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 bolüm 2'de yapılmıştır. Yapılan literatür araştırmasıyla bu çalışma da kullanılan metotların nasıl tercih edildiği belirlenmiştir. Büyük bir uygulama alanının mevcut olması ve benzer çalışmalarla makine öğrenme metotlarında ciddi başarılar elde edilmiş olması, bölüm 3'de açıklanan algoritmaların kullanılmasını sağlayarak tezin alt yapısını oluşturmuştur. Bu bağlamda bu bölümde uygulama alanı olarak beş ana başlık şeklinde bir akış izlenmiştir. İlk olarak, uygulama çalışmaların temel yapısını oluşturan ve önemi büyük olan veri seti hakkında detaylı bir bilgilendirme verilmiştir. Sonrasında, model geliştirme bölümüne geçmeden yeni bir başlık açılarak, geliştirilecek olan modellerin kabul doğruluklarının ölçümünün 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 başlık da ise, kullanılan dört algoritma ile uygulaması yapılan model geliştirme için bulunan sonuçların ortak bir tablo da karşılaştırması yapılarak bölüm sonlandırılmıştır. Ayrıca, tüm uygulama geliştirme ve veri işlem süreçlerin de kullanılan platform, programlama dili ve versiyon bilgileri için tablo 4.1'e ve uygulama çalışmalarında kullanılacak olan bilgisayarın teknik özeliklerine dair ise tablo 4.2' ye ayrıntılı olarak bakılabilir.

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 özelikleri

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 sağlayıcılığı 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.

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

- **FONTUR**: Fonların tür bilgisini içermektedir ve 31 çeşit olarak mevcuttur. Tür sayısı çok olduğundan dolayı Tablo 4.3'te bilgileri verilmektedir. Veri tipi, object olarak tutulmaktadır.
- **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 tutulmaktadır.
- **TEDAVPAYSAYISI**: Fonun satılmış olan pay adeti bilgisini içerir. Veri tipi, float64 olarak tutulmaktadır.
- **KISISAYISI**: Fonun yatırımcı sayısı bilgisini barındırır. Veri tipi, int64 olarak bulunmaktadır.
- **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 tutulmaktadır.
- **DOLARFIYAT**: Merkez Bankasının gün sonunda o tarih için en son verdiği dolar fiyatını bulundurmaktadır. Veri tipi, float64 olarak bulunmaktadır.
- **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 oranının, yüzdelik üzerinden gösterilmesi bilgisini içerir. Veri setindeki menkul oranı, 26 farklı menkul değerini kolon bilgisi olarak bulundurmaktadır. Verideki her bir menkul değerin 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	Kisa Vadeli Kira Sertifikalari Katilim 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	Dövize Ödemeli Tahvil
EUT	Eurobonds	Eurobonds
FB	Commercial Paper	Finansman Bonosu
FKB	Fund Participation Certificate	Fon Katılma Belgesi
GAS	Real Estate Certificate	Gayrı 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 toplanmasının ve her bir kolon için içerik ve tip bilgilerinin nasıl olduğudur. 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 bu verilerin ayrıntılı incelemesi yapılacaktır.

4.1.2 Veri Ön İşleme Süreçleri

Veri işleme bölümünde, hazırlanmış verinin ham formatı üzerinden ön işleme süreçleri yapılacaktır. Ön işlem süreçleri veri setinin, uygulama bölümünde geliştirilecek olan model için çok önemli bir yere sahiptir. Çünkü ham verinin toplanma sürecinde gereksiz, hatalı, tekrarlı, tutarsız ve boş verilerin olması mümkündür ve verinin bu hali ise geliştireceğimiz model için doğru bir sonuç elde edilmesini engeller. Bu sebepten ötürü, veri seti tutarlı, eksiksiz ve hatasız olmasını sağlayacak daha ideal bir yapıya dönüştürülür. Doğru bir veri seti 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 getirilmesini sağlayan sürece veri ön işleme adımı denir[1]. Veri ön işleme adımları aşağıdaki şekilde sıralanabilir.

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

Bu adımlar çerçevesinde, veri setimizin adım adım ön işleme süreçlerini yapıp ideal bir veri kümesine dönüşümü sağlanacaktır. Temizlenmiş veri kümesi üzerinden, her bir veri kolonunun ortalaması, maximum, minimum değerleri ve korelasyon matrislerine bakılarak gerekli-gereksiz veri olup olmama 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'de 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 veriden ve 38 kolondan oluştuğu bilgisi verilmiştir. Daha sonra dört kolondan oluşan yukarıdaki şekil 4.1'de ki 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 kullandığı bilgisi verilmiştir. Şekil 4.1'de anlaşılacağı 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 anlaşılmaktadır. Bu bilgiler ışığında, veri setimiz hatalı ve gereksiz kolonlardan temizlenecek ve object olan verilerin tiplerin 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   TARIH           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   FAIZ            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 adım da şekil 4.3'de ki veri setinde bulunan null değerleri üzerinde durulacaktır.

