TÜRKİYE CUMHURİYETİ
YILDIZ TEKNİK ÜNİVERSİTESİ
BİLGİSAYAR MÜHENDİSLİĞİ BÖLÜMÜ

ESKİŞEHİR KİRALIK KONUTLAR VERİ SETİ VE MAKİNE ÖĞRENMESİ UYGULAMALARI

16011059 — Ahmet Onur AKMAN

BLM4510 - YAPAY ZEKA DERSİ İKİNCİ DÖNEM ÖDEVİ

Doç. Dr. Mehmet Fatih Amasyalı

Mayıs, 2021

İÇİNDEKİLER

ŞEKİL LİSTESİ **iii**

1 Giriş **1**
1.1 Problem 1
1.2 Programlama Ortamı 1

2 Veri Seti **3**
2.1 Veri Setinin Özellikleri 3
2.1.1 Öznitelikler 3
2.2 Veri Seti Değerlendirmesi 5

3 Yaklaşım **6**
3.1 Algoritmalar 6
3.2 Çapraz Geçerleme 6
3.3 Data Bucketing 7
3.4 Özellik Seçimi 7
3.5 Özellik Dönüşümü 8
3.6 Başarı Tespiti 8
3.7 Kütüphaneler 8

4 Program Çıktıları **9**
4.1 Özellik Seçimi 9
4.1.1 Tüm Özniteliklerin Kullanılması 9
4.1.2 4 Özellik Seçimi ve Çapraz Geçerleme 11
4.1.3 2 Özellik Seçimi ve Çapraz Geçerleme 12
4.1.4 Tek Özellik Seçimi ve Çapraz Geçerleme 14
4.2 Özellik Dönüşümü 15
4.2.1 PCA İle 4 Özelliğe Dönüşüm 16
4.2.2 PCA İle 2 Özelliğe Dönüşüm 17

5 Sonuç **20**
5.1 Gözlem 20
5.1.1 Decision Trees 20

5.1.2 K-Nearest Neighbor Algoritması 20
5.1.3 Lineer Regresyon 21
5.1.4 Feature Selection'un Genel Etkileri 21
5.1.5 Feature Extraction'un Genel Etkileri 22
5.2 Çıkarım 22

ŞEKİL LİSTESİ

Şekil 1.1 Spyder, the Scientific Python Development Environment. 2
Şekil 3.1 10 Katlı Çapraz Geçerleme Uygulamasının Şematik Gösterimi . . 7
Şekil 4.1 Feature Selection Olmadan 10 Katlı Çapraz Geçerlemede Elde Edilmiş En İyi Modeller 10
Şekil 4.2 Feature Selection Olmadan 10 Katlı Çapraz Geçerlemede Elde Edilmiş En İyi Fold 10
Şekil 4.3 Feature Selection Olmadan 10 Katlı Çapraz Geçerlemede Elde Edilmiş En İyi Tahminleyici 11
Şekil 4.4 4 Features Selection İle 10 Katlı Çapraz Geçerlemede Elde Edilmiş En İyi Modeller 11
Şekil 4.5 4 Features Selection İle 10 Katlı Çapraz Geçerlemede Elde Edilmiş En İyi Fold 12
Şekil 4.6 4 Features Selection İle 10 Katlı Çapraz Geçerlemede Elde Edilmiş En İyi Tahminleyici 12
Şekil 4.7 2 Features Selection İle 10 Katlı Çapraz Geçerlemede Elde Edilmiş En İyi Modeller 13
Şekil 4.8 2 Features Selection İle 10 Katlı Çapraz Geçerlemede Elde Edilmiş En İyi Fold 13
Şekil 4.9 2 Features Selection İle 10 Katlı Çapraz Geçerlemede Elde Edilmiş En İyi Tahminleyici 14
Şekil 4.10 Yalnızca "size" Özniteliği İle 10 Katlı Çapraz Geçerlemede Elde Edilmiş En İyi Modeller 14
Şekil 4.11 Yalnızca "size" Özniteliği İle 10 Katlı Çapraz Geçerlemede Elde Edilmiş En İyi Fold 15
Şekil 4.12 Yalnızca "size" Özniteliği İle 10 Katlı Çapraz Geçerlemede Elde Edilmiş En İyi Tahminleyici 15
Şekil 4.13 PCA 4 Özelliğe Dönüşüm İle 10 Katlı Çapraz Geçerlemede Elde Edilmiş En İyi Modeller 16
Şekil 4.14 PCA 4 Özelliğe Dönüşüm İle 10 Katlı Çapraz Geçerlemede Elde Edilmiş En İyi Fold 17

Şekil 4.15 PCA 4 Özelliğe Dönüşüm İle 10 Katlı Çapraz Geçerlemede Elde Edilmiş En İyi Tahminleyici 17
Şekil 4.16 PCA 2 Özelliğe Dönüşüm İle 10 Katlı Çapraz Geçerlemede Elde Edilmiş En İyi Modeller 18
Şekil 4.17 PCA 2 Özelliğe Dönüşüm İle 10 Katlı Çapraz Geçerlemede Elde Edilmiş En İyi Fold 18
Şekil 4.18 PCA 2 Özelliğe Dönüşüm İle 10 Katlı Çapraz Geçerlemede Elde Edilmiş En İyi Tahminleyici 19

1
Giriş

Bu bölümde, proje kapsamında ele alınan problem ve bu problemin yapı taşları ele alınacaktır.

1.1 Problem

Bu çalışma, yapay zeka ve veri analizi uygulamaları başta olmak üzere birçok uygulamanın belki de en kritik ve en belirleyici adımı olan sıfırdan veri seti oluşturma deneyimi kazandırma ve oluşturulan bu veri setininin üzerinde farklı makine öğrenmesi algoritmalarının performans analizini yaptırma amacı taşımaktadır. Çalışmanın kapsam açıklamasında, sıfırdan bireysel olarak en az 200 örnekli ve en az 10 öznitelikli bir veri setini oluşturulması, daha sonra bu veri seti üzerinde en az 5 farklı tahminleyiciyi değişik yaklaşımlar altında çalıştırarak çıktıların gözlemlenmesi istenmiştir.

1.2 Programlama Ortamı

Bu proje kapsamında, hem şahsi aşinalığım, hem daha önce yaptığım benzer çalışmalardaki konfor ve başarı oranları, hem kütüphane desteğinin zengin olması, hem de veri görselleştirme konusunda yeterli desteğin sağlanması sebebiyle programlama dili Python, IDE olarak ise bu konuda önde gelen seçimlerden biri olan Anaconda Spyder kullanılmıştır.

Bunun yanı sıra veri setinin oluşturulması Microsoft Excel üzerinden gerçekleştirilmiştir.

Şekil 1.1 Spyder, the Scientific Python Development Environment.

2
Veri Seti

Bu bölümde çalışma kapsamında oluşturulan veri setinin detaylarından bahsedilecektir.

