**Öz**

Anormallik tespiti, anormallikleri tespit etmek için önemli bir araçtır.Finansal dolandırıcılık tespiti, bilgisayar ağına izinsiz giriş, insan davranış analizi, gen ekspresyon analizi ve çok daha fazlasını içeren birçok farklı alanda kullanılır. Son zamanlarda finans sektöründe dolandırıcılık faaliyetlerinin tespiti ile ilgili araştırmalara olan ilgi artmıştır.

Denetim veri kümelerindeki anormalliklerin tespiti, denetçi için inanılmaz zorlu bir süreci kapsıyor .Bir denetim sürecinde kullanılan başlıca teknikler olarak oran analizi,işlem testleri,örnekleme,veri modelleme ve veri analizi gösterilebilir.

Mali denetimlerde yaygın olarak uygulanan işlem verilerini analiz etme tekniği olarak ve veri setindeki düzensizlikleri tespit etmek için ekonomi dünyasında Benford yasası uygulanıyor.Benford yasası ile tekrar eden rakamlara bakılarak suistimal tespitleri yapılıyor.

Bu araştırma,denetim veri setlerindeki aykırı değerleri makine öğrenmesi algoritmaları aracılığıyla bulmayı amaçlıyor.Denetimsiz aykırı değer tespiti teknikleri kullanılarak ,denetçinin aykırı değerleri veya anormallığı bulmasına yardımcı olunması amaçlanıyor..

***Anahtar Kelimeler****:**unsupervised outlier detection,anomaly detection,fraud detection,*

**Abstract**

Sistematik literatür araştırmasına dayanarak, denetimsiz aykırı değer tespit teknikleri;

yakınlığa dayalı teknikler,kümeleme bazlı teknikler,istatiksel bazlı teknikler olarak sınıflandırılıyor.

Aykırı değer tespiti, denetimli öğrenme veya denetimsiz öğrenme veya yarı denetimli öğrenme yöntemi kullanılarak yapılabiliyor. Denetimli öğrenmede, aykırı değer tespit modelinin ve modelin dayandığı bu veri kümesinin aykırı değerlerinin bilindiği yani verilerin etiketlenmiş olduğu durumdur.Yarı denetimli olması durumunda ise yalnızca normal sınıftan oluşan ve anormallik içermeyen bir eğitim seti oluşturuluyor. Test seti de ayrılması gereken normal kayıtlardan ve anormallikerden oluşuyor. Denetimsiz öğrenme kullanılarak yapılan aykırı değer tespitlerinde ise, modeller veri setlerine göre eğitiliyor.Aykırı değerin özellikleri önceden tanımlı olmadığından dolayı bu öğrenme yönteminde, aykırı değerler bilinmemektedir.Fikir denetimsiz bir anormallik algılama yöntemi kullanılarak, verilerin yalnızca içsel özelliklere dayalı olarak puanlanmasıdır. Aykırı değer tespiti genellikle denetimsiz öğrenme yöntemi kullanılarak yapılıyor.Bu çalışma denetimsiz öğrenme üzerine odaklanmıştır.Denetimsiz aykırı değer algoritmalarından Pyod kütüphanesi referansıyla toplamda 14 farklı algoritma kullanılmıştır.(ABOD, CBLOF, HBOS, Feature Bagging, IF, LOF, KNN, OCSVM, MCD, PCA, LODA, SOD, LSCP, LMDD).

**1. Giriş**

Anormallik tespitili konusuyla ilgili toplam 110 farklı ülkeden 20.194 araştırmacı bulunuyor(www.wizdom.ai/topic/anomaly\_detection).Anomali tespiti ile en çok araştırılan konulara baktığımızda Isolation Forest,Cyber Attack,Fraud,Malware gibi konular görünüyor.