Şekil 4.3: Veri setinin ilk 5 satırlık formatı

	TARIH	FONTIP	FONTUR	FON	FONTOPLAMDEGER	TEDAVPAYSAYISI	KISISAYISI	FONFIYAT	FAIZ	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 × 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 değerin var olup olmadığı aşağıdaki şekil 4.4 sorgulanmıştır.

```
TARIH                0
FONTIP               0
FONTUR               0
FON                  0
FONTOPLAMDEGER       0
TEDAVPAYSAYISI       0
KISISAYISI           0
FONFIYAT             0
FAIZ                 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'de verilen bilgi, her değişken karşısında null olan toplam değer sayısıdır. 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 oranlar bilgisinde bulunmaktadır. Bunun nedeni veri tabanında menkul oranları tutulurken, bulunan menkul oranlarının yanısıra değeri olmayan menkuller için de değişkenlere null değeri atılmış olmasıdır. Menkul oranlar, hesaplanırken yüzdelik üzerinden 0 ile 100 arasında bir değer aldığından dolayı, null olan değerlerin 0 olacak şekilde dönüştürülmesinin 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 kümesindeki 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 değişkeninin veri setinden silinme işlemi yapılacaktır. Negatif değerler için ise, menkul değişken oranlarının 0 ile 100 arasında değer alacağı ifade edilmiştir. 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 bildirmiştir. 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 yönüne ve büyüklüğüne dair korelasyon matris ilişkileri incelenecektir. Korelasyon kat sayısı, -1 ile +1 arasında bir değer alır. Buradaki değerinin artı ya da eksi olması ilişkinin büyüklüğü hakkında bilgi vermez, artı değeri alan iki değişkenin, birlikte aynı yönde arttığını ya da azaldığını gösterir. Eksi değeri ise, tam tersi bir korelasyonun iki değişken arasında olduğunu ifade eder. Biri artarken, diğer değişkenin azaldığını veya tam tersinin olması durumudur [2]. İlk olarak tüm sayısal değişkenlerin korelasyon matrisi şekil 4.6'da 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	0.090018	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 × 22 columns

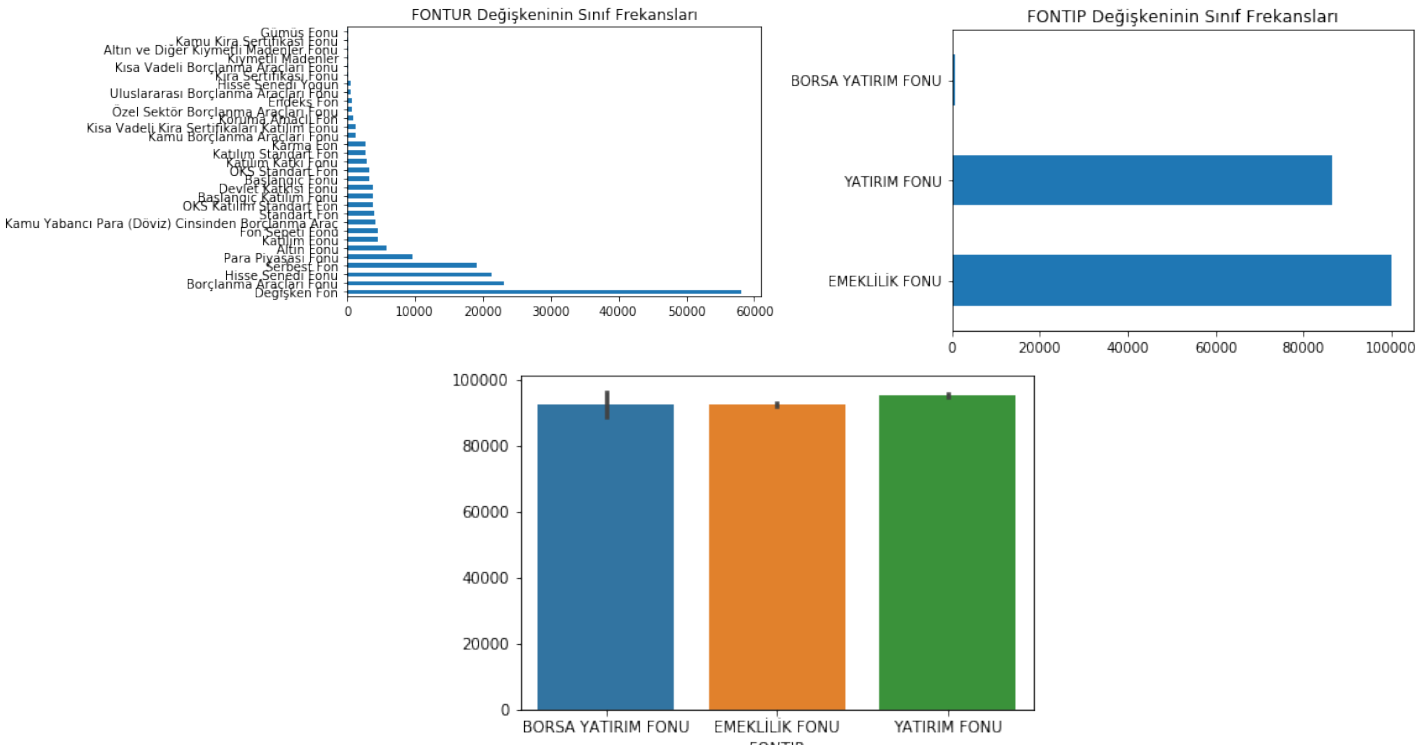
Şekil 4.6: Korelasyon matrisi

Korelasyon matrisine şekil 4.6'da bakıldığı zaman, değişkenlerin aralarındaki ilişkinin yön ve büyüklük değerlerini görmekteyiz. Genel olarak incelendiğinde, 22 satırda ve 22 kolon matrisinde değerler arasında ciddi bir tutarsızlığın görülmediği ve bazı değişkenler arasında korelasyon katsayısının daha iyi olduğu gözlemlenmektedir. Bu değişkenlerden, FONTOPLAMDEGER - TEDAVPAYSAYISI - KISISAYISI arasındaki ilişkinin diğer değişkenlere oranla daha güçlü olduğu görülmektedir. Veri seti özelikleri bölümünde veriler hakkındaki açıklamalar da, bu güçlü ilişkinin nedenini desteklemektedir. Bu değişkenlerin bağımsız değişken olarak seçilen FONFIYAT değerine olan katkıları dikkate alındığında diğer değişkenlere oranla daha iyi olduğu görülür ve korelasyon matrisinde menkul oran değişkenleri katsayıları da genel olarak normal gözlenmektedir. FONFIYAT değişkeninin FAIZ, DOLARFIYAT ve ALTINFIYAT ile doğrudan korelasyon ilişkisinin zayıf olduğu tespit edilmektedir. Ama ALTINFIYAT değişkeninin, FAIZ ile arasında ters yönde ve DOLARFIYAT ile de aynı yönde güçlü bir ilişkisi olduğu incelenmiştir.