2.1 Veri Setinin Özellikleri

Bu çalışma için hazırlanacak veri setinin içeriği, şahsen yaşadığım şehir olan Eskişehir'in merkezi içerisinde HepsiEmlak.com üzerinden kiralık olarak yayınlanmış apartman daireleri olarak seçilmiştir. İlanların tümü HepsiEmlak.com'dan elle alınmıştır ve tamamiyle halka açık verilerin toplanmasıyla veri seti oluşturulmuştur. Veri seti .csv formatında olup toplamda 211 örnek ve bu örneklerin her birinde eksiksiz doldurulmuş olan 12 öznitelikten oluşmaktadır.

2.1.1 Öznitelikler

Toplanacak özniteliklerin seçimi sürecinde seçilecek özelliklerin uygulanacak algoritmalar için kullanışlı olması ve HepsiEmlak.com'dan kolayca toplanabilecek bilgiler olması kriterler olarak belirlenmiştir. Bu iki isteği karşılayan 12 öznitelik 211 örneğin tümü için toplanmıştır. Bu öznitelikler aşağdaki gibidir.

- province: Eskişehir merkez bölgesindeki iki ilçe, Odunpazarı ve Tepebaşı.
- street: Mahalle.
- size: Dairenin metrekare cinsinden büyüklüğü.
- total_room: Dairedeki oda sayısı.
- total_floor: Apartmandaki toplam kat sayısı.
- floor: Dairenin apartmanında kaçıncı katta olduğu.
- age: Binanın yaşı.

- heating: Dairenin ısıtma şekli. Kombi veya merkezi ısıtma.
- facing: Cephe. Kuzeyden batıya doğru saat yönünde 1-4 arası rakamlarla kodlanmıştır ve cephelerin avantajlılık sırası 3, 2, 4, 1 olarak belirlenmiştir. Çoklu cepheli daireler için en avantajlı cephe yazılmıştır. (Ör. Güney ve batı çift cepheli bir daire için 3, yani güney yazılmıştır.)
- balcony: Dairede balkon veya teras olup olmama durumu. 1 ve 0 ile kodlanmıştır. Fransız tipi balkonlar balkon olarak değerlendirilmemiştir.
- furniture: Dairenin eşyalı veya eşyasız olarak kiraya verildiği. 1 ve 0 ile kodlanmıştır. Yarı veya tam eşyalı daireler bu alanda eşdeğer sayılmıştır.
- rent: Aylık kira ücreti. Program içerisinde target olarak kullanılmıştır.

Özniteliklerdeki çeşitlilik aşağıdaki gibi verilmiştir.

- province: 2 çeşit.
- street: 37 çeşit.
- size: 33 çeşit.
- total_room: 5 çeşit.
- total_floor: 7 çeşit.
- floor: 9 çeşit.
- age: 25 çeşit.
- heating: 2 çeşit.
- facing: 4 çeşit.
- balcony: 2 çeşit.
- furniture: 2 çeşit.
- rent: 35 çeşit.

2.2 Veri Seti Değerlendirmesi

Veri seti bu haliyle çalışmanın kapsam yönergelerini karşılamaktadır. Bunun yanında her bir örnek için bütün öznitelikler tanımlanmıştır ve eksik veri yoktur. Bütün ilanlar tek bir oturumda toplanmıştır, bu sebeple birkaç kez paylaşılmış aynı daire ilanları dışında tekrarı yoktur, bahsedilen tekrar senaryosu da nadirdir ve olmaması için dikkat edilmiştir.

Bunun yanı sıra, hem Eskişehir'deki yapılaşma, hem de pandemi dolayısıyla değişen şehir nüfus yapıları bu veri seti kapsamında da çok net gözlemlenmektedir. Eskişehir gibi orta büyüklükte olup 3 tane gözde üniversiteye ev sahipliği yapan, bu durumun getirdiği nüfus yapısı sebebiyle "öğrenci şehri" olarak anılan bir şehir de bu pandemiden en çok etkilenen şehirlerden biri olmuştur. Bu etkiyle beraber, veri setindeki 1 oda ve 1 salondan veya yalnızca 1 odadan oluşan dairelerin fazlalığı dikkat çekmektedir. Öyle ki salon ve oda sayısının toplamı 2 veya 1 olan daireler, veri setinin %65'ini oluşturmuştur. Bu durum aynı zamanda eşyalı dairelerin %35 civarında olmasından ve üniversitelere yakın birçok boş kalmış kiralık evlerin bulunmasından da gözlemlenebilmektedir.

Eskişehir merkez bölgesinin iki ilçesi olan Tepebaşı ve Odunpazarı'nın sakinlerinin genel profili ve bu bölgelerdeki yapıların eskilikleri de veri setine direkt olarak yansımıştır. Daire sakinlerinin o dairenin tapusuna da sahip olması durumunun çok daha yaygın olduğu Odunpazarı ilçesindeki dairelerin sayıca çok daha az, genellikle 20 yaşın üzerinde, metrekare olarak daha büyük, genellikle balkonlu ve daha yüksek kira bedellerine sahip olduğu gözlemlenmektedir. Bunun tersine veri setinin %75'ini oluşturan Tepebaşı ilçesinin konutları ise çok daha yeni ancak maliyet olarak nispeten daha ucuz evleri barındırdığı görülmektedir. Ayrıca yeni imar kurallarıyla gelen yapılarda balkon kısıtlamaları Tepebaşı ilçesindeki konutlarda net olarak gözlemlenebilmektedir.

Bu ve bunun gibi faktörler dolayısıyla, her ne kadar dengeli bir veri seti oluşturma gayreti içine girildiyse de, veri setinde kaçınılmaz dengesizlikler olduğu söylenebilir. Bu dengesizliklerin yanı sıra, sadece 211 adet örnek içermesi sebebiyle bu dengesizliği bazı örnekleri çıkararak ortadan kaldırmak imkansız olacaktır.

3
Yaklaşım

Bu bölümde çalışma boyunca temel olarak izlenen yöntemlerden ve bu yöntemlerin seçimindeki motivasyonlardan bahsedilecektir.

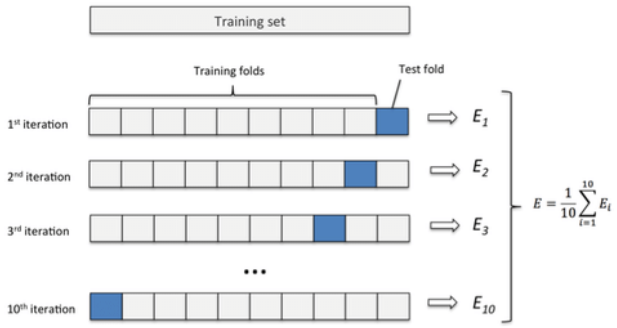
3.1 Algoritmalar

Çalışma sürecinde aynı veri seti üzerine performansı incelenen makine öğrenmesi algoritmaları aşağıda verilmiştir:

- Decision Tree, Gini Kriteriyle
- Decision Tree, Entropi Kriteriyle
- KNN Sınıflayıcı, Distance Ağırlıklandırmasıyla
- KNN Sınıflayıcı, Uniform Ağırlıklandırmasıyla
- Lineer Regresyon

3.2 Çapraz Geçerleme

Çalışma kapsamında da talep edildiği üzere, programda 10 katlı çapraz geçerleme kullanılmıştır. Basit bir şekilde açıklamak gerekirse, bu yaklaşımla beraber, veri setinin 10 parçaya bölünmesinin ardından bu 10 parçanın sırayla her biri test, geri kalanları training verisi olarak kullanılmış ve modellerin bu değişimlere verdiği sonuçlar gözlemlenmiştir.



