Pazarlama Karması Modeli (Marketing Mix Model)

Faruk Ayaz1

*1Akdeniz Üniversitesi, Yönetim Bilişim Sistemleri*

**Özet**

Bu makalenin amacı, Pazarlama Karması Modelinin (Marketing Mix Model) ne olduğu, neler içerdiği, nasıl kurulacağı, ne gibi katkılar sağladığı ve gereksinimlerinin neler olduğu sorularının cevaplanması ve bu konu hakkındaki araştırma sonuçlarının paylaşılmasıdır. Makale kapmasında yaptığımız araştımalar ve deneysel işlemler için bir veri kümesi kullanmıştır. Elde edilen veri kümesi üzerinde keşifçi veri analizi, öznitelik seçimi ve parametre optimizasyonu işlemleri yapılmıştır. Veri analizleri sonrasında Python programlama dili kullanılarak makine öğrenmesi modelleri oluşturulmuş ve bu modeller arasından en iyi performansı elde edilen modellerle aşırı parametre optimizasyonu yapılmıştır. Aynı zamanda değişkenlerin esneklik değerleri hesaplanmıştır. Son olarak veri kümesi Optiscorer Engine’yle skorlanmış ve Pyhton üzerinde elde edilen değerle karşılaştırılmıştır.

***Anahtar Kelimeler:*** *Pazarlama, Veri Bilimi, Veri Analizi*

**Summary**

The purpose of this article is to explain the Marketing Mix Model. It is the sharing of research results on what this modeling is, what it includes, how it will be set up, what contributions it provides and what its needs are. In our study, researches have been made on the Marketing Mix Model to address these questions. It used a dataset for the research and experimental procedures we have done in the article grab. Exploratory data analysis, feature selection and parameter optimization processes were performed on the obtained suitable data set. After data analysis, machine learning models were created using the Python programming language, and extreme parameter optimization was performed with the models that gave the best performance among these models. At the same time, the elasticity values ​​of the variables were calculated. Finally, the data set was scored with Optiscorer Engine and compared with the value obtained on Pyhton.

**Key Words**: Marketing, Data Science, Data Analysis

**1.Giriş ve Tanım**

**1.1 Pazarlama Karması Modeli Nedir?**

Pazarlama Karması Modeli (Marketing Mix Model, MMM), çok değişkenli regresyonlar, satış ve pazarlama değerleri içeren istatistiksel analizlerdir[1]. Şirketlerin marka değerlerine göre tüketicinin tepkisi hakkında "ne yaparsam satışlarım artar" sorusunun yanıtlanmasını sağlar. İyi bir modelleme tutarlı veri kaydı, tüketici tepkisi, hem kontrol edilebilir (fiyat, reklam harcaması vb.) değişkenler hem de kontrol edilemeyen ( mevsimsellik, gün, hafta vb.) değişkenler içermelidir [2]. Pazarlama karması modelinin kullanımı her geçen gün artmaktadır. Şirketler de modeli değerlendirmeye ve kullanmaya artan bir ilgi duymaktadırlar. Modelin kullanımının artması nedeniyle pazarlama karması modellemesi şirketlerin pazarlama bütçelerinde önemli bir rol almaya başlamıştır [3].

**1.2 Veri Bilimi ve Pazarlama Karması Modeli**

Pazarlama alanında yürütülen faaliyetlerin kayıt altına alınması veri biliminin pazarlama faaliyetlerinde kullanılmasına olanak sağlamıştır. Veri bilimi pazarlama faaliyetlerini belirlemede ve elde edilen sonuçların değerlendirilmesinde önemli rol oynamaya başlamıştır. Günümüzde Pepsi, Coca Cola, Starbucks gibi firmaların aktif olarak modellemeyi pazarlama faaliyetlerini belirleme de kullandıkları bilinmektedir [4].

Yapılan akademik çalışmalarla modelin işlevselliği konusunda araştırmalar yapılmıştır. Micheal J. Wolfe ve John C. Crotts [5] Çalışmalarında turizm sektöründe yapılan pazarlama yöntemlerinin etkisini ölçmek istemişlerdir. Makale kapsamında yapılan araştırmalarda kullanılan veri bir müzenin ziyaretçi veri kümesi kullanılmıştır. Müzeye yapılan daha önceki ziyaretlerinden oluşan veri kümesi kullanılarak bir model oluşturulmuştur. Bu modelden çıkan sonuçlara göre yeni bir pazarlama stratejisi belirlenip ve bu strateji uygulanmaya başlanmıştır. Yapılan bu pazarlama stratejisi 77 gün uygulanmış ve elde edilen yeni verilerin sonuçları bir önceki pazarlama stratejisine göre daha olumlu sonuçlar elde edildiği tespit edilmiştir .

Asgarnezhad Nouri ve Milad Soltani[6] makalelerinde kullanmış oldukları veri kümesi Kıbrıs'a tatile gelen turistlere sorulan sorular ve bu soruların cevaplarıyla oluşturulmuş bir veri kümesidir. Bu makaleye konu olan araştırma yabancı turistlerin Kıbrıs’ta tatil yapma konusundaki yönelimlerini olumlu yönde değiştirmeyi amaçlamaktadır. Makale sonucunda Kıbrıs'ta yabancı turistin artması için uluslararası seminerler ve uluslararası festivallere daha çok ev sahipliği yapması gerektiğini fikri ortaya konmuştur. Aynı zamanda turistlerin ülkeye gelebilmesi için vize ve rezervasyon işlemlerinin internet ortamında daha ulaşılabilir hale getirilmesi gerektiği sonucuna varılmıştır.

**1.3 Makale Metadolojisi**

Bu makalede yapılmak istenen araştırma veri bilimi tekniklerini kullanarak bir pazarlama karması modeli kurmaktır. Bu modeli kurmak için kullanacağım veri kümesi bir markanın belirli bir ürünün iki yıl boyunca haftalık toplam satışlarını ve o hafta içinde yapılan pazarlama harcamalarını içermektedir. Araştırmamızda kurmak istediğimiz modele uygun bir veri kümesi bulmak için Kaggle platformundan yararlanılmıştır. Aramalar sonucunda bulunan veri kümesi, keşifçi veri analizi faaliyetlerini gerçekleştirebilmek için Knime programı kullanılmıştır. Knime, kod yazmaya gerek duymadan veri analiz akışları çizebileceğimiz bir platformdur. Knime programında ön hazırlık işlemleri, öznitelik seçimi ve parametre optimizasyonu işlemleri yapılmıştır. Bu işlemler sonucunda elde edilen bilgiler doğrultusunda JupyterLab çalışma alanında Pyhton programlama dili kullanılarak uçtan uca veri analizi ve esneklik değerlerinin hesaplanma işlemleri yapılmıştır. Son olarak veri kümemizi AutoML yaklaşımıyla oluşturulan Optiscorer ürünüyle test edilip elde edilen bütün sonuçlar değerlendirilmiştir. Bu sonuçlar doğrultusunda pazarlama faaliyetlerinin satışlar üzerindeki etkilerini doğru olarak hesaplamak ve elde edilen sonuçların bir sonraki pazarlama stratejilerini düzenleme de kullanılabilir hale getirmek istenmektedir.

**2. Pazarlama Karması Modelinin Hedefleri ve Gereksinimleri**

Pazarlama Karması Modeli, projelerinin amacı daha önceki pazarlama performanslarını hesaplayarak gelecek pazarlama harcama getirilerini geliştirmektir. Modelden elde edilen sonuçlar pazarlama bütçesindeki taktikleri, ürünleri, segmentleri, zamanı ve pazarları yeniden düzenlememize yardım eder. Bütün pazarlama taktiklerinde kaliteli veri, yeterli zaman, ürün, demografik özellikler ve market çeşitliliği projeye dahil edilmelidir. Her bir proje veri toplamaktan başlayıp, optimizasyon ve gelecek stratejileriyle biten 4 farklı evreden oluşmaktadır. Bu evreler aşağıdaki gibidir.

1. Verinin Toplanması ve Bütünlüğü

2. Modelleme

3. Model Tabanlı İş Ölçümleri

4. Optimizasyon ve Simülasyon

**2.1. Verinin Toplanması ve Bütünlüğü**

Araştırmamız için istatiksel modellemeye uygun olacak şekilde çalışan verileri toplamamız gerekmektedir. Hangi ürünün analizinin yapılacağını belirlemek, hangi zaman aralığına bakılacağını belirlemek, kullanılacak birim zaman boyutunun belirlenmesi, hangi pazarın modellemesinin yapılacağını belirlenmesi, satış performansı ölçümlerinin belirlenmesi, marka marj oranları ve pazarlama taktik harcamaları maliyetleri belirlenmelidir.

**2.2. Modelleme**

Kullanılan istatistiksel model genel olarak satıcı tarafından belirlenir. Modelin işlevsel olması için reklam verenle beraber modelin sorularımıza doğru cevaplar verip veremeyeceğinden emin olmamız gerekmektedir. Marka müdürleri, Pazarlama Karması Modeli kurulurken kendi analiz ekipleriyle bu aşamada projeye dahil olmalıdır. Analiz ekibinin istatiksel detaylar ve özelleştirmelerin içinde bulunması çok önemlidir. Bu konuda oluşabilecek herhangi bir soru veya kaygı model kuran kişiyle paylaşılmalıdır.

**2.3. Model Tabanlı İş Ölçümleri**

Pazarlama Karması Modelinden elde ettiğimiz sonuçlarla bizim daha önceden cevaplanmasını istediğimiz sorularımızın karşılaştırılmasını yapmamız gerekmektedir. Projemiz satışımızı etkileyen her bir taktiğin sonuçlarını bize çıktı olarak verecektir ve bu sonuçları paylaşmadan önce satıcıyla görüşüp bu sonuçların daha önce cevaplamayı hedeflediğimiz sorulara cevap olup olmadığını kontrol etmemiz gerekmektedir.

Pazarlama Karması Modelinin temel çıktısı satışların ayrıştırılmasıdır. Her bir taktiğin satışlar üzerindeki etkisi gösterilir. Bu çıktı üzerinden farklılıkların asıl nedenlerini, artış sağlayabilecek pazarlama taktiklerini görebiliriz. Genel olarak bu nedenler satış takımı tarafından kontrol edilmeyen değerlerdir. Artış gözlemlenen taktikler ise tam tersidir pazarlama ve satış takımları tarafından kontrol edilebilir. Projemizde var olan bütün medya kampanya taktiklerinin artışlarını ölçümleyebilmemiz gerekmektedir. Bu şekilde var olan kampanyanın devam edip etmemesi gerektiğine karar verebileceğiz.

**2.4. Optimizasyon ve Simülasyon**

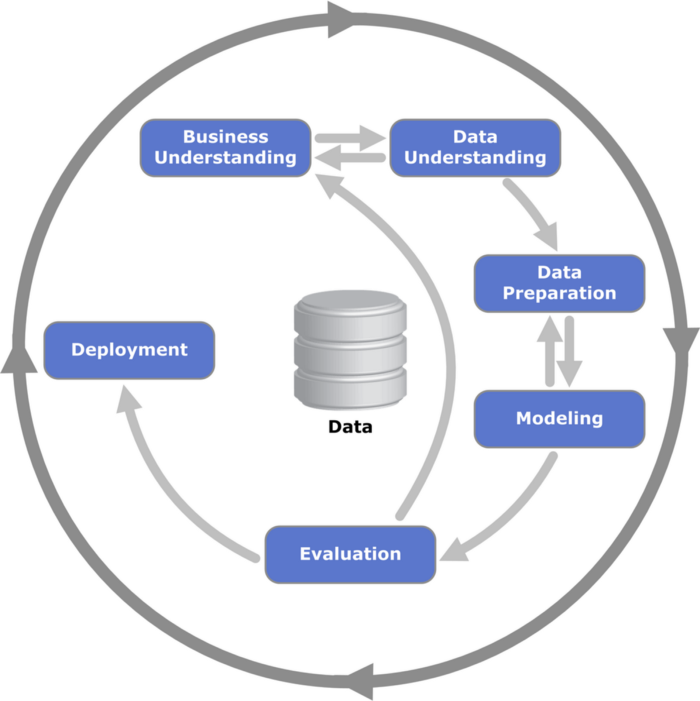
Pazarlama KarmasıModellemesinin son evresi elde ettiğimiz yeni çıktıları bir sonraki pazarlama planlamasının yeni girdileri haline gelecektir. Modeli kurduktan sonra optimizasyon ve simülasyon denemeleri yapabilirisiniz. Denemeler bir sonraki model oluşturma konusunda fikirler sağlayacaktır. Bu denemeler her bir pazarlama taktiğinin satışa olabilecek etkisini gösterecektir. En iyi senaryoyu oluşturabilecek pazarlama planın girdilerini bize çıktı olarak verecektir. Bu işlemler sonucu elde ettiğimiz değerler bizim ihtiyaçlarımızı karşılayacak durumda olacaktır. [7]

**3. Uygulama**

Uygulamalı olarak veri analizi yaptığım kaynaklardan birisi Knime programıdır. Knime, kod yazmaya ihtiyaç duymadan programdaki nodları kullanarak uçtan uca veri analizi yapabilme imkanı sağlayan bir programdır. Uçtan uca veri analizini yapabilmek için takip edilmesi gereken bir süreç vardır. Bu sürece CRISP-DM (Cross Industry Standard Process Model for Data Mining) denir.

**3.1 CRISP-DM**

CRISP-DM süreci yaygınlaşan veri analizi kullanımdan sonra bu kullanımın daha etkili ve doğru bir şekilde yapılmasını amaçlayan bir süreçtir. Bu süreçte olan 6 aşamanın sırasıyla uygulanması halinde sizi doğru bir veri analiz sonucuna götürmesi mümkündür. CRISP-DM sürecinin kullanılmasının bir başka sebebi ise endüstride veri analizi konusunda bir standartlaşma sağlanmaya çalışılmasıdır.[8]



Grafik 1 - CRISP-DM

**3.1.1 İşin Anlaşılması (Business Understanding)**

İşin Anlaşılması, CRISP-DM sürecinin ilk aşamasıdır. Yapılacak olan projenin ne amaçla yapılmak istendiğini anlamak veri analizi aşamalarında çok önem taşımaktadır. Bu aşamada yapılacaklar işin anlaşılması, işe dair daha önce yapılan araştırmalara göz atılması ve iş sektöründe karşılaşabileceğimiz sorunların önceden saptanmaya çalışılması gerekmektedir.

**3.1.2 Verinin Anlaşılması (Data Understanding)**

Verinin Anlaşılması alanında istediğimiz değerlere ulaşabildiysek ve projenin amacını anlayabildiysek bir sonraki aşama olan Verinin Anlaşılması aşamasına geçebiliriz. Bu aşama ise amacımıza uygun verileri toplamamız, verilerin neyi temsil ettiğini anlamamız ve elde ettiğimiz verilerin amacımıza uygun olup olmadığına karar vermemiz gerekmektedir.

**3.1.3 Veri Ön Hazırlık (Data Preprocessing)**

Verilerimizi topladıktan sonra elde ettiğimiz verilerde modelimizi oluştururken sorunlar çıkartabilecek veri tipleri, uç değerler ve eksik verilere dikkat etmemiz gerekmektedir. Bu tür sorunlar modelimizi oluştururken model de hatalara ve yanlılığa sebep olabilir. Bu tür sorunlar çözülürken kurulan modelin amacı göz önünde bulundurulup ona göre çözümler bulunmaya çalışılmalıdır.

**3.1.4 Modelleme (Modeling)**

Bu aşamadan önce yapılan işlemlerden emin olunup bu aşamaya geçilmesi gerekmektedir. Veri Ön Hazırlık aşamasında yapılanlar yanlışlar modelleme işlemi sırasında sorunlar ortaya çıkartabilir. Modelleme işlemi yapılırken kurulan modelin veriye uygun olduğu gözlemlenmelidir. Model sonrası çıkacak bilgilerin anlamlı ve faydalı olmasına özen gösterilmelidir.

**3.1.5 Değerlendirme (Evalutaion)**

Bu aşamada bir önceki modelleme aşamasında ve kurulan modellerden elde ettiğimiz bilgiler doğrultusunda çıkan değerlerin bizim istediğimiz sonuçlar olup olmadığı ve hangi modelin bu isteklerimizi daha iyi yerinde getirdiği gözlenmeye çalışılır. Eğer çıkan sonuçlar bizim elde etmek istediğimiz cevapları vermiyorsa CRISP – DM aşamalarından ilki olan İşin Anlaşılması aşamasına geri dönüp ortadaki sorun anlaşılmaya çalışılır.

**3.1.6 Uygulama (Deployment)**

Bu aşama CRISP-DM sürecinin son aşamasıdır. Bir önceki aşamadan gelen bilgiler doğrultusunda oluşturmak istediğimiz son modeli oluşturabiliriz. Aynı zamanda proje ilerlerken ortaya çıkan yeni ihtiyaçlar ve istekler doğrultusunda yeni modelimize yön verebilir ve daha önceki sorularımıza cevaplar bulabiliriz.

**3.2 Knime ile Keşifçi Veri Analizi**

Makalenin amacı Pazarlama Karması Modeli kavramını açıklamaktır. Bu amaç doğrultusunda oluşturulmuş olan modellerden biri Knime platformu üzerinde hazırlanmıştır. Bu platformda veriyi okuma, veri görselleştirme, veri ön işleme, modelleme ve değerlendirme işlemleri yapılmıştır.

