

4. Model Çalışması

Tez çalışmasının bu bölümünde, fonların fiyat tahmini için model geliştirme uygulamaları yapılacaktır. Model geliştirme için kullanacağımız makine öğrenme metotlarına dair literatür araştırması bölüm 2’de yapılmıştır. Yapılan literatür araştırmasında kendi çalışmamız için kullanacağımız metotları tercih etmemizi sağlayan, büyük bir uygulama alanının mevcut olunması ve benzer çalışmalarda makine öğrenme metotlarının ciddi başarılar elde etmesi, bölüm 3’de açıklamaları yapılmış olana algoritmaları kullanmamızı sağlayacak alt yapıyı oluşturmuştur. Bu bağlamda bu bölümde uygulama olarak beş ana başlık şeklinde bir akış izlenmiştir. İlk olarak, uygulama çalışmaların temel yapısını oluşturan ve önemi yüksek olan veri seti hakkında detaylı bir bilgilendirme verilmiştir. Sonrasında, model geliştirme bölümüne geçmeden yeni bir başlık açılarak, geliştirilecek olan modellerin kabul doğruluklarının ölçümünü yapılmasını sağlayan metrikler ile ilgi açıklamalar yapılmış ve daha sonra model geliştirme için kullanılacak olan algoritmaların işleyişi ve kodlama süreçleri için detaylı bilgi verilmiştir. Son ana başlığa ise, kullanılan dört algoritma ile uygulaması yapılan model geliştirme için bulunan sonuçları ortak tablo olarak karşılaştırması yapılarak bölüm sonlandırılmıştır. Ayrıca, tüm uygulama geliştirme ve veri işlem süreçlerin de kullanılan platform, programlama dili ve versiyon bilgileri için tablo 4.1’de ve uygulama çalışmalarında kullanılacak olan bilgisayarın teknik özelliklerine dair ise tablo 4.2’ de olmak üzere ayrıntılı bilgi verilmiştir.

Tablo 4.1: Uygulama geliştirme platform bilgileri

Uygulama Adı	Açıklama	Version
Anaconda Navigator	Veri bilimi masaüstü portalı	1.9.12
Python3	Nesne yönelimli, yorumlamalı ve etkileşimi yüksek seviyeli bir programlama dilidir.	3.7.7
Spyder	Python programlama geliştirme için açık kaynaklı bir geliştirme ortamıdır.	4.1.2
JupyterLab	Bir çok programlama dilinde etkileşimli bilgi işlemleri için geliştirilmiş açık kaynaklı bir geliştirme ortamıdır.	1.2.6

Tablo 4.2: Bilgisayar teknik özellikleri

İşlemci	CPU @ 2.7 GHz, Dual-Core Intel Core i5
Bellek	8 GB
Grafikler	Intel Iris Graphics 6100 1536 MB
Depolama	SSD 128 GB

4.1. Veri Hazırlama ve Ön İşleme Süreçleri

Bu bölüm iki alt başlıktan oluşmaktadır. İlk olarak model geliştirme için kullanılacak olan veri seti hakkında detaylı bir bilgi verilmiştir. İkinci olarak, veri ön işleme süreçleri için veri setinin oluşturulması, temizlenmesi ve verinin görselleştirme adımlarının tümü bu başlık altında incelenmiştir.

4.1.1 Veri seti hakkında

Bu çalışmada kullanılan veriler, Takas İstanbul(Takasbank)’un platform ve veri kaynağı sağlayacağı yaptığı Türkiye Elektronik Fon Dağıtım Platformu (TEFAS) web sitesi üzerinden genele açık olan bilgilerden elde edilmiştir. Her bir fon için 02.01.2019 – 31.12.2019 tarihleri arasında, tipi, türü, toplam değeri, tedavüldeki pay sayısı, pay alan kişi sayısı, fiyatı ve menkul(26 çeşit) oranları ile her bir tarih için faiz bilgisi, altın fiyatı ve dolar fiyatı baz alınarak veri seti hazırlanmıştır. Veri kümesi, toplam 187,438 veri ve 37 kolondan oluşmaktadır. Veri setini oluşturan her bir kolonun tipi ve içeriği hakkında detaylı bilgi ise aşağıda verilmiştir.

- **TARİH:** Fonun işlem gördüğü tarih bilgisini içerir. Veri tipi, date olarak tutulur.
- **FONTİP:** Fonun tip bilgisini içerir ve 3 çeşittir; Borsa Yatırım Fonu, Emeklilik Fonu, Yatırım Fonu. Veri tipi, object olarak tutulmaktadır. Veri setinde object olarak tutulan veriler, veri ön işlem sürecinde kategorik veri olacak şekilde dönüşümü yapılacaktır.

- **FONTUR:** Fonların tür bilgisini içermekte ve 31 çeşit olarak mevcuttur. Tür sayısı çok olduğundan dolayı Tablo 4.3'te bilgileri verilmektedir. Veri tipi, object olarak tutulmakta.
- **FON:** Veri setinde fonların uzun isimlendirmeleri mevcut değildir. Bunun yerine, Takasbank tarafından her bir fon'a ait bir kod bilgisi verilmiştir. Toplam fon sayısı 892 ve veri tipi olarakta object bilgisini içermektedir.
- **FONTOPLAMDEGER:** Fonun büyüklük bilgisini içerir ve hesaplama olarak; portföy büyüklüğü + alacaklar - borçlar şeklinde bulunur. Veri tipi, float64 olarak tutulur.
- **TEDAVPAYSAYISI:** Fonun satılmış olan pay adeti bilgisini içerir. Veri tipi, float64 olarak tutulur.
- **KISISAYISI:** Fonun yatırımcı sayısı bilgisini barındırır. Veri tipi, int64 olarak bulunur.
- **FONFIYAT:** Fonun o tarihteki fiyat bilgisini içerir. Veri tipi, float64 olarak tutulmaktadır.
- **FAIZ:** Tarih bazında Merkez Bankasının verdiği faiz bilgisini barındırır. Veri tipi, float64 olarak tutulur.
- **DOLARFIYAT:** Merkez Bankasının gün sonunda o tarih için en son verdiği dolar fiyatını bulundurmaktadır. Veri tipi, float64 olarak bulunur.
- **ALTINFIYAT:** Merkez Bankasının gün sonunda o tarih için son verdiği altın fiyat bilgisini bulundurmaktadır. Veri tipi ise, float64'tur.
- **MENKULORAN:** Fonun portföyündeki kıymetin portföy değerine oranın, yüzdelik üzerinden gösterilmesi bilgisini içerir. Veri setindeki menkul oranı, 26 farklı menkul değerini kolon bilgisi olarak bulundurmaktadır. Verideki her bir menkul değerinin kodu ve tanımı tablo 4.4'te gösterilmektedir. Veri tipleri ise, tüm menkul değerleri float64 olarak veri kümesinde tutulmaktadır.

Tablo 4.3: Veri setindeki fon türleri

FONTUR	
1	Altın Fonu
2	Gümüş Fonu
3	Hisse Senedi Fonu
4	Başlangıç Fonu
5	Başlangıç Katılım Fonu
6	Borçlanma Araçları Fonu
7	Devlet Katkısı Fonu
8	Değişken Fon
9	Endeks Fon
10	Fon Sepeti Fonu
11	Kamu Borçlanma Araçları Fonu
12	Kamu Kira Sertifikası Fonu
13	Kamu Yabancı Para (Döviz) Cinsinden Borçlanma Araç
14	Karma Fon
15	Katılım Katkı Fonu
16	Katılım Standart Fon
17	Kıymetli Madenler
18	OKS Katılım Standart Fon
19	OKS Standart Fon
20	Para Piyasası Fonu
21	Standart Fon
22	Uluslararası Borçlanma Araçları Fonu
23	Özel Sektör Borçlanma Araçları Fonu
24	Altın ve Diğer Kıymetli Madenler Fonu
25	Hisse Senedi Yoğun
26	Katılım Fonu
27	Kira Sertifikası Fonu
28	Kısa Vadeli Kira Sertifikaları Katılım Fonu
29	Koruma Amaçlı Fon
30	Kısa Vadeli Borçlanma Araçları Fonu
31	Serbest Fon

Tablo 4.4: Takasbank'tan alınan menkul kod tanımları

KOD	ACIKLAMAING	ACIKLAMATR
BB	Bank Bills	Banka Bonosu
DT	Government Bond	Devlet Tahvili
DB	FX Payable Bills	Döviz Ödemeli Bono
DÖT	Foreign Currency Bills	Döviz Ödemeli Tahvil
EUT	Eurobonds	Eurobonds
FB	Commercial Paper	Finansman Bonosu
FKB	Fund Participation Certificate	Fon Katılma Belgesi
GAS	Real Estate Certificate	Gayri Menkul Sertifikası
HB	Treasury Bill	Hazine Bonosu
HS	Stock	Hisse Senedi
KBA	Government Bonds and Bills (FX)	Kamu Dış Borçlanma Araçları
KKS	Government Lease Certificates	Kamu Kira Sertifikaları
KH	Participation Account	Katılım Hesabı
KM	Precious Metals	Kıymetli Madenler
OSKS	Private Sector Lease Certificates	Özel Sektör Kira Sertifikaları
OST	Private Sector Bond	Özel Sektör Tahvili
TR	Reverse-Repo	Ters-Repo
TPP	TMM	TPP
T	Derivatives	Türev Araçları
VM	Term Deposit	Vadeli Mevduat
VDM	Asset-Backed Securities	Varlığa Dayalı Menkul Kıymetler
YBA	Foreign Debt Instruments	Yabancı Borçlanma Aracı
YHS	Foreign Equity	Yabancı Hisse Senedi
YMK	Foreign Securities	Yabancı Menkul Kıymet
D	Other	Diğer
R	Repo	Repo

Bu bölümde buraya kadar anlatılan, verilerin toplanması ve her bir kolon için içerik ve tip bilgilerinden bahsedildi. Ayrıca, verilerin toplanma sürecinde herhangi bir ön işlem süreci

uygulanmadığı için mevcut şu an ki format verinin ham halini barındırmaktadır. Bundan dolayı, eksik, yanlış ve gereksiz verilerde bulunmaktadır. Veri işlem bölümünde ayrıntılı incelemesi yapılacaktır.