Şekil 3.1 10 Katlı Çapraz Geçerleme Uygulamasının Şematik Gösterimi

3.3 Data Bucketing

Veri setinde tutarlılık sağlamak adına, azınlık durumunda olan sayısal değerler kategorik tipe dönüştürülmüştür. Bu işlem için bucketing tekniği kullanılmıştır. Bu işlem için hazırda kullanılabilecek kütüphaneler bulunmaktadır, ancak özgünlük katmak amacıyla bu adım başka kütüphanelerin yardımıyla açık bir şekilde baştan implemente edilmiştir. Buna göre sayısal veri türüne sahip size, age ve rent öznitelikleri, belirlenen sayı aralıklarına yerleştirilmiştir.

Önceki bölümde veri setinin darlığından ve pandemi faktörü dolayısıyla yansıttığı dengesizliklerden bahsedilmişti. Bu kısımda, bu açığı olabildiğince kapatabilmek adına, feature öznitelikleri dar bucketlara konulurken, class özniteliği olan rent için bu değer 200 olarak belirlenmiştir. Böylelikle hem yapılan tahminin anlamlı olma durumu korunmuş, hem de modellerin nokta atışı yapma zorunluluğu ortadan kaldırılmıştır.

Bu işlemle beraber veri setindeki her bir özniteliğin ifade ettiği değerler kategorik hale gelmiştir.

3.4 Özellik Seçimi

Veri setindeki öznitelik sayısının başarıya olan etkisini gözlemlemek adına kullanılabilecek en iyi yöntemlerden biri, çalışma kapsam açıklamasında da talep edildiği üzere, feature selection yaklaşımıdır. Bu programda bu işlem için Ki-Kare testinden faydalanılmıştır. Bu testten geçirilen özniteliklerin sonuca olan etkileri matematiksel olarak değerlendirilmiş, istenen sayıda öznitelik seçilerek sınıflandırma adımlarına daha az ama daha kullanışlı özniteliklerle geçilmesi sağlanmıştır.

Bu yaklaşımın farklı öznitelik sayılarıyla verdiği sonuçlar bir sonraki bölümde verilmiştir.

3.5 Özellik Dönüşümü

Veri setinde bulunan floor ve total_floors özniteliklerinin tek başlarına ayırt edici olmayacakları, kullanışlı olabilmeleri için beraber kullanılmaları gerektiği görülmüştür. Bu sebeple bu iki kolon yerine "floor_rated" adında, çıkarılan iki kolondaki değerlerin birbirine oranlanmasıyla oluşturulmuş bir kolon kullanılmıştır. Böylece hem veri içerisinde o dairenin apartmandaki diğer dairelere göre yeri daha iyi ifade edilmiş, hem öznitelik sayısı azaltılmış, hem de iki ayırt edici özelliği düşük öznitelik ayırt ediciliği yüksek başka bir özniteliğe dönüşmüştür. "floor_rated" özniteliğinin altındaki değerler 0 ve 1 aralığındaki sayılardan oluşturulmuştur.

Bunun dışında eldeki verilerin dönüştürülmesi ve dimension olarak küçülmesi için PCA tekniği kullanılmıştır. PCA tekniği uygulandığı zaman farklı değerler için alınan sonuçlar bir sonraki bölümde paylaşılacaktır.

3.6 Başarı Tespiti

Başarı tespiti, modellerin test verisi üzerine yapmış olduğu tahminler sonucu ortaya koydukları skorlar ile ölçülmektedir. Bunun yanı sıra f1-skoru ve confusion matrisi görüntüleme seçenekleri de bulunmaktadır.

3.7 Kütüphaneler

Çalışma kapsamında kullanılan kütüphaneler aşağıdaki gibidir.

- numpy: Dizi işlemleri.
- pandas: Dosya ve dataframe yönetimi için.
- matplotlib: Grafik gösterimleri.
- sklearn: ML algoritmaları, feature selection, feture extraction ve başarı ölçümleri.
- seaborn: Confusion Matrix çizimi için.
- statistics: Yalnızca mean() fonksiyonu için.

4
Program Çıktıları

Bu bölümde programın verdiği başarı oranları paylaşılacaktır. Bu çıktılar bir sonraki bölümde değerlendirilecektir.

4.1 Özellik Seçimi

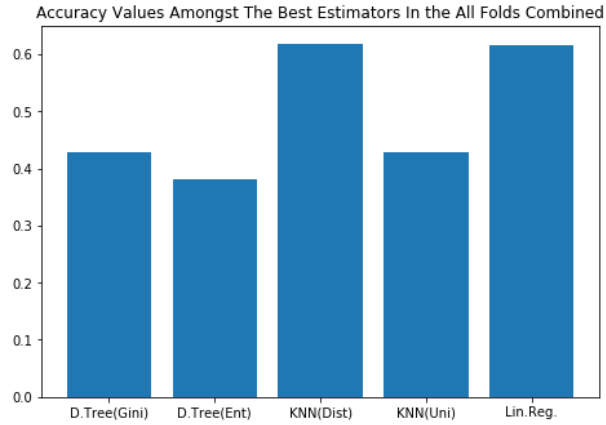
Bu başlık altında modellerin PCA uygulanmadan sadece farklı sayıda özelliklerin seçimiyle 10 katlı K-Fold boyunca verdikleri sonuçlar paylaşılacaktır.

4.1.1 Tüm Özniteliklerin Kullanılması

Bu kısımda özellik seçimi olmadan 5 farklı model için 10 katlı K-Fold ile alınan sonuçlar paylaşılacaktır.

10 farklı training-test veri kümesi kombinasyonu sonucu elde edilmiş her türden modeller ve accuracy değerleri Şekil-4.1'de verilmiştir. Aynı zamanda aynı şekilde her prediction sonucu toplanmış doğruluk skorlarının her model özelinde ortalaması da paylaşılmıştır.

```
In all folds, best D.Tree(Gini) with accuracy: 0.429 average: 0.26
In all folds, best D.Tree(Ent) with accuracy: 0.381 average: 0.279
In all folds, best KNN(Dist) with accuracy: 0.619 average: 0.323
In all folds, best KNN(Uni) with accuracy: 0.429 average: 0.294
In all folds, best Lin.Reg. with accuracy: 0.616 average: 0.31
```

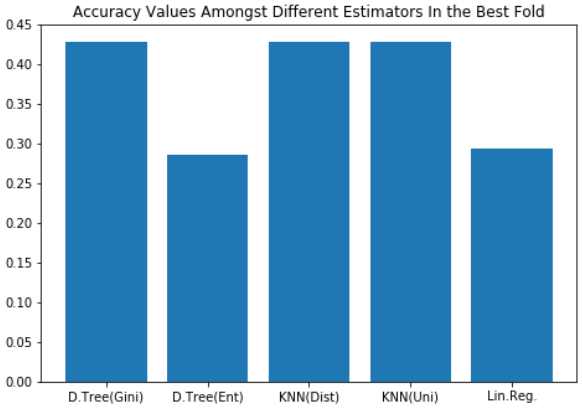


Şekil 4.1 Feature Selection Olmadan 10 Katlı Çapraz Geçerlemede Elde Edilmiş En İyi Modeller

Çapraz geçerleme adımlarının her bir tanesinde alınan sonuçların birbiriyle karşılaştırılması sonucu ortalama skor değerine göre belirlenen en iyi fold hakkında bilgiler Şekil-4.2'de verilmiştir.