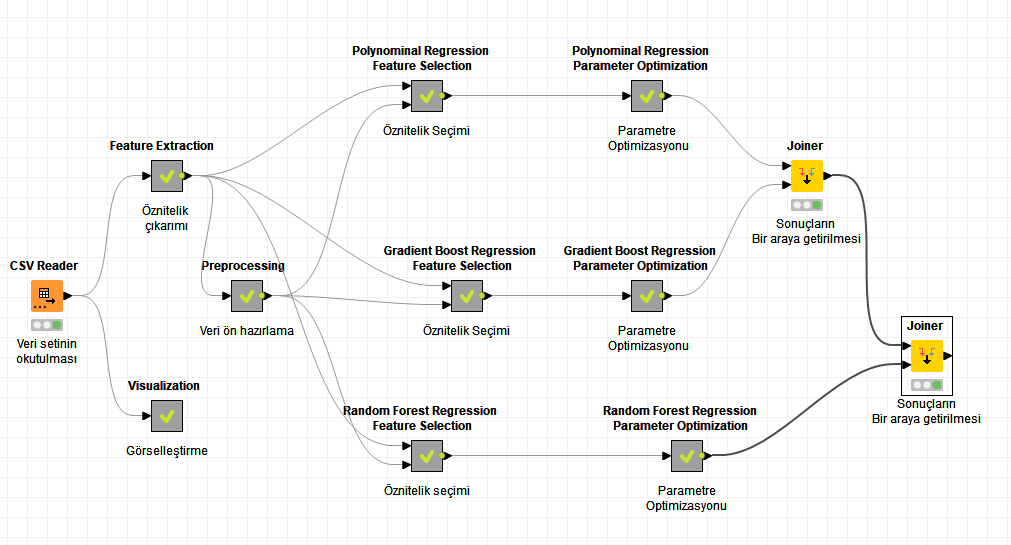
**3.2.1 Veri Kümesinin Açıklanması**

* Veri kümesine Kaggle platformunda rastlanılmıştır [9].
* Veri kümesiyle Kaggle platformunda yapılmış sadece bir proje bulunmaktadır. Bu proje de veri kümesiyle uçtan uca analiz yapılmıştır. Bu analiz veri kümesi hakkında ilk izlenimlerimizi edinmemizi sağlamaktadır.
* Veri kümesinin bir virgülle ayrılmış değerler “coma separatted values, csv” dosyasıdır.
* Veri kümesinin belirli bir markanın belirli bir ürününün 2 yıl boyunca yapılan satış değerlerini içermektedir.
* Her satır 2 yıl boyunca haftalık olarak yapılan satışlardan oluşmaktadır.
* Değişkenlerimizden bazıları ise satışlar için yapılan pazarlama faaliyetlerinin maliyet değerlerini taşımaktadır.
* Satış yapılan bu iki yılın hangi yıllar olduğu bilinmemektedir.

**3.2.2. Veri Kümesindeki Değişkenlerin Açıklanması**

* **NewVolSales:** Haftalık toplam satış miktarı.
* **Base\_Price:** Satılan ürünün o haftaki birim satış fiyatı.
* **Radio:** Radyo kanalı üzerinden yapılan pazarlama harcamaları.
* **InStore:** Mağaza içerisinden yapılan promosyon ve pazarlama harcamaları.
* **NewspaperInserts:** Kategorik bir değişkendir. O hafta gazetelere herhangi bir harcama yapılıp yapılmadığını göstermektedir.
* **Discount:** Haftalık indirim oranı.
* **TV:** Televizyon üzerinden yapılan o haftaki pazarlama harcamaları.
* **Stout:** Ürünün dayanaklılığıdır. Burada dayanıklılıktan kasıt ürünün stoklardaki tükenme tarihi bölü ürüne yapılan ziyaret sayısıdır.
* **Website\_Campaign:** O hafta hangi internet sitesi üzerinden kampanya yapıldığı gösteren kategorik değişkendir.

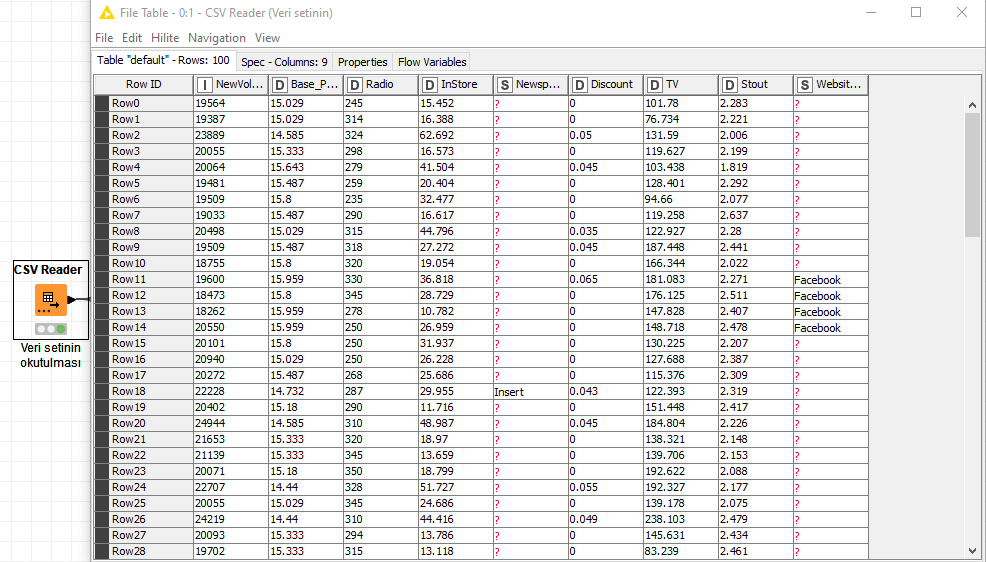
**Projenin Genel Görünümü**



Şekil 1 – Genel Görünüm

Projemizin genel görünüşü yukarıda görüldüğü gibidir. Görüntü olarak kalabalık gözükmemesi adına aynı işlem içerisinde sayılabilecek nodlarımızı birer Metanod içinde toplanmıştır. Makale boyunca bu MetaNod’lar teker teker açıklanacaktır.

**3.2.3 Verinin Okunması**

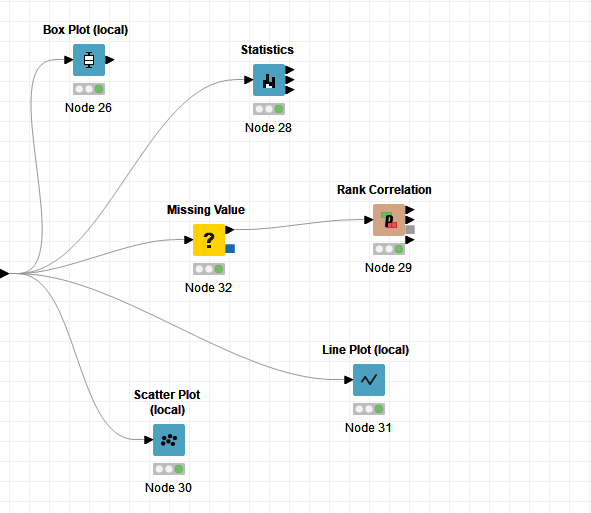


Şekil 2 – Verilerin Okunması

Verimizi okutmak için Knime’da bulunan CSV Reader nod’unu kullandık. Veriyle ilgili ilk gözlem olarak NewspaperInserts ve Website\_Campaign kolonlarındaki kayıp veriler olmuştur.

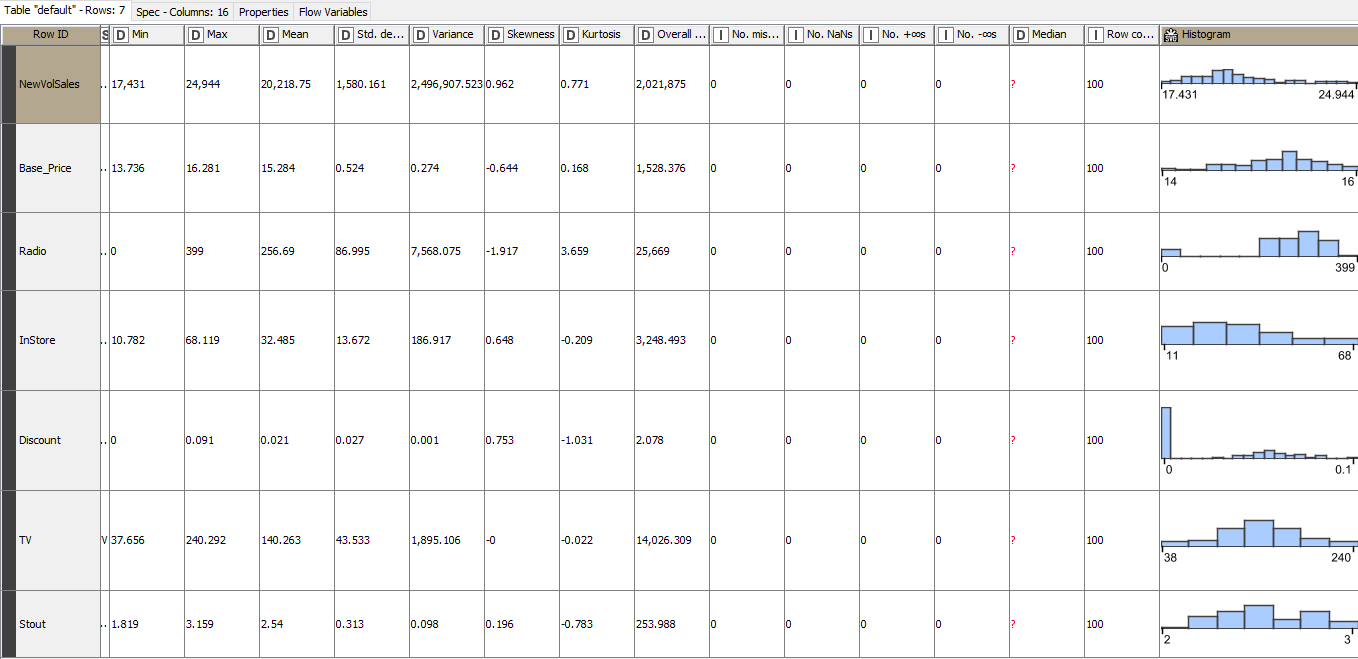
**3.2.4 Verinin Görselleştirilmesi ve Anlaşılması**

Veri görselleştirilmesi veriyi anlamak için bize önemli bir fayda sağlamaktadır. Elde edilen grafikler bize veri ön işleme aşamasında yardımcı olacak ve hangi işlemleri yapmamız gerektiği konusunda bilgiler verecektir.



Şekil 3 – Görselleştirme

1. **Veri Kümesinin İstatistikleri**

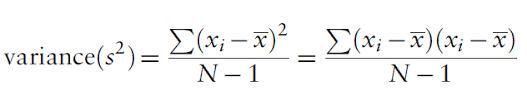


Şekil 4 – istatistikler

Şekil 4’te görüldüğü üzere bütün istatistiklerimiz bu şekildedir. Buradan sayısal değişkenlerde kategorik değişken olmadığı sonucunu çıkartabiliriz. Ayrıca ‘0’ değeri ile başlayan değerler görülüyor. Kayıp veri durumu söz konusu olursa ortalama değer yerine medyan değeri kullanmamız daha mantıklı olacaktır.

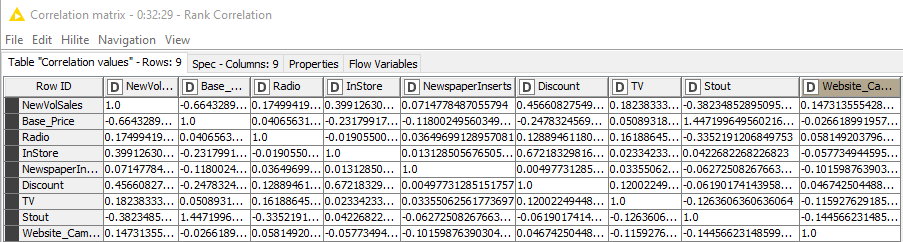
1. **Değişkenler Arasında Korelasyon Analizi**

Korelasyon analizi değişkenler arası ilişkilerin yönünü ve derecesini tanımlamak için kullanılan bir istatistiksel analiz tekniğidir. İstatistik analizi uygulamalarında verilerimiz nicel veya sıralayıcı (ordinal) ölçeğe sahip olduğunda Spearman korelasyon analizine başvurabiliriz. Spearman korelasyon analizi, normal dağılım koşulunu gerektirmediği için son derece esnek bir tekniktir. Ancak parametrik olmayan tüm analizlerin bir bedeli var. Pratik istatistik analizi uygulamalarında Spearman korelasyon katsayıları bağlamında olası problemlere hazır olmamız gerekmektedir.[10]



Formül 1 – Kolerasyon Formülü

Kolaresyon formülü Formül 1’de görüldüğü gibidir. Knime programında bu analizi bir nod ile hesaplayabilmekteyiz.

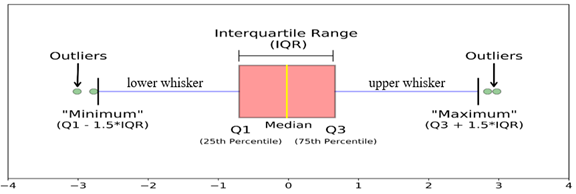


Şekil 5 - Korelasyon

Korelasyon ilişkileri Şekil 5’te görüldüğü gibidir. Base\_Price ile NewVolSales değişkenleri arasında %66’lık bir negatif korelasyon ilişkisi gözlemlenmiştir.

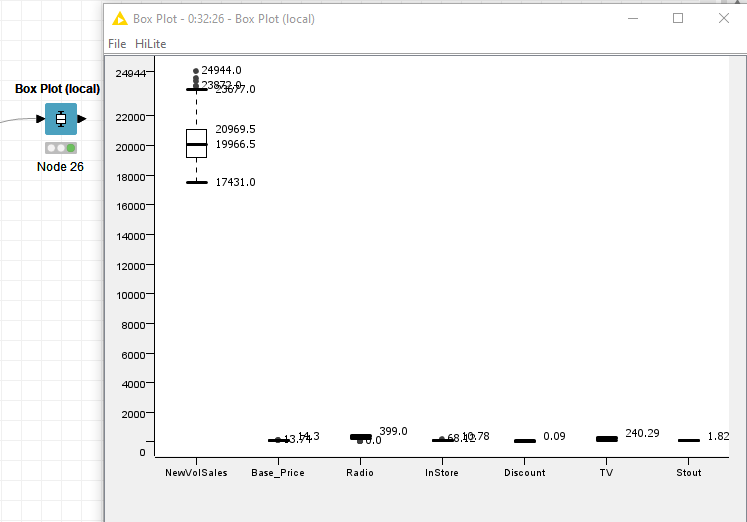
1. **Aykırı Değer Tespiti**

Veri kümesi karakteristiğine uygun olmayan ve veri kümesindeki diğer gözlemlerden belirgin bir şekilde sapan gözlemler aykırı gözlem olarak adlandırılmaktadır. Veri kümesinin gözlenen veya teorik olarak tahmin edilen örüntüsünün aksine farklı örüntüler içeren aykırı gözlemler, verinin üretilme veya ölçüm sürecinden kaynaklandığı gibi, örneklem hatasından veya örneklem dağılımına ilişkin varsayımlardan da kaynaklanmaktadır. Veri kümesinin kalitesini olumsuz olarak etkileyen aykırı değerlerin veri kümesinden çıkartılması, gerçekleştirilecek istatistiki analizlerin sonuçlarının güvenilirliği açısından önemli olacağı için, veri kümesinde bulunan aykırı değerlerin doğru olarak tespit edilerek veri kümesinden çıkartılması, normalize edilmesi veya dönüştürülerek homojenliğin sağlanması önerilmektedir.[11]



Formül 2 – Aykırı Değer Tespiti

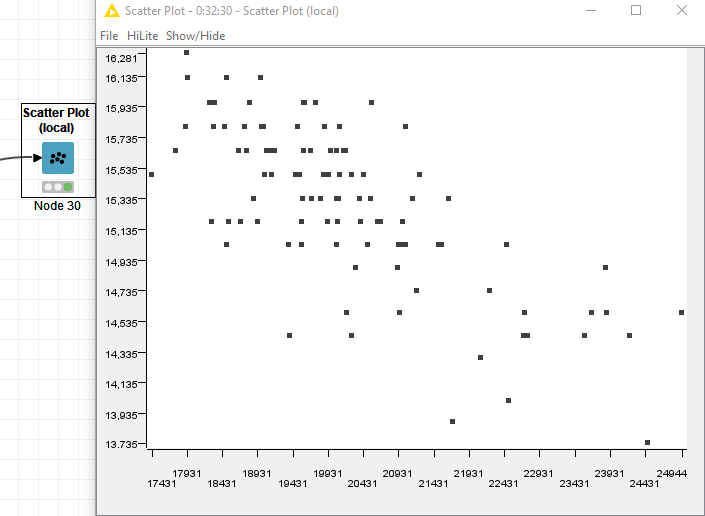
Aykırı değerlerin tespiti Formül 2’de görüldüğü gibidir. Knime programında bu analizi bir nod ile hesaplayabilmekteyiz.



Şekil 6 – Aykırı Değerler 1

Şekil 6’da görüldüğü üzere numerik değişkenlerimizin bazılarında aykırı değerler bulunuyor. Gerekli işlemler veri ön hazırlık aşamasında yapılacaktır.

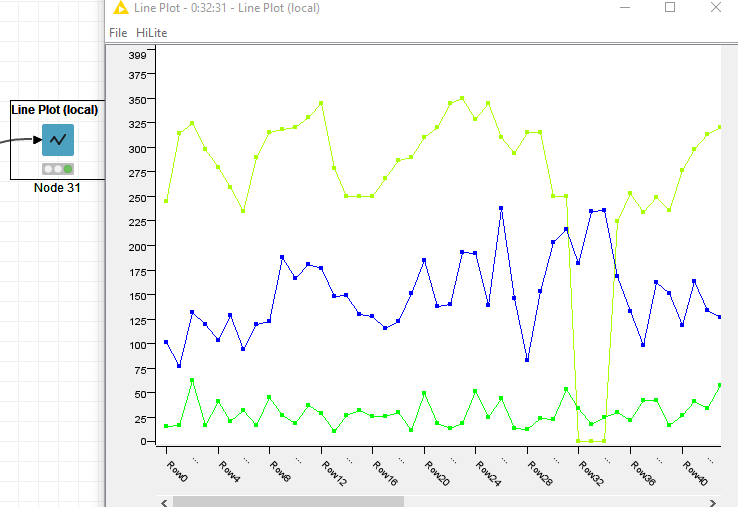
1. **Dağılım Grafiği**



Şekil 7 – Dağılım Grafiği 1

Şekil 7’de x eksenimiz hedef değişkenimiz olan NewVolSales değişkeni, y ekseni ise Haftalık birim fiyatı belirten Base\_Price değişkenimizdir. Bu durumda bu şekilden haftalık birim fiyatın düşük olduğu haftalarda toplam satışın yüksek olduğunu, haftalık birim satış fiyatının yüksek olduğu haftalarda ise toplam satışın az olduğunu gözlemleyebiliriz. Daha önce gördüğümüz bu iki değişken arasındaki negatif korelasyon ilişkisi bu şekilde açıklanabilir.