Aykırı değer, bir veri kümesindeki diğer gözlemlerden büyük ölçüde farklı olan herhangi bir veri noktasıdır. Anormallik tespiti, veri madenciliğinin önemli bir yönüdür.Ana amaç anormal veya olağandışı verileri tanımlamaktır.Aykırı değerlerin var olmasının birçok nedeni vardır.Hatalı veri girişi,sensörden okunan hatalı veriler veri setinde anormal değerler oluşmasına sebep olabilir.Örneğin MR görüntüsündeki anormallik,kötü huylu bir tümörün varlığını gösterebilir.Benzer şekilde bir kredi kartı işlemindeki anormal davranış dolandırıcılık faaliyetlerini gösterebilir.Bir ağdakı alışılamadık bir trafik modeli,bir bilgisyarın saldırıya uğradığı anlamına gelebilir.Bu çalışma finansal denetim sürecinde anormallik tespitine odaklanmıştır.Finansal verilerin yetersizliğinden dolayı sentetik verilerin nasıl üretilip üzerinden değerlendirme yapılacağı bu çalışmada belirtiliyor.

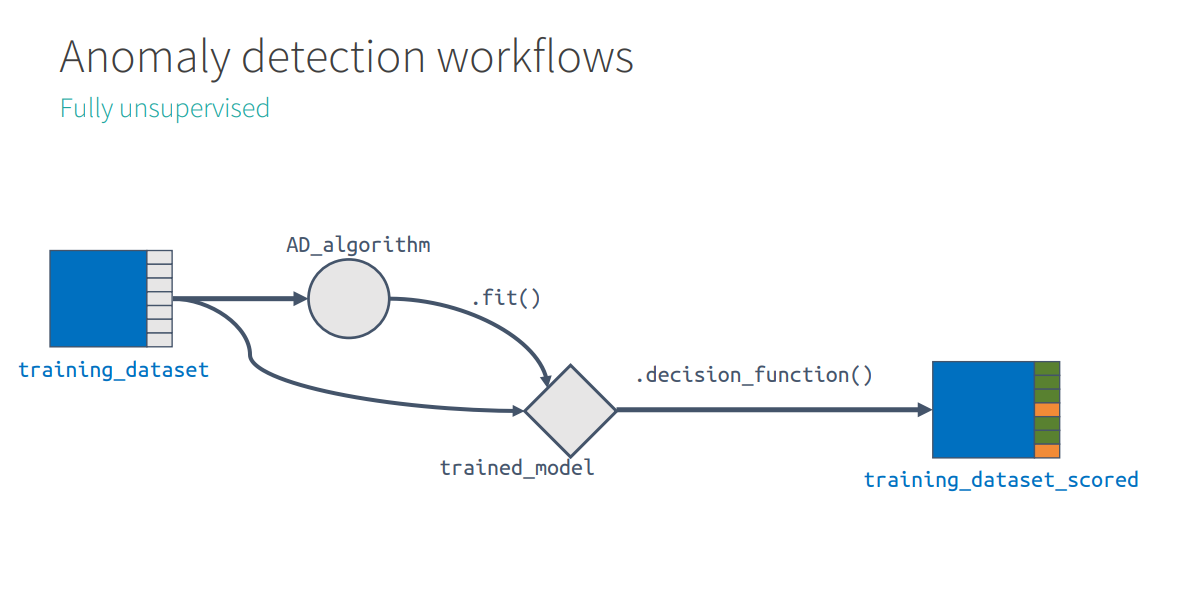
Bölüm 2, anomali tespiti ve anomali algılama algoritmaları hakkında bir arka plan sağlar.Bölüm 3 çalışma gerçekleştirilirken izlenen metadoloji hakkında bilgi verir.Bölüm 4 model kurulmasını içerir.Bölüm 5 kurulan modellerin değerlendirilmesi hakkında bilgi verir.Bölüm 6 yapılan çalışmanın Sckit-Learn ve Pycaret kütüphaneleri kullanılarak Power Bı programında dashboard oluşturulmasını içerir.Bölüm 7 sonuç ve önerileri içerir.Bölüm 8 dolandırıcılık tespit türleri üzerine tartışmayı içerir.