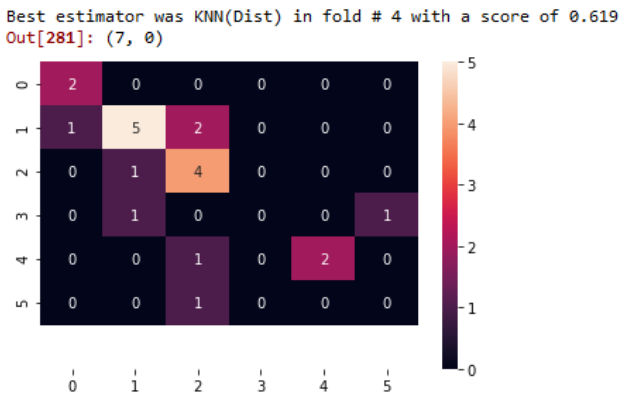
```
Best fold was # 1  with an average score amongst the models of  0.373

In the best fold, D.Tree(Gini) with accuracy: 0.429
In the best fold, D.Tree(Ent) with accuracy: 0.286
In the best fold, KNN(Dist) with accuracy: 0.429
In the best fold, KNN(Uni) with accuracy: 0.429
In the best fold, Lin.Reg. with accuracy: 0.293
```



Şekil 4.2 Feature Selection Olmadan 10 Katlı Çapraz Geçerlemede Elde Edilmiş En İyi Fold

Bütün tahminleyiciler ve foldlar arasında elde edilmiş en iyi tahminleyicinin Confusion Matrix çizimi Şekil-4.3'de verilmiştir.

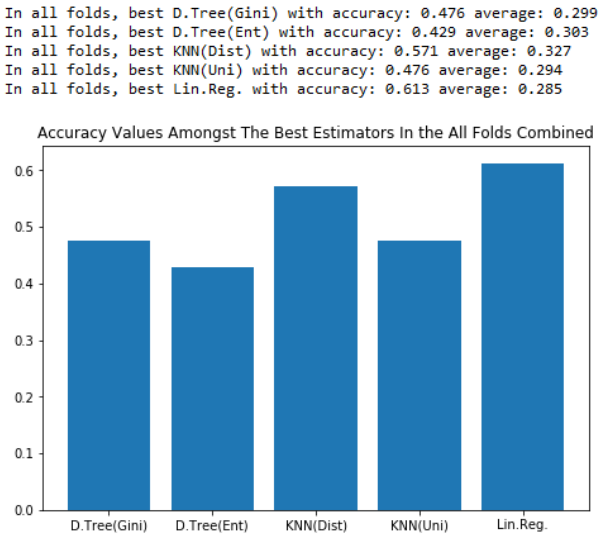


Şekil 4.3 Feature Selection Olmadan 10 Katlı Çapraz Geçerlemede Elde Edilmiş En İyi Tahminleyici

4.1.2 4 Özellik Seçimi ve Çapraz Geçerleme

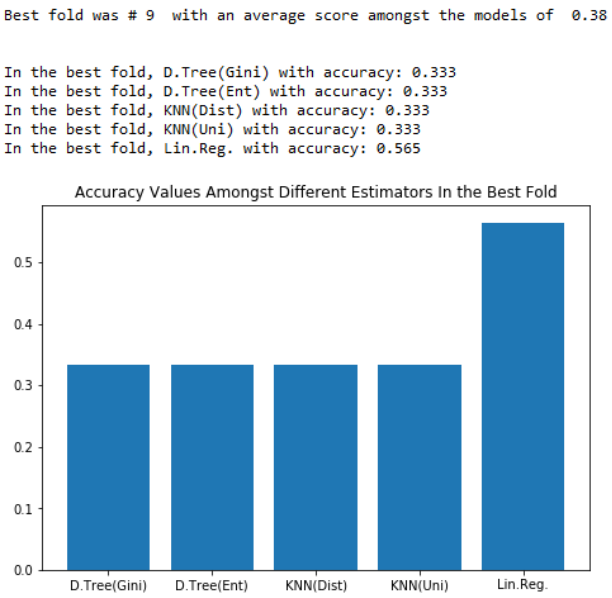
Bu kısımda 4 özelliğin seçilmesiyle yapılan sınıflandırmalarda 5 farklı model için 10 katlı K-Fold ile alınan sonuçlar paylaşılacaktır.

10 farklı training-test veri kümesi kombinasyonu sonucu elde edilmiş her türden modeller ve accuracy değerleri Şekil-4.4'de verilmiştir. Aynı zamanda aynı şekilde her prediction sonucu toplanmış doğruluk skorlarının her model özelinde ortalaması da paylaşılmıştır.



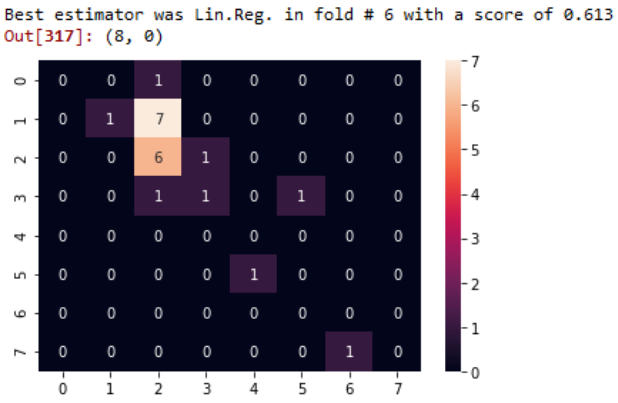
Şekil 4.4 4 Features Selection İle 10 Katlı Çapraz Geçerlemede Elde Edilmiş En İyi Modeller

Çapraz geçerleme adımlarının her bir tanesinde alınan sonuçların birbiriyle karşılaştırılması sonucu ortalama skor değerine göre belirlenen en iyi fold hakkında bilgiler Şekil-4.5'de verilmiştir.



Şekil 4.5 4 Features Selection İle 10 Katlı Çapraz Geçerlemede Elde Edilmiş En İyi Fold

Bütün tahminleyiciler ve foldlar arasında elde edilmiş en iyi tahminleyicinin Confusion Matrix çizimi Şekil-4.6'de verilmiştir.