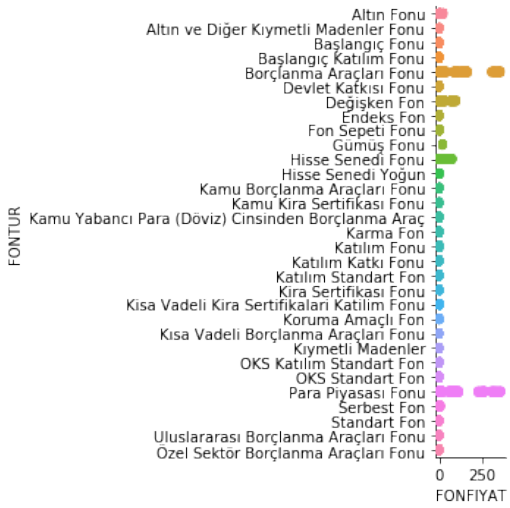
4.1.2.3 Veri Görselleştirme

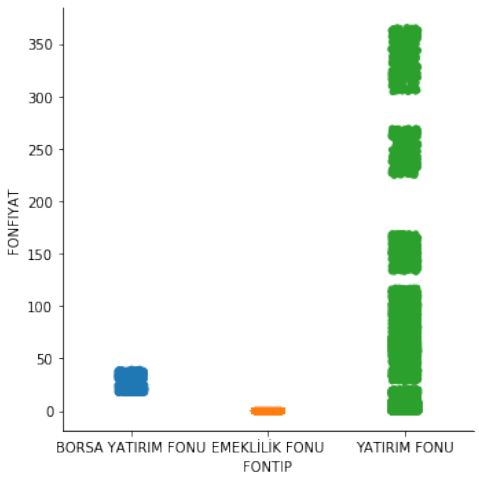
Bu bölümde veri setindeki kategorik ve sayısal değişkenler için görselleştirme çalışmaları yapılacaktır. Veri ön işlemenin bir önceki adımlarında verinin ideal bir veri setine dönüştürülme süreçleri yapılmış ve bu adımda ise bazı değişkenler üzerinden temel bazı grafikselleştirme işlemleri yapılarak bölüm sonlandırılacaktır.

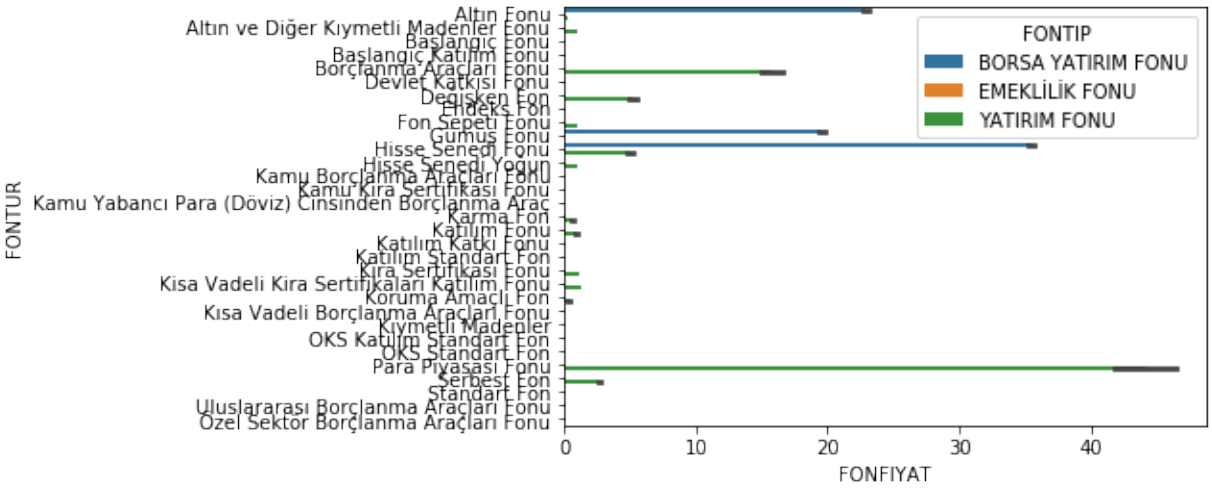
Şekil 4.7: Kategorik değişken dağılım grafikleri

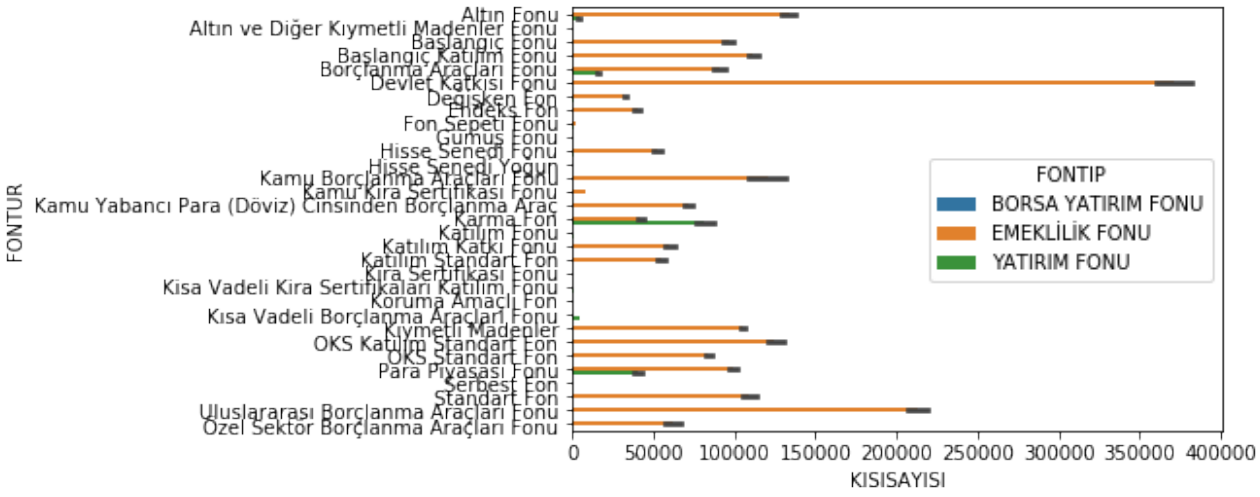


Şekil 4.8: Fon fiyat ve kişi sayısının kategorik değişken grafikleri



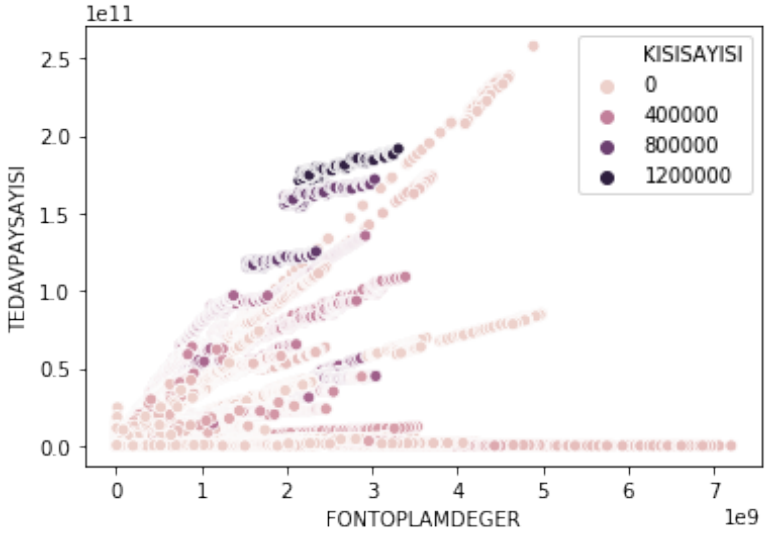




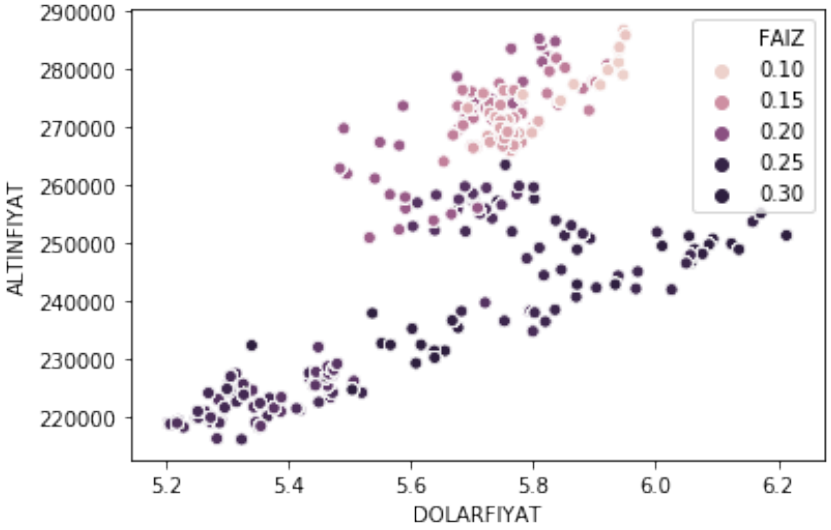


Veri görselleştirme amacımız gereği, belli değişkenlerin veri ile olan ilişkilerinin daha anlaşılır olmasını sağlamak için grafikleştirme çalışmaları yapıldı ve bu çalışmalardan, şekil 4.7 ve 4.8'de kategorik değişkenlerin dağılımı ve bağımsız değişken ile olan bağlantılarının görselleştirme işlemleri yapılmıştır. Son olarak yapılan şekil 4.9 ve 4.10'daki grafiklerde, Toplam-Pay-Kisi (FONTOPLAMDEGER-TEDAVPAYSAYISI-KISISAYISI) ve veri setimizde mevcut olan 2019 yılına ait Faiz-Altın-Dolar(FAIZ-ALTINFIYAT-DOLARFIYAT) ilişkisinin korelasyon grafiği verilmiştir. Bir önceki bölüm olan veri korelasyon matris değerleri incelenirken altın değişkeninin, 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 algoritmalarının değerlendirme sonuçları ile ilgili yapılan çalışmalar incelendiğinde genel olarak çok sık kullanılan ölçüm metrikleri;

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

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

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

4.1, 4.2 ve 4.3'de ki denklem değişkenleri;

Gözlem sayısı: $\frac{1}{n}$

Gerçek değerler: y_i

Gerçek değerlerin ortalaması: $\bar{y}$

Tahmin edilen değerler: $\hat{y}_i$

MSE değeri, veri setimizdeki bağımsız değişkenlerin tahmin ettiği değer ile gerçek değerin 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 şeklinde 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. Ayırca RMSE'nin MSE metoduna göre

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

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

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

Ortalama mutlak hata, gerçek değerden tahmin edilen değerin farkının mutlak ölçümü alınarak bulunur. Hesaplanan MAE değerleri daha kolay yorumlanabilir oldukları için regresyon ve zaman serisi gibi problemlerde daha çok kullanılmaktadır[3,4]. Matematiksel olarak gösterimi;

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

4. R Kare Oranı(Belirleme Katsayısı) - R Squared (R^2)

R-kare (R2), bağımsız bir değişken veya bir regresyon modelindeki değişkenler tarafından açıklanan bağımlı bir değişkenin varyans oranını temsil eden istatistiksel bir ölçüdür[3,4]. Yani, bağımsız değişkenin bağımlı değişkendeki, değişikliği açıklama başarısıdır. Matematiksel denklemi;

$$R2 = \frac{\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}{\sum_{i=1}^{n}(y_i - \bar{y}_i)^2} \quad \textbf{(4.4)}$$

Kısaca 4 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