**2.1 Relate Works**

Anormallik tespitini içeren çoğu çalışma, verileri anomali veya anomali değil olarak etiketler. Etiketleme, araştırmacıların anormalliği eğitmesine yardımcı olur algılama algoritmaları ve algoritmik performansı değerlendirme test verisi üzerinden yapılır. Bitcoin ağının yeniliği nedeniyle, yalnızca sınırlı dolandırıcılık olarak bildirilen işlem sayısı bu denetimli öğrenme tekniğinin kullanımını olanaksız kılar. Hem hileli kullanıcıları hem de ilişkili işlemlerini tespit etme öncülüğünde, k-ortalama kümeleme, Mahalanobis yöntemi kullanıldı.Denetimsiz Destek Vektör Makineleri (SVM) Bitcoin veri kümesinde kullanıldı. Kullanılan algoritmalar, 30 bilinen vakadan 3'ünü tespit etti. Sonraki çalışmalarında, ağdan çıkarılan tam veri kümesini kullanarak 2013 yılının Nisan ayında , k-ortalamalı kümelemenin benimsenmesi, Yerel Aykırı Değer Faktörü (LOF) kullanıldı..Dahası modellerine benzeyen sentetik düğüm verileri oluşturuldu. Modelin performansı sentetik verilere dayalı olarak, bir doğruluk seviyesine ulaştığı bulundu.

Literatüre bağlı kalarak bu çalışmada pyod kütüphanesi ile sentetik veriler üretilip sentetik veriler üzerinden değerlendirilme yapılmıştır.Pyod kütüphanesinden 14 farklı algoritma kullanılarak veri setinin genel olarak %10’u aykırı değer olarak tespit edilmiştir.

**3. Methadology**

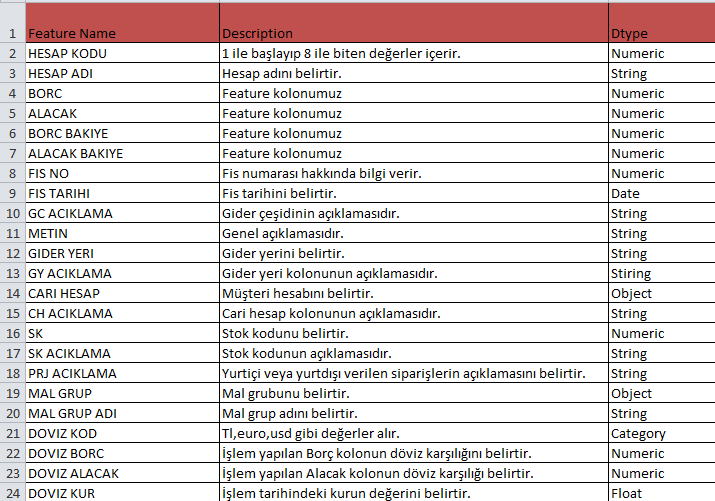


(Anomaly detection workflows)

Veri seti model oluşturulmadan önce, scikit-learn kütüphanesinden referans alan minmaxscaler ve robustscaler yöntemleri ile scale edildi.Oluşturulan modele algoritmaya göre fit edilerek train işlemi gerçekleştirilir.Train işleminden sonra decision\_function() fonksiyonuyla anormallik skoru hesaplanır.predict() fonksiyonu ile tahmin etme işlemi gerçekleştirilir.

**3.1 Data Description**

Bu veri setinde 135344 satır,11017 adet tekrarlayan veri ve toplam 23 kolon bulunuyor.Bunlardan 7 tanesi kategorik 9 tanesi text türünden oluşuyor.Veri seti toplamda belllek 23.7 mb yer kaplıyor.



**3.2 Denetimsiz Aykırı değer tespit türleri**

**3.2.1. Açı Bazlı Aykırı Değer Tespiti (ABOD) :** Her nokta ile komşuları arasındaki ilişkiyi dikkate alır. Bu komşular arasındaki ilişkileri dikkate almıyor. Ağırlıklı kosinüs puanlarının tüm komşulara olan varyansı, dıştaki puan olarak görülebilir.

ABOD çok boyutlu veriler üzerinde iyi performans gösteriyor.3 farklı parametre alıyor.Veri setindeki aykırı gözlem değerini ifade eden contamination parametresini 0.1 olarak ,komşu küme sayısını ifade eden n\_neighbors küme sayısını 5 olarak seçiyoruz.