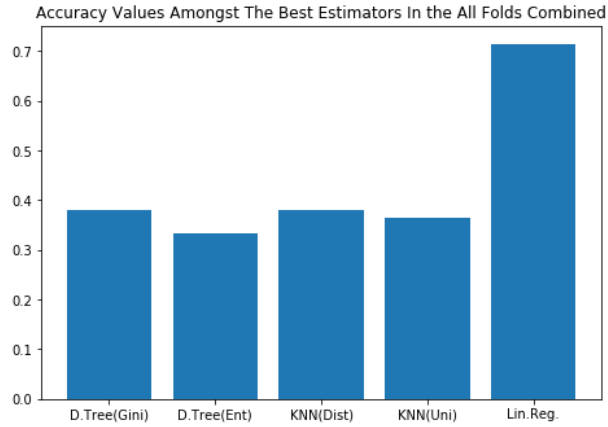
Şekil 4.6 4 Features Selection İle 10 Katlı Çapraz Geçerlemede Elde Edilmiş En İyi Tahminleyici

4.1.3 2 Özellik Seçimi ve Çapraz Geçerleme

Bu kısımda 2 özelliğin seçilmesiyle yapılan sınıflandırmalarda 5 farklı model için 10 katlı K-Fold ile alınan sonuçlar paylaşılacaktır.

10 farklı training-test veri kümesi kombinasyonu sonucu elde edilmiş her türden modeller ve accuracy değerleri Şekil-4.7'de verilmiştir. Aynı zamanda aynı şekilde her prediction sonucu toplanmış doğruluk skorlarının her model özelinde ortalaması da paylaşılmıştır.

```
In all folds, best D.Tree(Gini) with accuracy: 0.381 average: 0.256
In all folds, best D.Tree(Ent) with accuracy: 0.333 average: 0.252
In all folds, best KNN(Dist) with accuracy: 0.381 average: 0.242
In all folds, best KNN(Uni) with accuracy: 0.364 average: 0.255
In all folds, best Lin.Reg. with accuracy: 0.716 average: 0.271
```

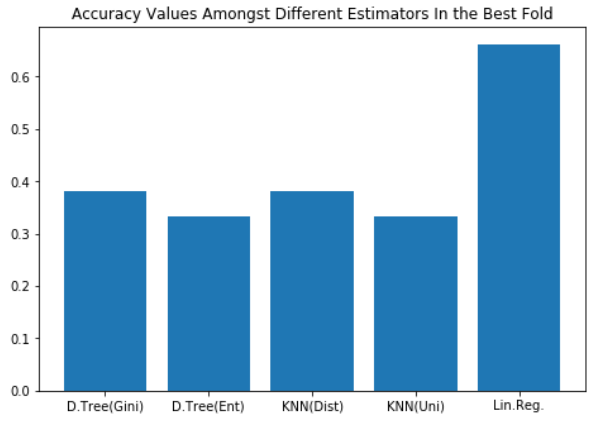


Şekil 4.7 2 Features Selection İle 10 Katlı Çapraz Geçerlemede Elde Edilmiş En İyi Modeller

Çapraz geçerleme adımlarının her bir tanesinde alınan sonuçların birbiriyle karşılaştırılması sonucu ortalama skor değerine göre belirlenen en iyi fold hakkında bilgiler Şekil-4.8'de verilmiştir.

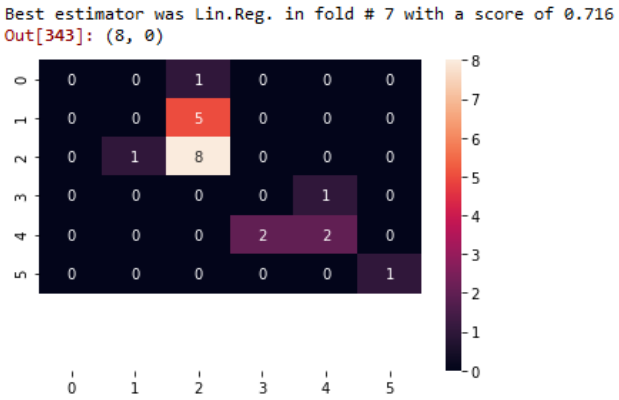
```
Best fold was # 9  with an average score amongst the models of  0.418

In the best fold, D.Tree(Gini) with accuracy: 0.381
In the best fold, D.Tree(Ent) with accuracy: 0.333
In the best fold, KNN(Dist) with accuracy: 0.381
In the best fold, KNN(Uni) with accuracy: 0.333
In the best fold, Lin.Reg. with accuracy: 0.662
```



Şekil 4.8 2 Features Selection İle 10 Katlı Çapraz Geçerlemede Elde Edilmiş En İyi Fold

Bütün tahminleyiciler ve foldlar arasında elde edilmiş en iyi tahminleyicinin Confusion Matrix çizimi Şekil-4.9'de verilmiştir.

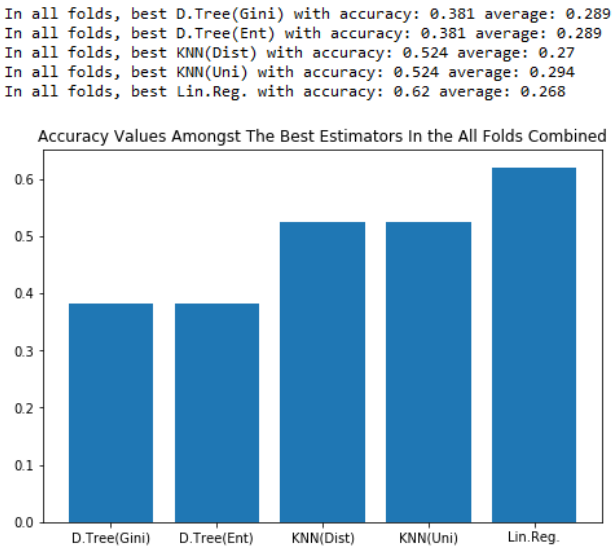


Şekil 4.9 2 Features Selection İle 10 Katlı Çapraz Geçerlemede Elde Edilmiş En İyi Tahminleyici

4.1.4 Tek Özellik Seçimi ve Çapraz Geçerleme

Bu kısımda tek özelliğin seçilmesiyle yapılan sınıflandırmalarda 5 farklı model için 10 katlı K-Fold ile alınan sonuçlar paylaşılacaktır. Ki-Kare testi ile en ayırt edici özelliğin size olduğuna karar verilmiştir.