1. **Çizgi Grafiği**



Şekil 8 – Çizgi Grafiği

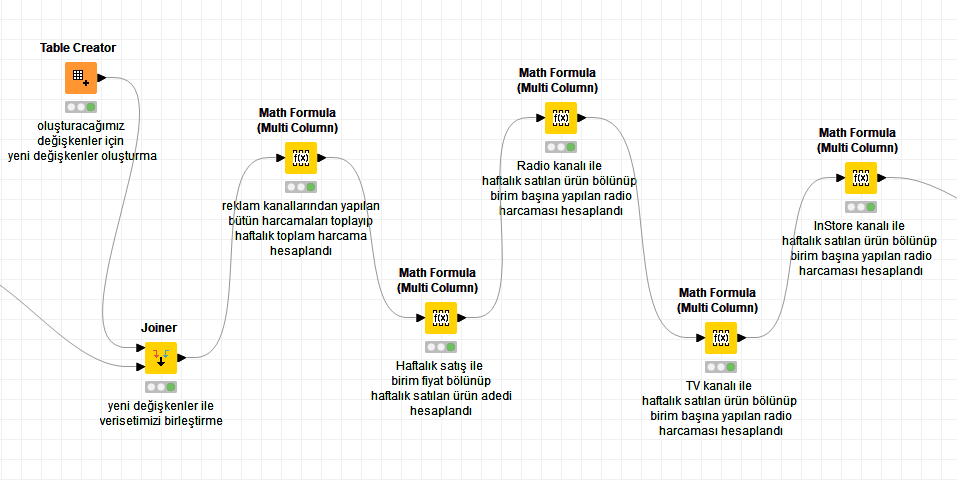
Çizgi grafiğine sadece reklam kanallarını içeren değişkenler eklenmiştir. Bu sayede 3 farklı kanala ne kadar harcama yapıldığı aynı anda gözlemlenmiştir. Sarı çizgiler Radio harcamalarını, mavi çizgiler TV harcamalarını ve yeşil çizgiler ise Instore harcamalarını göstermektedir. En çok harcama yapılan kanal önce Radio sonra TV ve en az harcama yapılan kanal ise Instore harcamalarıdır.

**3.2.5 Öznitelik Mühendisliği (Feature Engineering)**

Öznitelik mühendisliği nedir ve neden ihtiyacımız vardır. Basitçe anlatmak gerekirse bütün makine öğrenmesi algoritmaları verileri girdi olarak alıp çıktıları bilgi olarak bize geri vermektedir. Girdi için kullanılan veriler genellikle kolonlar şeklindedir ve bazen bu veri kümeleri bizim istediğimiz sonuçları çıkarmakta güçlük çekebilmektedir. Bu durumlarda algoritma daha fazla veri veya değişkene ihtiyaç duyar. Öznitelik seçimine burada ihtiyaç duyulmaktadır. Veri kümesinde halı hazırda bulunan verileri kaybetmeden onlardan yeni değişkenler oluşturarak modelimizin öğrenmesini güçlendirmekteyiz. Öznitelik mühendisliğinin iki ana hedefi bulunmakdatır. Veri kümesine yeni girdiler eklemek ve eklenen yeni girdilerle algoritma ihtiyaçlarını karşılamak ve makine öğrenmesi modellerinin performansını yükseltmektir.[12]

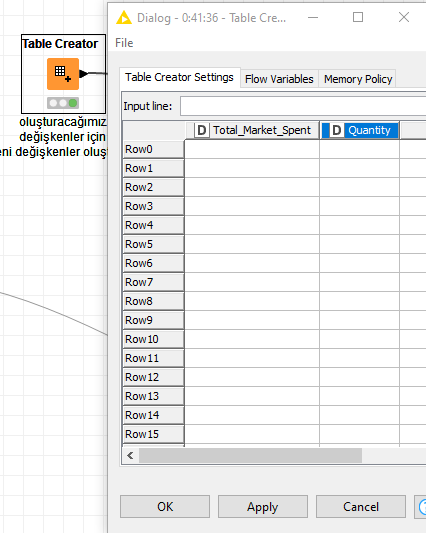
1. **Feature Engineering Metanod’u**

Verimizin daha iyi eğitebilmek için oluşturulan değişkenlerden oluşan Metanod’dur.



Şekil 9 – Öznitelik Mühendisliği

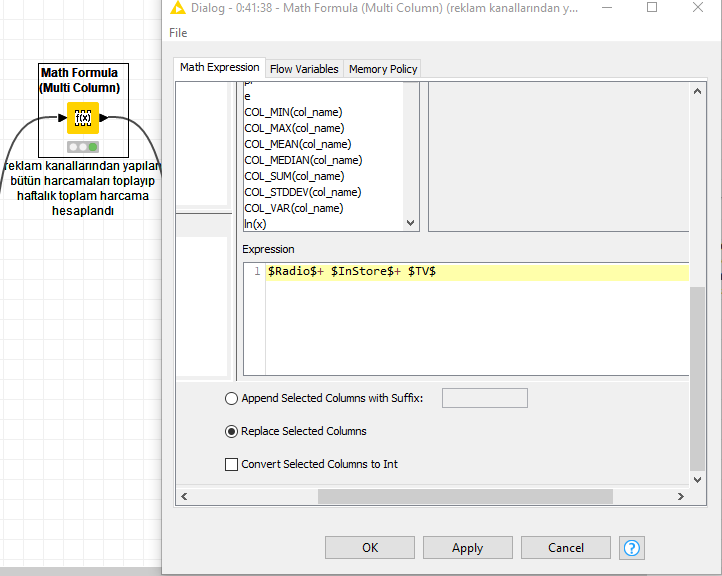
1. **Öznitelik Mühendisliği ile Yeni Değişkenler Oluşturulması**



Şekil 10 – Değişken Oluşturma

Biz burada veri kümemize ekleyeceğimiz yeni değişkenler için yeni kolonlar oluşturduk.

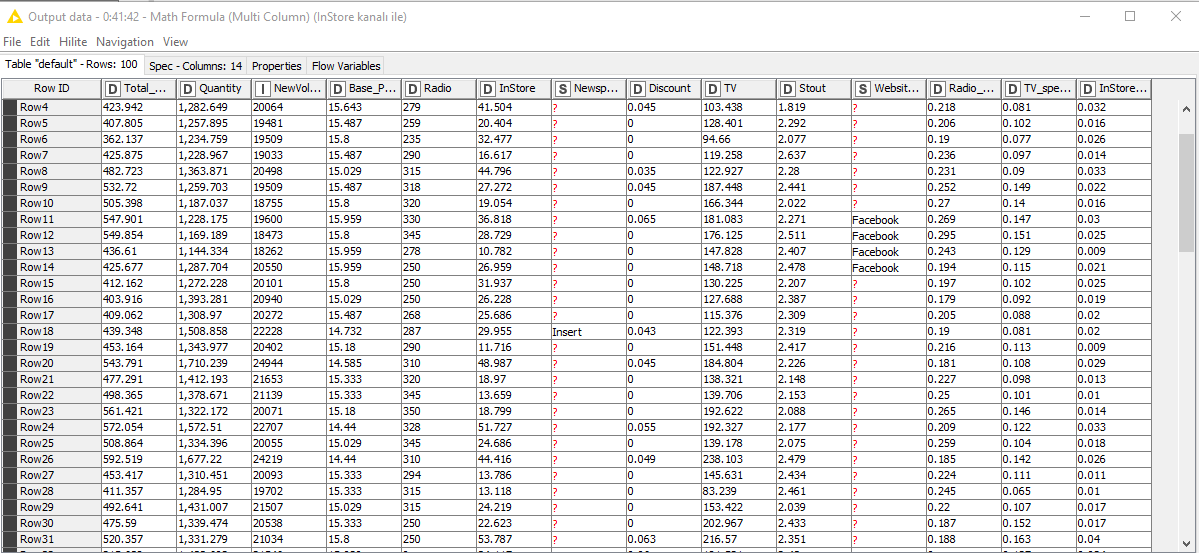
1. **Oluşturulan Yeni Değişkenlerin Değerlerinin Hesaplanması**



Şekil 11 – Yeni Değişken Förmülü

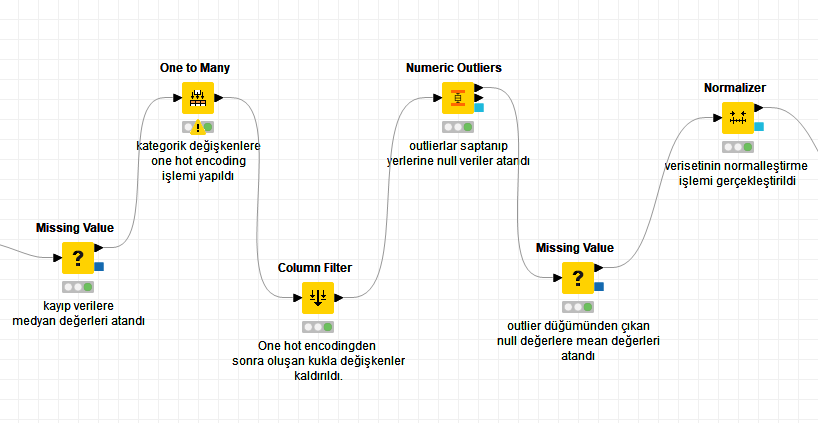
Şekil 11’de görüldüğü gibi yeni değişkenler oluşturmak için Math Formula Nod’unun Expression alanına işlem yapmak istediğiniz kolonları ve yapılmasını istediğiniz işlemleri belirterek yeni değişkenler oluşturmamız mümkündür.

**3.2.6 Veri Ön İşleme İşlemleri**



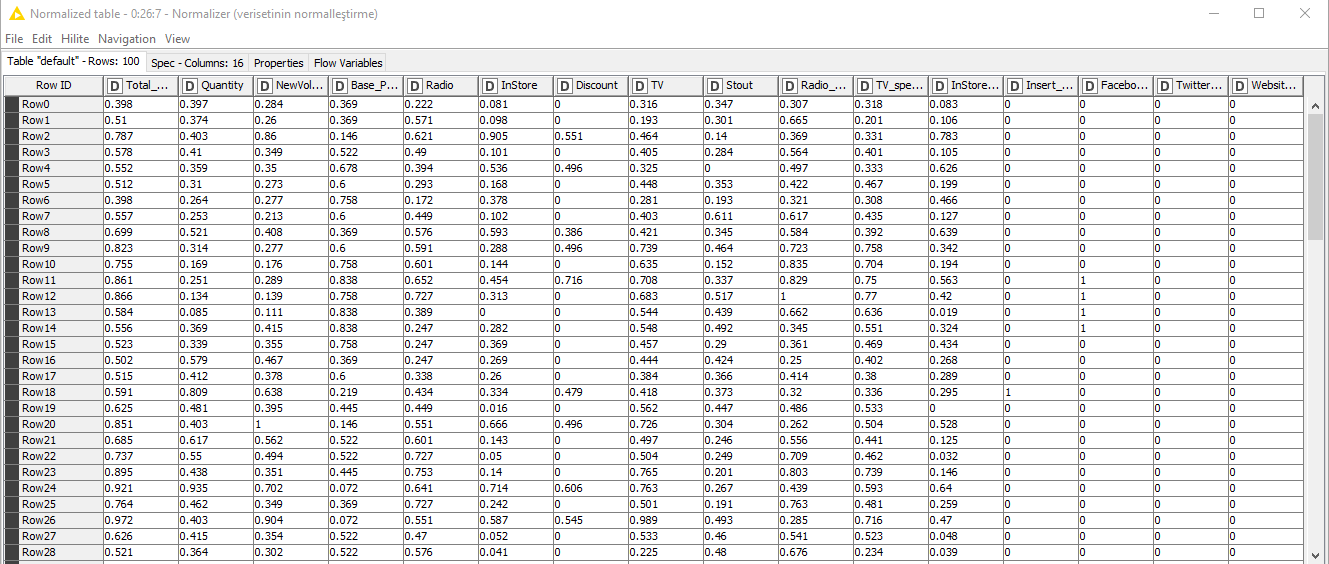
Şekil 12 – Veri Ön Hazırlık Öncesi

Veri Ön Hazırlık işlemlerinden önce veri kümemiz şekil 12’deki gibi gözükmektedir.



Şekil 13 – Veri Ön Hazırlık

Şekil 13’teki işlemlerin ardından veri kümemiz eğitim için modele hazır hale getirildi.



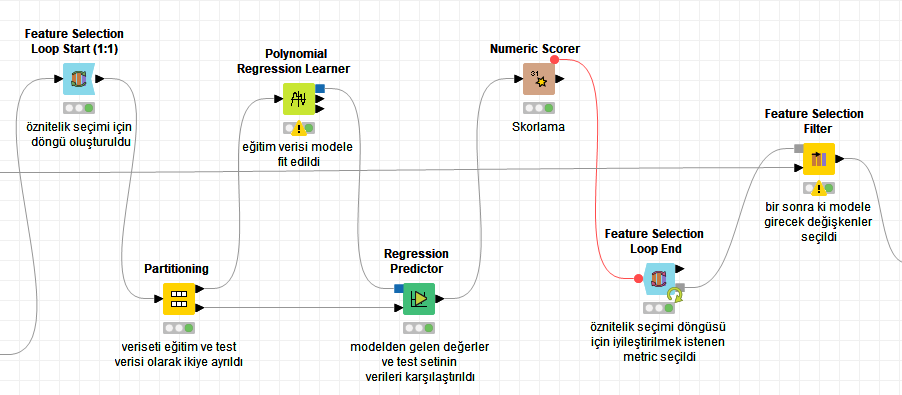
Şekil 14 – Veri Ön Hazırlık Sonrası

Şekil 14’te ile Şekil 12’ye bakarak yapılan veri ön hazırlık işlemlerini gözle görebilmekteyiz.

**3.2.7 Polinominal Regresyon**

Polinominal Regresyon projemizde kullandığımız 3 makine öğrenmesinden birincisidir. Önce Öznitelik Seçimi sonra ise Parametre optimizasyonu yaparak bu makine öğrenmesi algoritmasında verim elde etmeye çalışacağız.

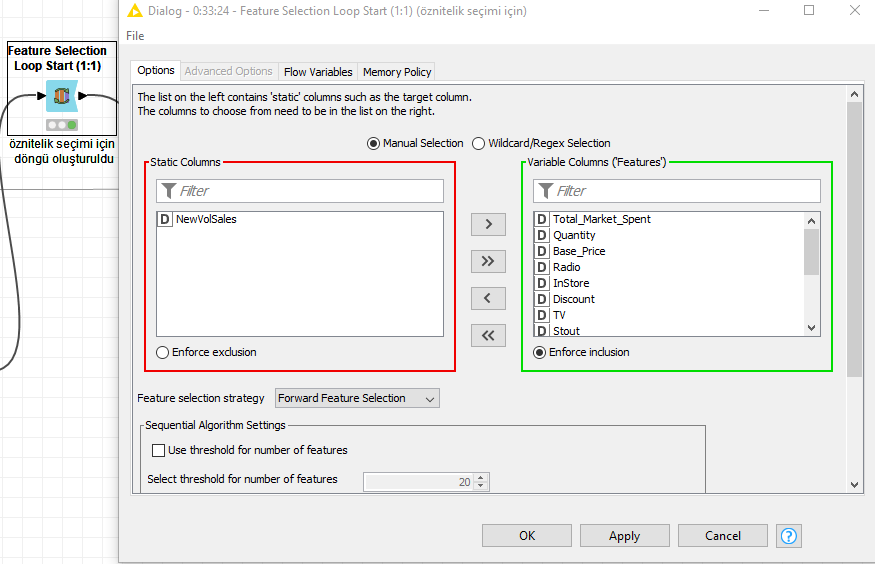
**3.2.7.1 Polynomial Regression Feature Selection MetaNod’u**



Şekil 15 – Polinominal Regresyon Öznitelik Seçimi

Feature Selection MetaNod’u şekil 15’te görüldüğü gibidir.

1. **Öznitelik Seçimi Döngüsü Başlangıcı**

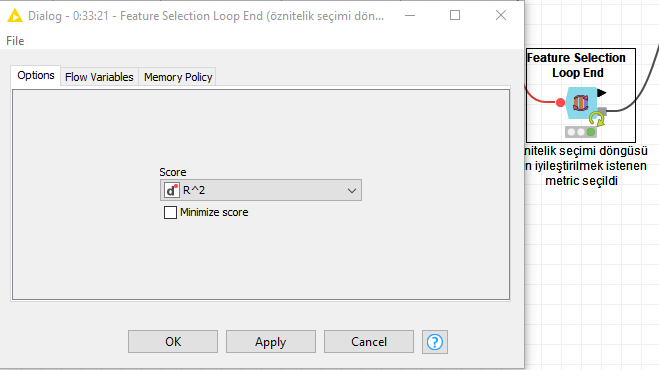


Şekil 16 – Öznitelik Seçim Döngüsü Başlangıcı 1

Feature Selection döngüsünü çalıştırmak için hedef değişkenimizi Static Columns sekmesine atıyoruz ve Feature Selection Strategy seçeneklerinden Forward Feature Selection’ı seçiyoruz.

1. **Öznitelik Seçimi Döngüsü Bitişi**

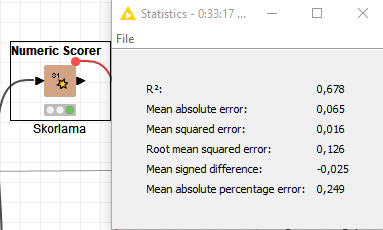
Makine öğrenmesi algoritmasını çalıştırmadan önce döngünün biteceği Nod’da maksimize veya minimize etmek istediğimiz metriği seçmemiz gerekmektedir.



Şekil 17 – Öznitelik Seçim Döngüsü Sonu 1

Öznitelik seçimi döngüsünün sonunda R2 metriğinin maksimize edilmesini tercih ettik bizim başarı metriğimiz R2metriği olacaktır.

1. **Makine Öğrenmesi Performansının Skorlanması**



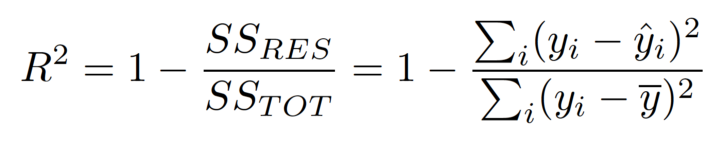
Şekil 18 – Skorlama 1

Polinominal Regresyon algoritmasından aldığımız ilk sonuçlar bu şekildedir. Elde Edilen sonuçların iyileştirilmeye ihtiyaç duyduğunu görmekteyiz.

Şekil 18’ de elde ettiğimiz değerlere metrikler denmektedir. Bu metrikler kurmuş olduğumuz modelin performansının nasıl olduğu hakkında bize bilgiler vermektedir. Bu araştırma kapsamında kurulan modeller için R2 metriğini hedef metriği olarak ele alacağız. Bunun sebebi ise R2 değerinin genel olarak 0 ile 1 arasında geldiği için anlaması ve yorumlanması daha kolay olduğunu düşünülebilir.