**3.2.2. İsolation Forest** : Scikit-learn kütüphanesini dahili olarak kullanır. Bu yöntemde, veri bölümleme bir dizi ağaç kullanılarak yapılır. İzolasyon Ormanı, noktanın yapı içinde ne kadar izole olduğuna bakan bir anormallik puanı sağlar. Anormallik puanı daha sonra normal gözlemlerden aykırı değerleri belirlemek için kullanılır.İzolasyon Ormanı çok boyutlu veriler üzerinde iyi performans gösteriyor.Sckit learn,Pyod,Pycaret kütüphanesi olmak üzere 3 farklı kütüphanede kullanılmıştır.

**3.2.3. Clustering based Local outlier detection** : Verileri küçük kümeler ve büyük kümeler halinde sınıflandırır. Anormallik puanı daha sonra noktanın ait olduğu kümenin büyüklüğüne ve en yakın büyük kümeye olan mesafeye göre hesaplanır.Alfa ve beta parametrelerini kullanarak kümeleri küçük kümeler ve büyük kümeler halinde sınıflandırır. Anormallik puanı daha sonra noktanın ait olduğu kümenin boyutuna ve en yakın büyük kümeye olan mesafeye göre hesaplanır.

**3.2.4 Histogram based local outlier :** Özellik bağımsızlığını varsayan ve histogramlar oluşturarak aykırı değer puanını hesaplayan etkili, denetimsiz bir yöntemdir.

Çok değişkenli yaklaşımlardan çok daha hızlıdır fakat precision oranı çok daha azdır.

Her özellik için yüksekliğin bulunduğu bir histogram oluşturulur. (namic bin-width histograms) Histogramlar daha sonra normalleştirilir, böylece maksimum yükseklik 1.0'dır ve her biri özelliğin ağırlığına eşittir.

**3.2.5 K-Nearest Neighbors Detector : Herhangi** bir veri noktası için, k'inci en yakın komşusuna olan uzaklık, dıştaki puan olarak görülebilir.

Pratik bir kural olarak, k, 10 <k aralığında olacaktır

PyOD, üç kNN dedektörünü destekler :

En büyük: Aykırı değer olarak k'inci komşunun mesafesini kullanır

Ortalama: Uç değer olarak tüm k komşularının ortalamasını kullanır

Medyan: Aykırı değer olarak k komşuya olan mesafenin medyanını kullanır

**3.2.6 Local Outlier Factor :** Her örneğin anormallik skoruna Yerel Aykırı Değer Faktörü adı verilir. Komşularına göre belirli bir numunenin yerel yoğunluk sapmasını ölçer.

Anomali skorunun, nesnenin çevredeki mahalleye göre ne kadar izole olduğuna bağlı olması yereldir.

Daha kesin olarak yerellik, uzaklığı yerel yoğunluğu tahmin etmek için kullanılan k-en yakın komşular tarafından verilir.

Bir numunenin yerel yoğunluğu, komşularının yerel yoğunlukları ile karşılaştırılarak, komşularından önemli ölçüde daha düşük yoğunluğa sahip numuneler belirlenebilir. Bunlar aykırı değerler olarak kabul edilir.

Bu algoritma için aynı kütüphaneden fit\_predict fonksiyonu kullanıldı.Decision\_function() ile anormallik skoru hesaplandı.

LOF puanını hesaplamak için,üç adım hesaplanmalıdır:

1. Her x kaydı için k en yakın komşular bulunmalıdır .

2. Bu k en yakın komşular N k kullanılarak , bir kaydın yerel yoğunluğu hesaplama yoluyla tahmin edilir.(yerel ulaşılabilirlik yoğunluğu (LRD))

3. Son olarak, LOF skor kendi içinde LRD kaydını hesaplar.

LOF puanı bu nedenle temelde yerel yoğunlukların bir oranıdır

**3.2.7 One-Class SVM :** Bir sınıfı modellerken, algoritma çoğunluk sınıfının yoğunluğunu yakalar ve yoğunluk fonksiyonunun uç noktaları üzerindeki örnekleri aykırı değerler olarak sınıflandırır. SVM'nin bu modifikasyonu, Tek Sınıflı SVM olarak adlandırılır.