Algoritma 1: NIPALS

Input: $E_0 = X$ Bağımsız değişken değerleri

Output: $P = [p_1, \ldots, p_H], T = [t_1, \ldots, t_H]$

for all $h = 1, \ldots, H$ **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 kodun 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 değerininde keşime noktası$(t_h; e_p)$ olmayan veri kümesinden eğimi, e_p'nin p t_h kolununda E değer 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 incelecektir. Model geliştirme alt başlığında, veri setimizin bağımsız değişkenlerinden 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 bölümü tamamlanmış olacaktır.

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(['TARIH','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 × 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 ayrımının, 4 algoritmanın model geliştirme aşamasının ortak ilk adımı olduğu için sadece bu başlıkta detaylı anlatımı yapılarak, 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 eğtim 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 işlemi, veri setimizin genelinde %75 eğitim ve %25 test verisi oluşturacak formatta yapılmıştır. Bölünmüş olan verinin, bağımlı (y_train, y_test) ve bağımsız (X_train, X_test) değişkenler bazında olacak şekilde eğtim 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 mevcuttur ve bu parametre değişiklikleri optimizasyon bölümünde yapılacaktır. İkinci satırda modelin bağımsız değişken katsayıları verilmiştir. Burada öğrenme modeli için hesaplanan çoklu doğrusal bir fonksiyon denklemin de, 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 sonuçların daha iyi olduğu görülse de, model için daha doğru bir değerlendirme referansı ise test verisinden elde edilen sonuçlar ile olmuştur.

4.3.1.2 Model optimizasyonu

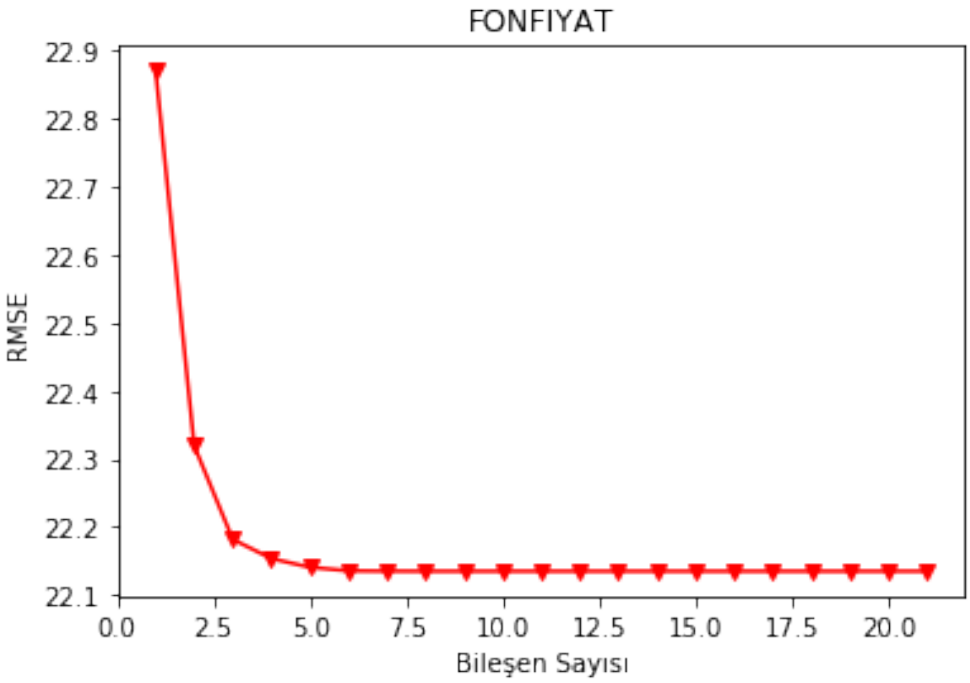
Kısmi En Küçük Kareler Regresyonu (PLSR) kullanılarak elde edilen tahmin sonuçlarında model optimizasyon parametreleri için yapılan literatür ve örnek çalışmalar incelendiğinde[5], PLSR temel çalışma yapısından hareketle, bağımsız değişkenlerin daha az sayıda ve aralarında çoklu bağlantı problemi olmayan bileşenlere indirgenip model kurma fikrine dayanıyor olması, 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 seçimi için kullanılan değer aralığı, 5 ile 10 arasında olmaktadır[6]. Bu çalışma 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. Tahmin başarısının, RMSE sonuç çıktısına göre düşük olduğu değerlendirilmiştir.

4.3.2 Ridge Regresyonu (RR)

Bu bölümde Ridge Regresyonu (RR) ile model geliştirme çalışmaları yapılacak, daha önce bölüm 3'de ayrıntılı olarak RR algoritması hakkında bilgilendirilme yapılmıştır. Burada ise, sözde kodun (pseudocode) yapısı ve iki alt başlık olarak; model geliştirme ve optimizasyon konuları incelenecektir.

4.3.2.1 Model geliştirme

Bu kısımda yapılan model çalışmasının, PLSR model geliştirme aşamalarıyla benzer olması ve tekrar olmaması amacıyla aynı adımlarda ayrıntılı bir anlatım yapılmayacaktır. Ayrıca, uygulamanın birinci adımı olan veri kümesinin test ve eğitim seti olarak ayrılma aşaması da ortak olduğu için bu adım atlanarak, 2'inci adımdan başlayarak model çalışmamız gerçekleştirilmiş olacaktır.

2.Adım: Eğitim seti ile model kurma;

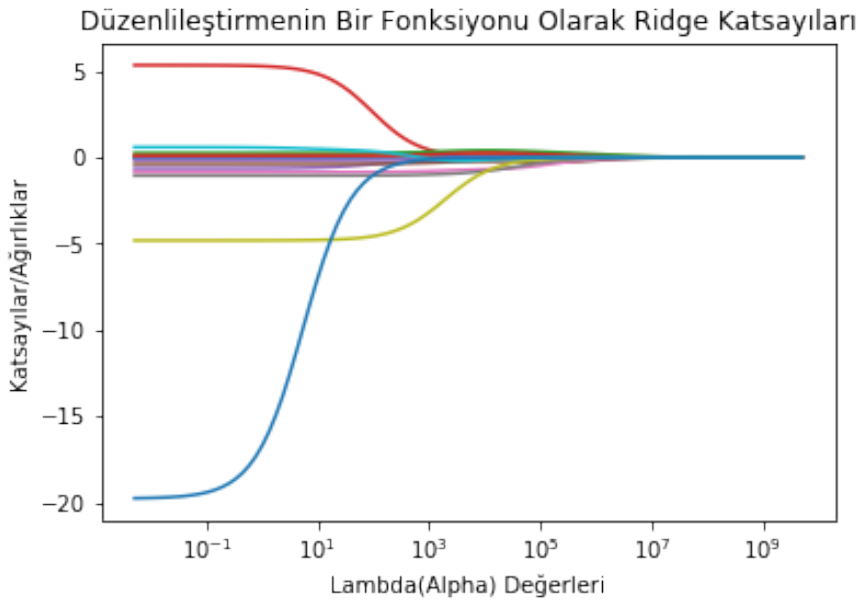
```
ridge_model