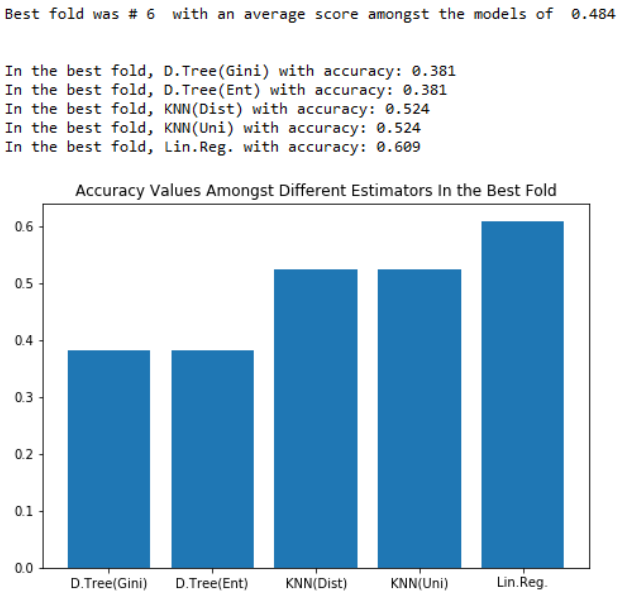
10 farklı training-test veri kümesi kombinasyonu sonucu elde edilmiş her türden modeller ve accuracy değerleri Şekil-4.10'de verilmiştir. Aynı zamanda aynı şekilde her prediction sonucu toplanmış doğruluk skorlarının her model özelinde ortalaması da paylaşılmıştır.



Şekil 4.10 Yalnızca "size" Özniteliği İle 10 Katlı Çapraz Geçerlemede Elde Edilmiş En İyi Modeller

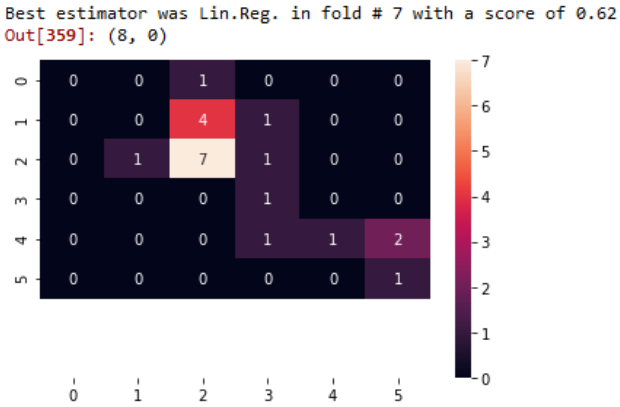
Çapraz geçerleme adımlarının her bir tanesinde alınan sonuçların birbiriyle karşılaştırılması sonucu ortalama skor değerine göre belirlenen en iyi fold hakkında

bilgiler Şekil-4.11'de verilmiştir.



Şekil 4.11 Yalnızca "size" Özniteliği İle 10 Katlı Çapraz Geçerlemede Elde Edilmiş En İyi Fold

Bütün tahminleyiciler ve foldlar arasında elde edilmiş en iyi tahminleyicinin Confusion Matrix çizimi Şekil-4.12'de verilmiştir.



Şekil 4.12 Yalnızca "size" Özniteliği İle 10 Katlı Çapraz Geçerlemede Elde Edilmiş En İyi Tahminleyici

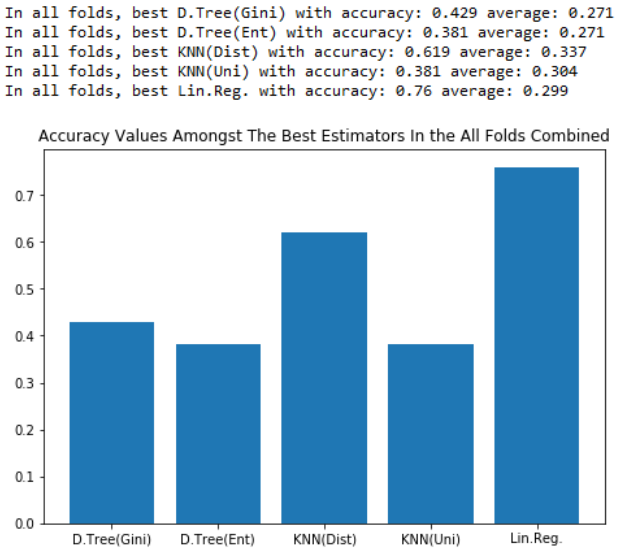
4.2 Özellik Dönüşümü

Bu başlık altında modellerin K-Kare ile 8 adet özelliğin seçilmesinin ardından PCA ile dönüştürülerek sırasıyla 4, 2 ve 1 taneye düşürülmesi sonucu elde edilen başarı oranları paylaşılacaktır.

4.2.1 PCA İle 4 Özelliğe Dönüşüm

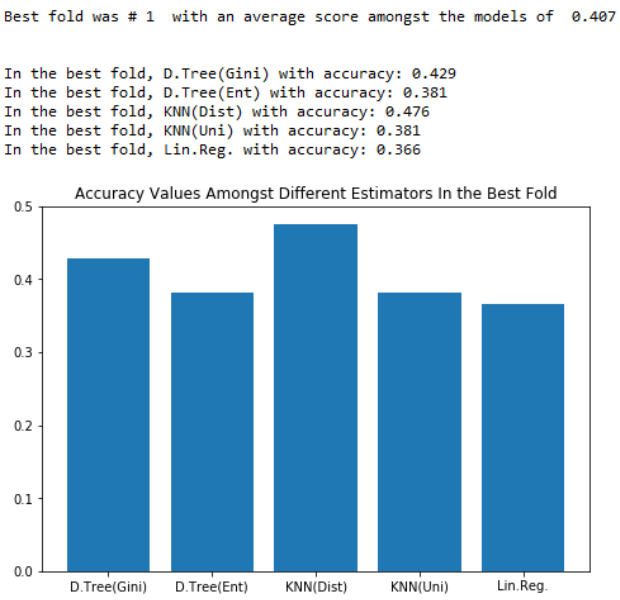
Bu kısımda seçilmiş 8 özellikten PCA ile dört özelliğe dönüşüm sonucu yapılan sınıflandırmalarda 5 farklı model için 10 katlı K-Fold ile alınan sonuçlar paylaşılacaktır.

10 farklı training-test veri kümesi kombinasyonu sonucu elde edilmiş her türden modeller ve doğruluk skorları Şekil-4.13'de verilmiştir. Aynı zamanda aynı şekilde her prediction sonucu toplanmış doğruluk skorlarının her model özelinde ortalaması da paylaşılmıştır.



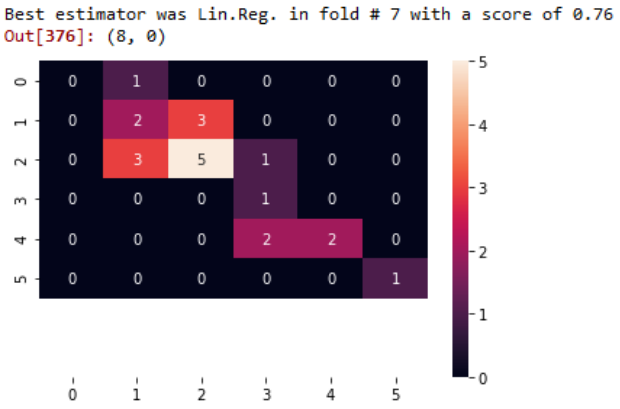
Şekil 4.13 PCA 4 Özelliğe Dönüşüm İle 10 Katlı Çapraz Geçerlemede Elde Edilmiş En İyi Modeller

Çapraz geçerleme adımlarının her bir tanesinde alınan sonuçların birbiriyle karşılaştırılması sonucu ortalama skor değerine göre belirlenen en iyi fold hakkında bilgiler Şekil-4.14'de verilmiştir.



