J., Wang C.-Y., Hsu P.F. Developing the Computer Programming Self-Efficacy Scale for Computer Literacy Education, *Journal of Educational Computing Research* 2019, 56(8): 1345–1360, doi: 10.1177/0735633117746747.
- [61] Atman Uslu N. How do computational thinking self-efficacy and performance differ according to secondary school students’ profiles? The role of computational identity, academic resilience, and gender, *Educ Inf Technol* 2023, 28(5): 6115–6139.

- [62] Barr V. Stephenson C. Bringing computational thinking to K-12,” ACM Inroads 2011, 2(1): 48–54, doi: 10.1145/1929887.1929905.
- [63] Bergin S., Mooney A., Ghent J., Quille K. Using Machine Learning Techniques to Predict Introductory Programming Performance, 2015, Accessed: Jun. 16, 2023. [Online]. Available: <https://mural.maynoothuniversity.ie/8682/>
- [64] Yong, B. Jiang X., Lin J., Sun G., Zhou Q. Online practical deep learning education,” Educational Technology & Society 2022, 25(1): 193–204, 2022.

# Ekler

# Ek A

## Tezden Üretilmiş Yayınlar

### **Konferans Bildirileri**

#### 1. Bildiri 1

Durak, A., & Bulut, V. (2023, July). Programlama eğitiminde öğrenci performansının makine öğrenmesi algoritmaları ile tahminlenmesi. In International Conference on Applied Engineering and Natural Sciences (Vol. 1, No. 1, pp. 118-120).

### **Makaleler**

#### 1. Makale 1

Durak, A., & Bulut, V. (2023). Classification and prediction-based machine learning algorithms to predict students' low and high programming performance. Computer Applications in Engineering Education. <https://doi.org/10.1002/cae.22679>

Ek B

Dıř Cilt Kapađı

T.C.  
İzmir Kâtip Çelebi Üniversitesi  
Fen Bilimleri Enstitüsü

Programlama Eğitiminde Öğrenci  
Performansının, PLS-SEM ve Makine  
Öğrenmesi Algoritmalarını Kullanan  
Karşılaştırmalı Bir Yaklaşımla  
Tahminlenmesi

Yazılım Mühendisliği Ana Bilim Dalı  
Yüksek Lisans

Aykut DRAK

ORCID 0000-0001-7070-9048

Tez Danışmanı: Doç. Dr. Vahide BULUT

Kasım 2023



**AYKUT DURAK Programlama Eğitiminde Öğrenci Performansının, PLS-SEM ve Makine  
Öğrenmesi Algoritmalarını Kullanan Karşılaştırmalı Bir Yaklaşımla Tahminlenmesi**  
**YÜKSEK LİSANS TEZİ 2023**

# Özgeçmiş

Adı Soyadı: Aykut DURAK

## Eğitim:

2005–2009 Hacettepe Üniversitesi, Eğitim Fakültesi, Bilgisayar ve Öğretim Teknolojileri Eğitimi Bölümü- Lisans

2017–2019 Bartın Üniversitesi Eğitim Bilimleri Enstitüsü, Yaşam Boyu Öğrenme Ana Bilim Dalı Tezli Yüksek Lisans

## İş Deneyimi:

2022 – .. Öğretmen-Fatih Ortaokulu (KONYA/EREĞLİ)

2021 – 2022 Öğretmen-Bartın Bilim ve Sanat Merkezi (BARTIN/MERKEZ)

2020 – 2021 Öğretmen-Fatih Ortaokulu (BARTIN/MERKEZ)

2017 – 2020 Müdür Yardımcısı-Kıranpazarı Ortaokulu (BARTIN/MERKEZ)

2017 – 2017 Bilişim Teknolojileri Rehber Öğretmenliği - Kıranpazarı Ortaokulu (BARTIN/MERKEZ)

2016 – 2017 Öğretmen-Kıranpazarı Ortaokulu (BARTIN/MERKEZ)

2014 – 2016 Öğretmen-İlçe Milli Eğitim Müdürlüğü (ANKARA/ALTINDAĞ)