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) değişkenin, model katsayıları değerlerine olan etkisini deneysel olarak göstermek için şekil 4.16 ve 4.17'deki çalışmalar yapılmıştır.

```python
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('Düzenlileştirmenin 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 katsayı/ağırlık değerlerinde değişim olduğu görülmektedir. Deneysel çalışma için yapılmış olan şekil 4.16'daki kod bloğu ve sonuç grafiğinden anlaşılacağı üzere, alpha parametresinin alacağı değerin katsayılara olan 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 bakıldığında, PLSR test ve eğitim sonuçlarına göre daha iyi olduğu görülmektedir. Ayrıca, eğitim ve test değerleri arasında da küçük farklılıkların olduğu görülmektedir. Bazı ölçüm metriklerinin eğitim seti üzerinden alınan sonuçlarının, test verisinden daha iyi olduğu görülse de, model için daha doğrulanmış sonuçlar test verisinden elde edilen sonuçlar baz alınarak yapılmaktadır.

4.3.2.2 Model optimizasyonu

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

1. **Adım:** Optimum alpha(lambda) değerinin 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+0...
       3.28966612e-01, 2.48851178e-01, 1.88246790e-01, 1.42401793e-01,
       1.07721735e-01, 8.14875417e-02, 6.16423370e-02, 4.66301673e-02,
       3.52740116e-02, 2.66834962e-02, 2.01850863e-02, 1.52692775e-02,
       1.15506485e-02, 8.73764200e-03, 6.60970574e-03, 5.00000000e-03]),
        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 4.18: RR optimum alpha(lambda) değeri ve kod bloğu

Kod bloğu şekil 4.18'de anlaşılacağı üzere, model geliştirme çalışmasında elde ettiğimiz deneysel sonuç dizisini, eğtim 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. Tahmin başarısının, RMSE sonuç değerine göre düşük olduğu saptanmaktadır.

4.4. Doğrusal Olmayan Çoklu Regresyon Algoritmaları ile Model Geliştirme

Model geliştirme çalışmalarının ikinci adımı olan doğrusal olmayan çoklu regresyonlar ile tahmin işleminde; Destek Vektör Regresyonu (SVR) ve Yapay Sinir Ağları(YSA) algoritmalarıyla model geliştirme süreçlerinin devamı yapılacaktır. Bölüm 3'de teorik kapsamlarından bahsedilen SVR ve YSA algoritmaların her birinin kendi bölüm başlığı altında sözde kod (pseudocode) yapılarından, model geliştirme ve optimizasyon adımlarından bahsedilecektir.

4.4.1 Destek Vektör Regresyonu (SVR)

Bu bölümde Destek Vektör Regresyonu (SVR) ile model geliştirme çalışmaları yapılacaktır. İlk olarak, bu bölüm başlığı altında SVR algoritmasının sözde kod(pseudocode) yapısından bahsedilecektir. Daha sonrada, iki alt başlıkta; model geliştirme ve optimizasyon konuları incelenecektir.

4.4.1.1 Model Geliştirme

Bu kısımda yapılacak model çalışması, toplamda 2 adımdan oluşmaktadır. Daha önceki model çalışmalarında, PLSR model geliştirme adımlarından olan veri kümesinin test ve eğitim seti olarak ayrılma aşaması ortak olduğu için bu adım atlanarak, 2'inci adımdan başlayarak model çalışması gerçekleştirilecektir.

2.Adım: Eğitim seti ile model kurma;