**R2 skoru nedir ?**

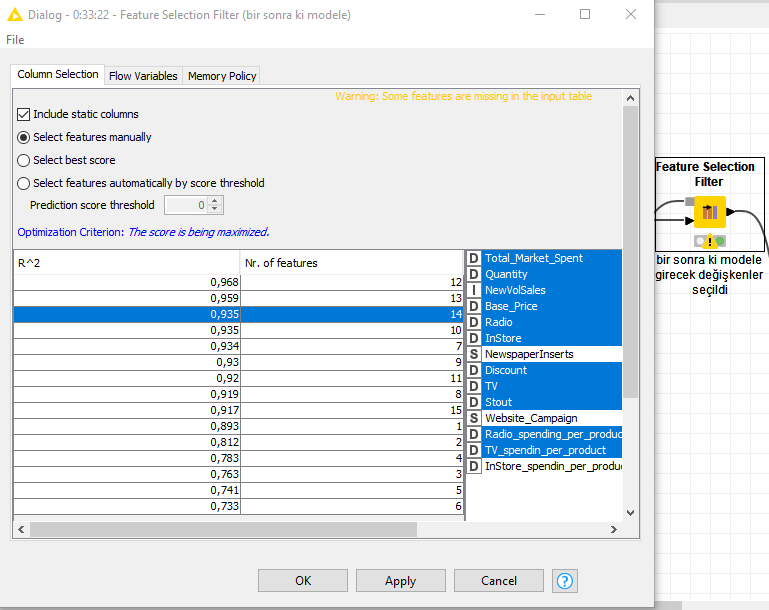
R², verilerin yerleştirilmiş regresyon hattına ne kadar yakın olduğunun istatistiksel bir ölçüsüdür. Ayrıca belirleme katsayısı veya çoklu regresyon için çoklu belirleme katsayısı olarak da bilinir. Daha basit bir dilde söylemek gerekirse R2, doğrusal regresyon modelleri için uygunluk ölçüsüdür.



Formül 3 – R Kare Formülü

Burada, SSres, model tarafından açıklanan Varyans olarak da bilinen Karesel Regresyonun Toplamıdır. SStot, Toplam Kareler Toplamıdır. “yi” Gerçek Gözlemdir. “yi\_cap” Öngörülen Gözlemdir. “y\_bar” Gerçek değerin ortalamasıdır [13].

1. **Öznitelik Seçimi Döngüsü Sonrası Değişken Seçimi**



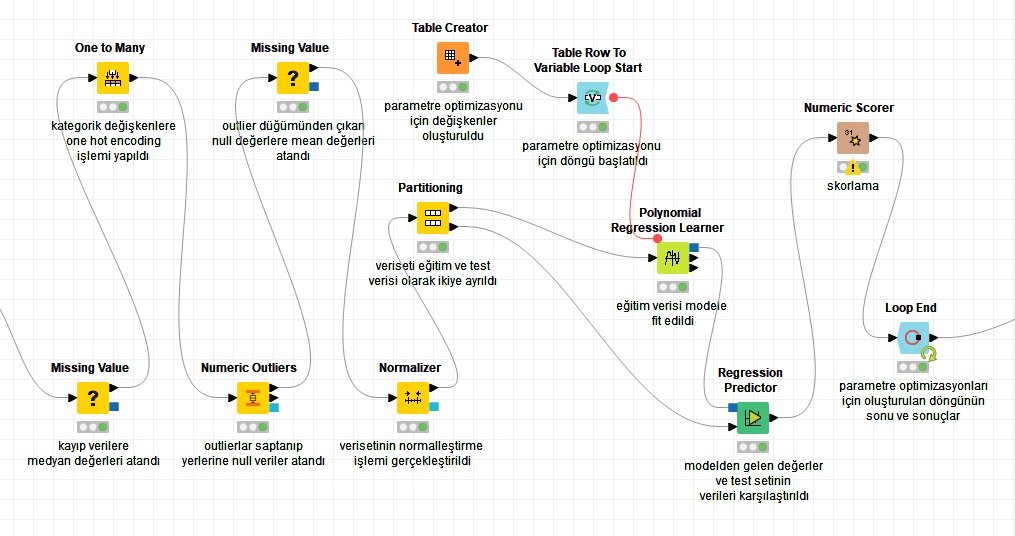
Şekil 19 – Öznitelik Seçim Filtresi

Şekil 19’da da görüldüğü üzere Feature Selection Filter Nod’unda değişkenlerin R2 metrik değerleri, bu değerlerin ortaya çıkmasını sağlayan değişkenler ve kaç değişken kullanıldığı gösterilmektedir. Bu değişkenler arasından istediğimiz değişken kombinasyonlarından birini seçip yeni kuracağımız modele yönlendirebiliriz.

**3.2.7.2 Polynomial Regression Parameter Optimization Metanod’u**

Günümüzde optimizasyon işlemleri, mühendislik tasarımlarında kullanılan önemli bir yöntemdir. Optimizasyon yöntemlerini kullanmanın önemli avantajları olduğu bilinmektedir. Gerçek sistem tasarımı için, geleneksel yöntemler deneysel formülleri kullanarak yalnızca yapılabilirliğini esas alan çözümler içerirken, optimizasyon yöntemleri gerçekçi fiziksel modellere dayalı optimum sonuçlar arar. Mühendislik tasarımında kullanılan optimizasyon yöntemleri çok sayıda işlem yapmayı gerektirir. Çünkü, optimizasyon yöntemlerinin çözmeye çalıştıkları karmaşık fonksiyonların farklı potansiyel çözümler için irdelenmesi gerekmektedir [14].

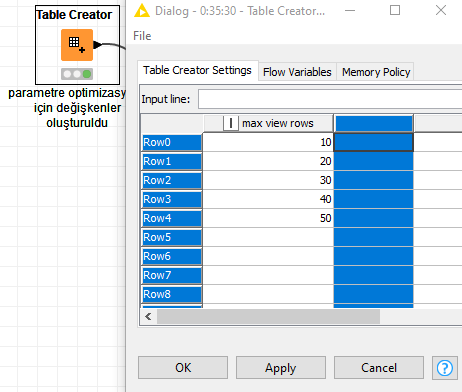
Bu çalışmada üzerinde durulan Parametre ayarlama problemi bir optimizasyon problemi olup, gerçek zamanlı optimizasyon modelleme uygulamalarını içermektedir. Dinamik verilerin kullanıldığı büyük parametre uzaylarına sahip modellerin parametrelerinin ayarlanması, sistemlerin tasarımı ve kontrolü için önemlidir. Parametre ayarlama problemlerinde çözüme ulaşmak için farklı yaklaşımlardan yararlanılabilinir. Ancak bütün yaklaşımlarda amaç, deneysel ve modelden alınan veriler arasındaki farkı en aza indirgemektir. Model parametrelerinin belirlenmesinde uygun deneysel verilerin elde edilmesi önemlidir. Modelin süreç gösterim kabiliyeti, modelde yer alan parametrelerin doğru bir şekilde belirtildiği gerçeği ile büyük oranda orantılıdır [15].



Şekil 20- Parameter Optimizasyonu 1

Şekil 20’de görüldüğü üzere bu MetaNod’da bulunan ilk 5 nod daha önce yaptığımız veri ön hazırlık işlemleridir.

1. **Parametre Optimizasyonu İçin Değişken Oluşturma**

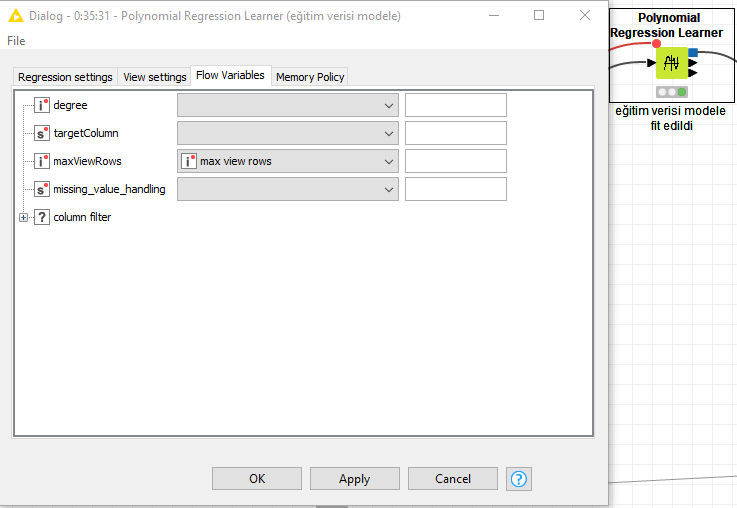


Şekil 21 – Parametre Oluşturma

Algoritmamızın içine yerleştirmek istediğimiz yeni parametremiz Şekil 21’de görüldüğü gibidir. Döngü bu değerleri teker teker deneyip sonuçlarını bize getirecektir.

1. **Parametrenin Algoritmaya Yerleştirilmesi**

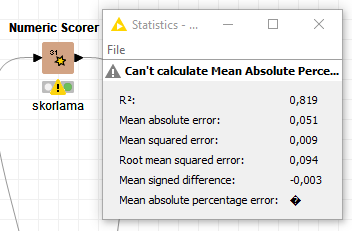
Makine öğrenmesi algoritmamızın parametrelerini optimize edebilmek için oluşturduğumuz değerleri modelimize eklememiz gerekmektedir.



Şekil 22 – Parametre Yerleştirme 1

Polynomial Regressin Learner Nod’unda bulunan Flow Variables sekmesindeki hazır olarak gelen değişkenlerle daha önceden hazırladığımız değişkenleri değiştiriyoruz.

1. **Makine Öğrenmesi Performansının Skorlanması**.



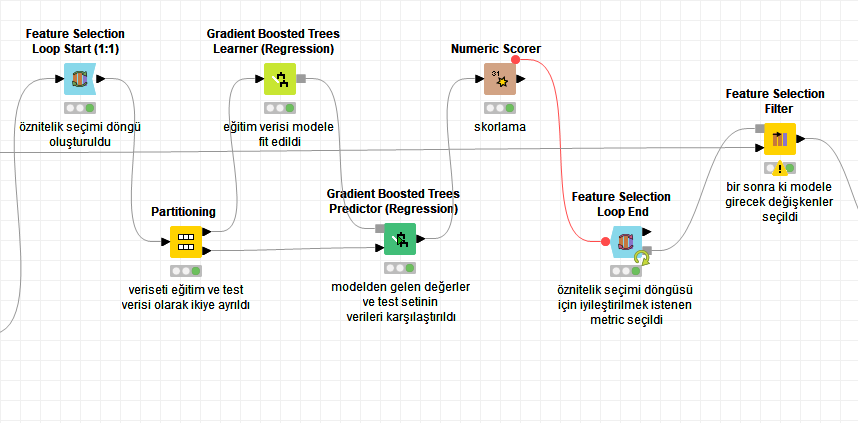
Şekil 23 – Skorlama 2

Polinominal Regresyon algoritmasının son skoru şekil 23’te gibidir. Görüldüğü üzere 0,678’den 0,819 oranına kadar artmış bir R2 skoru gözükmektedir. Yaptığımız uygulamaların işe yaradığını var sayabiliriz.

**3.2.8 Gradyan Destekli Ağaç Regresyonu**

Gradyan Destekli Ağaç Regresyonu projemizde kullandığımız 3 makine öğrenmesinden ikincisidir. Önce Öznitelik Seçimi sonra ise Parametre optimizasyonu yaparak bu makine öğrenmesi algoritmasında verim elde etmeye çalışacağız.

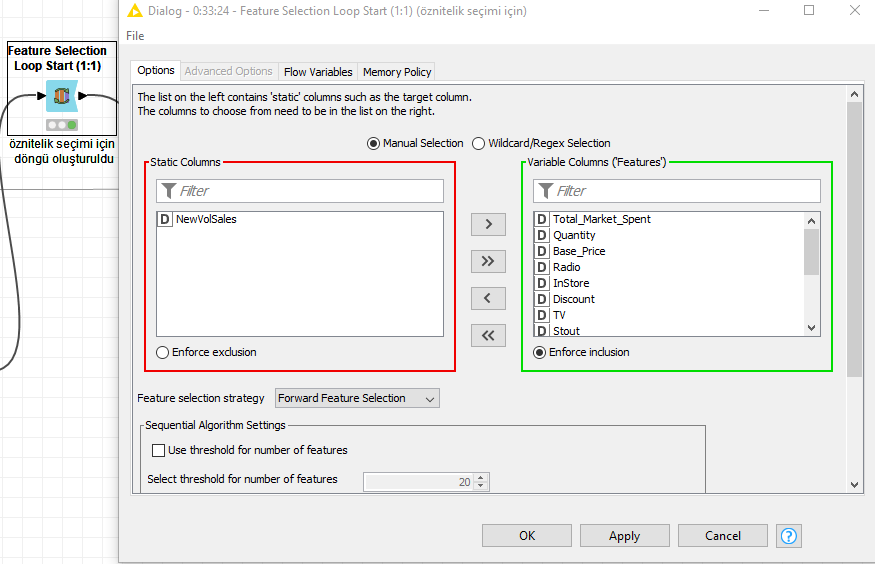
**3.2.8.1 Gradient Boost Regression Feature Selection MetaNod’u**



Şekil 24 – Öznitelik Seçimi 2

Feature Selection MetaNod’u şekil 24’de görüldüğü gibidir.

1. **Öznitelik Seçimi Döngüsü Başlangıcı**

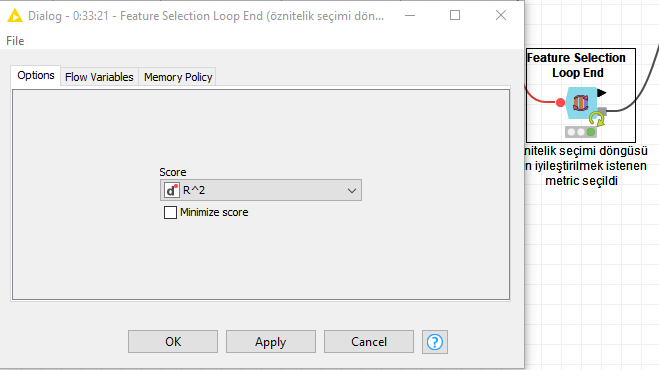


Şekil 25– Öznitelik Seçim Döngüsü Başlangıcı 2

Feature Selection döngüsünü çalıştırmak için hedef değişkenimizi Static Columns sekmesine atıyoruz ve Feature Selection Strategy seçeneklerinden Forward Feature Selection’ı seçiyoruz.

1. **Öznitelik Seçimi Döngüsü Bitişi**

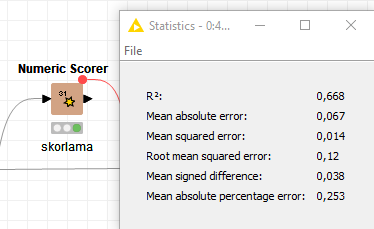
Makine öğrenmesi algoritmasını çalıştırmadan önce döngünün biteceği Nod’da maksimize veya minimize etmek istediğimiz metriği seçmemiz gerekmektedir.



Şekil 26 – Öznitelik Seçim Döngüsü Başlangıcı 2

Öznitelik seçimi döngüsünün sonunda R2 metriğinin maksimize edilmesini tercih ettik bizim başarı metriğimiz R2 metriği olacaktır.

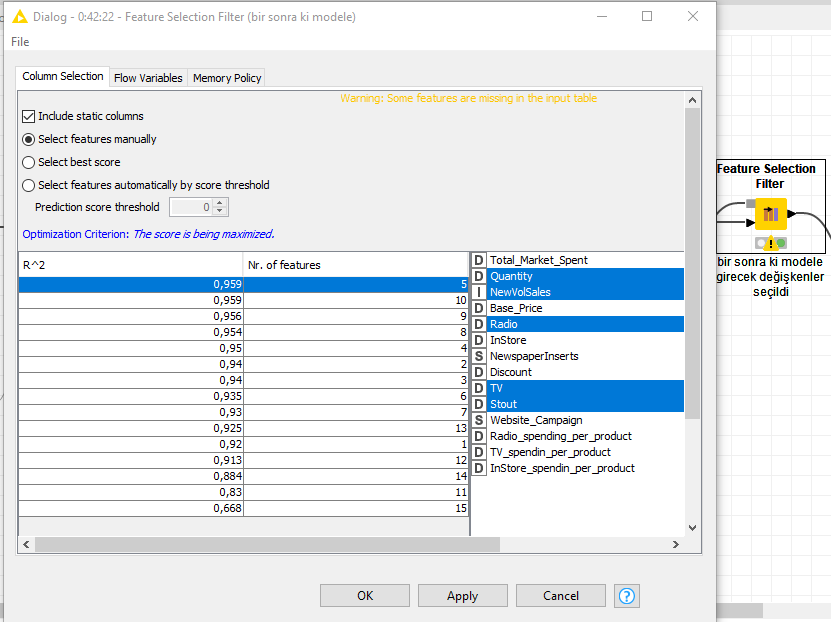
1. **Makine Öğrenmesi Performansının Skorlanması**



Şekil 27- Skorlama 3

Gradyan Destekli Ağaç Regresyonu algoritmasından aldığımız ilk sonuçlar bu şekildedir. Elde Edilen sonuçların iyileştirilmeye ihtiyaç duyduğunu görmekteyiz.

1. **Öznitelik Seçimi Döngüsü Sonrası Değişken Seçimi**

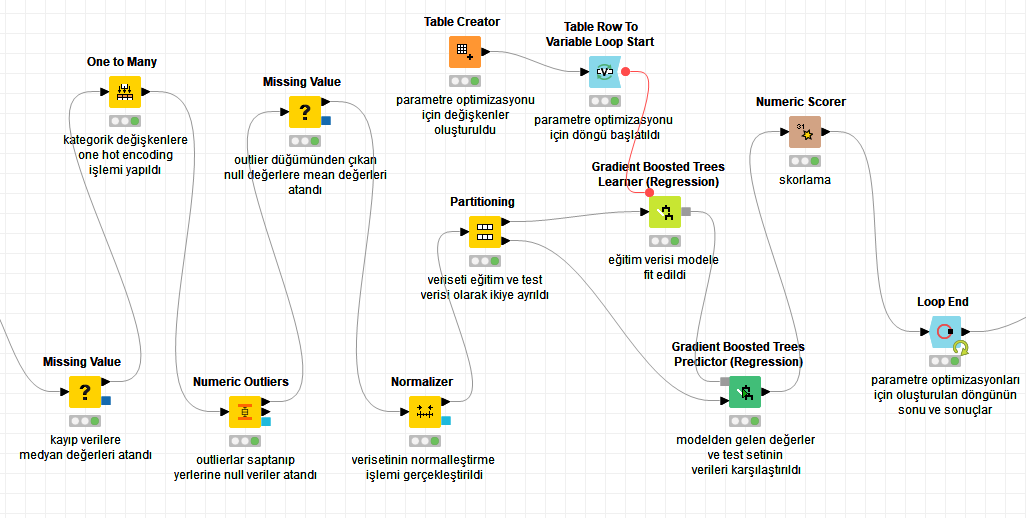


Şekil 28 – Öznitelik Filtresi 2

Şekil 28’de de görüldüğü üzere Feature Selection Filter Nod’unda değişkenlerin R2 metrik değerleri, bu değerlerin ortaya çıkmasını sağlayan değişkenler ve kaç değişken kullanıldığı gösterilmektedir. Bu değişkenler arasından istediğimiz değişken kombinasyonlarından birini seçip yeni kuracağımız modele yönlendirebiliriz.