2014 – 2014 Öğretmen-Meşeiçi Şehit Bülent Deveci Ortaokulu (ŞIRNAK/MERKEZ)

2013 – 2014 Müdür Vekili -Kızılsu Suluklu Ortaokulu (ŞIRNAK/MERKEZ)

2012 – 2013 Müdür Yardımcısı-Kızılsu Suluklu Ortaokulu (ŞIRNAK/MERKEZ)

2011 – 2012 Öğretmen-Kızılsu Suluklu İlköğretim Okulu (ŞIRNAK/MERKEZ)

## Yayımlar (varsa):

Uluslararası Makaleler (SSCI/SCI-Exp./ESCI/ SCOPUS kapsamında taranan dergilerde)

1. **[SCI-Exp.] Durak, A. & Bulut, V. (2023).** Classification and prediction-based machine learning algorithms to predict students' low and high programming

performance. *Computer Applications in Engineering Education*.  
<https://doi.org/10.1002/cae.22679>

2. [SSCI] **Durak, A.**, & Kaygin, H. (2020). Parental mediation of young children's internet use: Adaptation of parental mediation scale and review of parental mediation based on the demographic variables and digital data security awareness. *Education and Information Technologies*, 25, 2275–2296.
3. [ERIC/ ESCI] Yildiz Durak, H., Saritepeci, M., & **Durak, A.** (2023). Modeling of Relationship of Personal and Affective Variables with Computational Thinking and Programming. *Technology, Knowledge and Learning*. doi: <https://doi.org/10.1007/s10758-021-09565-8>.
4. [SCOPUS] Yildiz Durak, H., Şimşir Gökalp, Z., Saritepeci, M., Dilmaç, B. & **Durak, A.** (2023). Investigation of personal, technology usage status, vaccine-related, social media-specific epistemological beliefs, media literacy, social impact strategies variables affecting vaccine hesitancy beliefs in the Covid-19 pandemic. *Journal of Public Health*. doi: <https://doi.org/10.1007/s10389-023-01872-x>
5. [ULUSLARARASI DİĞER] Saritepeci, M., & **Durak, A.** (2022). Adaptation of T-STEM CT Scale to Turkish: Teacher self-efficacy and outcome expectancy for teaching computational thinking. *Research on Education and Psychology (REP)*, 6(Special Issue), 47-56.
6. [ULUSLARARASI DİĞER] Yıldız Durak, H., & **Durak, A.** (2020). Öğretim teknolojisi ve hayat boyu öğrenme: hayat boyu öğrenme bağlamında teknoloji kullanımını ele alan tezlerde ortaya çıkan eğilimler, fırsatlar ve zorluklar. *Hayatboyu Öğrenme ve Öğretim Teknolojileri Dergisi*, 1(1), 88 – 106.

### **Uluslararası Kitap Bölümü**

1. Yıldız Durak, H., Saritepeci, M., Topçu, A. & **Durak, A.** (2020). Investigation of Variables Related to Computational Thinking Self-Efficacy Level in Middle School Students: Are Demographic Variables, Academic Success or Programming Related Variables More Important?. In M. Kalogiannakis and S. Papadakis (Eds.) *Tools for Teaching Computational Thinking in P-12 Education* (Chapter , pp. 61-99). IGI Global.

2. Saritepeci, M., Yıldız Durak, H. & **Durak, A.** (2021). Hayat Boyu Öğrenme Bağlamında Endüstri 4.0. H. Yıldız Durak ve M. Saritepeci (Ed.) içinde Dijital Çağda Hayat Boyu Öğrenme, (10. Bölüm). Pegem Yayıncılık.