Şekil 4.14 PCA 4 Özelliğe Dönüşüm İle 10 Katlı Çapraz Geçerlemede Elde Edilmiş En İyi Fold

Bütün tahminleyiciler ve foldlar arasında elde edilmiş en iyi tahminleyicinin Confusion Matrix çizimi Şekil-4.15'de verilmiştir.



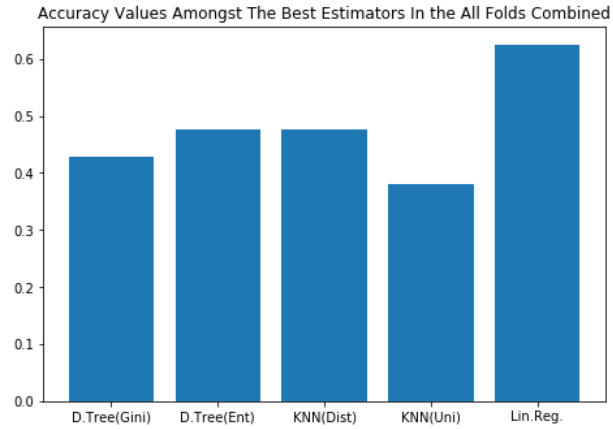
Şekil 4.15 PCA 4 Özelliğe Dönüşüm İle 10 Katlı Çapraz Geçerlemede Elde Edilmiş En İyi Tahminleyici

4.2.2 PCA İle 2 Özelliğe Dönüşüm

Bu kısımda seçilmiş 8 özellikten PCA ile iki özelliğe dönüşüm sonucu yapılan sınıflandırmalarda 5 farklı model için 10 katlı K-Fold ile alınan sonuçlar paylaşılacaktır.

10 farklı training-test veri kümesi kombinasyonu sonucu elde edilmiş her türden modeller ve doğruluk skorları Şekil-4.16'de verilmiştir. Aynı zamanda aynı şekilde her prediction sonucu toplanmış doğruluk skorlarının her model özelinde ortalaması da paylaşılmıştır.

```
In all folds, best D.Tree(Gini) with accuracy: 0.429 average: 0.303
In all folds, best D.Tree(Ent) with accuracy: 0.476 average: 0.299
In all folds, best KNN(Dist) with accuracy: 0.476 average: 0.304
In all folds, best KNN(Uni) with accuracy: 0.381 average: 0.27
In all folds, best Lin.Reg. with accuracy: 0.626 average: 0.269
```

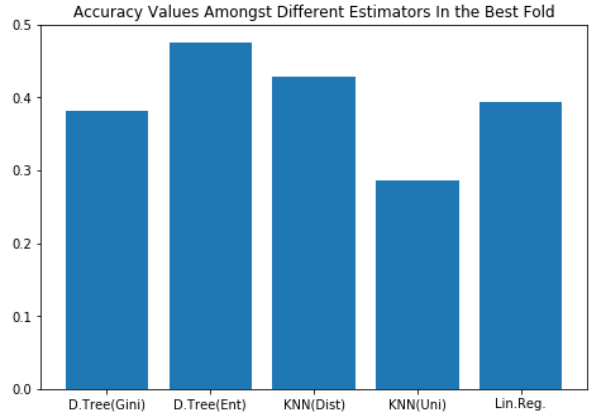


Şekil 4.16 PCA 2 Özelliğe Dönüşüm İle 10 Katlı Çapraz Geçerlemede Elde Edilmiş En İyi Modeller

Çapraz geçerleme adımlarının her bir tanesinde alınan sonuçların birbiriyle karşılaştırılması sonucu ortalama skor değerine göre belirlenen en iyi fold hakkında bilgiler Şekil-4.17'de verilmiştir.

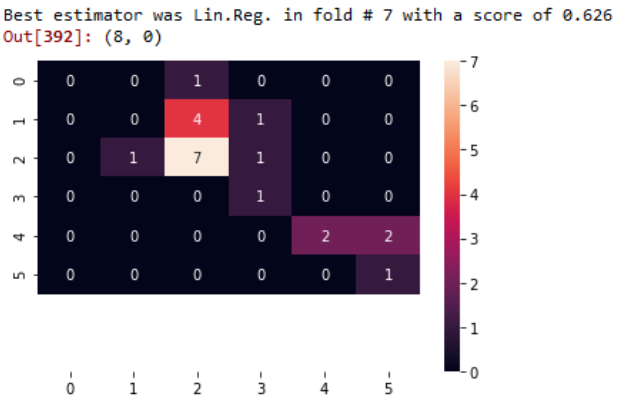
```
Best fold was # 1  with an average score amongst the models of  0.393

In the best fold, D.Tree(Gini) with accuracy: 0.381
In the best fold, D.Tree(Ent) with accuracy: 0.476
In the best fold, KNN(Dist) with accuracy: 0.429
In the best fold, KNN(Uni) with accuracy: 0.286
In the best fold, Lin.Reg. with accuracy: 0.394
```



Şekil 4.17 PCA 2 Özelliğe Dönüşüm İle 10 Katlı Çapraz Geçerlemede Elde Edilmiş En İyi Fold

Bütün tahminleyiciler ve foldlar arasında elde edilmiş en iyi tahminleyicinin Confusion Matrix çizimi Şekil-4.18'de verilmiştir.



Şekil 4.18 PCA 2 Özelliğe Dönüşüm İle 10 Katlı Çapraz Geçerlemede Elde Edilmiş En İyi Tahminleyici

5
Sonuç

Bu bölümde programın verdiği sonuçların nedenleri irdelenecektir.

5.1 Gözlem

5.1.1 Decision Trees

İki farklı kriterle oluşturulmuş iki farklı Decision Tree tahminleyicilerin değişen veri yapıları karşısında genel olarak ortalama doğruluk oranlarını korudukları gözlemlenmiştir. 10 katlı geçerlemenin her bir iterasyonunda elde edilen başarıların ortalama değerlerine bakıldığı zaman, ortalama doğruluğun PCA uygulanarak özniteliklerin ikiye indirgendiği denemede %30 bandına yükseldiği görülmüştür ancak bu değerin yapının tutarlılığı sayesinde Decision Tree'lerin diğer doğruluk skorlarına göre çok da farklı bir değer olmadığı belirtilmelidir.

Bunun dışında elde edilmiş en yüksek doğruluk skoruna sahip ağaç yapısı dört özellik seçimi uygulaması ve PCA ile iki özellik çıkarımı uygulamalarında, hem Gini hem de entropi kriterlerine sahip ağaçlarda ortaya çıkmıştır. Veri setinin darlığı, birçok öznitelik değeri için yalnızca tek bir örneğin bulunuyor olması ve veri setinde fiyat ve diğer bazı özniteliklerde homojenliğin sağlanmamış olması göz önünde bulundurulduğunda, böyle bir problemde neredeyse %50 başarı oranıyla çalışan bir ağaç yapısı elde etmiş olmak olumlu bir sonuç olarak değerlendirilebilir.