**3.2.8.2 Gradient Boost Regression Parameter Optimization Metanod’u**

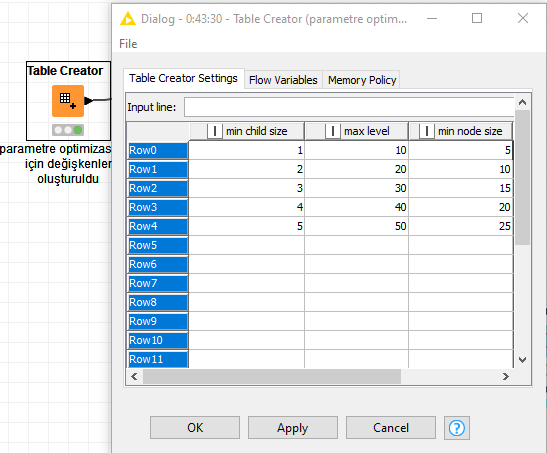
Parametre optimizasyonu, algoritmamız eğitim işlemi gerçekleştirirken kullandığı hazır parametreler yerine dışardan manuel olarak farklı parametreler girilmesine denir. Bu işlem sonucunda algoritmamızın başarısının artması beklenmektedir.



Şekil 29 – Parametre Optimizasyonu 2

Şekil 29’da görüldüğü üzere bu MetaNod’da bulunan ilk 5 nod daha önce yaptığımız veri ön hazırlık işlemleridir.

1. **Parametre Optimizasyonu İçin Değişken Oluşturma**

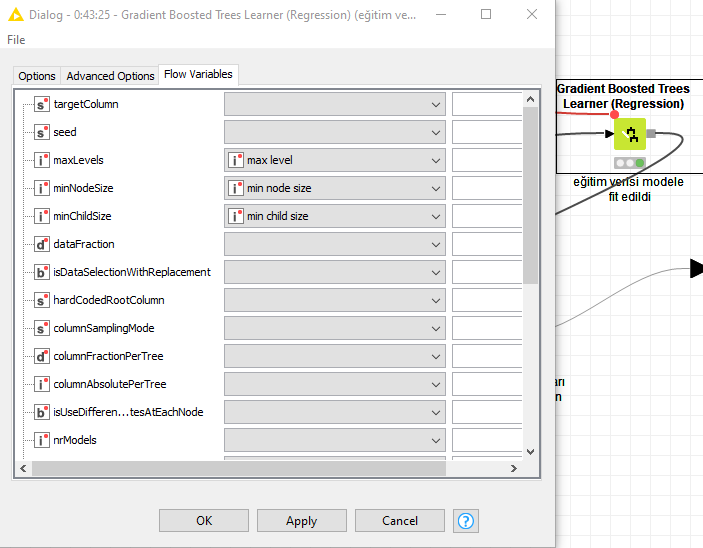


Şekil 30 – Parametre Oluşturma 2

Algoritmamızın içine yerleştirmek istediğimiz yeni parametrelerimiz hazırdır. Döngü bu değerleri teker teker deneyip sonuçlarını bize getirecektir.

1. **Parametrenin Algoritmaya Yerleştirilmesi**

Makine öğrenmesi algoritmamızın parametrelerini optimize edebilmek için oluşturduğumuz değerleri modelimize eklememiz gerekmektedir.

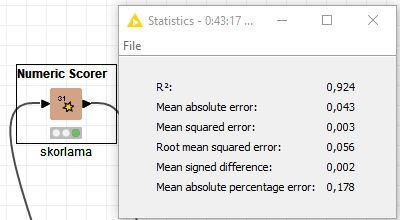


Şekil 31 – Parametre Yerleştirme 2

Gradient Boost Trees Learner Nod’unda bulunan Flow Variables sekmesindeki hazır olarak gelen değişkenlerle daha önceden hazırladığımız değişkenleri değiştiriyoruz.

1. **Makine Öğrenmesi Performansının Skorlanması**

Yaptığımız parametre optimizasyonu ve öznitelik seçimi uygulamaların sonucunu göreceğimiz son skorlama nod’u olacaktır.



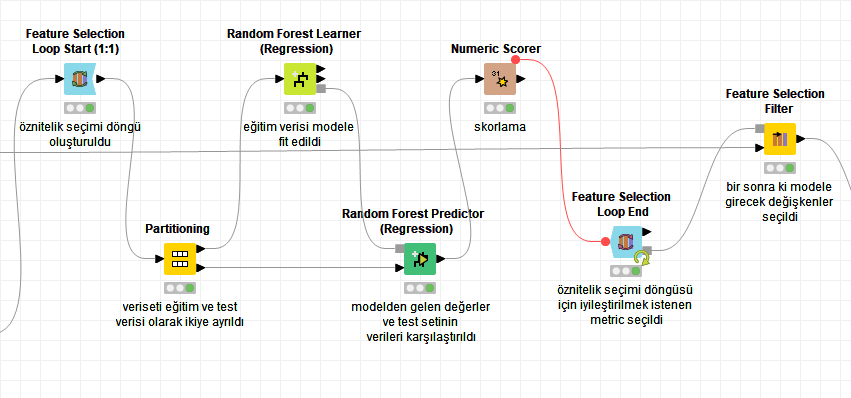
Şekil 32 – Skorlama 4

Gradyan Destekli Ağaç Regresyonu algoritmasının son skoru şekil 32’deki gibidir. Görüldüğü üzere 0,668’den 0,924 oranına kadar artmış bir R2 skoru gözükmektedir. Yaptığımız uygulamaların işe yaradığını var sayabiliriz.

**3.2.9 Rassal Orman Regresyonu**

Rassal Orman Regresyonu projemizde kullandığımız 3 makine öğrenmesinden sonuncusudur. Önce Öznitelik Seçimi sonra ise Parametre optimizasyonu yaparak bu makine öğrenmesi algoritmasında verim elde etmeye çalışacağız.

**3.2.9.1 Random Forest Regression Feature Selection Metanod’u**

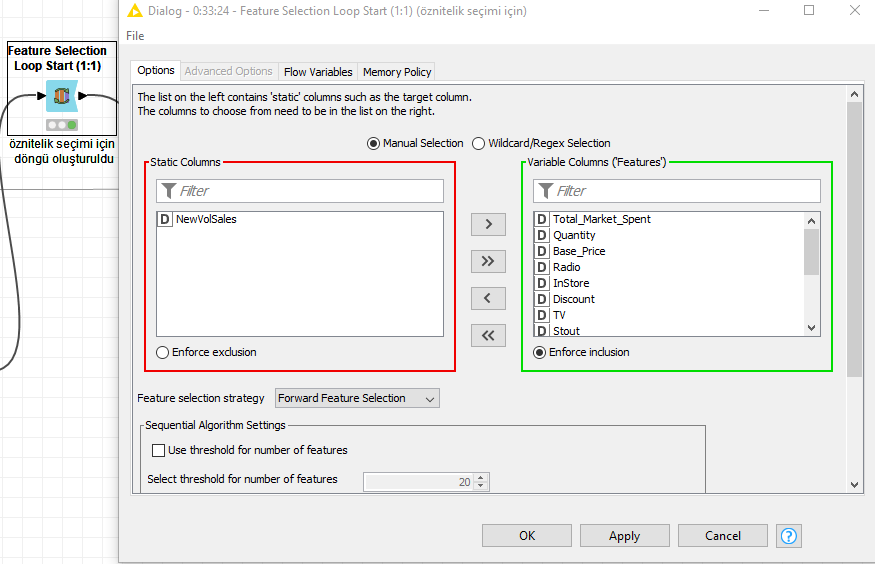


Şekil 33 – Öznitelik Seçimi 3

Feature Selection MetaNod’u şekil 33’te görüldüğü gibidir.

1. **Öznitelik Seçimi Döngüsü Başlangıcı**

Feature Selection Loop Start Nod’uyla öznitelik seçimi döngümüzü başlatıyoruz.

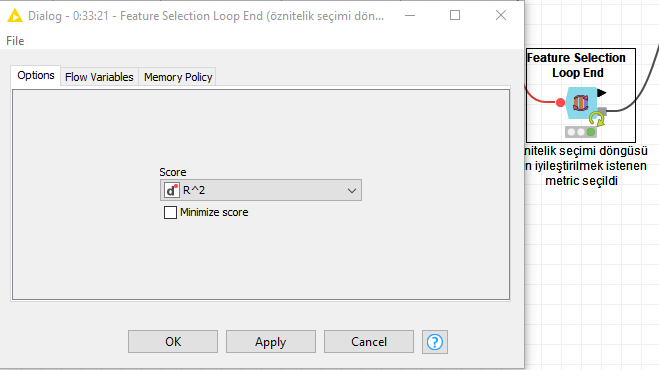


Şekil 34 – Öznitelik Seçim Döngüsü Başlangıcı 3

Feature Selection döngüsünü çalıştırmak için hedef değişkenimizi Static Columns sekmesine atıyoruz ve Feature Selection Strategy seçeneklerinden Forward Feature Selection’ı seçiyoruz.

1. **Öznitelik Seçimi Döngüsü Bitişi**

Makine öğrenmesi algoritmasını çalıştırmadan önce döngünün biteceği Nod’da maksimize veya minimize etmek istediğimiz metriği seçmemiz gerekmektedir.

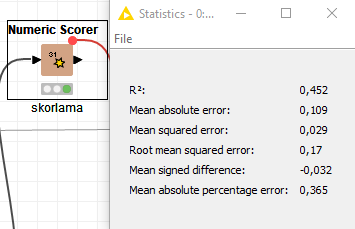


Şekil 35 – Öznitelik Seçim Döngüsü Başlangıcı 3

Öznitelik seçimi döngüsünün sonunda ‘R2’ metriğinin maksimize edilmesini tercih ettik bizim başarı metriğimiz ‘R2’ metriği olacaktır.

1. **Makine Öğrenmesi Performansının Skorlanması**

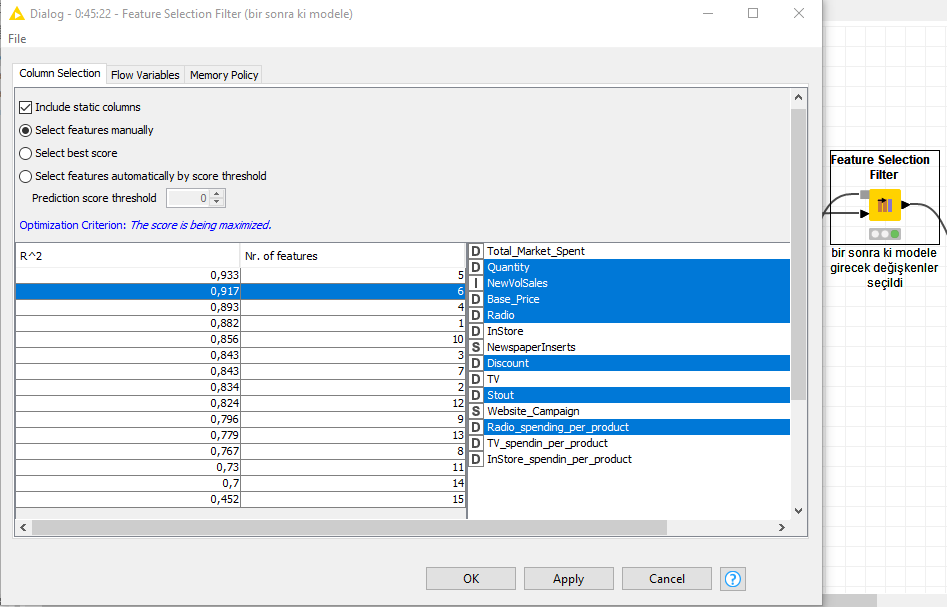
Numeric Scorer Nod’unda eğitimden çıkmış verilerle test verilerini karşılaştırıp sonuçlara bakabildiğimiz Nod’dur.



Şekil 36 - Skorlama 5

Rassal Orman Regresyonu algoritmasından aldığımız ilk sonuçlar bu şekildedir. Elde Edilen sonuçların iyileştirilmeye ihtiyaç duyduğunu görmekteyiz ve bu algoritmanın şuana kadar en düşük başarı skorunu verdiğini söyleyebiliriz.

1. **Öznitelik Seçimi Döngüsü Sonrası Değişken Seçimi**

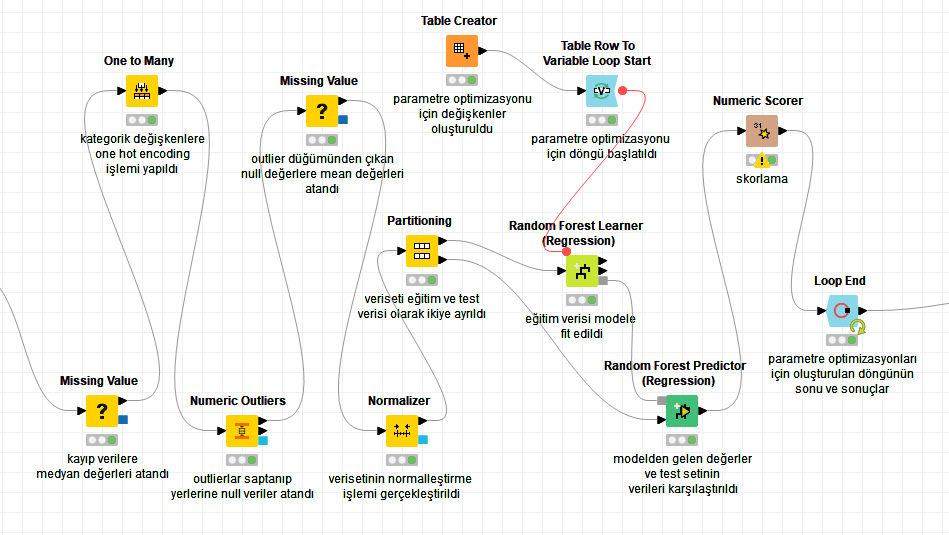


Şekil 37 – Öznitelik Seçimi 3

Şekil 36’da da görüldüğü üzere Feature Selection Filter Nod’unda değişkenlerin R2 metrik değerleri, bu değerlerin ortaya çıkmasını sağlayan değişkenler ve kaç değişken kullanıldığı gösterilmektedir. Bu değişkenler arasından istediğimiz değişken kombinasyonlarından birini seçip yeni kuracağımız modele yönlendirebiliriz.

**3.2.9.2 Random Forest Regression Parameter Optimization Metanod’u**

Parametre optimizasyonu, algoritmamız eğitim işlemi gerçekleştirirken kullandığı hazır parametreler yerine dışardan manuel olarak farklı parametreler girilmesine denir. Bu işlem sonucunda algoritmamızın başarısının artması beklenmektedir.

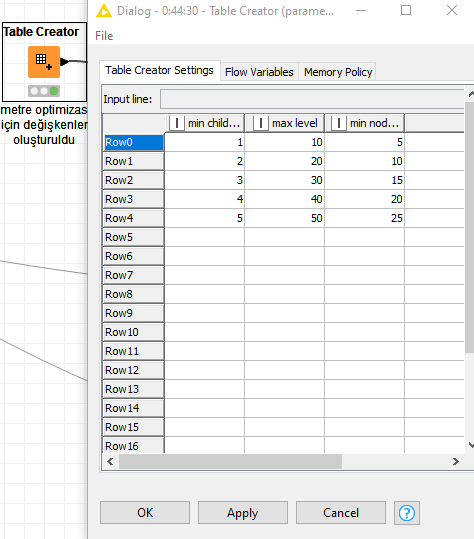


Şekil 38 – Parametre Optimizasyonu 3

Şekil 38’de görüldüğü üzere bu MetaNod’da bulunan ilk 5 nod daha önce yaptığımız veri ön hazırlık işlemleridir.

1. **Parametre Optimizasyonu İçin Değişken Oluşturma**

Daha öncede kullanmış olduğumuz bu nod Knime içerisinde manuel olarak tablolar oluşturmaya imkan sağlıyor.

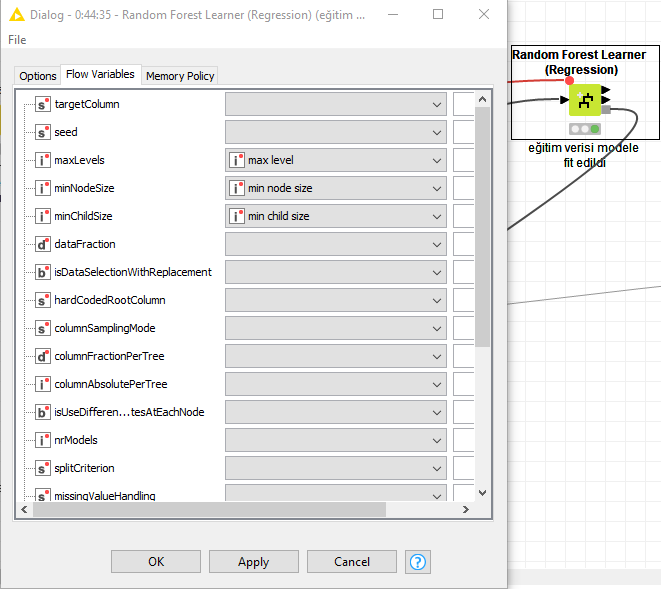


Şekil 39 – Parametre Oluşturma 3

Algoritmamızın içine yerleştirmek istediğimiz yeni parametrelerimiz hazır. Döngü bu değerleri teker teker deneyip sonuçlarını bize getirecektir.

1. **Parametrenin Algoritmaya Yerleştirilmesi**

Makine öğrenmesi algoritmamızın parametrelerini optimize edebilmek için oluşturduğumuz değerleri modelimize eklememiz gerekmektedir.

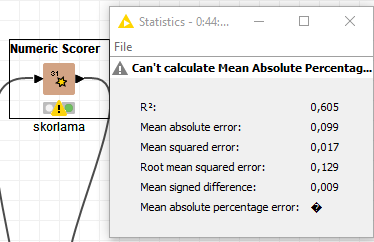


Şekil 40- Parametre Yerleştirme 3

Random Forest Learner Nod’unda bulunan Flow Variables sekmesindeki hazır değişkenlerle daha önceden hazırladığımız değişkenlerle değiştiriyoruz.