4.1.2 Veri Ön İşleme Süreçleri

Veri işleme bölümünde, hazırlanmış verinin ham formatı üzerinde ön işleme süreçleri yapılacaktır. Ön işlem süreçleri veri setinin, uygulama bölümünde geliştirilecek olan model için çok önemli bir yere sahiptir. Çünkü ham veri, toplanma sürecinde gereksiz, hatalı, tekrarlı, tutarsız ve boş verilerinin olması mümkün ve verinin bu formatı ise geliştireceğimiz model için doğru bir sonuç elde edilmesini engeler. Bu sebepten ötürü, veri setini tutarlı, eksiksiz ve hatasız olmasını sağlayacak daha ideal bir yapıya dönüştürülür. Doğru bir veri setinin ise, model geliştirme çalışmalarımız için daha performanslı ve doğru sonuçlar elde edilmesini sağlar. Bu kapsamda, verilerin uygulama geliştirme ve analizler için daha uygun bir yapıya getirmesini sağlayan süreçte veri ön işleme adımı denir[1]. Veri ön işleme adımlarını aşağıdaki şekilde sıralanabilir.

- Veri temizleme
- Veri bütünleştirme
- Veri indirgeme
- Veri dönüştürme
- Veri madenciliği tekniklerinin uygulanması
- Bulunun sonuçların değerlendirilmesi, uygulanılması[1].

Bu adımlar çerçevesinde, veri setimizi adımı adım ön işleme süreçlerini yapıp ideal bir veri kümesine dönüşümü sağlanacaktır. Temizlenmiş veri kümesi üzerinden, her bir veri kolonun ortamlaması, maximum, minimum değerleri ve korelasyon matrislerine bakılarak gerekli-gereksiz veri olup olmam konusu incelenecektir. Son olarak ise, veri analizinin daha iyi anlaşılması için görselleştirme çalışmaları yapılacaktır.

4.1.2.1 Veri Temizleme ve Dönüştürme

Veri setinin düzenlenmemiş formatı şekil 4.1 verilerek veri üzerinden, temizleme ve dönüştürme işlemleri sırasıyla yapılmaya başlanacaktır.

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 187438 entries, 0 to 187437
Data columns (total 38 columns):
#   Column              Non-Null Count  Dtype
---  -
0   TARIH                187438 non-null  datetime64[ns]
1   FONTIP               187438 non-null  object
2   FONTUR               187438 non-null  object
3   FON                  187438 non-null  object
4   Unnamed: 4           0 non-null       float64
5   FONTOPLAMDEGER       187438 non-null  float64
6   TEDAVPAYSAYISI       187438 non-null  float64
7   KISISAYISI           187438 non-null  int64
8   FONFIYAT             187438 non-null  float64
9   FAIZ                187438 non-null  float64
10  DOLARFIYAT           187438 non-null  float64
11  ALTINFIYAT           187438 non-null  float64
12  BB                   187438 non-null  float64
13  DT                   185256 non-null  float64
14  DB                   173992 non-null  float64
15  DÖT                  148075 non-null  float64
16  EUT                  113791 non-null  float64
17  FB                   84322 non-null   float64
18  FKB                  55378 non-null   float64
19  GAS                  34803 non-null   float64
20  HB                   21222 non-null   float64
21  HS                   11863 non-null   float64
22  KBA                  6322 non-null    float64
23  KKS                  2693 non-null    float64
24  KH                   1216 non-null    float64
25  KM                   548 non-null     float64
26  OSKS                 211 non-null     float64
27  OST                  11 non-null      float64
28  TR                   0 non-null       float64
29  TPP                  0 non-null       float64
30  T                    0 non-null       float64
31  VM                   0 non-null       float64
32  VDM                  0 non-null       float64
33  YBA                  0 non-null       float64
34  YHS                  0 non-null       float64
35  YMK                  0 non-null       float64
36  D                    0 non-null       float64
37  R                    0 non-null       float64
dtypes: datetime64[ns](1), float64(33), int64(1), object(3)
memory usage: 54.3+ MB
```

Şekil 4.1 : Veri setinin ön işlem yapılmamış formatı

İlk olarak, şekil 4.1’de veri setinin toplam 187,438 veri ve 38 kolondan oluştuğu bilgisi verilmiş. Daha sonra dört kolondan oluşan yukarıdaki şekil 4.1 ise kolon bilgileri sırasıyla, verinin veri kümesindeki kolon sırası, adı, mevcut kolonda bulunan veri sayısı ve tip bilgilerini içermektedir.

Son olarak, bulunan verilerin tip sayısı ve ne kadar hafıza kullanıldığı bilgisi verilmiştir. Şekil 4.1’de anlaşılabacağı üzere, dördüncü sıradaki kolonun tanımsız olduğu ve boş veri içerdiği görülmektedir. Ayrıca, 28’inci sıradaki TR’den başlayarak 37’inci kolandaki alana kadar herhangi bir veri içermediği ve gereksiz olduğu anlaşıyor. Bu bilgiler ışığında, veri setimiz hatalı ve gereksiz kolonlardan temizlenecek ve object olan verilerin tipleri kategorik olarak dönüşüm işlemleri bu adımda yapılarak, şekil 4.2 ‘deki son halini almış olacaktır.

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 187438 entries, 0 to 187437
Data columns (total 27 columns):
#   Column                Non-Null Count  Dtype
---  ---
0   TARİH                 187438 non-null  datetime64[ns]
1   FONTIP               187438 non-null  category
2   FONTUR               187438 non-null  category
3   FON                  187438 non-null  category
4   FONTOPLAMDEGER       187438 non-null  float64
5   TEDAVPAYSAYISI      187438 non-null  float64
6   KISISAYISI           187438 non-null  int64
7   FONFIYAT             187438 non-null  float64
8   FAİZ                 187438 non-null  float64
9   DOLARFIYAT           187438 non-null  float64
10  ALTINFIYAT           187438 non-null  float64
11  BB                   187438 non-null  float64
12  DT                   185256 non-null  float64
13  DB                   173992 non-null  float64
14  DÖT                  148075 non-null  float64
15  EUT                  113791 non-null  float64
16  FB                   84322 non-null   float64
17  FKB                  55378 non-null   float64
18  GAS                  34803 non-null   float64
19  HB                   21222 non-null   float64
20  HS                   11863 non-null   float64
21  KBA                   6322 non-null    float64
22  KKS                   2693 non-null    float64
23  KH                    1216 non-null    float64
24  KM                     548 non-null     float64
25  OSKS                  211 non-null     float64
26  OST                    11 non-null      float64
dtypes: category(3), datetime64[ns](1), float64(22), int64(1)
memory usage: 35.1 MB
```

Şekil 4.2: Veri setinin ön işlem yapılmış formatı

Veri setimiz gereksiz kolon değişkenlerinden temizlenerek, mevcut değişken kolon sayısı 27 olacaktır. Bir sonraki adımda şekil 4.3’te veri setinde bulunan null değerleri üzerinden durulacaktır.

Şekil 4.3: Veri setinin ilk 5 satırlık formatı

	TARİH	FONTİP	FONTUR	FON	FONTOPLAMDEGER	TEDAVPAYSAYISI	KISISAYISI	FONFIYAT	FAİZ	DOLARFIYAT	...	FKB	GAS	HB	HS	KBA	KKS	KH	KM	OSKS	OST
0	2019-01-02	BORSA YATIRIM FONU	Altın Fonu	FGA	8.722176e+07	4.350000e+06	0	20.050979	0.2302	5.2905	...	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
1	2019-01-02	BORSA YATIRIM FONU	Gümüş Fonu	FGS	1.238294e+07	7.000000e+05	0	17.689916	0.2302	5.2905	...	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
2	2019-01-02	BORSA YATIRIM FONU	Hisse Senedi Fonu	DJA	1.491130e+07	4.800000e+05	0	31.065216	0.2302	5.2905	...	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
3	2019-01-02	EMEKLİLİK FONU	Altın Fonu	AEA	1.465104e+09	6.335717e+10	325392	0.023125	0.2302	5.2905	...	0.0	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
4	2019-01-02	EMEKLİLİK FONU	Altın Fonu	AEL	1.784293e+09	7.415738e+10	325460	0.024061	0.2302	5.2905	...	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN

5 rows x 27 columns

İlk 5 veri bilgisi verilmiş olan veri seti incelendiğinde, verilerde NaN değerlerin olduğu görülmektedir. Bu bağlamda veri setinin genelinde NaN ya da null değer var olup olmadığını aşağıdaki şekil 4.4 sorgulanmıştır.