5.1.2 K-Nearest Neighbor Algoritması

KNN algoritmasının böyle bir problemde kullanılmasında kümeleme algoritmalarından faydalanılarak sınıflandırma yapma mentalitesi takip edilmiştir. KNN algoritması bu sınıflandırma problemi için Decision Tree ile aşağı yukarı benzer doğruluk oranı ortaya koymuştur. Genel foldlar arası ortalamaya bakıldığında aşağı yukarı %30 ortalaması yakalandığı görülmektedir. Ancak yapılan denemelerden birinde ortaya konmuş olan Distance ağırlıklandırmalı KNN yapısı oldukça

dikkat çekicidir. Tüm özelliklerin kullanıldığı, feature selection veya extraction uygulamalarının devre dışı bırakıldığı denemede, KNN yapılarından biri %62 başarı oranı yakalamıştır. Bu denemedeki en iyi tahminleyici olma başarısı gösteren bu tahminleyicinin Confusion Matrix'i Şekil-4.3'de verilmiştir.

5.1.3 Lineer Regresyon

Yapılan denemelerde en çok dikkat çeken tahminleyiciler Lineer Regresyon modelleri olmuştur. Problemi basit matematiksel bir eşitliğe çevirme mentalitesiyle çalışan bu basit yapı, yapılan denemelerde ilki hariç diğer hepsinde en iyi model örneğini vermeyi başarmıştır. Bu değerler arasında zirvede %76 başarı oranı ile seçilen 8 özniteliğin PCA ile 4'e indirgendiği çalışmada 7. foldda ortaya çıkan model vardır. Bu tahminleyicinin Confusion Matrix'i Şekil-4.15'de verilmiştir. Böylesine ucu açık bir problemi, az veri ve basit bir mantıkla %80'e yakın bir başarı oranı ile çözebilmek çok olumlu bir sonuç olarak yorumlanmıştır.

Duruma başka bir perspektifte bakmak gerekirse, Lineer Regresyon tahminleyicilerinin denemeler boyunca hit-or-miss durumunda oldukları görülmektedir. Her denemede en iyi sonuçları veren ve bar grafiklerinde genelde üstünlüğünü açık bir şekilde fark ettiğimiz Lineer Regresyon tahminleyicileri, konu foldlar arası genel ortalamaya geldiğinde çoğu zaman diğerlerinin gerisinde kalmıştır. Bu ortalama hesabında, az önce bahsettiğimiz %60-70 seviyelerinde doğruluk oranlarının da olduğu düşünüldüğünde, bazı foldlarda karşımıza çıkan tahminleyicilerin çözüme hiçbir şekilde yakınsayamadığı, tam tersine çok uzakta kaldığı görülmektedir. Bu sebeple bu problem özelinde bize en iyi sonuçları veren bu tahminleyicilerin tutarlı bir performans ortaya koymadıkları anlaşılmaktadır.

5.1.4 Feature Selection'un Genel Etkileri

Bu çalışmaya benzer yaklaşıma sahip diğer çalışmalarda kullanılan veri setlerinin içeriği ve özniteliklerin problemin amacıyla olan bağlantısı değişkenlik gösterebilir. Bu problem için hazırlanan veri seti, spesifik olarak bu problem akılda tutularak oluşturulmuştur. Hatta, problemin daha iyi çözülmesi için eldeki verinin özniteliklerinin yeterli olmadığı bile savunulabilir. Feature Selection uygulamasının her ne kadar genelde olumlu bir etki yarattığı görülse de, genel ortalama skorlara bakıldığı zaman yaptığı etkinin bütün modelleri kapsamadığı anlaşılmaktadır. Seçilen özellik sayısı arttıkça doğruluk skorlarında ortaya çıkan etki için doğrusal bir mantık oluşturmak mümkün değildir.

Feature selection ile özellik sayısı bire düşürüldüğü zaman en ayırt edici özniteliğin

"size" özniteliği olduğu görülmüştür. Bölüm-4.1.4'te incelenen örnekteki Lineer Regresyon modeli, sadece size özniteliğinden yola çıkarak bir dairenin kira bedelini >%60 doğruluk oranıyla +-100 Türk Lirası hassaslığında tahmin edebilmektedir.

5.1.5 Feature Extraction'un Genel Etkileri

Denemelerde en net gözlemlenen durumlardan biri de doğru uygulanan bir Feature Extraction'ın bazı tahminleyicilerin başarılarına olan direkt etkisi olmuştur. Bütün denemelerde ortaya çıkan en iyi modellerden iki tanesi, PCA Feature Extraction ile seçilmiş 8 özniteliğin 4'e indirgenmesiyle ortaya çımıştır.

5.2 Çıkarım

Bir evin kira bedelini tahmin etme problemi, gerçek hayatta bile birçok açıdan bakmakla değerlendirilebilecek bir problemdir. Gerçek hayatta bir evi tutmadan önce insanlar için metrekare büyüklüğü veya evin hangi cepheye baktığı gibi durumlardan çok daha önemli etmenler de vardır. Belirtilen emlak sitesinden toplanan bu veri setinin içeriğindeki özniteliklerin yanı sıra, evlerin etrafındaki okullar ve işletmeler, bölgedeki suç oranı, toplu taşıma ile ulaşım imkanları, evin içinin yapılı veya bakımlı olup olmama durumu, (Özellikle Eskişehir'in coğrafi konumu için) evin 1999 depremini görüp görmemiş olması, bölgede fiber internet bağlantısının yapılıp yapılmamış olması ve bunun gibi daha onlarca öğe, bir konutun kira bedelini direkt olarak etkilemektedir. Bütün bunların yanı sıra, evin emlakçı yardımıyla veya direkt sahibi tarafından kiraya verilmiş olması, pazarlık payı bırakılıp bırakılmamış olması veya ev sahibinin ne kadar aciliyetle kiracı arıyor olması da önemli etmenlerdir. Üstelik oluşturulan veri seti 200 daire içermektedir ve bu sayı bir insan beyni için bile kesin tahmin yapmak için yeterli olmayabilir. Bu zorluklar, bucket genişlikleri ile olabildiğince yumuşatılmıştır ve bunun doğruluğa doğrudan etki ettiği görülmüştür.

Veri setinin zayıflıklarının yanı sıra, kullanılan algoritmalar her ne kadar veri biliminde oldukça popüler yaklaşımlar içerse de, genel olarak basit bir mantığa sahip temel düzey algoritmalar oldukları unutulmamalıdır. Bu etmenlerin hepsi göz önünde bulundurulduğunda, ortaya konmuş olan %70-80 bandındaki doğruluk skorları oldukça başarılı olarak değerlendirilmiştir.

Algoritmaların mantıklarındaki farklılıklar, farklı parametrelerle yapılan denemelerde her birinin farklı zamanlarda başarılı sonuçlar vermiş olmasından anlaşılmaktadır. Bu sebeple bu çalışma özelinde yalnızca kısmen daha başarılı veya kısmen daha başarısız deneme ayrımı yapılabilir, en iyisini veya en kötüsünü belirlemek mümkün olmayacaktır.