1. **Makine Öğrenmesi Performansının Skorlanması**

Yaptığımız parametre optimizasyonu ve öznitelik seçimi uygulamaların sonucunu göreceğimiz son skorlama nod’u olacaktır.



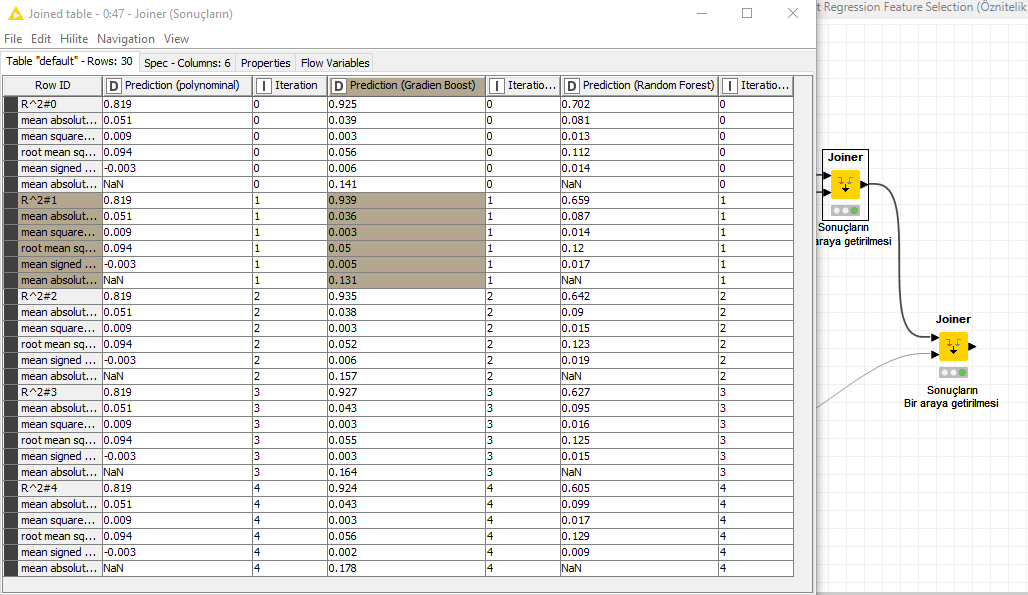
Şekil 41 – Skorlama 6

Rassal Orman Regresyonu algoritmasının son skoru şekil 41’deki gibidir. Görüldüğü üzere 0,452’den 0,605 oranına kadar artmış bir R2 skoru gözükmektedir. Yaptığımız uygulamaların işe yaradığını var sayabiliriz. Ama bu artışın yeterli olduğunu söylememiz çok güçtür.

**3.2.10 Elde Edilen Bütün Sonuçların Karşılaştırılması**

Kurduğumuz 3 ayrı modelden farklı farklı sonuçlar almış bulunmaktayız. Bu sonuçları karşılaştırmak için bir araya getirmemiz gerekmektedir.

**3.2.10.1 Elde Edilen Bütün Sonuçların Birleştirilmesi**



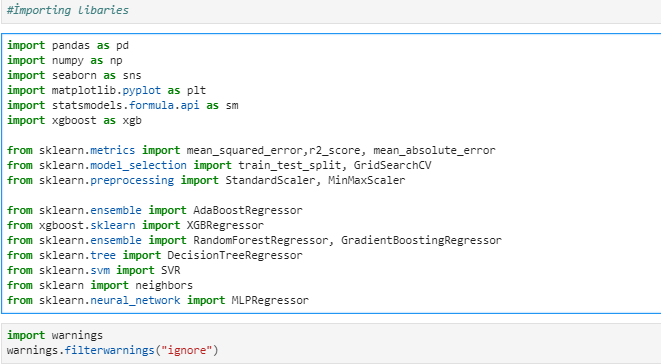
Şekil 42 – Sonuçların Birleştirilmesi

Modellerimizin Parametre Optimizasyon Metonod’larını joinerlarla birleştirerek şekil 42’deki değerleri elde ettik. En iyi metrik değerlerini Gradyan Destekli Ağaç Regresyonu Algoritmasının 2. İterasyonunda elde ettiğimiz gözüküyor. Polinomnial Regresyon algoritmasında iterasyon sayısındaki değişim sonuçlara etki etmediğini gözlemleniyor. Rassal Orman Regresyon Algortimasında ise iterasyon değerleri arttıkça sonuçların azaldığını söylenebilir.

**3.3 Python ile Uçtan Uca Veri Analizi ve Esneklik Değerlerinin Hesaplanması**

**3.3.1 Kütüphanelerin Yüklenmesi**

Python programlama diliyle yapacağımız işlemleri kütüphaneler aracılığıyla yapacağız. Bunun için bu kütüphaneleri projemize yüklememiz etmemiz gerekmektedir.

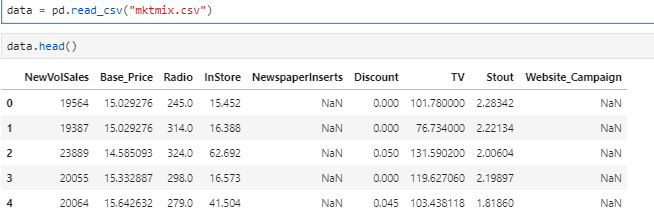


Kod 1 – Kütüphanelerin yüklenmesi

Kod 1’de de görüldüğü üzere veri ön hazırlık işlemlerinden değerlendirme işlemlerine kadar kullanmamız gereken bütün kütüphaneleri projemize yükledik.

**3.3.2 Verimizi okutmak**

Verimizi projemizin içinde okutmak için “pandas” kütüphanesinin “read” methodunu kullanacağız.



Kod 2 – Verinin okunması

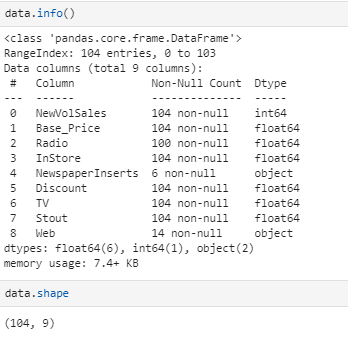
Pandas kütüphanesinin “read” methodunu kullanarak verimizi projemize okuttuk ve “head” methoduyla görünümünü gözden geçirdik.

**3.3.3 Veri Hakkında Bilgi Edinmek**

Veri hakkında genel bilgi edinmek için birkaç sorgulama işlemi yapacağız ve veri ön hazırlığı için hangi işlemleri yapmamız gerektiğine karar vereceğiz.

1. **Veri tiplerinin öğrenilmesi**

Veri kümemizdeki veri tiplerini öğrenmek için “info” komutunu kullanacağız.



Kod 3 – Değişken Tipleri

Kod 3’de de görüldüğü üzere verimizde bulunan değişkenlerden 6’sı Float, 1’i İnteger ve 2 değişkenin de Object tipinde olduğu görülmüştür. Float değişkenlerin numerik değişkenler, Object değişkenlerin kategorik değişkenler ve Integer değişkenin ise bizim tahmin etmeyi hedeflediğimiz değişken olduğunu söyleyebiliriz.

Aynı zamanda “shape” komutuyla verimizde kaç adet gözlem ve kaç adet değişken olduğunu görebiliyoruz. 104 adet gözlemimiz ve 9 adet değişkenimiz olduğu gözlemlenmiştir.

1. **Kayıp Verilerin Gözlenmesi ve Düzeltilmesi**

Veri kümelerinde bulunan kayıp veriler veri ön hazırlık aşamalarında sorunlar çıkarmaktadır. Bu sorunların üstesinden gelmek için veri kümemizde olabilecek kayıp verilerin kontrolü ve bu kayıp verilerin düzenlenmesi işlemlerini yapmamız gerekmektedir.

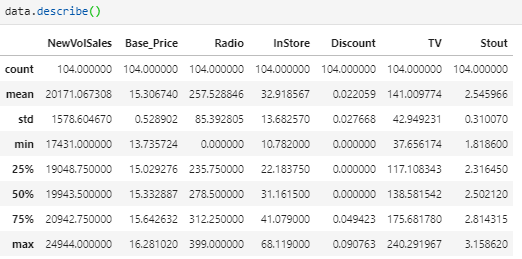


Kod 4 – Kayıp Veri Kontrolü

Kod 4’de de görüldüğü üzere “Radio” değişkeninde 4 adet, “NewspaperInserts” değişkeninde 98 adet ve “Web” değişkeninde 90 adet kayıp veri bulunmaktadır. Bu değişkenlerdeki kayıp verilerin düzetilmesi işlemi “fillna” metoduyla yapılmıştır. “Radio” değişkenin medyan değerleri ve diğer iki değişkenlerdeki kayıp verileri de “other” değeri atanmıştır.

1. **Numerik Değerlerin İstatistiklerinin Gözlemlenmesi**

Veri kümemizde bulunan istatistiksel değerlere bakarak veri hakkında birçok değeri gözlemleyebiliriz. Bunun için “describe” metodunu kullanacağız.



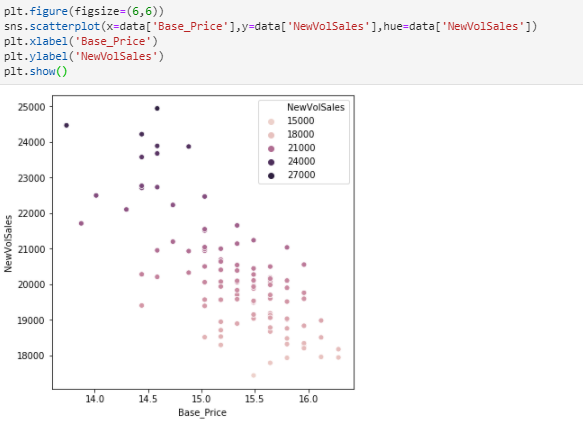
Kod 5 – İstatistikler

Kod 5’de veri kümemizdeki numerik değişkenlerin birçok istatistiksel değerlerini görmekteyiz. Veri kümemizde standartlaştırma işlemini yapmamız gerektiğini anlayabiliyoruz. Bu kanıya varılmasının sebebi “Radio” değişkenin değerleri ile “Discount” değişkenlerinin arasında bulunan sayısal farklılıklardır. Bu sayısal farklılıklar düzenlenmezse makine öğrenmesi sürecinde düşük değerlerin etkileri de düşük olacaktır. Bu durum algoritmamızı yanlış yönlendireceği için bu durumu standartlaştırma işlemi yaparak düzenlememiz gerekmektedir.

**3.3.4 Görselleştirme**

Veri kümemiz üzerinde görselleştirmeler yaparak kullanacağımız veri kümesi hakkında daha çok bilgi sahibi olabiliriz.

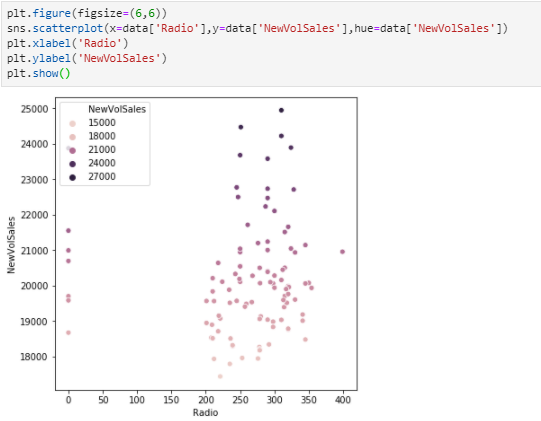
1. **Dağılım Grafiği 1**



Kod 6 – Görsel 1

Kod 6’da x eksenimizde hedef değişkenimiz olan NewVolSales değişkeni, y ekseninde ise Haftalık birim fiyatı belirten Base\_Price değişkenimizdir. Bu durumda bu şekilden haftalık birim fiyatın düşük olduğu haftalarda toplam satışın yüksek olduğunu, haftalık birim satış fiyatının yüksek olduğu haftalarda toplam satışın az olduğunu gözlemleyebiliriz.

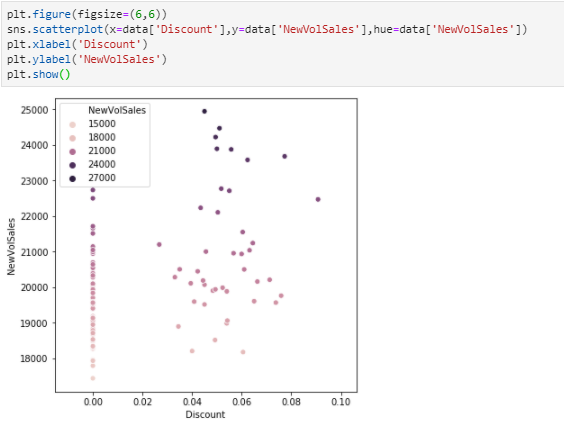
1. **Dağılım Grafiği 2**



Kod 7 – Görsel 2

Kod 7’deki görsele bakınca “Radio” değişkenin de bazı gözlemlerin 0 değeri içerdiğini gözlemleyebiliriz.

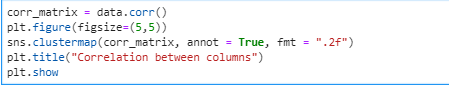
1. **Dağılım Grafiği 3**

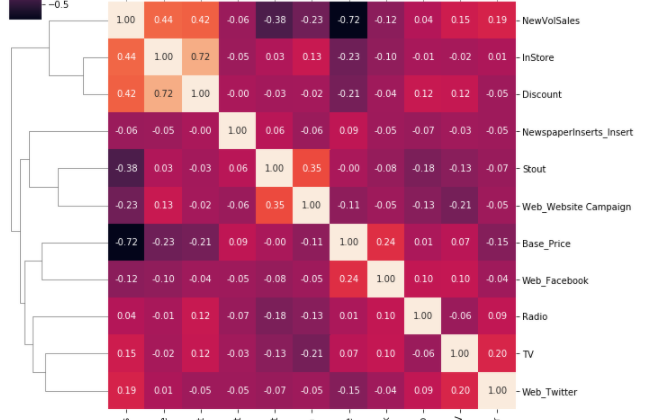


Kod 8 – Görsel 3

Kod 8’ deki görsele bakınca “Discount” gözlemlerinden bazılarının 0 değeri içerdiği gözlemlenmiştir.

1. **Korelasyon Matrisi**





Kod 9 – Görsel 4

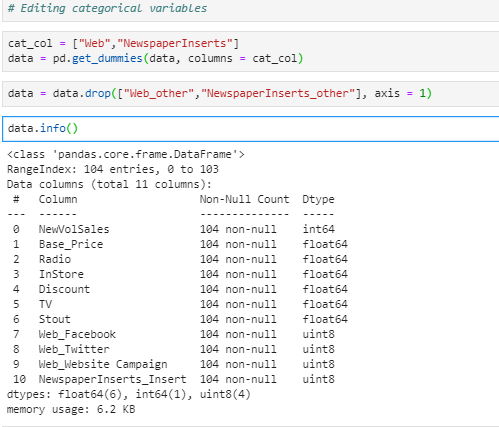
Kod 9’daki görselde de görüldüğü “Discount” ve “InStore” değişkenleri arasında ilişki 0.72’lik pozitif korelasyon, “NewVolSales” ile “Base\_Price” değişkenleri arasında da 0.72’lik negatif ilişki gözlemlenmektedir.

**3.3.5 Veri Ön Hazırlık**

Modelimizi kurmaya geçmeden önce bir veri ön hazırlık işlemlerini yapmamız gerekmektedir.

1. **Kategorik Değişkenlerin Numerik Değişkenlere Dönüştürülmesi (One – Hot Encoding)**

Veri kümemizde 2 adet kategorik değişken bulunmaktaydı bu değerleri modelimizi kurarken olumsuz etkilememesi için numerik değerlere çevireceğiz.

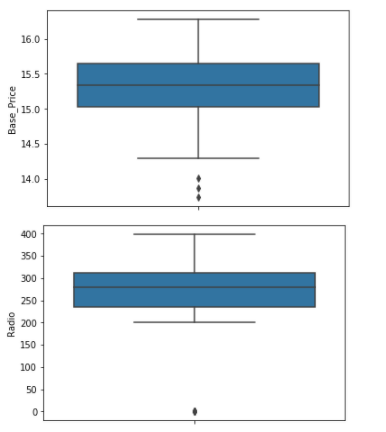


Kod 10 – Kategorik Değişkenlerin Düzenlenmesi

Kod 10’da da görüldüğü üzere ilk önce kategorik değişkenlerimizi bir değere atadık. Bu değere atama işlemini yaptıktan sonra “pandas” kütüphanesinin “get\_dummies” methoduyla bu değişkenler içinde bulunan kategorik değişkenleri numerik değişkenlere çevirdik. Bu çevirme işleminin sonunda kukla değişken tuzağına düşmemek için her iki değişken grubundan birer değişkeni veri kümesinden düşürdük.

1. **Aykırı Değerlerin Düzenlenmesi**





Kod 11 – Aykırı Değer Tespiti

Veri kümemizdeki aykırı değerleri görebilmek için “box plot” grafiğinden yararlandık. İlk önce verimizde bulunan numerik değişkenleri bir değere atadık bu değerleri “for” döngüsüyle birer birer “box plot” grafiklerini oluşturduk. Bu grafiklerde görüldüğü üzere “Radio” ve “Base\_Price” değişkenlerinde aykırı değerler gözlemlenmiştir.



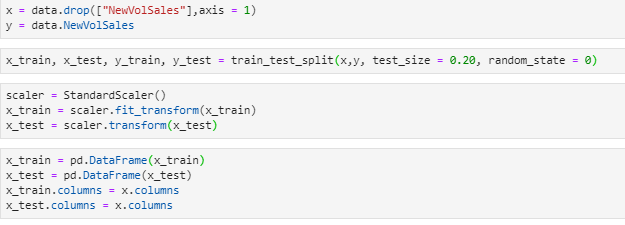
Kod 12 – Aykırı Değerlerin Düzeltilmesi

Kod 12’de de görüldüğü üzere “box plot” grafiklerinden elde ettiğimiz bilgiyle aykırı değer içeren değişkenleri bir değere atadık. Bu işlemden sonra bir fonksiyon tanımlıyoruz. Bu fonksiyon sayesinde verimizin normal dağılım eğrisindeki yüzde 5’lik, 25’lik, 75’lik ve 95’lik değerlerini “numpy” kütüphanesinin “percentile” metoduyla belirlemiş olduk. Eğer aykırı değerimiz %5lik değerler içerisinde ise bu değeri “low\_level” değeri ile eşitlemesini istedik. Eğer aykırı değerimiz %95’lik değerler içerisinde ise bu değeri “up\_level” değişkeni ile eşitlemesini istedik. Bu fonksiyon tanımlama işleminden sonra daha sonra belirlemiş olduğumuz aykırı değerlerini bir “for” döngüsü ile çalıştırıp aykırı değerlerinden kurtulmuş olduk.