Standart bir SVM'den temel farkı, denetimsiz bir şekilde uyması ve C gibi marjı ayarlamak için normal hiperparametreleri sağlamamasıdır .

Bunun yerine, destek vektörlerinin hassasiyetini kontrol eden ve verilerdeki aykırı değerlerin yaklaşık oranına, örneğin% 0,01'e ayarlanması gereken bir hiperparametre " nu " sağlar.

**3.2.8 PCA :** Temel bileşen analizi (PCA), aykırı değerleri tespit etmek için kullanılabilir. PCA, veriyi daha düşük boyutlu bir alana yansıtmak için verinin Tekil Değer Ayrıştırmasını kullanan doğrusal bir boyut indirgemesidir.

Bu prosedürde, verilerin kovaryans matrisi, özdeğerlerle ilişkili özvektörler adı verilen ortogonal vektörlere ayrıştırılabilir. Yüksek özdeğerlere sahip özvektörler, verilerdeki varyansın çoğunu yakalar.

Bu nedenle, k özvektör tarafından oluşturulan düşük boyutlu bir hiper düzlem, verilerdeki varyansın çoğunu yakalayabilir. Bununla birlikte, aykırı değerler normal veri noktalarından farklıdır ve bu, küçük özdeğerlere sahip özvektörler tarafından oluşturulan hiper düzlemde daha belirgindir.

Bu nedenle, aykırı değer puanları, tüm özvektörler üzerinde bir örneğin öngörülen mesafesinin toplamı olarak elde edilebilir.

Score(X) = Her örnek ile seçilen özvektörler tarafından oluşturulan hiper düzlem arasındaki ağırlıklı öklid mesafesinin toplamını belirtir.

**3.2.9 Elliptic Envelope :** Aykırı değer tespitini gerçekleştirmenin yaygın bir yolu, normal verilerin bilinen bir dağıtımdan geldiğini varsaymaktır (örneğin, veriler Gauss'a göre dağıtılmıştır). Bu varsayımdan yola çıkarak, genellikle verilerin "şeklini" tanımlamaya çalışırız ve dıştaki gözlemleri, uygun şekilden yeterince uzakta duran gözlemler olarak tanımlayabiliriz.

Scikit-learn covariance.EllipticEnvelope, verilere sağlam bir kovaryans tahminine uyan bir nesne sağlar ve böylece merkezi modun dışındaki noktaları göz ardı ederek merkezi veri noktalarına bir elips sığdırır.

Örneğin, başlangıç ​​verilerinin Gauss dağılımlı olduğunu varsayarsak, başlangıç ​​konumunu ve kovaryansı sağlam bir şekilde (yani aykırı değerlerden etkilenmeden) tahmin edecektir. Bu tahminden elde edilen Mahalanobis mesafeleri, uzaklığın bir ölçüsünü türetmek için kullanılır.

**3.2.10 Feature Bagging :** Bir özellik torbalama dedektörü, veri kümesinin çeşitli alt örneklerine bir dizi temel dedektöre uyar. Tahmin doğruluğunu iyileştirmek için ortalama alma veya diğer kombinasyon yöntemlerini kullanır

Varsayılan olarak, Yerel Aykırı Değer Faktörü (LOF) temel tahmin aracı olarak kullanılır. Bununla birlikte, kNN ve ABOD gibi herhangi bir tahmin edici, temel tahmin edici olarak kullanılabilir.

Özellik torbalama ilk olarak , özelliklerin bir alt kümesini rastgele seçerek n alt örnek oluşturur. Bu, temel tahmin edicilerin çeşitliliğini ortaya çıkarır. Son olarak, tahmin puanı, tüm temel dedektörlerin ortalamasının alınması veya maksimumunun alınmasıyla oluşturulur.