```
scaler = StandardScaler()
scaler.fit(X_train_s)
```

```
StandardScaler(copy=True, with_mean=True, with_std=True)
```

```
X_train_scaled_s = scaler.transform(X_train_s)
```

```
X_test_scaled_s = scaler.transform(X_test_s)
```

```
from sklearn.svm import SVR
```

```
svr_rbf = SVR("rbf").fit(X_train_scaled_s, y_train_s)
```

```
svr_rbf
```

```
SVR(C=1.0, cache_size=200, coef0=0.0, degree=3, epsilon=0.1, gamma='scale',
    kernel='rbf', max_iter=-1, shrinking=True, tol=0.001, verbose=False)
```

Şekil 4.19: SVR model yapısı ve veri standartlaştırma

SVR'nin dikkat edilmesi gereken kritik özelliği, marjinal verilere karşı duyarlı bir yapıya sahip olmasıdır. Bu sebepten ötürü, verilerden daha iyi tahmin değeri sağlamak için, verilerin standartlaştırılması gerekmektedir. Şekil 4.19'da, modelin ilk satırında, StandardScaler metodu kullanılarak, değişkenlerin ortalaması sıfır ve standart sapması bir olacak şekilde standartlaşma işlemi yapılmıştır[8]. Bir sonraki kod satırında, X bağımsız değişkeninin eğtim ve test setleri üzerinde veri standartlaştırma yapılmıştır. Ardından SVR algoritmasının, rbf çekirdek (kernel) fonksiyonu ile svr_rbf model nesnesi oluşturulmuştur.

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

Tablo 4.9: SVR ölçüm metrik sonuçları - 1

Ölçüm Metrikleri	Eğitim	Test
MSE	531.8675777625025	542.8368894117968
RMSE	23.06225439462722	23.298860259931104
MAE	3.677658112113757	3.711111249858252
R2	0.06766281319477463	0.07045020958408388

Tablo 4.9'da model ölçüm metrik değerlerine bakıldığında, eğitim ve test sonuçlarının PLSR ve RR model sonuçlarına göre daha düşük olduğu görülmektedir. Son olarak, SVR model eğitim ve test değerleri arasında da farklılıkların olduğu görülmektedir. Model kabul değerleri, test verisinden elde edilen sonuçlar baz alınarak yapılmaktadır.

4.4.1.2 Model optimizasyonu

Destek Vektör Regresyonu (SVR) model optimizasyon bölümünde, şekil 4.19'daki model parametreleri üzerinde optimizasyon işlemleri yapılacaktır. Tez çalışmasının daha önceki bölümleri olan literatür incelmesinde SVR üzerine yapılan çalışmalar incelenmiş ve metot bölümünde ayrıntılı olarak SVR teorik ve matematiksel denklem yapısı incelenmişti. Bu çalışmalardan yola çıkılarak SVR modelin çekirdek fonksiyonu rfb(Radial Basis Function) için, karmaşıklık ya da ceza değeri olarak ifade edilen ***C*** parametresi optimizasyon problemin konusu olmuştur[8,9]. Model optimizasyon işlemleri toplam 2 adımdan oluşacaktır.

1. **Adım:** Optimum ***C*** parametresi bulunması;

```
svr_rbf
```

```
SVR(C=1.0, cache_size=200, coef0=0.0, degree=3, epsilon=0.1, gamma='scale',
    kernel='rbf', max_iter=-1, shrinking=True, tol=0.001, verbose=False)
```

```
#'C': [0.1, 1, 10, 100, 1000]
svr_params = {"C": np.arange(0.1,2,0.1)}
svr_cv_model = GridSearchCV(svr_rbf,svr_params, cv = 10)
svr_cv_model.fit(X_train_scaled_s, y_train_s)
```

```
pd.Series(svr_cv_model.best_params_)[0]
```

```
svr_tuned = SVR("rbf", C = pd.Series(svr_cv_model.best_params_)[0]).fit(X_train,
                                                                        y_train)
```

```
y_pred = svr_tuned.predict(X_test)
```

Şekil 4.20: SVR optimum C parametre değeri ve kod bloğu

Şekil 4.20'deki kod bloğunun birinci satırında, ***C*** parametresi için deneysel çalışmalarda kullanılan örnek veriler verilmiştir. ***C*** değişken optimizasyonu ile ilgili incelen bir çok çalışmada verilen değer aralıkları uyarlanacak probleme göre belli bir sayı seti verilerek işlemler yapılmıştır. Yapılan çalışmalarda, ***C*** parametresi (ceza parametre) aralığının çok küçük ya da büyük olması durumunda model öğrenemsi için ciddi minimizasyon problemlerine yol açmaktadır[10,11,12]. Bu bağlamda, uygulamda kullandığımız ***C*** parametre aralığını veri setimiz için daha uygun olduğu değerlendirilmiştir. Daha sonra, Cross-validation metodunun hiper parametre optimizasyon tekniği olan GridSearchCV kullanılarak her bir parametre değeri için en doğru model buluncaya kadar, eğtim seti üzerinden çalışarak ***C*** parametresinin optimum değer bulunur. Son olarak, bulunan optimum değerler ile nihayi doğrulanmış model elde edilir.

2**.Adım:** Doğrulanmış model tahmin sonuçları;

Tablo 4.10: SVR ölçüm metrik sonuçları - 2

Ölçüm Metrikleri	
MSE	497.0262344776716
RMSE	22.294085190419267
MAE	6.69722792342646
R2	0.14889602917267908

Tablo 4.10'daki SVR doğrulanmış model sonuçları, RMSE değerine göre fon fiyat tahmini için birim başına düşen hata payı oranının artı eksi(+/-) 22.29 olarak, bir sapma değerinin olduğu görülmektedir. SVR model sonuçları incelendiğinde, PLSR ve RR model geliştirme çalışmalarının sonuçlarına göre çok daha düşük olduğu gözlemlenmektedir. Tahmin başarısının, RMSE çıktısına göre en düşük performansa sahip model olarak incelenmiştir.

4.4.2 Yapay Sinir Ağları(YSA)

Yapay Sinir Ağları(YSA) ile model geliştirme bölümünde, algoritmanın sözde kod(pseudocode) yapısından bahsedilecek ve iki alt başlıkta; model geliştirme ve optimizasyon konuları incelenerek model geliştirme süreçleri tamamlanmış olacaktır.

4.4.2.1 Model geliştirme

Bu bölümde yapılacak olan Yapay Sinir Ağları(YSA) ile model geliştirme 2 adımdan oluşacaktır. Bu tez çalışmasında, model geliştirmenin ilk örneği olan PLSR ile model geliştirmede, veri setinin eğtim ve test seti olarak ayrılmasına dair ayrıntılı açıklamalar model çalışmalarının ortak noktası olduğundan önceki bölümlerde yapıldığı dikkate alınarak direk 2. adımdan başlanacaktır.

2.Adım: Eğitim seti ile model kurma;

```
scaler = StandardScaler()
scaler.fit(X_train_y)
```

```
StandardScaler(copy=True, with_mean=True, with_std=True)
```

```
X_train_scaled = scaler.transform(X_train_y)
```

```
X_test_scaled = scaler.transform(X_test_y)
```

```
from sklearn.neural_network import MLPRegressor
```

```
mlp_model = MLPRegressor(hidden_layer_sizes = (100,20)).fit(X_train_scaled, y_train_y)
```

```
mlp_model
```

```
MLPRegressor(activation='relu', alpha=0.0001, batch_size='auto', beta_1=0.9,
             beta_2=0.999, early_stopping=False, epsilon=1e-08,
             hidden_layer_sizes=(100, 20), learning_rate='constant',
             learning_rate_init=0.001, max_fun=15000, max_iter=200,
             momentum=0.9, n_iter_no_change=10, nesterovs_momentum=True,
             power_t=0.5, random_state=None, shuffle=True, solver='adam',
             tol=0.0001, validation_fraction=0.1, verbose=False,
             warm_start=False)
```

Şekil 4.21: YSA model yapısı ve veri standartlaştırma

Yukarıdaki şekil 4.21'de YSA model yapısı verilen kod bloğunun birinci satırında, veri standartlaştırma işlemleri yapılmaktadır. YSA ile model kurma aşamasında birden fazla katman ve hücre yapısının olmasından dolayı, verilerdeki aşırı uç değerlerin, aykırılıkların bulunması veya değişken ölçeklerinin varyans yapılarının birbirinden çok farklı olması, bulunacak sonuçların

güvenirliliğini düşürmektedir. Özelikle bu sebepten kaynaklı, YSA model geliştirme aşamasında veri standartlaştırma işlemi yapılmaktadır[13]. Burada kullanılacak olan StandardScaler metodu SVR model geliştirme adımında bahsedilmiştir. Ayrıca, ayrıntılı olarak YSA model parametreleri görülmektedir. Model parametrelerinden, gizli katman sayısı deneysel çalışmalar için manuel olarak (100,20) verilmiştir. Son olarakta, aktivasyon fonksiyonu, alpha ve diğer değişkenlerin default değeri bulunmaktadır.

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

Tablo 4.11: YSA ölçüm metrik sonuçları - 1

Ölçüm Metrikleri	Eğitim	Test
MSE	27.809947088094503	39.321943328021554
RMSE	5.273513732616471	6.270721117066326
MAE	1.6469758061055841	1.7579658612687985
R2	0.9512505576249017	0.93266540116882

Tablo 4.11'de model ölçüm metrik değerlerine bakıldığında, eğitim ve test sonuçlarının daha önceki model geliştirme bölümlerindeki tablo 4.5, 4.7 ve 4.9'daki sonuçlarına göre çok daha iyi olduğu gözlemlenmektedir. Ayrıca, eğitim ve test değerleri arasında farklılıklar olduğu görülmektedir. Model değerlendirme için test sonuçları referans alınacaktır.

4.4.2.2 Model optimizasyonu

Yapay Sinir Ağları(YSA) model optimizasyon kısmında, şekil 4.21'de verilen model parametreleri için optimizasyon çalışmaları yapılacaktır. YSA hakkında, literatür ve metot bölümden yapılan araştırmalarda optimizasyon problemlerine konu olunan parametrelerin; aktivasyon(activation), cezalandırma katsayısı(alpha) ve gizli katman sayısı(hidden_layer_sizes) olduğu tespit edilmiştir[13,14,15,16]. Deneysel çalışmalar yapılarak, optimizasyon yapılacak olan parametrelerin optimum değerlerinin nasıl elde edildiği, toplam 2 adımdan oluşacak şekilde incelenecektir.

1. **Adım:** Optimum activation, alpha ve hidden_layer_size parametrelerinin bulunması;