**3.3.6 Modelleme**

**a) Modellemeye Hazırlık**

Projemizin ön hazırlık aşamasını bitirdikten sonra makine öğrenmesi modellerimizi oluşturacağımız modelleme aşamasına geldik. Burada makine öğrenmesi algoritmaları oluşturup bu modellerin sonuçlarını gözlemleyeceğiz.



Kod 13 – Modelleme Hazırlığı

Kod 13’de görülmekte olan kod satırlarında ilk önce hedef değişkenimiz veri kümemizden ayrılıp “y” değişkenine atandı ve geriye kalan veri kümemize ise “x” değeri verildi.

Veri kümemizi makine öğrenmesi algoritmasına sokabilmemiz için verimizi test ve eğitim kümeleri olarak ikiye ayırmamız gerekmektedir. Bu işlemi “train\_test\_split” komutuyla yapıyoruz. Bu komutun içinde bulunan “test\_size” değişkenine “0.20” değerini atadık. Böylece verimiz eğitim için veri kümemizin %80’ini, test için ise %20’sini kullanacaktır.

Veri kümemizin standartlaştırma işleminden geçmesi gerektiğini daha önce bahsetmiştik. Bu işlemi “StandardScaler” metoduyla gerçekleştiriyoruz.

Veri kümemiz standartlaştıktan sonra “DataFrame” olma özelliğini kaybedip “Numpy” dizisine dönüşmektedir. Bu dönüşme işlemi ilerleyen işlemlerde bizim için sorunlar oluşturacağı için standardize edilmiş verimizi tekrar “DataFrame” haline getirip kolon isimlerini de tekrar atıyoruz.

1. **Kullanılacak Algoritmalar ve Modelleme**
2. Destek Vektör Regresyonu (Support Vector Regression) : İki sürekli değişken arasında doğrusal ilişkiyi inceleyen istatistiksel bir algoritmadır. Genellikle veriler arasında ilişkiyi açıklayabilecek bir doğru oluşturmaya çalışır [16].
3. AdaBoost Regresyonu (AdaBoost Regression): AdaBoost Regresyonu, orjinal veri kümesine bir bağımsız değişken yerleştirerek başlamaktadır. Bu yerleştirme işleminin ardından aynı veri kümesine bağımsız değişkenin ek kopyalarını yerleştirir ve değişkenlerin ağırlıklarını bu yerleştirmeler sonucu oluşan hatalara göre belirleyen bir algoritmadır [17].
4. Aşırı Gradyan Artırma Regresyonu (Extereme Gradient Boosting Regression, XGBRegression): Son derece verimli ve esnek olacak şekilde tasarlanmış optimize edilmiş bir gradyan eğilimli algoritmadır. Gradyan artırma çerçevesi altında makine öğrenmesi algoritmalarını uygulamaktadır. [18].
5. Rassal Orman Regresyonu (Random Forest Regression) : Veri kümesinin çeşitli alt örneklerine bir dizi sınıflandırma karar ağacına uygun hale getirmektedir. Tahmin doğruluğu ve aşırı öğrenmeyi kontrol etmek için ortalama değeri kullanan bir regresyon algoritmasıdır [19].
6. Karar Ağacı Regresyonu (Decision Tree Regression) : Karar ağaçları veriyi bir sinüs eğrisine uygun hale getirmek için ek olarak gürültülü gözlemler kullanmaktadır. Buna sonuç olarak sinüs eğrisine yaklaşan yerel doğrusal regresyonları öğrenmektedir [20].
7. Gradyan Eğilimli Ağaç Regresyon (Gradient Boosting Tree Regression): Gradyan Eğilimli Regresyon ileriye dönük aşamalı bir tarzda katkı modeli oluşturmaktadır. İsteğe bağlı türevlenebilir yapısı kayıp fonksiyonların optimizasyonuna izin verir. Her aşamada verilen kayıp fonksiyonunun negatif gradyanına bir regresyon ağacı yerleştirilmektedir [21].
8. K En Yakın Komşu Regresyonu ( K – Nearest Neighbors Regression) : KNN regresyonu, sezgisel bir şekilde, aynı mahalledeki gözlemlerin ortalamasını alarak bağımsız değişkenler ile sürekli sonuç arasındaki ilişkiye yaklaşan parametrik olmayan bir yöntemdir [22].
9. Çok Katmanlı Algılayıcı Regresyonu (Multi Layer Perceptron Regression): Bir Çok Katmanlı Algılayıcı Regresyonu, en az üç düğüm katmanından oluşur: bir giriş katmanı, bir gizli katman ve bir çıktı katmanı. Giriş düğümleri dışında, her düğüm doğrusal olmayan bir aktivasyon işlevi kullanan bir nörondur. Çok Katmanlı Algılayıcı Regresyonu, eğitim için geri yayılım adı verilen denetimli bir öğrenme tekniğini kullanır. Çoklu katmanları ve doğrusal olmayan aktivasyonu, Çok Katmanlı Algılayıcı Regresyonu’nu doğrusal bir algılayıcıdan ayırır. Doğrusal olarak ayrılamayan verileri ayırt edebilir [23].
10. Polinominal Regresyon (Polynimal Regression): Belirtilen dereceye eşit veya daha düşük dereceye sahip özelliklerin tüm polinom kombinasyonlarıyla yeni bir özellik matrisi oluşturan regresyon algoritmasıdır [24].

Verilerimizi makine öğrenmesi modellerine hazır hale getirdik ve modelleme işlemine başlayabilmek için önümüzde bir engel kalmadığına göre modelleme işlemine başlayabiliriz.



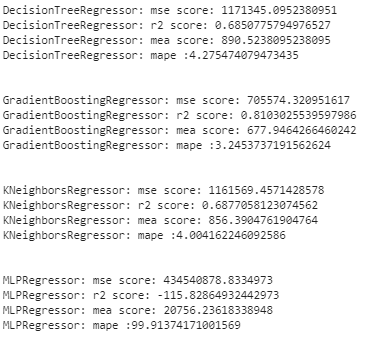
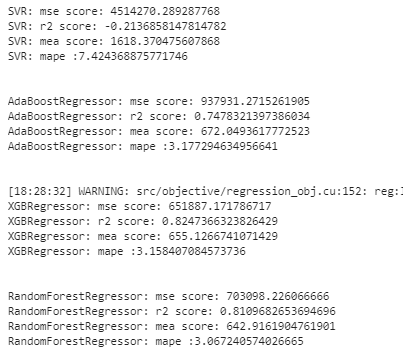
Kod 14 – Modelleme

Kod 14’de de görüldüğü üzere daha önceden “import” edilen makine öğrenmesi modellerinin her birini bir değere atadık. Bu değerlerden oluşan bir liste oluşturduk. Aynı zamanda bu değerlerin isimlerinden oluşan başka bir liste daha hazırladık.

Oluşturduğumuz bu değerleri bir “for” döngüsü içinde birer birer döndürerek elde ettiğimiz sonuçları yazdırmayı amaçladık.

1. **Skorlama**

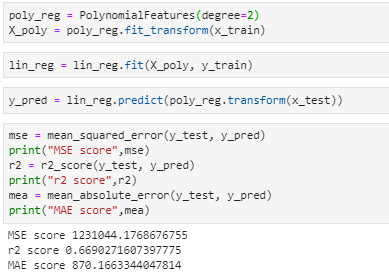
Verimizi modelleme işlemlerine soktuktan sonra bu makine öğrenmesi işlemlerinden hangisini kullanmamız gerektiğine bu sonuçlara bakarak karar vereceğiz.



Kod 15 – Skorlama

Oluşturduğumuz “for” döngüsü sayesinde bütün makine öğrenmesi algoritmalarının sonuçlarını bir arada görebiliyoruz. Bu algoritmalardan elde ettiğimiz metriklerden “R2 score’unu” hedef metriğimiz olarak ele alacağız.

Bu sonuçlar doğrultusunda en iyi algoritmalarımız “XGBRegressor” ,”RandomForestRegressor” ve “GradientBoostRegressor” algortimalarının en iyi değerlere sahip olarak görülmektedir.



Kod 16 – Polinominal Regresyon

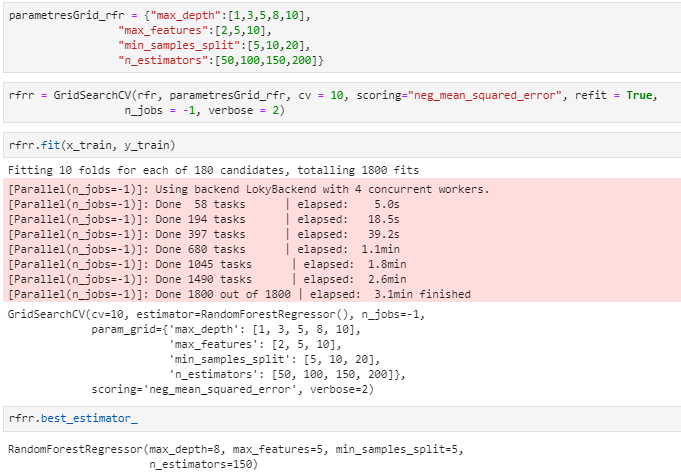
Polinominal Regresyon algoritmasının çalışma prensibi diğer algoritmalara göre değişik olduğunu için o algoritmayı “for” döngüsü içinde çalıştıramadık. Ama Polinominal Regresyon algoritmasından çıkacak sonucu öğrenmek için yine de bu algoritmayı oluşturup sonucuna baktık.

Polinominal Regresyon algoritmasını oluşturmak için eğitim kümemizi polinominal değerlere çevirmemiz gerekmektedir. Bu çevirme işlemini gerçekleştirdikten sonra Doğrusal Regresyon algoritması yardımıyla tahmin etme işlemini yapmaktayız. Elde ettiğimiz sonuç daha önceki sonuçlardan düşük olduğu için daha önceki algoritmalarla projemize devam edeceğiz.

**3.3.7 Model Geliştirme İşlemleri**

Modelleme işlemlerinden sonra seçmiş olduğumuz birbirine yakın sonuçlar veren algoritmaların parametrelerini optimize ederek maksimum başarı oranına ulaşmaya çalışacağız.

1. **Rassal Orman Regresyonunun Parametre Optimizasyonu**



Kod 17 – Parametre Optimizasyonu 1

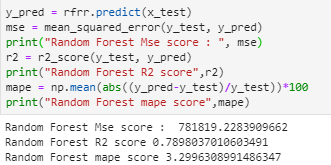
Kod 17’de de görüldüğü üzere ilk olarak Rassal Orman Regresyonu algoritmasının parametrelerine belirli değerler atanmıştır. Bu değerler teker teker denenip elde edilen sonuçlardan başarı oranları yüksek olan parametre değerlerini kullanacağız.

**Çapraz Doğrulama Algoritması**

Çapraz Doğrulama Algoritması, genel anlamda aşırı-parametre uzayının tanımlanmış alt kümelerine dayanan kapsamlı bir aramadır. Aşırı parametreler, minimum değer, maksimum değer ve adım sayıları girilerek belirlenmektedir. Belirtme işlemi sonrasında algoritma bu değerler aralığında her bir kombinasyonu hesaplamaktadır. her kombinasyonun performansı belirli metriklerce değerlendirilip optimal sonuca ulaşmayı amaçlamaktadır [25].

Çapraz Doğrulama algoritmasının “cv” değerini 10 olarak belirledik. Bunun anlamı bu algoritma verimizi 10 parçaya bölüp geriye kalan 9 parçayla öğrenme işlemini gerçekleştirecektir. Bu öğrenme işlemi bütün parçalar, hem eğitim hem de test verisi olarak kullanıldıktan sonra bitecektir. Bu veri kümesinin 10 defa çalışması demektir.

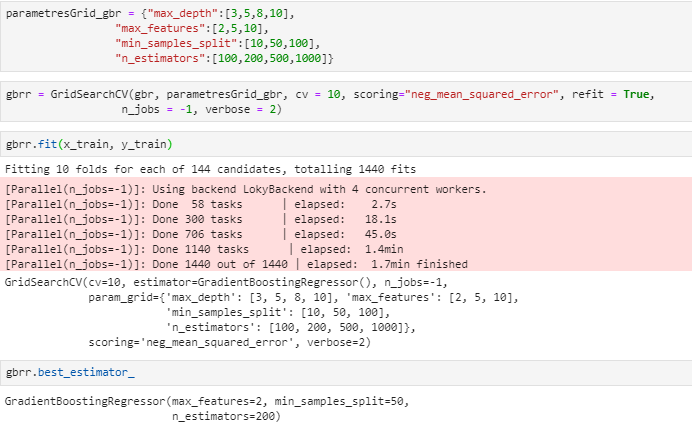
İşlem bittikten sonra “best\_estimator\_” komutuyla algoritma da en yüksek başarıya ulaşan parametreleri görmüş olduk.



Kod 18 – Skor 1

Parametre optimizasyonu sonrasında Rassal Orman Regresyonu algoritmasının başarı oranı 0.810’dan 0.781’e düşmüş olduğunu gözlemlenmiştir.

1. **Gradyan Eğilimli Ağaç** **Regresyonu Parametre Optimizasyonu**

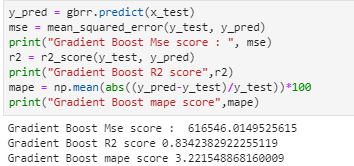


Kod 19 – Parametre Optimzasyonu 2

Kod 19’de de görüldüğü üzere ilk olarak Gradyan Eğilimli Ağaç Regresyonu algoritmasının parametrelerine belirli değerler atanmıştır. Bu değerler teker teker denenip elde edilen iyi sonucu verenler parametreleri kullanacağız.

Çapraz Doğrulama algoritmasının “cv” değerini 10 olarak belirledik. Bunun anlamı bu algoritma verimizi 10 parçaya bölüp geriye kalan 9 parçayla öğrenme işlemini gerçekleştirecektir ve bu öğrenme işlemi bütün parçalar hem eğitim hem de test verisi olarak kullanıldıktan sonra bitecektir. Bu veri kümesinin 10 defa çalışması demektir.

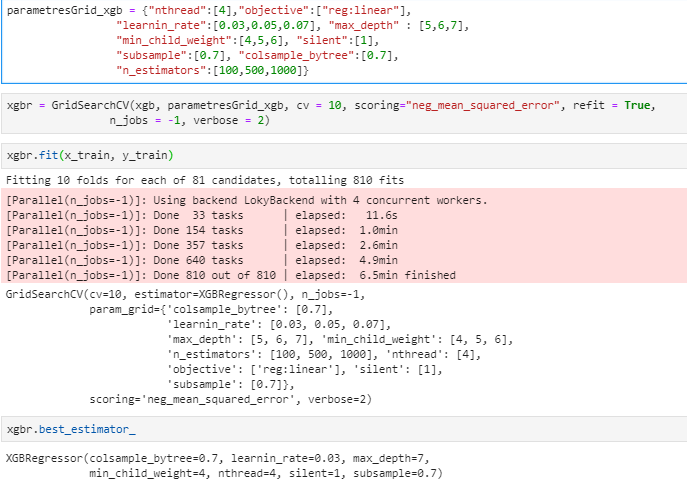
İşlem bittikten sonra “best\_estimator\_” komutuyla algoritma da en yüksek başarıya ulaşan parametreleri görmüş olduk.



Kod 20 – Skor 2

Parametre optimizasyonu sonrasında Gradyan Eğilimli Ağaç Regresyonu algoritmasının başarısı 0.810’dan 0.834’e arttığı gözlemlenmiştir.

1. **Aşırı Gradyan Artırma Regresyonu Parametre Optimizasyonu**

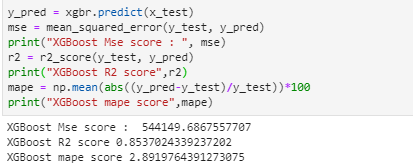


Kod 21 – Parametre Optimizasyonu 3

Kod 21’de de görüldüğü üzere ilk olarak Aşırı Gradyan Artırma Regresyonu algoritmasının parametrelerine belirli değerler atanmıştır. Bu değerler teker teker denenip elde edilen iyi sonucu verenler parametreleri kullanacağız.

Çapraz Doğrulama algoritmasının “cv” değerini 10 olarak belirledik. Bunun anlamı bu algoritma verimizi 10 parçaya bölüp geriye kalan 9 parçayla öğrenme işlemini gerçekleştirecektir ve bu öğrenme işlemi bütün parçalar hem eğitim hem de test verisi olarak kullanıldıktan sonra bitecektir. Bu veri kümesinin 10 defa çalışması demektir.

İşlem bittikten sonra “best\_estimator\_” komutuyla algoritma da en yüksek başarıya ulaşan parametreleri görmüş olduk.



Kod 22 – Skor 3

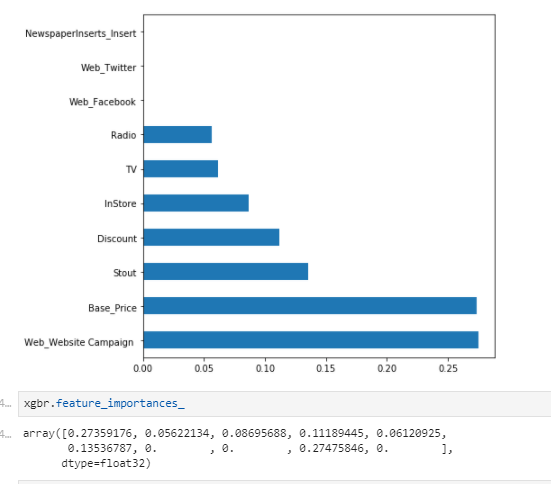
Parametre optimizasyonu sonrasında Aşırı Gradyan Artırma Regresyonu algoritmasının başarısı 0.824’den 0.853’3 arttığını gözlemlenmiştir.

Yapılan parametre optimizasyonlarından sonra elde edilen R2 sonuçları;

Aşırı Gradyan Artırma Regresyonu Algoritması : 0.853

Gradyan Eğilimli Ağaç Regresyonu Algoritması : 0.834

Rassal Orman Regresyonu : 0.789 sonuçlarını vermişlerdir. En iyi çalışan algoritmamız Aşırı Gradyan Artırma Regresyonu algoritması oluştur. Projemize Aşırı Gradyan Artırma Regresyonu algoritması ile devam edilecektir.