TARİH	0
FONTİP	0
FONTUR	0
FON	0
FONTOPLAMDEGER	0
TEDAVPAYSAYISI	0
KISISAYISI	0
FONFIYAT	0
FAİZ	0
DOLARFIYAT	0
ALTINFIYAT	0
BB	0
DT	2182
DB	13446
DÖT	39363
EUT	73647
FB	103116
FKB	132060
GAS	152635
HB	166216
HS	175575
KBA	181116
KKS	184745
KH	186222
KM	186890
OSKS	187227
OST	187427
dtype:	int64

Şekil 4.4: Veri setinde toplam null değer sayısı

Yukarıdaki şekil 4.4'te verilen bilgi, her değişken karşısında null olan toplam değer sayısı verilmiştir. Veri ön işlem sürecinde dikkat edilmesi gereken bir diğer önemli konu ise, veri içinde

bütünlüğü bozan null değerlerinin olup olmaması durumudur. Veri setimizde olan null değerleri, şekil 4.4'te görüldüğü üzere sadece menkul oranları bilgisinde bulunmaktadır. Bunun nedeni veri tabanında menkul oranları tutulurken, bulunan menkul oranlarında değeri olmayan menkul için değişkene null değeri atılmıştır. Menkul oranları hesaplanırken yüzdelik üzerinden 0 ile 100 arasında bir değer aldığından dolayı, null olan değerleri 0 olacak şekilde dönüştürülmesi veri yapısını bozmayacağı görülmüştür.

Şekil 4.5: Sayısal değerlerin ortalama bilgileri

	count	mean	std	min	25%	50%	75%	max
FONTOPLAMDEGER	187438.0	2.394037e+08	5.509213e+08	0.000000	6.929778e+06	3.815812e+07	1.786791e+08	7.207539e+09
TEDAVPAYSAYISI	187438.0	6.301665e+09	1.704090e+10	0.000000	3.456783e+07	5.147614e+08	4.347861e+09	2.577214e+11
KISISAYISI	187438.0	4.325684e+04	1.049055e+05	0.000000	5.100000e+01	1.269000e+03	3.306150e+04	1.064438e+06
FONFIYAT	187438.0	3.982973e+00	2.395512e+01	0.000000	1.523725e-02	3.057850e-02	5.684475e-01	3.657033e+02
FAIZ	187438.0	2.034597e-01	4.610195e-02	0.105100	1.630000e-01	2.295000e-01	2.418000e-01	2.550000e-01
DOLARFIYAT	187438.0	5.683442e+00	2.214480e-01	5.203800	5.532600e+00	5.723800e+00	5.813300e+00	6.213800e+00
ALTINFIYAT	187438.0	2.527867e+05	2.080213e+04	216132.634958	2.323975e+05	2.559016e+05	2.715878e+05	2.868514e+05
BB	187438.0	4.071823e+01	3.596555e+01	-90.330000	7.260000e+00	2.794000e+01	7.994000e+01	2.195700e+02
DT	187438.0	2.295568e+01	2.804406e+01	-119.570000	2.960000e+00	9.650000e+00	3.410000e+01	1.074500e+02
DB	187438.0	1.496090e+01	2.223612e+01	-16.150000	7.100000e-01	5.340000e+00	1.806000e+01	1.561000e+02
DÖT	187438.0	8.487016e+00	1.478798e+01	-23.390000	0.000000e+00	1.760000e+00	9.640000e+00	1.000500e+02
EUT	187438.0	5.873882e+00	1.284448e+01	-83.930000	0.000000e+00	1.000000e-02	6.260000e+00	9.918000e+01
FB	187438.0	3.238396e+00	8.946145e+00	-16.570000	0.000000e+00	0.000000e+00	1.790000e+00	9.824000e+01
FKB	187438.0	1.616070e+00	5.469517e+00	-40.620000	0.000000e+00	0.000000e+00	0.000000e+00	9.756000e+01
GAS	187438.0	1.045527e+00	4.623233e+00	-17.890000	0.000000e+00	0.000000e+00	0.000000e+00	9.670000e+01
HB	187438.0	6.632977e-01	3.542419e+00	-15.650000	0.000000e+00	0.000000e+00	0.000000e+00	5.857000e+01
HS	187438.0	3.054617e-01	2.177117e+00	-16.250000	0.000000e+00	0.000000e+00	0.000000e+00	4.899000e+01
KBA	187438.0	1.008381e-01	9.393350e-01	-16.760000	0.000000e+00	0.000000e+00	0.000000e+00	2.425000e+01
KKS	187438.0	2.969931e-02	5.083447e-01	-12.150000	0.000000e+00	0.000000e+00	0.000000e+00	3.756000e+01
KH	187438.0	5.369669e-03	1.304408e-01	0.000000	0.000000e+00	0.000000e+00	0.000000e+00	6.530000e+00
KM	187438.0	2.637352e-03	1.100750e-01	0.000000	0.000000e+00	0.000000e+00	0.000000e+00	6.940000e+00
OSKS	187438.0	5.484480e-05	7.166541e-03	0.000000	0.000000e+00	0.000000e+00	0.000000e+00	1.000000e+00
OST	187438.0	0.000000e+00	0.000000e+00	0.000000	0.000000e+00	0.000000e+00	0.000000e+00	0.000000e+00

Veri kümesineki sayısal değişkenlerin ortalama, standart sapma, minimum, maximum değerleri ve verinin dağılım oranlarının bilgisi şekil 4.5'te verilmiştir. Bu bilgilerden de açıkça görülmektedir ki, OST değişkeninin tüm verilerinin 0 olduğu ve geri kalan diğer menkul oran değişkenlerinin de minimum değerlerinin negatif olduğu bilgisi mevcuttur. Gereksiz veri kapsamında OST değişkenini veri setinden silinme işlemi yapılacaktır. Negatif değerler ise, menkul değişken oranlarının 0 ile 100 arasında değer alınacağı ifade edilmişti. Lakin, Takasbank'tan alınan bilgiye göre müşteri kendi

portföyünü hazırlarken toplam menkul değerinin 100 olması için negatif değerle dengeyi kurmaya çalışmış ve değişken oranlarını bu dengeyi sağlayacağı formatta Takasbank’a bildirilmiştir. Veri setinin orijinal yapısının ve bütünlüğünün bozulmaması için negatif değerlere dair bir işlem yapılmamıştır.

4.1.2.2 Veri Korelasyon Matrisi

Veri ön işleminin bu adımında değişkenler arasındaki bağlantı yönü ve büyüklüğüne dair korelasyon matris ilişkileri incelenecektir. Korelasyon kat sayısı, -1 ile +1 arasında bir değer alır. Buradaki değerinin artı ya da eksi olması ilişkin büyüklüğü hakkında bilgi vermez. Artı değeri alan iki değişkenin, birlikte aynı yönde artıp ya da azaldığını gösterir. Eksi değerinin ise, tam tersi bir ilişkinin iki değişken arasında olduğunu ifade eder. Biri artar iken, diğer değişkenin azaldığını veya tersinin olması durumudur[2]. İlk olarak tüm sayısal değişkenlerin korelasyon matrisi şekil 4.6’te verilerek, matris grafiği hakkında değerlendirmeler yapılmıştır.