```
mlp_params = {'alpha': [0.1, 0.01,0.02,0.005],
              'hidden_layer_sizes': [(20,20),(100,50,150),(300,200,150)],
              'activation': ['relu','logistic']}

mlp_cv_model = GridSearchCV(mlp_model, mlp_params, cv = 10)

mlp_cv_model

mlp_cv_model.fit(X_train_scaled, y_train)

mlp_cv_model.best_params_

mlp_tuned = MLPRegressor(alpha = 0.02, hidden_layer_sizes = (100,50,150))

mlp_tuned.fit(X_train_scaled, y_train)

y_pred = mlp_tuned.predict(X_test_scaled)
```

Şekil 4.22: YSA optimum parametre değerleri ve kod bloğu

Yukarıdaki şekil 4.22'nin birinci satırında, optimizasyonu yapılacak parametreler için deneysel çalışmalarda kullanılan örnek veriler verilmiştir. Burada kullanılan deneysel çalışma örnekleri, alpha(ceza terimi katsayısı) için verilecek değer aralığı çok büyük seçilirse modelin eksik öğrenmesi aynı şekilde çok küçük seçildiğinde de aşırı öğrenme problemlerine yol açacağı ifade edilmiştir [16,17]. Bundan dolayı, model çalışmamız için örnek olarak kullanılan, sayısı dizisi belli bir aralık olarak verilmiş en optimum değer bulunmaya çalışılmıştır. Aynı konsepte, gizli katman sayısı içinde örnek deneysel veriler kullanılarak model için doğru değerler bulunmuştur. Aktivasyon fonksiyonu ise, en sık kullanılan ve tercih edilen fonksiyonlar parametre setine verilmiştir[18]. Daha sonra, Cross-validation metodunun hiper parametre optimizasyon tekniği olan GridSearchCV kullanılarak örnek parametre seti için en doğru modeli bulacak parametreler, best_params_ yöntemi ile elde edilmiştir. Son olarak, tüm değerler için bulunan optimum değerler ile nihai doğrulanmış model elde edilmektedir.

2.**Adım:** Doğrulanmış model tahmin sonuçları;

Tablo 4.12: YSA ölçüm metrik sonuçları - 2

Ölçüm Metrikleri	
MSE	39.321943328021554
RMSE	6.270721117066326
MAE	1.7579658612687985
R2	0.93266540116882

Tablo 4.12'de YSA doğrulanmış model sonuçları, RMSE değerine göre fon fiyat tahmini için birim başına düşen hata payı oranının artı eksi(+/-) 6.2 olarak, bir sapma değerinin olduğu görülmektedir. YSA model sonuçları incelendiğinde, PLSR, RR ve SVR model geliştirme çalışmalarının sonuçlarına göre çok daha iyi değerler elde ettiği gözlemlenmektedir. Tahmin başarısının, RMSE değerine göre en başarılı model olduğu saptanmaktadır.

4.5. Doğrulanmış Model Sonuç Karşılaştırmaları

Model çalışmasının bu son bölümünde, 4 algoritmanın doğrulanmış model sonuçları karşılaştırma tablosu yapılarak değerlendirilecektir. Ayrıca, her bir model için tahmin edilen değerle gerçek değer arasındaki uzaklık ilişkisini de gösteren grafiklere yer verilerek bölüm sonlandırılacaktır.

Tablo 4.13: Model karşılaştırma ölçüm metrik sonuçları

Ölçüm Metrikleri	PLSR	RR	SVR	YSA
MSE	497.0027614702058	497.0262344776716	542.8368894117968	39.321943328021554
RMSE	22.29355874395575	22.294085190419267	23.298860259931104	6.270721117066326
MAE	6.705839394559914	6.69722792342646	3.711111249858252	1.7579658612687985
R2	0.14893622417341656	0.14889602917267908	0.07045020958408388	0.93266540116882

Yukarıdaki tablo 4.13'te model geliştirme çalışmaları yapılmış olan 4 algoritmanın doğrulanmış model ölçüm metrik sonuçları bulunmaktadır. Bu sonuçlar her değerlendirme metriği için ayrı olarak incelenecektir.

- **MSE:** Basitçe, gerçek değer ile tahmin edilen değer arasındaki çizilecek olan regresyon eğrisinin ne kadar gerçek değerlere yakın olduğunu söylemektedir. Kısaca, tahmin fonksiyonun performansını ölçer. Karşılaştırma tablosuna baktığımızda, YSA modelinin en iyi tahmin başarısına sahip olduğu görülmektedir. Daha sonra, PLSR ve RR modellerinin birbirlerine çok yakın değer tahminleri yaptıkları, SVR modeline göre daha iyi oldukları ama genel olarak 3 modelin de çok düşük performansa sahip olduğu tespit edilmiştir.

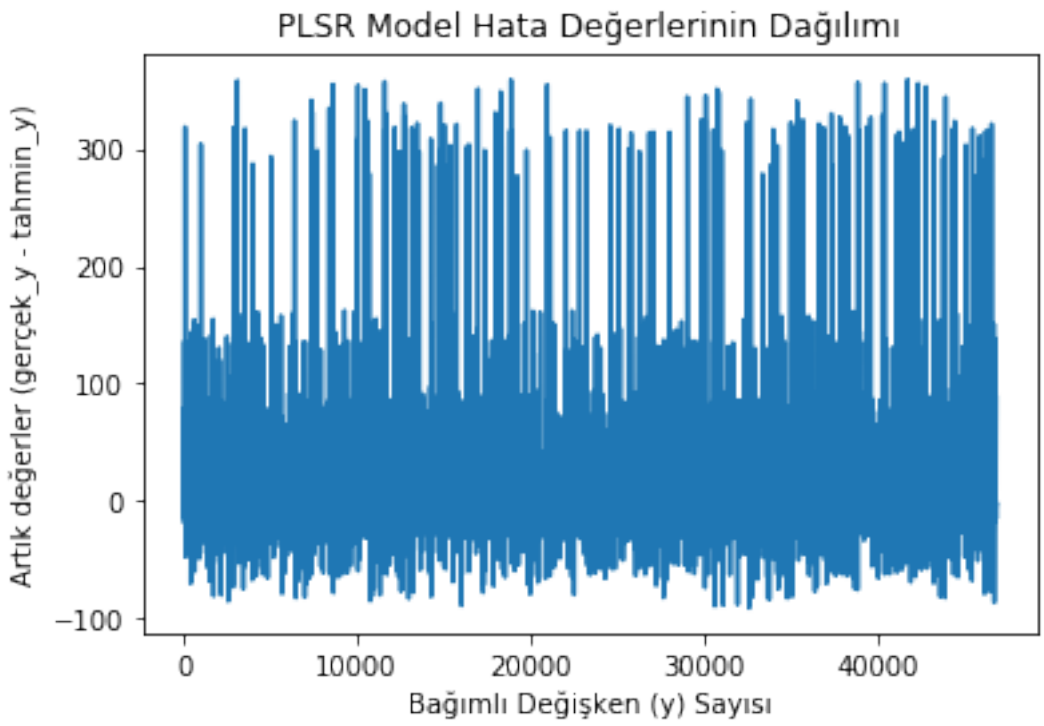
- **RMSE:** MSE değerinin karekökü alınmış formatıdır. RMSE'nin MSE metoduna göre daha avantajlı olduğu model ölçüm metrikleri bölümünde anlatılmıştır. Kısaca, RMSE tahmin edilen değerlerin, gerçek değere olan uzaklığının (+/-) birim pay başına düşen değeridir. Buna göre, YSA modelinin diğer üç modele göre çok başarılı olduğu gözlemlenmektedir. PLSR ve RR model değerlerinin birbirine çok yakın olduğu ama düşük performansa sahip oldukları incelenmiştir. Ayırca, en düşük performansın SVR modeline ait olduğu görülmüştür.

- **MAE:** Kısaca, iki sürekli değişken arasındaki fark değerinin performans ölçüsüdür. Model ölçüm metrikleri bölümünde regresyon problemlerinde sıkça kullanılan ve çıktılarının yorumlanması kolay olan bir model olduğundan bahsedilmiştir. Buradan hareketle model sonuçlarına göre, YSA modelinin yüksek bir başarı performansına sahip olduğu incelenmektedir. SVR de, diğer ölçüm metrik sonuçlarına göre çok iyi bir sonuca ulaşmıştır. Bunun nedeni, negatif yönelimli tahmin puanlarında MAE yönteminin daha iyi performans gösterdiği bilinmekte ve model tahmin değerlerimizin daha çok negatif yönelimli puanlara sahip olmasına bağlı olduğu değerlendirilmektedir. Son olarak, PLSR ve RR modellerinin diğer modellere göre çok düşük performansa sahip oldukları gözlemlenmiştir.