**3.2.11 Minimum covariance determinant :** Giriş değişkenleri bir Gauss dağılımına sahipse, aykırı değerleri tespit etmek için basit istatistiksel yöntemler kullanılabilir.

Örneğin, veri kümesinin iki giriş değişkeni varsa ve her ikisi de Gauss ise, o zaman özellik alanı çok boyutlu bir Gauss oluşturur ve bu dağılımın bilgisi, dağılımdan uzak olan değerleri tanımlamak için kullanılabilir.

Bu yaklaşım, normal verileri kapsayan bir hiper küre (elipsoid) tanımlanarak genelleştirilebilir ve bu şeklin dışında kalan veriler bir aykırı değer olarak kabul edilir. Bu tekniğin çok değişkenli veriler için verimli bir şekilde uygulanması, Minimum Kovaryans Belirleyicisi veya kısaca MCD olarak bilinir.

Minimum Kovaryans Belirleyici kovaryans tahmincisi, Gauss'a göre dağıtılmış verilere uygulanacaktır, ancak yine de tek modlu, simetrik bir dağılımdan alınan verilerle ilgili olabilir. Çok modlu verilerle kullanılması amaçlanmamıştır (bir MinCovDet nesnesine uymak için kullanılan algoritma böyle bir durumda büyük olasılıkla başarısız olacaktır). Çok modlu veri kümeleriyle başa çıkmak için projeksiyon izleme yöntemleri dikkate alınmalıdır.

**3.2.12 Subspace Outlier Detection :** Alt uzay aykırı değer tespiti (SOD) şeması, yüksek boyutlu bir özellik uzayının değişen alt uzaylarında aykırı değeri tespit etmeyi amaçlar.

Her veri nesnesi için SOD, veri nesnesinin komşuları tarafından yayılan eksene paralel alt uzayı araştırır ve nesnenin bu alt uzaydaki komşularından ne kadar saptığını belirler.

**3.2.13 Paralel Aykırı Değer Topluluklarının (LSCP) Yerel Olarak Seçmeli Kombinasyonu** :

LSCP, bir test örneğinin yerel bölgesinde yetkin dedektörleri seçen, denetimsiz bir paralel aykırı değer algılama grubudur. Bu uygulama, Ortalama Maksimum stratejisi kullanır. İlk olarak, temel dedektörlerin heterojen bir listesi eğitim verilerine uydurulur ve ardından her tren örneği için sözde bir temel gerçek oluşturulur ve maksimum aykırı değer puanı alınarak oluşturulur.

Her test örneği için:

1) Yerel bölge, birden çok yinelemede tanımlanmış bir eşikten daha sık meydana gelen rastgele örneklenmiş özellik alt uzaylarındaki en yakın eğitim noktaları kümesi olarak tanımlanır.

2) Yerel bölge kullanılarak, yerel bir sözde yer gerçeği tanımlanır ve her bir taban dedektörünün eğitim aykırı değer puanları ile sözde yer gerçeği arasında pearson korelasyonu hesaplanır.

3) Bir histogram, pearson korelasyon skorlarından oluşturulmuştur; En büyük bölmedeki dedektörler, verilen test örneği için yetkin temel dedektörler olarak seçilir.

4) Seçilen yetkin dedektörlerin ortalama aykırı puan puanı nihai puan olarak alınır.