	FONTOPLAMDEGER	TEDAVPAYSAYISI	KISISAYISI	FONFIYAT	FAIZ	DOLARFIYAT	ALTINFIYAT	BB	DT	DB	...	FB	FKB	GAS	HB	HS	KBA	KKS	KH	KM	OSKS
FONTOPLAMDEGER	1.000000	0.623976	0.572384	0.240230	-0.046102	0.028570	0.045445	-0.117865	-0.027456	0.005659	...	0.129416	0.106025	0.128855	0.111821	0.121980	0.096114	0.097218	0.121421	0.027718	0.033211
TEDAVPAYSAYISI	0.623976	1.000000	0.775371	-0.060899	-0.008602	0.005140	0.006192	-0.017060	-0.051967	-0.005084	...	0.064690	0.041446	0.053225	0.049651	0.113149	0.052736	0.048886	0.058034	0.011181	0.015839
KISISAYISI	0.572384	0.775371	1.000000	-0.032681	0.005122	-0.004760	-0.006781	-0.015445	-0.041986	-0.012995	...	0.030175	0.040284	0.068447	0.079492	0.128956	0.112067	0.123169	0.161027	0.038945	0.041491
FONFIYAT	0.240230	-0.060899	-0.032681	1.000000	-0.007930	0.004703	0.008218	-0.048665	-0.015605	-0.016653	...	0.083380	0.091493	0.065059	0.019361	-0.016518	-0.012317	-0.009550	-0.006741	-0.003937	-0.001252
FAIZ	-0.046102	-0.008602	0.005122	-0.007930	1.000000	-0.245448	-0.765568	-0.033752	0.036919	0.018948	...	-0.005592	0.011795	0.001221	0.007406	0.003255	0.006380	0.001681	0.002117	-0.015502	-0.003672
DOLARFIYAT	0.028570	0.005140	-0.004760	0.004703	-0.245448	1.000000	0.637137	0.013332	-0.022635	0.001280	...	0.006523	-0.022790	-0.002092	-0.002664	-0.009928	-0.010439	-0.003190	0.010967	0.006252	-0.002566
ALTINFIYAT	0.045445	0.006192	-0.006781	0.008218	-0.765568	0.637137	1.000000	0.029033	-0.046377	-0.010399	...	0.014475	-0.017570	-0.002947	-0.005810	-0.001857	-0.008426	0.002367	0.004487	0.014997	0.002182
BB	-0.117865	-0.017060	-0.015445	-0.048665	-0.033752	0.013332	0.029033	1.000000	-0.486768	-0.417391	...	-0.230900	-0.159207	-0.134248	-0.126064	-0.097665	-0.082703	-0.050781	-0.043330	-0.024688	-0.008640
DT	-0.027456	-0.051967	-0.041986	-0.015605	0.036919	-0.022635	-0.046377	-0.486768	1.000000	-0.181136	...	-0.137992	-0.119830	-0.080377	-0.057854	-0.053092	-0.039780	-0.005085	0.022366	0.015303	0.011778
DB	0.005659	-0.005084	-0.012995	-0.016653	0.018948	0.001280	-0.010399	-0.417391	-0.181136	1.000000	...	-0.046250	-0.059735	-0.033105	-0.034325	-0.035612	-0.025799	-0.019604	-0.012832	-0.011097	-0.005074
DÖT	0.041470	0.005734	-0.003340	0.045750	-0.011628	0.001866	0.014714	-0.330324	-0.140319	-0.043228	...	0.072753	0.063816	0.019115	0.032271	0.008000	0.036889	0.006667	-0.005970	0.002188	-0.003420
EUT	0.087064	0.042417	0.039501	0.024862	-0.010221	0.017357	0.022112	-0.297759	-0.141279	-0.027267	...	0.090035	0.059019	0.042512	0.049625	0.022105	0.024676	0.016664	0.020945	-0.001334	-0.001435
FB	0.129416	0.064690	0.030175	0.083380	-0.005592	0.006523	0.014475	-0.230900	-0.137992	-0.046250	...	1.000000	0.172609	0.082908	0.085865	0.143654	0.079199	0.009244	-0.003426	0.000199	0.003417
FKB	0.106025	0.041446	0.040284	0.091493	0.011795	-0.022790	-0.017570	-0.159207	-0.119830	-0.059735	...	0.172609	1.000000	0.166006	0.124785	0.141729	0.156235	0.064302	0.007911	0.002714	-0.001466
GAS	0.128855	0.053225	0.068447	0.065059	0.001221	-0.002092	-0.002947	-0.134248	-0.080377	-0.033105	...	0.082908	0.166006	1.000000	0.096039	0.106244	0.090018	0.089072	0.057383	0.067329	-0.000141
HB	0.111821	0.049651	0.079492	0.019361	0.007406	-0.002664	-0.005810	-0.126064	-0.057854	-0.034325	...	0.085865	0.124785	0.096039	1.000000	0.110060	0.090707	0.068995	0.048971	0.009890	-0.000064
HS	0.121980	0.113149	0.128956	-0.016518	0.003255	-0.009928	-0.001857	-0.097665	-0.053092	-0.035612	...	0.143654	0.141729	0.106244	0.110060	1.000000	0.202821	0.130826	0.096938	0.022058	0.014396
KBA	0.096114	0.052736	0.112067	-0.012317	0.006380	-0.010439	-0.008426	-0.082703	-0.039780	-0.025799	...	0.079199	0.156235	0.090707	0.090707	0.202821	1.000000	0.185762	0.123108	0.078569	0.047130
KKS	0.097218	0.048886	0.123169	-0.009550	0.001681	-0.003190	0.002367	-0.050781	-0.005085	-0.019604	...	0.009244	0.064302	0.089072	0.068995	0.130826	0.185762	1.000000	0.174660	0.096591	0.055314
KH	0.121421	0.058034	0.161027	-0.006741	0.002117	0.010967	0.004487	-0.043330	0.022366	-0.012832	...	-0.003426	0.007911	0.057383	0.048971	0.096938	0.123108	0.174660	1.000000	0.344127	0.061897
KM	0.027718	0.011181	0.038945	-0.003937	-0.015502	0.006252	0.014997	-0.024688	0.015303	-0.011097	...	0.000199	0.002714	0.067329	0.009890	0.022058	0.078569	0.096591	0.344127	1.000000	0.196173
OSKS	0.033211	0.015839	0.041491	-0.001252	-0.003672	-0.002566	0.002182	-0.008640	0.011778	-0.005074	...	0.003417	-0.001466	-0.000141	-0.000064	0.014396	0.047130	0.055314	0.061897	0.196173	1.000000

22 rows x 22 columns

Şekil 4.6: Korelasyon matrisi

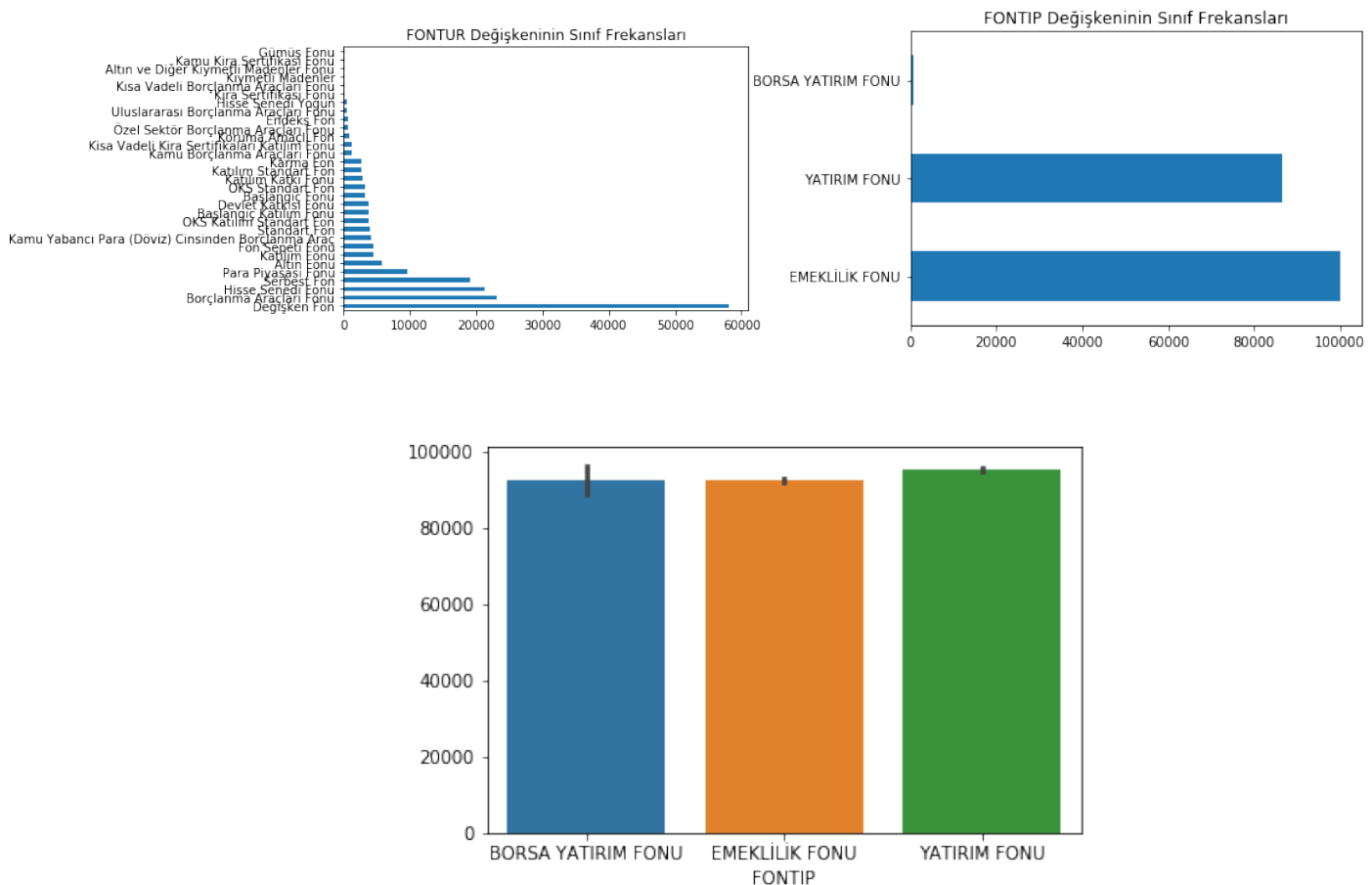
Korelasyon matrisine şekil 4.6 bakıldığı vakit, değişkenlerin aralarındaki ilişkin yön ve büyüklük değerlerini görmekteyiz. Genel olarak incelendiğinde, 22 satıra ve 22 kolon matrisinde değerler arasında ciddi bir tutarsızlığın görülmediği ve bazı değişkenler arasında korelasyon katsınının daha iyi olduğu gözlemlenmektedir. Bu değişkenlerden, FONTOPLAMDEGER - TEDAVPAYSAYISI - KISISAYISI arasındaki ilişkin diğer değişkenlere oranla daha güçlü olduğu görülmektedir. Veri seti

özellikleri bölümünde veriler hakkındaki açıklamalarda, bu güçlü ilişkinin nedeni desteklemektedir. Bu değişkenlerin bağımsız değişken olarak seçilen FONFIYAT değerine olan katkıları da diğer değişkenlere oranla daha iyi olduğu görülür ve korelasyon matrisinde menkul oran değişkenleri katsayıları da genel olarak normal gözlenmektedir. FONFIYAT değişkenin FAİZ, DOLARFIYAT ve ALTINFIYAT ile doğrudan korelasyon ilişkisin zayıf olduğu tespit edilmektedir. Ama ALTINFIYAT değişkenin, FAİZ ile arasında ters yönde ve DOLARFIYAT ile de aynı yönde güçlü bir ilişki olduğu incelenmiştir.