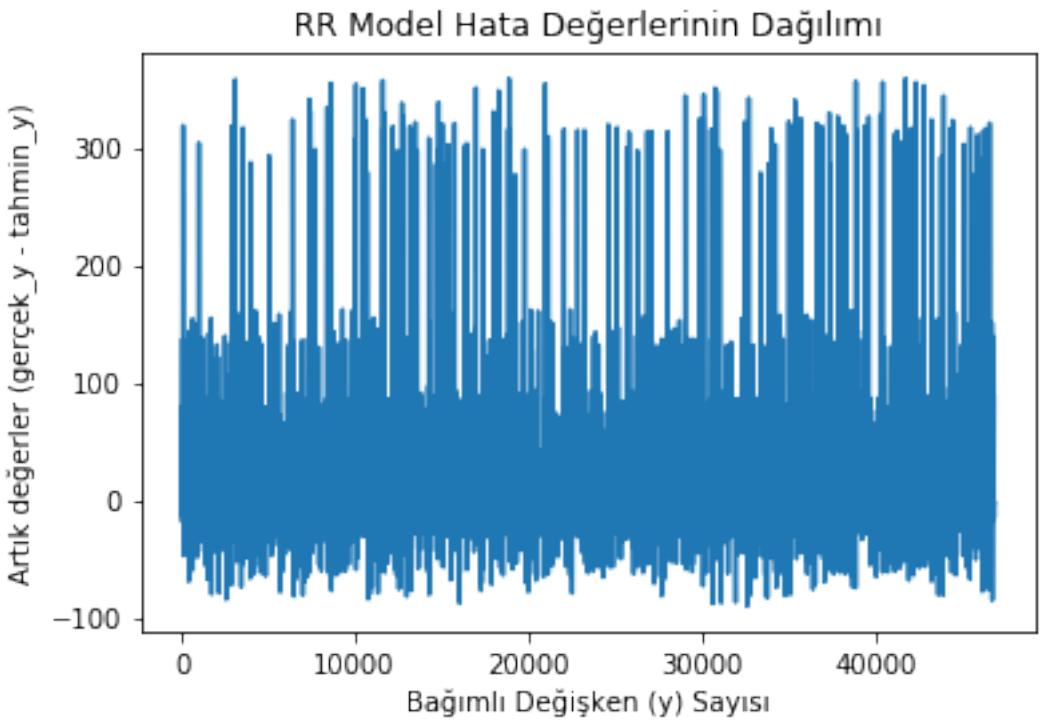
- **R2:** Model için veri uyumluluğu skoru olarak ifade edilebilir. Alacağı en iyi skor değerinin 1 olduğu belirtilmiştir. Model ölçüm metrikleri bölümünde ayrıntılı anlatımı yapılmıştır. Tablo 4.13'teki çıktılara baktığımızda, YSA modelin çok yüksek bir performansa sahip olduğu ve başarı skoru olarak % 90'nın üstünde olduğu görülmektedir. PLSR ve RR'nin birbirine çok

yakın sonuçlara sahip olduğu ve başarı skorlarının düşük olduğu belirtilmektedir. Ayrıca, burada SVR modelinin diğer üç modelin sonuçlarına göre en kötü skor değerini elde ettiği incelenmiştir.

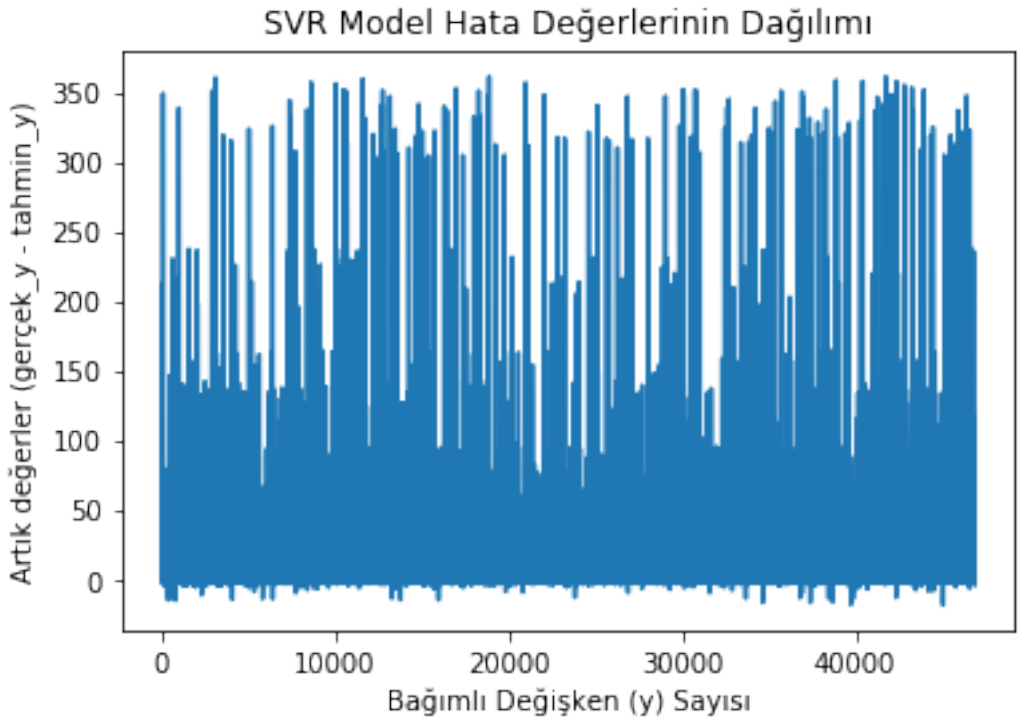
Doğrulanmış model sonuçlarının, ölçüm metriklerine göre değerlendirilme adımları yukarıda yapıldıktan sonra, tahmin edilen değerlerin gerçek değerlerden ne kadar uzakta olduğunu ifade eden artık verileri görselleştirme işlemleri yapılacaktır. Modellerin görselleştirme grafikleri tahmin edilen değer verilerinin dağılımını daha iyi okumamızı sağlayacaktır.



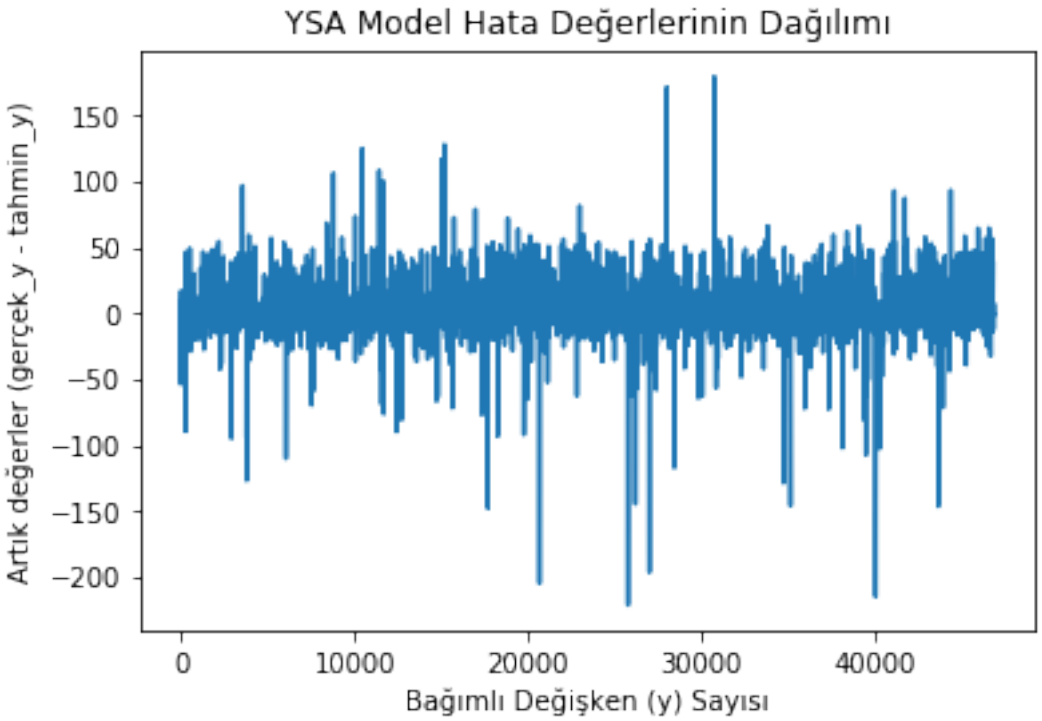
Şekil 4.23: PLSR model artık veri grafiği



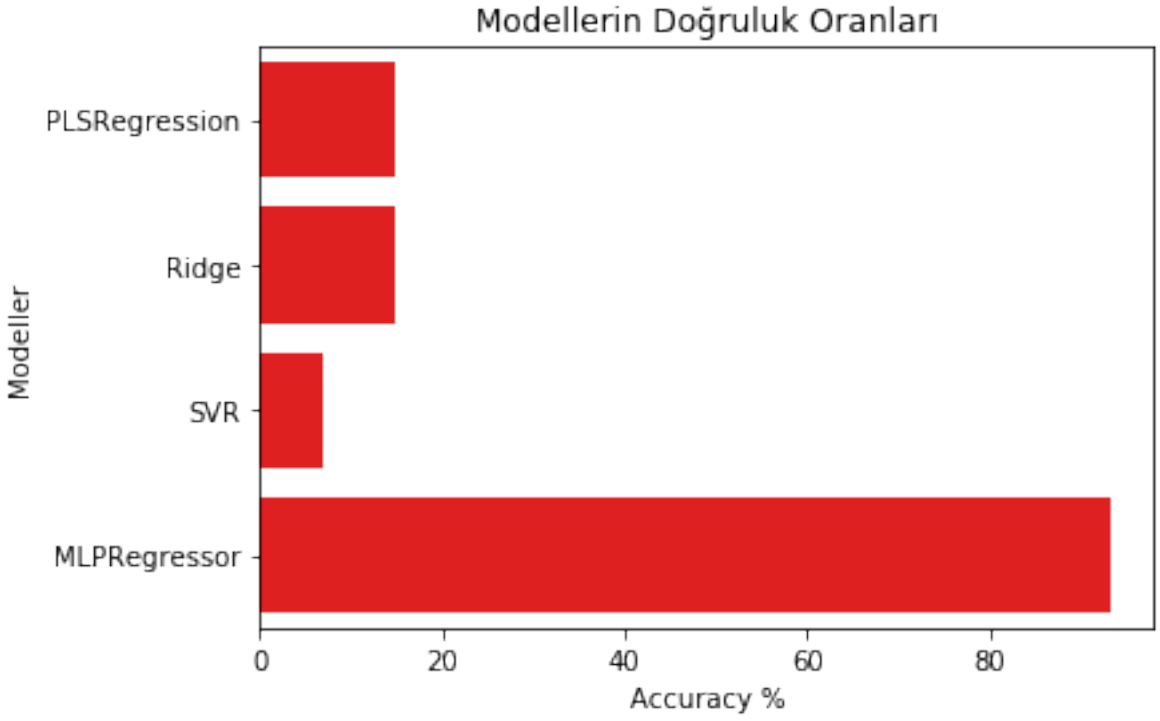
Şekil 4.24: RR model artık veri grafiği



Şekil 4.25: SVR model artık veri grafiği



Şekil 4.26: YSA model artık veri grafiği



Şekil 4.27: 4 modelin doğruluk oran grafiği

Yukarıdaki 4 algoritmayla geliştirilen model hata dağılım grafiklerine baktığımızda, YSA model tahmin veri grafik dağılımında, tahmin edilen değerlerin yayılımının dengeli olduğu görülmektedir. Ayrıca, YSA modelinin ölçüm metrik sonuçlarına göre de en yüksek tahmin başarısına sahip olduğu gözlemlenmiştir. Sonuç olarak, fonların fiyat tahmin için geliştirilen model çalışmasında, YSA modeli tahmin edilen fon fiyat değerinin birim pay başına düşen hata oranı, RMSE değerine göre (+/-) 6.2 olarak bulunmuştur. Model başarı yüzdesi ise, R2 ölçüm metrik sonucuna bakılarak %90 üzerinde bir tahmin başarısı olarak elde edilmiştir. Son olarak, 4 modelin tahmin başarısı R2 ölçüm metriğine göre yapılmış olup, şekil 4.27'de de model karşılaştırması yapılmıştır. Ancak nihai model karşılaştırılmasının yüzdelik sonucu R2 ile değerlendirilemez. Diğer ölçüm metrik sonuçları da göz önüne alındığında regresyon problemleri için PLSR, RR ve SVR modellerinin kullanılabilirliklerini yitirmedikleri tespit edilmiştir. Daha doğru sonuçlar için PLSR, RR ve SVR modellerin optimizasyon bölümlerinde modeller daha geniş deneysel değer aralıkları ile test edilebilir ve veri setlerinden korelasyon ilişkisi zayıf olan değişkenler çıkarılarak modeller tekrar değerlendirilebilir.

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
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
8. !!!!Trafalis, T.B, Ince, H. (2000). Support Vector Machine for Regression and Applications to Financial Forecasting. In: IJCNN 2000: Proceedings of the IEEE-INNS-ENNS International Joint Conference on Neural Networks: Volume 6 edited by Shun-Ichi Amari, et al., içinde 6348, IEEE Computer Society. !!!!—metot bolümden kullanılmış(ref: 24)
9. !!! Chen, Y., Tan, H. (2017). Short-term Prediction of Electric Demand in Building Sector Via Hybrid Support Vector Regression, Applied Energy, Volume 204, Pages 1363-1374. !!!!—-!!!! —metot bölümde kullanılmıştır(ref:27)
10. Lameski, P., Zdravevski, E., Mingov, R., & Kulakov, A. (2015). SVM Parameter Tuning with Grid Search and Its Impact on Reduction of Model Over-fitting. *RSFDGrC*.
11. Kor, K. (2015). Penetration Rate Optimization With Support Vector Regression Method, Yüksek Lisans Tezi, Fen Bilimleri Enstitüsü, İstanbul Teknik Üniversitesi, İstanbul.
12. https://scikit-learn.org/stable/auto_examples/svm/plot_rbf_parameters.html, SVM hiper-parametre
13. !!!!Adıyaman, F. (2007). Talep Tahmininde Yapay Sinir Ağlarının Kullanılması, Yüksek Lisans Tezi, Fen Bilimleri Enstitüsü, İstanbul Teknik Üniversitesi, İstanbul. !!!!—metot bölümde kullanılmıştır(ref:31)

14. !!!Ataseven, B. (2013) Yapay Sinir Ağları ile Öngörü Modellemesi. Dergipark, s.101-115, İstanbul. !!!!—metot bölümde kullanılmıştır(ref:32)
15. !!!Akkurt, A. (2005). Yapay Sinir Ağları Ve Türkiye Elektrik Tüketimi Tahmin Modeli, Yüksek Lisans Tezi, Fen Bilimleri Enstitüsü, İstanbul Teknik Üniversitesi, İstanbul. !!!!—metot bölümde kullanılmıştır(ref:33)
16. !!!Kuş,Z.(2019). MikrokanonikalOptimizasyonAlgoritmasıileKonvolüsyonelSinirAğlarında Hiper Parametrelerin Optimize Edilmesi, Yüksek Lisans Tezi, Fen Bilimleri Enstitüsü, Fatih Sultan Mehmet Vakıf Üniversitesi, İstanbul. !!!!—metot bölümde kullanılmıştır(ref:35)
17. Schilling, N., Wistuba, M., Drumond, L., & Schmidt-Thieme, L. (2015). Hyperparameter Optimization with Factorized Multilayer Perceptrons. *ECML/PKDD*.
18. Weissbart,L., Picek, S., Batina,L. (2019). On the Performance of Multilayer Perceptron in Profiling Side-channel Analysis, IACR Cryptol, ePrint Arch, Report 2019/1476