Kod 23 – Öznitelik Önem Katsayıları

Aşırı Gradyan Artırma Regresyonu Algoritmasının öznitelik önem değerleri Kod 23’teki gibidir.

**3.3.9 Makine Öğrenmesi Algoritmalarının Performanslarının Karşılaştırılması**

Araştırma boyunca elde ettiğimiz makine öğrenmesi performanslarının karşılaştırmasını yapılacaktır. Bu sayede veri kümemizin hangi algoritmalarda daha iyi çalıştığını gözlemleme şansımız olacaktır.

|  |  |
| --- | --- |
| **Makine Öğrenmesi Algoritmaları** | **R Kare skorları** |
| Polinominal Regresyon (Knime) | 0.819 |
| Gradyan Eğilimli Ağaç Regresyonu (Knime) | 0.939 |
| Rassal Orman Regresyonu (Knime) | 0.702 |
| Destek Vektör Regresyonu | -0.213 |
| AdaBoost Regresyonu | 0.747 |
| Aşırı Gradyan Artırma Regresyonu | 0.853 |
| Rassal Orman Regresyonu | 0.818 |
| Karar Ağacı Regresyonu | 0.722 |
| Gradyan Eğilimli Ağaç Regresyon | 0.839 |
| K En Yakın Komşu Regresyonu | 0.687 |
| Çok Katmanlı Algılayıcı Regresyonu | -115.870 |
| Polinominal Regresyon | 0.669 |

Tablo 1 – Performans Karşılatırması

Bütün makine öğrenmesi algoritmalarının performansların en iyi olduğu değerleri alarak bu tablo hazırlanmıştır. Tablo 1’de de görüldüğü üzere algoritmalar arasında en iyi performans gösteren algoritma Knime Programında oluşturmuş olduğumuz Gradyan Eğilimli Ağaç Regresyonu’dur fakat bu sonuca ulaşmak için yaptığımız öznitelik seçimi işlemlerinde veri kümemizden çıkan değişkenlerimiz olduğu için bu performansı en başarılı performans olduğunu söyleyemeyiz. Bir sonraki en iyi performans ve Gradyan Eğilimli Ağaç Regresyonu algoritmasını geliştirilerek oluşturulmuş olan Aşırı Gradyan Artırma Regresyonu’nun veri kümemize en iyi uyan algoritma olduğu söylenebiliriz. Ayrıca R2 skorları arasında eksi değerler bulunmaktadır. Bu eksi değerler genellikle o algoritmanın başarısının, bütün tahmin değerlerine ortalama değer verilerek oluşturulacak bir değerlendirmeden daha kötü olduğu şeklinde yorumlanmaktadır.

**3.3.10 Fiyat Esnekliği (Price Elasticity)**

Esneklik, pazarlama aktivitelerinin bizim satışımıza olan olumlu veya olumsuz olarak etkileri nelerdir sorusunu cevaplamaktadır. Bu işlem diğer pazarlama unsurları içinde geçerlidir. Doğrusal Regresyon modellerinde “beta\*(x.mean/y.mean)” olarak hesaplanmaktadır[26].

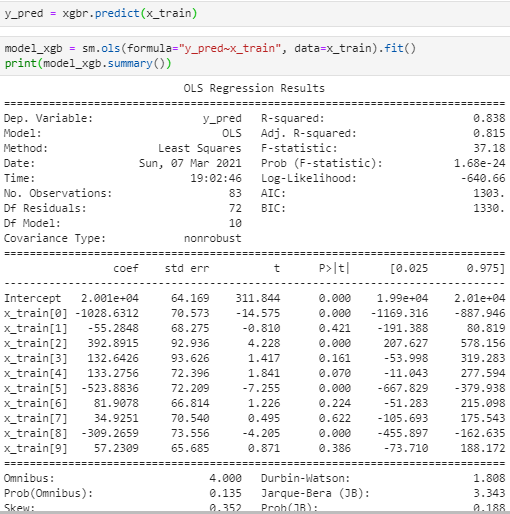
Bu hesaplama işlemini yapamıyoruz çünkü Aşırı Gradyan Artırma Regresyonu algoritmasından beta katsayılarını alamıyoruz bunun için başka bir yöntemle beta katsayılarımızı oluşturacağız.

1. **Sıradan En Küçük Kareler Regresyonu (Ordinary Least Squares Regression)**

Sıradan en küçük kareler regresyonu, bir veya birden fazla bağımsız değişken ile bir bağımlı değişken arasındaki ilişkiyi tahmin eden istatistiksel bir analiz yöntemidir. Yöntem; düz bir çizgi olarak yapılandırılmış bağımlı değişkenin gözlemlenen ve tahmin edilen değerleri arasındaki farkların kareleri toplamını en az indirerek ilişkiyi tahmin etmektedir [27].

Modelimizin beta katsayılarına erişmek için Aşırı Gradyan Artırma Regresyonu modeliyle tahmin ettiğimiz değişkenleri Sıradan En Küçük Kareler Regresyonu modeline sokarak elde edeceğiz.

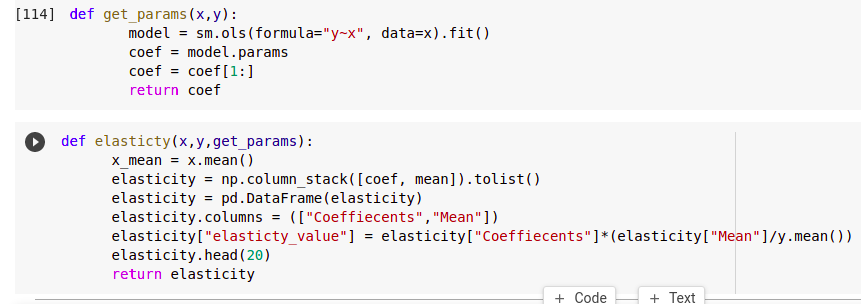
Sıradan En Küçük Kareler Regresyonu modelini çalıştırmak için “stastmodel” kütüphanesini kullanacağız.



Kod 24 – Sıradan En Küçük Kareler Regresyonu

Sıradan En Küçük Kareler Regresyonu modelini oluşturmak için “x\_train” değişkenimizi Aşırı Gradyan Artırma Regresyonu ile tahmin ettik ve model içinde “y\_pred” değişkeni olarak bu değeri girdik. “x” değişkeni olarak “x\_train” değerini girdik ve sonra olarak “data” değişkenine de “x\_train” verisi atayıp modelimizi “fit” ettik. Artık beta katsayılarını elde ettiğimize göre esneklik hesaplamasını yapabiliriz.

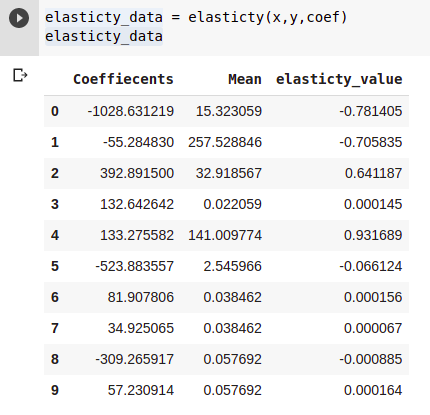
1. **Esneklik hesaplama**

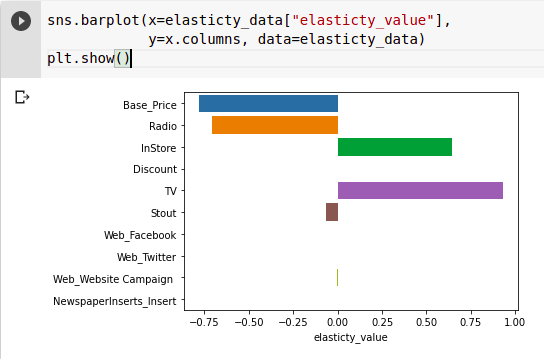


Kod 25 – Esneklik Hesaplama

Kod 25’te görüldüğü üzere iki adet fonksiyon oluştarak önce veri kümemizde bulunan değişkenlerin beta katsıyalarını hesapladık. İkinci fonksiyonda ise bir önceki fonksiyonda hesaplamış olduğumuz beta katsayılarını kullanarak değişkenlerimizin esneklik değerlerinin hesaplanmasını sağladık.

1. **Esneklik Değerleri ve Grafiği**

****



Kod 26 – Esneklik Sonuçları

**4. Sonuçların Değerlendirilmesi**

Kod 26’da da görüldüğü üzere değişkenlerimizin Esneklik değerleri gözükmektedir. Bu değerler bu değişkenlerde yapılacak %1’lik artışın fiyatlar üzerinde nasıl etkisi olacağını bize göstermektedir.

* Base\_Price değişkeninin Esneklik değeri -0.781 olarak hesaplanmıştır. Bu değerin anlamı bu birim fiyatta yapılacak olan %1’lik artış bizim toplam satışımızı %0.78 oranında düşüreceği anlamına gelmektedir
* Radio değişkenin Esneklik değeri -0.705 olarak hesaplanmıştır. Bu değerin anlamı radyoya pazarlama kanalına yapılacak %1’lik artışın toplam satışımızı %0.70 oranında düşüreceği anlamına gelmektedir.
* Instore değişkenin Esneklik değeri 0.641 olarak hesaplanmıştır. Bu değerin anlamını mağaza içi pazarlama harcamalarında yapacağımız %1’lik artışın toplam satışımızı %0.64 oranında arttıracağı anlamına gelmektedir.
* TV değişkenin Esneklik değeri 0.931 olarak hesaplanmıştır. Bu değerin anlamı televizyon pazarlama kanalına yapılacak %1’lik artışın toplam satımızı %0.931 oranında arttıracağı anlamına gelmektedir.

Bu veriler ışığında bu satışları yapmış olan satıcıya ilerleyen dönemlerdeki pazarlama harcamalarında ürünün birim fiyatında bir miktar indirime gidilmesi, Radyo kanalına yaptığı pazarlama harcamalarını kısmasını tavsiye edebiliriz. Radyo kanalından kısılan pazarlama bütçesini televizyon ve mağaza içi promosyonlara yönelik kullanması da bu sonuçlardan elde edebileceğimiz çıkarımlardan biridir. Aynı zamanda diğer küçük etkileri olan verilere bakıldığında “Stout” değişkeninde eksi değer gözlenmektedir. Bu nedenle mağaza stok sayılarının artırılmaması toplam satışımızın artmasında etkili olabilir. Ve son olarak elde edilen web reklamlarının çıktılarına bakılarak web site üzerinden yapılan pazarlama harcamaları diğer web kanallarından daha olumlu sonuçlar getirdiği söylenebilir.

KAYNAKÇA

[1]   Gian M. Fylgoni << The Journal of Advertising Research vol 58 issue 4 >>  1 Aralık 2018. [Çevrimiçi] – Ulaşılabilir: <http://www.journalofadvertisingresearch.com/content/58. Sayı /4. Makale /sayfa 390>

[2] Frances, P. H. (2005). <<The use of econometric models for policy simulation in marketing. Journal of Marketing Research,>> . [Çevrimiçi] – Ulaşılabilir; https://journals.sagepub.com/doi/abs/10.1509/jmkr.42.1.4.56891 42. Sayı, sayfa 4-14.

[3] Neff, J. (2007). << Representing PR in the Marketing Mix A Study on Public Relations Variables in Marketing Mix Modeling),>> [Çevrimiçi] – Ulaşılabilir; https://instituteforpr.org/wp-content/uploads/BG\_SmithKetchum.pdf 78(10), sayfa 8

[4] Micheal J. Wolfe, John C. Crotts << Marketing Mix Modeling for the Tourism Industry'dir" >> 16 Nisan 2011. [Çevrimiçi] – Ulaşılabilir file; // <https://www.tandfonline.com/doi/abs/10.1080/15980634.2011.11434633>, sayfa 2

[5] Micheal J. Wolfe, John C. Crotts << Marketing Mix Modeling for the Tourism Industry'dir" >> 16 Nisan 2011. [Çevrimiçi] – Ulaşılabilir file; // <https://www.tandfonline.com/doi/abs/10.1080/15980634.2011.11434633>

[6] Bagher Asgarnezhad Nouri ve Milad Soltani << Evaluating the Effect of Tourism Marketing Mix on Buying Holiday Homes in Cyprus>> 7 Eylül 2015. [Çevrimiçi] – Ulaşılabilir; <http://citeseerx.ist.psu.edu/viewdoc-/download?doi>= 10.1.1.1025.1031&rep=rep1&type=pdf

[7] Nielsen Holdings << MARKETING MIX MODELING WHAT MARKETING PROFESSIONALS NEED TO KNOW >> Mart 2014. [Çevrimiçi] – Ulaşılabilir; <https://www.nielsen.com/wp-content/uploads/sites/3/2019/04/marketing-mix-modeling-what-marketers-need-to-know.pdf> , sayfa 8-14

[8] Şadi Evren ŞEKER << CRISP-DM: Endüstriler Arası Standart İşleme – Veri Madenciliği için (Cross Industry Standard Processing – Data Mining) >> Temmuz 2018. [Çevrimiçi] – Ulaşılabilir; <https://ybsansiklopedi.com/wp-content/uploads/2018/08/crispdm.pdf> , sayfa 11-13

[9] <https://www.kaggle.com/veer06b/marrket-mix-dataset>

[10] İstatistik Merkezi << SPEARMAN KORELASYON ANALİZİ VE TEMEL PROBLEMLERİ >> [Çevrimiçi] – Ulaşılabilir; <https://www.istmer.com/spearman-korelasyon-analizi-ve-temel-problemleri/>

[11] M. Fevzi ESEN ve Mehpare Timor << ÇOK DEĞİŞKENLİ AYKIRI DEĞER TESPİTİ İÇİN KLASİK VE DAYANIKLI MAHALANOBİS UZAKLIK ÖLÇÜTLERİ: FİNANSAL VERİ İLE BİR UYGULAMA  
>> 21 Ağustos 2019 sayfa – 2 [Çevrimiçi] – Ulaşılabilir; :http://161.9.143.160:8080/bitstream/handle/20.500.12627/497/%C3%87OK%20DE%C4%9E%C4%B0%C5%9EKENL%C4%B0%20AYKIRI%20DE%C4%9EER%20TESP%C4%B0T%C4%B0%20%C4%B0%C3%87%C4%B0N%20KLAS%C4%B0K%20VE%20DAYANIKLI%20MAHALANOB%C4%B0S%20UZAKLIK%20%C3%96L%C3%87%C3%9CTLER%C4%B0%20F%C4%B0NANSAL%20VER%C4%B0%20%C4%B0LE%20B%C4%B0R%20UYGULAMA%20.pdf?sequence=1&isAllowed=y

[12] Emre Rençberoğlu << Fundamental Techniques of Feature Engineering for Machine Learning >> 1 Nisan 2019. [Çevrimiçi] – Ulaşılabilir; <https://towardsdatascience.com/feature-engineering-for-machine-learning-3a5e293a5114>

[13] Matthew Kramer << R2 STATISTICS FOR MIXED MODELS >> 2005 [Çevrimiçi] – Ulaşılabilir; <https://pdfs.semanticscholar.org/1b57/e7fa9a8b8f6921cc26eb33b0b70afc0a9849.pdf>, sayfa 3

[14] Deliktaş, B. Türker, H.T., Coşkun, H., Bikçe, M.,Özdemir, E. 2005. Genetik Algoritma Parametrelerinin Betonarme Kiriş Tasarımı Üzerine Etkisi. Bölgesel Jeoloji-Tektonik ve Sismotektonik Deprem Kaynak Mekanizmaları ve Dalga Yayınımı Sempozyumu, sayfa 23-25 Mart, Kocaeli, sayfa 46-54

[15] Yanıkoğlu, H., Özkara, E., Yüceer, M. 2010. Kinetik Model Parametrelerinin Belirlenmesinde Kullanılan Optimizasyon Tekniklerinin Kıyaslanması. 9. Ulusal Kimya Mühendisliği Kongresi(UKMK-9), 22-25 Haziran, Ankara.

[16] [https://heartbeat.fritz.ai/support-vector-regression-in-python-using-scikit-learn-89cc18e933b7#:~:text=Support%20vector%20regression%20(SVR)%20is,relationship%20between%20two%20continuous%20variables.&text=In%20the%20realm%20of%20machine,other%20common%20and%20popular%20algorithms](https://heartbeat.fritz.ai/support-vector-regression-in-python-using-scikit-learn-89cc18e933b7" \l ":~:text=Support vector regression (SVR) is,relationship between two continuous variables.&text=In the realm of machine,other common and popular algorithms).

[17] <https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.AdaBoostRegressor.html>

[18] <https://xgboost.readthedocs.io/en/latest/index.html>

[19] https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestRegressor.html

[20] <https://scikit-learn.org/stable/auto_examples/tree/plot_tree_regression.html>

[21] https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.GradientBoostingRegressor.html

[22] <https://bookdown.org/tpinto_home/Regression-and-Classification/k-nearest-neighbours-regression.html>

[23] <https://en.wikipedia.org/wiki/Multilayer_perceptron>

[24] <https://scikit-learn.org/stable/modules/generated/sklearn.preprocessing.PolynomialFeatures.html>

[25] Iwan Syarif, Adam Prugel-Bennett, Gary Wills << SVM Parameter Optimization Using Grid Search and Genetic Algorithm to Improve Classification Performance >> Aralık 2016 [Çevrimiçi] – Ulaşılabilir; <https://core.ac.uk/download/pdf/295538475.pdf>, sayfa 2

[26] Deepanshu Bhalla << A COMPLETE GUIDE TO MARKETING MIX MODELING >> Aralık 2019. [Çevrimiçi] – Ulaşılabilir: <https://www.listendata.com/2019/09/marketing-mix-modeling.html>

[27] Dudley L. Poston Jr . << Ordinary Least Squares Regression >> [Çevrimiçi] – Ulaşılabilir: <https://www.encyclopedia.com/social-sciences/applied-and-social-sciences-magazines/ordinary-least-squares> regression#:~:text=Ordinary%20least%20squares%20-(OLS)%20regression,and%20predicted%20values%20of%20the

[28] Ş. E. ŞEKER, «Her Seviyeye Uygun Uçtan Uca Veri Bilimi, Knime ile,» UDEMY, [Çevrimiçi] – Ulaşılabilir:: https://www.udemy.com/veribilimi/learn/v4/content. [Erişildi: Şubat 2021].

[29] Ş. E. ŞEKER, «Python ile Makine Öğrenmesi,» Udemy, [Çevrimiçi] – Ulaşılabilir; https://www.udemy.com/makine-ogrenmesi/learn/v4/content. [Erişildi: Şubat 2021].