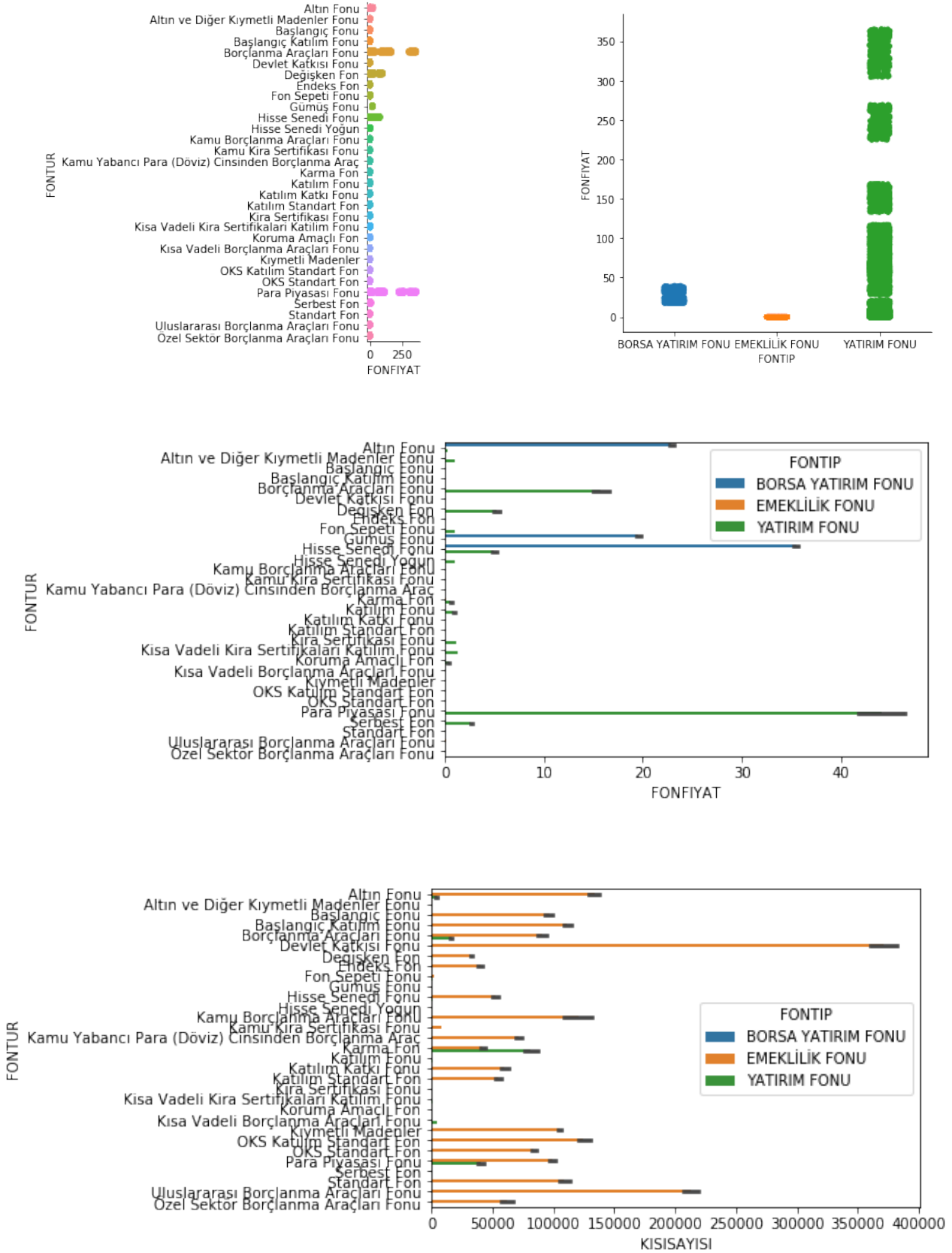
4.1.2.3 Veri Görselleştirme

Bu bölümde veri setindeki kategorik ve sayısal değişkenler için görselleştirme çalışmaları yapılacaktır. Veri ön işlemenin bir önceki adımlarından verinin ideal bir veri setine dönüştürülme süreçleri yapılmış ve bu adımda ise bazı değişkenler üzerinden temel bazı grafiksel işlemleri yapılarak bölümü bitirmiş olacağız.

Şekil 4.7: Kategorik değişken dağılım grafikleri

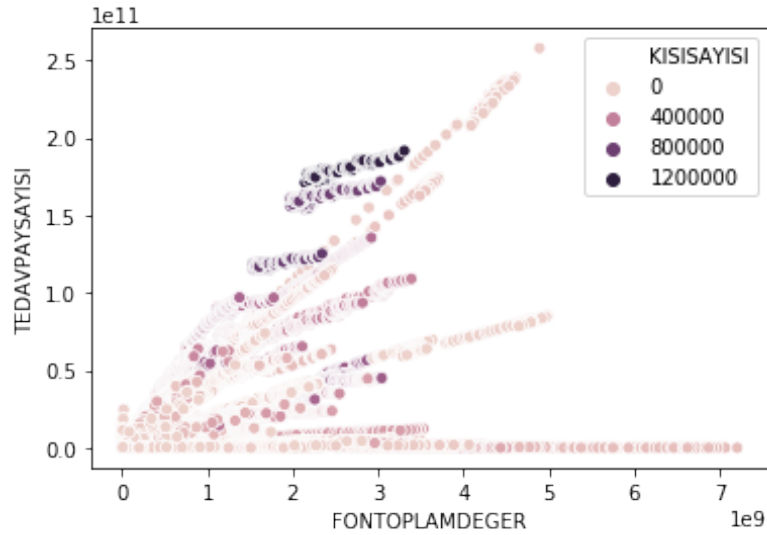


Şekil 4.8: Fon fiyat ve kişi sayısının kategorik değişken grafikleri

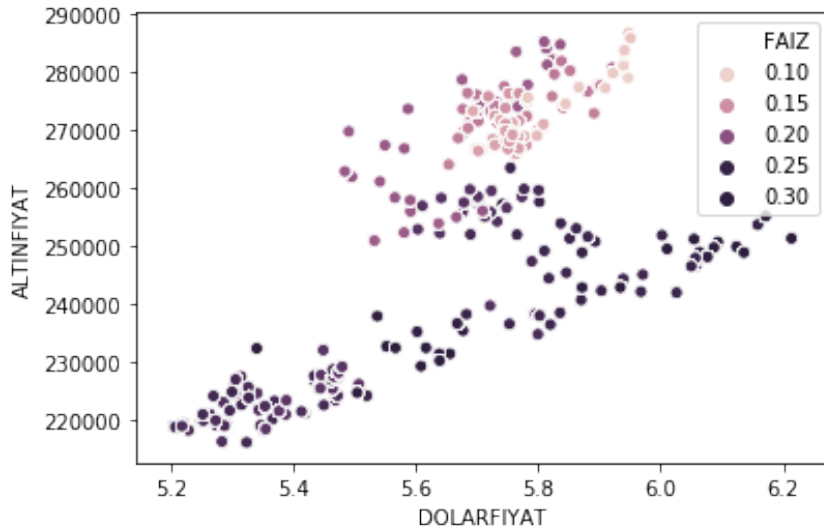


Veri göselleştirme amaçımız gereği, belli değişkenlerin veri ile olan ilişkilerini daha anlaşılır olmasını sağlamak için grafikleştirme çalışmaları yapıldı ve bu çalışmalardan, şekil 4.7 ve 4.8’de kategorik değişkenlerin dağılımı ve bağımsız değişken ile olan bağlantılarının görselleştirme işlemleri yapılmıştır. Son olarak yapılan şekil 4.9 ve 4.10’daki grafiklerde, Toplam-Pay-Kisi(FONTOPLAMDEGER-TEDAVPAYSAYISI-KISISAYISI) ve veri setimizde mevcut bulunan 2019 yılına ait Faiz-Altın-Dolar(FAIZ-ALTINFIYAT-DOLARFIYAT) ilişkisinin korelasyon grafiği verilmiştir. Bir önceki bölüm olan veri korelasyon matris değerleri incelenirken altın değişkenin, dolar ile pozitif ama faiz değişkeni ile de negatif bir ilişkisinin olduğu ifade edilmişti, bulunan şekil 4.10’daki grafik bu yorumu desteklemektedir.

Şekil 4.9: Toplam-Pay-Kisi sayısı korelasyon grafiği



Şekil 4.10: 2019 Altın-Dolar-Faiz korelasyon grafiği



4.2. Model Ölçüm Metrikleri

Fon fiyatları tahmini için model geliştirirken, yapılan uygulama sonuçlarının başarı değerlendirmesini ölçecek metriklere ihtiyaç duyulmaktadır. Bu bölümde geliştireceğimiz model çalışmaları regresyon problemlerine dahil olduğundan regresyon model değerlendirme metrikleri kullanılacaktır. Model için kullanılacak 4 algoritmanın sonuç değerlendirmesi aynı ölçüm metriklerine göre değerlendirileceğinden, bu başlık altında genel olarak kullanılacak olan metotlardan bahsedilecektir. Literatür bölümünde, makine öğrenme algoritmaların değerlendirme sonuçları ile ilgil yapılan çalışmalar incelendiğinde genel olarak çok sık kullanılan ölçüm metrikleri;

1. Hata Karelerin Ortalaması - Mean Squared Error(MSE)

Hata kare ortalaması, regresyon problemlerinde tahmin eğrisinin gerçek değer noktalarına ne kadar yakın olduğunu belirtir. Matematiksel gösterimi;

$$MSE = \frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2 \quad (4.1)$$

4.1, 4.2 ve 4.3'te ki denklem değişkenleri;

Gözlem sayısı: $\frac{1}{n}$

Gerçek değerler: y_i

Tahmin edilen değerler: \hat{y}_i

MSE değeri, veri setimizdeki bağımsız değişkenlerin tahmin ettiği değer ile gerçek değerlerin farkının karesinin ortalaması alınarak bulunur. Bulunan değer, birim başına düşen hata payı olarak değerlendirilir. MSE değerinin sıfıra yakın bulunması başarı ölçütü olarak daha iyi bir performans gösterdiği ifade edilebilir[3,4].