#### **Ulusal Makaleler (ULAKBİM TR DİZİN kapsamında taranan dergilerde)**

1. **Durak, A.** & Karaoğlan Yılmaz, F. G. (2019). Artırılmış Gerçekliğin Eğitsel Uygulamaları Üzerine Ortaokul Öğrencilerinin Görüşleri. Abant İzzet Baysal Üniversitesi Eğitim Fakültesi Dergisi.
2. Erçetin, Ş. Ş., & **Durak, A.** (2017). Ortaokullarda bilişim teknolojileri ve yazılım dersinin işlenişi, yaşanan problemler ve çözüm önerileri: Öğretmen görüşleri. Bartın Üniversitesi Eğitim Fakültesi Dergisi, 6(1), 159-176.
3. **Durak, A.**, & Kaygın, H. (2018). Teknoloji Kullanımının Yansımaları: Teknoloji Kullanımıyla Kaybolan Değerler Hakkında Ebeveyn Görüşleri. Bartın Üniversitesi Eğitim Fakültesi Dergisi, 7(3), 1035-1053.

#### **Uluslararası kongre bildirisi**

1. **Durak, A.**, & Bulut, V. (2023, July). Programlama eğitiminde öğrenci performansının makine öğrenmesi algoritmaları ile tahminlenmesi. In International Conference on Applied Engineering and Natural Sciences (Vol. 1, No. 1, pp. 118-120).
2. Saritepeci, M., & Yıldız Durak, H., **Durak, A.** (2019). Öğretmen Adaylarının Hayat Boyu Öğrenme Eğilimleri ile Çevrim-içi Bilgi Arama Stratejileri Arasındaki İlişkinin İncelenmesi. 7. Uluslararası Öğretim Teknolojileri ve Öğretmen Eğitimi Sempozyumu, 30 Ekim-1 Kasım 2019, Antalya, Türkiye.
3. **Durak, A.**, & Kaygın, H. (2018). Yetişkin Eğitimde Yeni Teknolojilerin Kullanımı. 12. Uluslararası Bilgisayar ve Öğretim Teknolojileri Eğitimi Sempozyumu, 2-4 Mayıs 2018, Ege Üniversitesi, İzmir.
4. **Durak, A.**, & Karaoğlan Yılmaz, F.G. (2017). Artırılmış gerçekliğin eğitsel uygulamaları üzerine ortaokul öğrencilerinin görüşlerinin incelenmesi. 5th International Instructional Technologies and Teacher Education Symposium (5. Uluslararası Öğretim Teknolojileri ve Öğretmen Eğitimi Sempozyumu - ITTES, 2017), 11-13.10.2017, Anadolu Üniversitesi, İzmir.

5. Yildiz-Durak, H., Sarıtepeci, M. & **Durak, A.** (2016). Farklı Branşlardaki Öğretmenlerin Eğitimde Teknoloji Kullanım Durumlarıyla İlgili Görüşleri. World Congress on Lifelong Education, WCLE- 2016, 16-17 December 2016 , Porto Bello Hotel Resort& SPA, Antalya.
6. Yildiz-Durak, H., Sarıtepeci, M. & **Durak, A.** (2016). Medya okur-yazarlığı dersinde medya mesajlarının öğretmen adayları tarafından anlamlandırma sürecinin yansıtılması. World Congress on Lifelong Education, WCLE- 2016, 16-17 December 2016 , Porto Bello Hotel Resort& SPA, Antalya.
7. Yildiz-Durak, H., Sarıtepeci M. & **Durak, A.** (2017). Öğretmen Adaylarının Eğitim Teknolojileri Öz-Yeterliliklerinin ISTE Standartlarına Göre Çeşitli Değişkenler Bakımından İncelenmesi. Uluslararası Bilgisayar ve Öğretim Teknolojileri Sempozyumu (ICITS-2017), 24-26 Mayıs 2017, İnönü Üniversitesi, Malatya.
8. Sarıtepeci M., Yildiz-Durak, H. & **Durak, A.** (2017). Öğretmen Adaylarının Problemlerini İnternet Kullanımlarının Sosyal Medya Kullanım Alışkanlıkları Çerçevesinde İncelenmesi. Uluslararası Bilgisayar ve Öğretim Teknolojileri Sempozyumu (ICITS-2017), 24-26 Mayıs 2017, İnönü Üniversitesi, Malatya.