2. Hata Kare Ortalamasının Karekökü - Root Mean Square Error(RMSE)

Hata karelerinin ortalamasının karekökü, MSE değerinin karekökü alınarak bulunur. RMSE değeri tahmin hatalarının standart sapması olarakta ifade edilebilir. Ayrıca RMSE'nin MSE metoduna göre

daha avantajlı olan özeliği, bazı durumlarda büyük hataları daha fazla cezalandırma işlevine sahip olduğu belirtilmektedir[3,4]. RMSE matematiksel denklemi;

$$RMSE = \sqrt{\frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2} \quad (4.2)$$

3. Ortalama Mutlak Hata - Mean Absolute Error(MAE)

Ortalama mutlak hata, gerçek değerden tahmin edilen değer farkının mutlak ölçümü alınarak bulunur. Hesaplanan MAE değerleri daha kolay yorumlanabilir oldukları için regresyon ve zaman serisi gibi problemlerde daha çok kullanılmaktadır[4]. Matematiksel olarak gösterimi;

$$MAE = \frac{1}{n} \sum_{i=1}^n |y_i - \hat{y}_i| \quad (4.3)$$

Kısaca 3 madde halinde tanımlamaları ve matematiksel denklemleri verilen yukarıdaki model ölçüm metrikleri, bu tez çalışmasında önerilen model geliştirme sonuçlarını değerlendirme ölçütünde referans alınarak işlemler yapılacaktır.

4.3. Doğrusal Çoklu Regresyon Algoritmaları ile Model Geliştirme

Model geliştirme çalışmalarının ilk adımı olan doğrusal çoklu regresyonlar ile tahmin işleminde; Kısmi En Küçük Kareler Regresyon (PLSR) ve Ridge Regresyon (RR) algoritmalarıyla uygulamalar yapılacaktır. Bölüm 3’de teorik kapsamlarından bahsedilen PLSR ve RR algoritmaların her birinin kendi bölüm başlığı altında sözde kod(pseudocode) yapılarından, model geliştirme ve optimizasyon adımlarından bahsedilecektir.

4.3.1 Kısmi En Küçük Kareler Regresyonu (PLSR)

Bu kısımda ilk olarak, PLSR sözde kodu(pseudocode) yapısından bahsedilecektir. Bölüm 3’de ayrıntılı bir şekilde PLSR matematiksel denklem yapısı ve algoritma olarak kullanacağımız NIPALS adımları anlatılmıştı, bu bölümde ise sözde kodu şekil 4.11’de verilerek algoritmanın temel çalışma yapısı gösterilmiştir.

Şekil 4.11: NIPALS algoritma sözde kodu[5]

```
Algoritma 1: NIPALS


---


Input:  $E_0 = X$  Bağımsız değişken değerleri
Output:  $P = [p_1, \dots, p_H], T = [t_1, \dots, t_H]$ 
  for all  $h = 1, \dots, H$  do
    Step 0: Initialize  $t_h$ 
    Step 1:
      repeat
        Step 1.1:  $p_h = E_{h-1}' t_h / (t_h' t_h)$ 
        Step 1.2:  $p_h = p_h / \|p_h\|$ 
        Step 1.3:  $t_h = E_{h-1} p_h / (p_h' p_h)$ 
      until convergence of  $p_h$  Sistem dengeye ulaşana kadar;
    Step 2:  $E_h = E_{h-1} - t_h p_h'$ 
  end for
```

Şekil 4.11'deki sözde kodu ana özelliği, vektör çiftleri arasındaki skaler değerler aracılığıyla çalışmasıdır. Geometrik açıdan, bu skaler değerler En Küçük Kareler Yöntemi (Ordinary Least Squares (OLS)) regresyon çizgilerinin eğimleri olarak yorumlanabilir. Özellikle, her bir t_{ih} ve t_h değeri, en küçük kareler çizgisinin kesişme noktası($p_h; e_i$) olmayan değer kümesinden geçerek eğimi, e_i 'nin i t_h satırında E_h aktarımı olur. Benzer şekilde, p_h değerinde keşime noktası($t_h; e_p$) olmayan veri kümesinden eğimi, e_p 'nin p t_h kolunda E değeri aktarımı yapılır. Geometrik olarak yapılan bu ikili aktarım işlemi, eksik verileri her regresyon satırında yerine yerleştirerek algoritma çalışma akışını tamamlar[5]. Daha sonra uygulama için, model geliştirme ve model optimizasyon adımları incelenecektir. Model geliştirme alt başlığında, veri setimizin bağımsız değişkenlerden bağımlı değişken olan fiyat parametresinin tahmini için ilk uygulama adımları incelenecek ve bir sonraki aşama olarak ise, geliştirilmiş olan model üzerinden optimizasyon işlem adımlarından bahsedilerek, PLSR ile model geliştirme bölümünü bitirmiş olacağız.

4.3.1.1 Model geliştirme

Kısmi En Küçük Kareler Regresyonu (PLSR) ile model geliştirme bölümünde, model kurma aşamalarını adımlar halinde örnek kod blokları ve elde edilen sonuçları verilerek detaylı bir şekilde anlatımı yapılacaktır. Toplam 3 adımda model geliştirme çalışması gerçekleştirilecektir.

1.Adım : Veri setinde bağımlı ve bağımsız değişken ile eğitim ve test seti ayrımı;

```
X_p = df_pls.drop(['TARİH', 'FONTIP', 'FONTUR', 'FON', 'FONFIYAT'], axis = 1)
y_p = df_pls["FONFIYAT"]

X_train_p, X_test_p, y_train_p, y_test_p = train_test_split(X_p, y_p, test_size=0.25, random_state=42)

X_p.head()
```

	FONTOPLAMDEGER	TEDAVPAYSAYISI	KISISAYISI	FAIZ	DOLARFIYAT	ALTINFIYAT	BB	DT	DB	DÖT	...	FB	FKB	GAS	HB	HS	KBA	KKS	KH	KM	OSKS
0	8.722176e+07	4.350000e+06	0	0.2302	5.2905	219270.871675	100.00	0.00	0.00	0.00	...	0.00	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
1	1.238294e+07	7.000000e+05	0	0.2302	5.2905	219270.871675	100.00	0.00	0.00	0.00	...	0.00	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
2	1.491130e+07	4.800000e+05	0	0.2302	5.2905	219270.871675	99.83	0.17	0.00	0.00	...	0.00	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
3	1.465104e+09	6.335717e+10	325392	0.2302	5.2905	219270.871675	1.01	3.44	85.25	3.07	...	7.23	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
4	1.784293e+09	7.415738e+10	325460	0.2302	5.2905	219270.871675	0.25	99.75	0.00	0.00	...	0.00	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0

5 rows x 21 columns

Şekil 4.12: Bağımlı-bağımsız değişken ve eğitim-test seti ayrımı

Veri kümesi üzerinden ilk olarak yapılacak olan bağımlı, bağımsız değişken ve veri setinin eğitim, test olarak ayrımı, 4 algoritmanın model geliştirme aşamasının ilk adımı olup ve 4 algoritmanın da ortak adımları olacağı için sadece burada detaylı anlatımı yapılacak, diğer 3 algoritmanın model geliştirme aşamasında anlatımı yapılmayacaktır. Bu bağlamdan hareketle veri üzerinden, bağımlı değişken fon fiyat bilgisi y ile bağımsız değişkenler ise X parametresine değer ataması yapılarak, tahmin işlemi için girdi ve çıktı verilerinin ayrıştırma süreçleri gerçekleştirilir. Daha sonra ayrımı yapılmış değişkenler üzerinden, veri setinin eğitim ve test olarak bölünmesi gerekir. Eğitim seti ile ilk model kurma aşamasında öğrenme süreci gerçekleştirilir. Öğrenme süreci yapılmış modelin, öğrenme başarı performansını ölçmek için ayrılmış olan test verisi ile öğrenme durumu değerlendirilir. Uygulama bölümü için önemli olan bu ayrıştırma işlemini, veri setimiz genelinde %75 eğitim ve %25 test verisi oluşturulacak formatta yapılmış. Bölünmüş olan verinin, bağımlı(y_train, y_test) ve bağımsız(X_train, X_test) değişkenler bazında olacak şekilde eğitim ve test olarak aktarım işlemleri yapılmıştır.

2.Adım: Eğitim seti ile model kurma;

Bu adımda, X_train bağımsız değişkenler ile y_train bağımlı değişken üzerinden ilk kurulan PLSR modeli eğitilir. Bu eğitilen modelin, bağımsız değişken girdilerinin katsayıları ve model parametre yapıları şekil 4.12’de gösterilmektedir.

```

pls_model

PLSRegression(copy=True, max_iter=500, n_components=2, scale=True, tol=1e-06)

pls_model.coef_

array([[ 8.84294439],
       [-4.69831641],
       [-3.03243924],
       [ 0.0337065 ],
       [-0.09314651],
       [-0.02433704],
       [-0.33800515],
       [-0.31991584],
       [-0.81996284],
       [ 0.91092399],
       [-0.09762054],
       [ 1.74953948],
       [ 2.26792273],
       [ 1.19959689],
       [-0.41431445],
       [-1.57986807],
       [-1.26811436],
       [-0.8361369 ],
       [-0.61910158],
       [-0.24914284],
       [-0.09490851]])

```

Şekil 4.12: PLSR Model yapısı ve bağımsız değişken katsayıları

İlk satırda pls_model ile NIPALS algoritmasının parametreleri görülmekte, içerik olarak maximum iterasyon ve bileşen sayıların default değerleri mevcut ve bu parametre değişiklikleri optimizasyon bölümünde yapılacaktır. İkinci satırda modelin bağımsız değişken katsayıları verilmiştir. Burada öğrenme modeli için hesaplanan çoklu doğrusal bir fonksiyon denklemin, birden fazla olan bağımsız değişkenlerin katsayı değerleri bulunmuştur.

3.Adım: Model eğitim ve test seti tahmin bilgileri;

Tablo 4.5: PLSR ölçüm metrik sonuçları

Ölçüm Metrikleri	Eğitim	Test
MSE	497.8090200774829	506.0252001342653
RMSE	22.31163418661849	22.495003892737277
MAE	7.099829131737095	7.119836719007614
R2	0.1273657565331895	0.13348626833437138

Yukarıdaki 4.5'teki tablodan görüleceği üzere, ölçüm metrik sonuçları eğitim ve test verisine göre elde edilmiştir. Ayrıca, modelin mevcut default değerleri üzerinden herhangi bir optimizasyon işlemi yapılmadan bu sonuçlara ulaşılmıştır. Tablodaki sonuçlardan, eğitim ve test değerleri

arasında küçük farklılıkların olduğu görülmektedir. Her ne kadar eğitim seti üzerinden alınan bazı sonuçlar daha iyi olduğu görülsede, model için daha doğru bir değerlendirme referansı test verisinden elde edilen sonuçlar olmuştur.

4.3.1.2 Model optimizasyonu

Kısmi En Küçük Kareler Regresyonu (PLSR) kullanılarak elde edilen tahmin sonuçlarında model optimizasyon parametreleri için yapılan literatür ve örnek çalışmalar incelendiğinde[5], PLSR temel çalışma yapısından hareketle, bağımsız değişkenlerin daha az sayıda ve aralarında çoklu bağlantı problemi olmayan bileşenlere indirgeyip model kurma fikrine dayanıyor olaması, bileşen sayısını optimizasyon konusu yapmıştır. Model doğrulama ve optimum parametre için Cross-validation metodu kullanılacaktır. Cross-validation, makine öğrenmenin veriler üzerinde doğru ve objektif bir öğrenme süreci için yaygın olarak kullanılan, model seçimi ve performans değerlendirmede tercih edilen basit ve etkili bir yöntemdir[6]. Burada, cross-validation yöntemlerinden K-Fold yöntemi kullanılacak ve optimizasyon çalışması toplam 2 adımdan oluşacaktır.

1.Adım: Optimum bileşen sayısı bulmak;

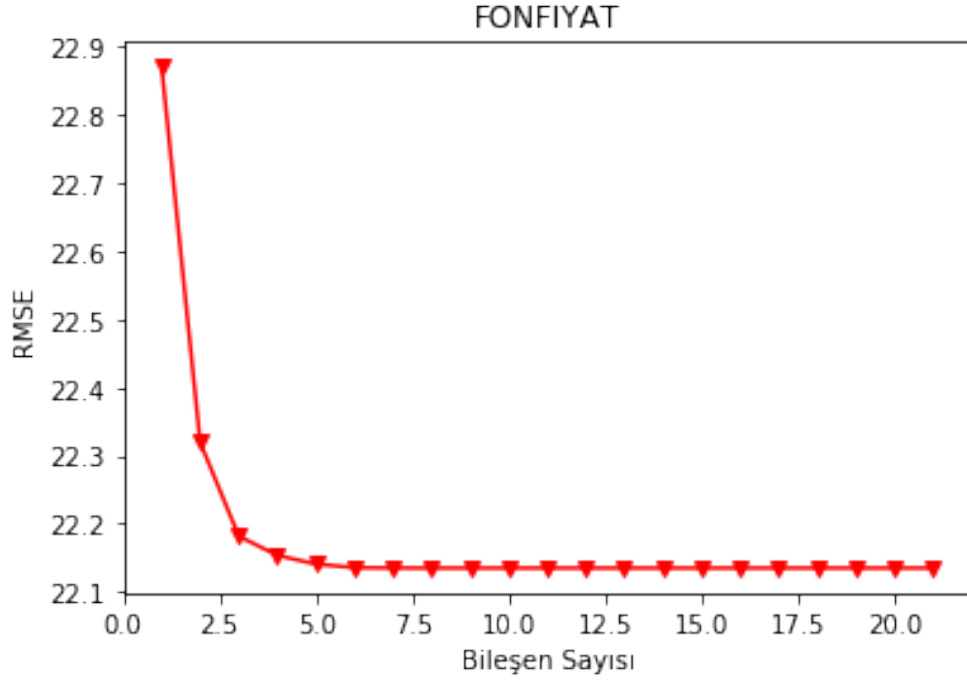
```
#CV
cv_10 = model_selection.KFold(n_splits=10, shuffle=True, random_state=1)

#Hata hesaplamak için döngü
RMSE = []

for i in np.arange(1, X_train_p.shape[1] + 1):
    pls = PLSRegression(n_components=i)
    score = np.sqrt(-1*cross_val_score(pls, X_train_p, y_train_p, cv=cv_10, scoring='neg_mean_squared_error').mean())
    RMSE.append(score)

#Sonuçların Görselleştirilmesi
plt.plot(np.arange(1, X_train_p.shape[1] + 1), np.array(RMSE), '-v', c = "r")
plt.xlabel('Bileşen Sayısı')
plt.ylabel('RMSE')
plt.title('FONFIYAT');
```

Şekil 4.13: PLSR optimum bileşen sayısı kod bloğu



Şekil 4.14: PLSR optimum bileşen sayısı

Şekil 4.13'teki kod bloğundan görüleceği üzere, k-flod yönteminde veri üzerinde işlem yapılırken veriyi bölme sayısı(n_splits) parametresinin secimi genel olarak yapılan çalışmalarda, kullanılan değer aralığı 5 ile 10 arasında olmaktadır[6]. Kendi çalışmamız için yapılan deneysel sonuçlarda ise, en optimum değer 10 olarak bulunmuştur. Kod bloğun çıktısı olan şekil 4.14'te, bileşen sayısı 5 değerinin bir kırılma noktası olduğu ve sonrası değerler için sabit RMSE oranları elde edildiği görülmektedir. Bu bağlamda, tercih edilen en optimum bileşen sayısı 6 olarak seçilmiştir.

2.Adım: Doğrulanmış model tahmin sonuçları;

Tablo 4.6: PLSR ölçüm metrik sonuçları - 2

Ölçüm Metrikleri	
MSE	497.0027614702058
RMSE	22.29355874395575
MAE	6.705839394559914
R2	0.14893622417341656

Tablo 4.6 ‘da optimizasyon işlemleri yapıldıktan sonra doğrulanmış model verilerinin sonuçlarına bakıldığında, fon fiyat tahmini ile gerçek değer arasındaki farkın açıklanma metriği RMSE değerini baz alarak değerlendirdiğimizde, fon fiyat tahmini yaptığımızda birim başına düşen hata payı oranının artı eksi 22.29 olarak bir sapma değerinin olduğu görülmektedir. Yüzdelik olarak tahmin başarısının ise, %77-78 civarında olduğu saptanmaktadır.

4.3.2 Ridge Regresyonu (RR)

Bu bölümde Ridge Regresyonu (RR) ile model geliştirme çalışmaları yapılacak, daha önceki bölüm 3’de ayrıntılı olarak RR algoritması hakkında bilgilendirilme yapılmıştı. Burada ise, sözde kodu(pseudocode) yapısı ve iki alt başlık olarak; model geliştirme ve optimizasyon konuları incelencektir.

4.3.2.1 Model geliştirme

Bu kısımda yapılan model çalışması, PLSR model geliştirme aşamalarının benzeri olacağından dolayı tekrar olmaması için aynı adımlarda ayrıntılı bir anlatım yapılmayacaktır. Ayrıca, uygulamanın birinci adım olan veri kümesinin test ve eğitim seti olarak ayrılma aşaması ortak olduğu için bu adım atlanarak, 2’inci adımdan başlayarak model çalışmamız gerçekleştirilmiş olacaktır.

2.Adım: Eğitim seti ile model kurma;

```
ridge_model  
  
Ridge(alpha=0.1, copy_X=True, fit_intercept=True, max_iter=None,  
      normalize=False, random_state=None, solver='auto', tol=0.001)  
  
ridge_model.coef_  
  
array([ 1.99548506e-08, -4.36501444e-10, -1.01864979e-05,  5.33789312e+00,  
       -6.33703278e-01,  2.45254214e-06, -1.37536605e-03, -1.26321075e-02,  
       -2.43176494e-02,  3.68163153e-02, -1.56943704e-02,  1.05959126e-01,  
        2.52256301e-01,  1.04918931e-01, -1.40978035e-01, -3.76084541e-01,  
       -8.49985257e-01, -1.04985067e+00, -4.80938207e+00,  5.91301679e-01,  
       -1.93941104e+01])
```

Şekil 4.15: RR Model yapısı ve bağımsız değişken katsayıları

Şekil 4.15'teki modelin birinci satırında model yapısı ve parametreleri mevcut iken, ikinci satırda tahmin işlemi yapılacak olan fonksiyon değişken katsayıları bulunmaktadır. Burada, model parametresi olan alpha(lambda) değişkeni, tahmin fonksiyonunda ceza teriminin katsayı değeridir. Deneysel çalışmalar için manuel 0.1 değeri verilmiştir. Diğer model parametreleri default değerlerdir. Alpha(lambda) değişkeninin, model katsayıları değerlerine olan etkisini deneysel olarak göstermek için şekil 4.16 ve 4.17'deki çalışmalar yapılmıştır.

```
lambdalar = 10**np.linspace(10,-2,100)*0.5

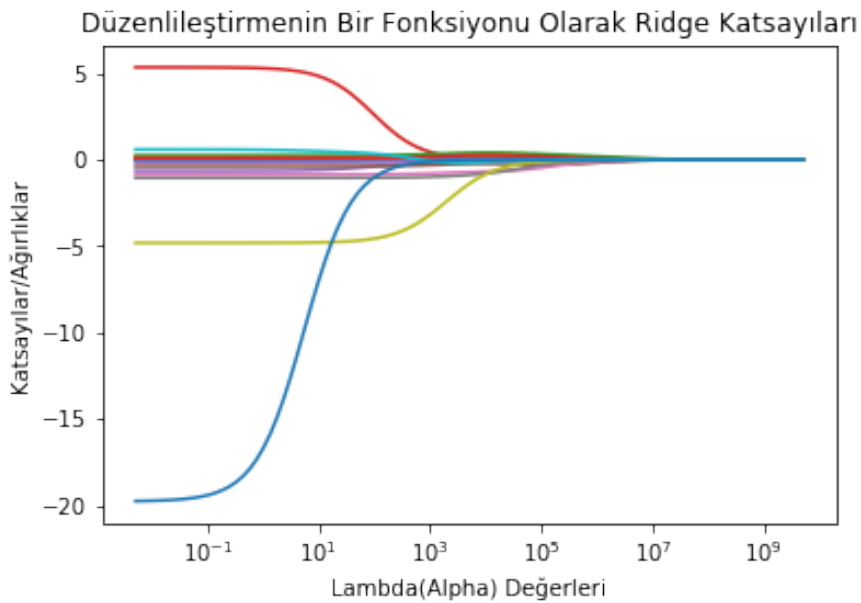
ridge_model = Ridge()
katsayilar = []

for i in lambdalar:
    ridge_model.set_params(alpha = i)
    ridge_model.fit(X_train_r, y_train_r)
    katsayilar.append(ridge_model.coef_)

ax = plt.gca()
ax.plot(lambdalar, katsayilar)
ax.set_xscale('log')

plt.xlabel('Lambda(Alpha) Değerleri')
plt.ylabel('Katsayılar/Ağırlıklar')
plt.title('Düzenlileştirimin Bir Fonksiyonu Olarak Ridge Katsayıları');
```

Şekil 4.16: RR lambda(alpha) örnek kod bloğu



Şekil 4.17: RR lambda(alpha) örnek grafiği

Şekil 4.17'deki grafik çıktısına bakıldığında her bir renk, bir değişkenin katsayı değerini ifade etmekte ve lambda(alpha) değeri değiştikçe katsayılar/ağırlıklar değerlerinde değişim olduğu görülmektedir. Deneysel çalışma için yapılmış olan şekil 4.16'daki kod bloğu ve sonuç grafiğinden anlaşılabacağı üzere, alpha parametresinin alacağı değerin katsayılar üzerindeki etkisi gözlemlenmektedir.

3. Adım: Model eğitim ve test seti tahmin bilgileri;

Tablo 4.7: RR ölçüm metrik sonuçları - 1

Ölçüm Metrikleri	Eğitim	Test
MSE	489.71681475048916	496.9772456519872
RMSE	22.12954619395728	22.292986467765758
MAE	6.685656583569678	6.706254687410646
R2	0.1415509865886838	0.14897991726786186

Tablo 4.7 'de model ölçüm metrik sonuçlarına bakıldığı vakit, PLSR test ve eğitim sonuçlarına göre daha iyi olduğu görülmektedir. Ayrıca, eğitim ve test değerleri arasında küçük farklılıkların olduğu görülmektedir. Bazı ölçüm metriklerinin eğitim seti üzerinden alınan sonuçları, test verisinden daha iyi olduğu görülsede, model için daha doğrulanmış sonuçlar test verisinden elde edilen sonuçlar baz alınır.

4.3.2.2 Model optimizasyonu

Ridge Regresyonu (RR) model parametre optimizasyon kısmında alpha(lambda) değişkeninin optimize edilmesi ile ilgili çalışmalar yapılacaktır. literatür bölümünde yapılan araştırmalar ve model geliştirme bölümündeki birinci adımda gerçekleştirilen deneysel çalışmalardan anlaşılabacağı üzere, model tahmin fonksiyonu üzerinde çok önemli bir etkisi olan ceza teriminin katsayısı alpha değişkeni optimizasyon konusu olacaktır[6,7]. Optimum değer bulmak için Cross-validation metodu kullanılacak ve yapılacak işlem sayısı toplam 2 adımdan oluşacaktır.

1. Adım: Optimum alpha(lambda) değerin bulunması;

```
lambdalar = 10**np.linspace(10,-2,100)*0.5

lambdalar[0:5]

array([5.00000000e+09, 3.78231664e+09, 2.86118383e+09, 2.16438064e+09,
       1.63727458e+09])

from sklearn.linear_model import RidgeCV
ridge_cv = RidgeCV(alphas = lambdalar,
                   scoring = "neg_mean_squared_error",
                   normalize = True)

ridge_cv.fit(X_train_r, y_train_r)

RidgeCV(alphas=array([5.00000000e+09, 3.78231664e+09, 2.86118383e+09, 2.16438064e+09,
       1.63727458e+09, 1.23853818e+09, 9.36908711e+08, 7.08737081e+08,
       5.36133611e+08, 4.05565415e+08, 3.06795364e+08, 2.32079442e+08,
       1.75559587e+08, 1.32804389e+08, 1.00461650e+08, 7.59955541e+07,
       5.74878498e+07, 4.34874501e+07, 3.28966612e+07, 2.48851178e+07,
       1.88246790e+07, 1.42401793e+07, 1.07721735e+07, 8.14875417e+06,
       6.16423370e+06, 4.66301673e+06, 3.52740116e+06, 2.66834962e+06,
       2.01850863e+06, 1.52692775e+06, 1.15506485e+06, 8.73764200e+05,
       6.60970574e+05, 5.00000000e+05]),
        cv=None, fit_intercept=True, gcv_mode=None, normalize=True,
        scoring='neg_mean_squared_error', store_cv_values=False)

ridge_cv.alpha_

0.005
```

Şekil 4.13: RR optimum alpha(lambda) değeri ve kod bloğu

Kod bloğundan anlaşılacağı üzere, model geliştirme çalışmasında elde ettiğimiz deneysel sonuç dizisini, eğitim verileri üzerinden cross-validation metodu kullanılarak optimum alpha değeri 0.005 olarak bulunmuştur.

2.Adım: Doğrulanmış model tahmin sonuçları;

Tablo 4.8: RR ölçüm metrik sonuçları - 2

Ölçüm Metrikleri	
MSE	497.0262344776716
RMSE	22.294085190419267
MAE	6.69722792342646
R2	0.14889602917267908

Tablo 4.6'daki PLSR model optimizasyon sonuçları ile Tablo 4.8'deki RR doğrulanmış model verilerinin sonuçlarına bakıldığında, birbirine çok yakın değerler olduğu görülmektedir. RR modelin tahmin fonksiyonu, RMSE değerine göre fon fiyat tahmini için birim başına düşen hata payı oranının artı eksi 22.29 olarak, bir sapma değerinin olduğu görülmektedir. Yüzdelik olarak ise, tahmin başarısının %77-78 civarında olduğu saptanmaktadır.

4.4. Doğrusal Olmayan Çoklu Regresyon Algoritmaları ile Model Geliştirme

Model geliştirme çalışmalarının ilk adımı olan doğrusal çoklu regresyonlar ile tahmin işleminde; Kısmi En Küçük Kareler Regresyon (PLSR) ve Ridge Regresyon (RR) algoritmalarıyla uygulamalar yapılacaktır. Bölüm 3'de teorik kapsamlarından bahsedilen PLSR ve RR algoritmaların her birinin kendi bölüm başlığı altında sözde kod(pseudocode) yapılarından, model geliştirme ve optimizasyon adımlarından bahsedilecektir.

4.4.1 Destek Vektör Regresyonu (SVR)

4.4.1.1 Model Geliştirme

4.4.1.2 Model optimizasyonu

4.4.2 Yapay Sinir Ağları(YSA)

4.4.2.1 Model geliştirme

4.4.2.2 Model optimizasyonu

4.5. Algoritmaların Model Karşılaştırması

Kaynakça

1. Özkan, Y. (2008). Veri Madenciliği Yöntemleri, İstanbul: Papatya Yayıncılık.(bak)
2. Özata, M. (2014). Regresyon, Korelasyon ve Faktör Analizi, Sosyal Hizmette İleri İstatistik Uygulamaları Dersi.(web erişim adresi)
3. <https://scikit-learn.org/stable/modules/classes.html>, Regression metrics
4. Aydemir, E. (2013). Kusurlu Ürünleri İçeren ekonomik Üretim Miktarı Modelinin Gri Sistem Teorisi Yaklaşımıyla Geliştirilmesi, Doktora Tezi, Fen Bilimleri Enstitüsü, Süleyman Demirel Üniversitesi, Isparta.
5. Esposito Vinzi, V. and Russolillo, G. (2013), Partial least squares algorithms and methods. WIREs Comp Stat, 5: 1-19. doi:[10.1002/wics.1239](https://doi.org/10.1002/wics.1239)
6. Cawley, C.G., Talbot, C.L.N. (2010). On Over-fitting in Model Selection and Subsequent Selection Bias in Performance Evaluation, Journal of Machine Learning Research, 11(70):2079–2107.
7. https://scikit-learn.org/stable/modules/generated/sklearn.linear_model.RidgeCV.html, RidgeCV