**3.3 Temel veri analizi ve veri setinin temizlenmesi**

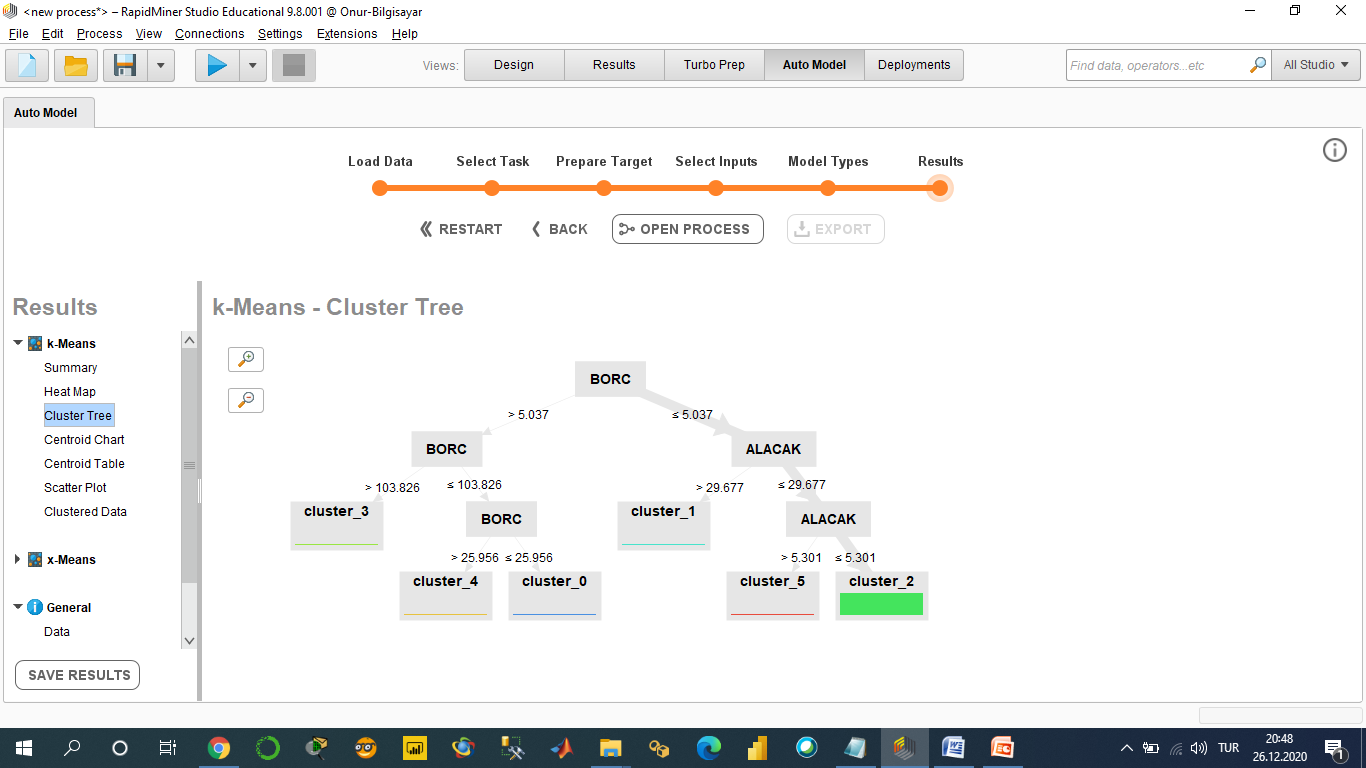
Borc ve borc bakiye kolonlarındaki hatali veri girişleri kaldılırılarak 0 değeri ile degiştirildi.Veri tipi numeric olarak belirlenen kolonlar “0” degeri ile veri tipi categoric olarak belirlenen kolonlar “yok” degişkeni ile dolduruldu.Bu bölümde pandas profiling,dtale,sweetviz,pandas,numpy kütüphaneleri kullanılarak veri seti analizi yapıldı.Veri setindeki kolonların betimsel istatistik değerlerine bakılarak,Borc,Alacak ve diğer kolonların değer aralığına,standart sapmasına,ortalamasına,1.çeyrek,2.çeyrek,3.çeyrek değerleri hakkında bilgi edinildi.Yapılan çalışma “EDA\_deneme\_csv.ipynb” dosyalı olarak kaydedildi.

Yapılan çalışma sonrası elde edilen çıktılar to\_html komutuyla html dosyası olarak kaydedildi.Anomali tespiti yapacağımız kolonlardan borc kolonu incelendiğinde borc bakiye ve döviz borc kolonu ile arasında pozitif korelasyon olduğu belirlenmiştir.Borc kolonun frekansına baktığımızda %22.3 ile 0 değerinin olduğu gözüküyor.

**3.4 Kümeleme**

Bu aşamada Rapidminer programı ve Google Colab olmak üzere 2 farklı çalışma yapıldı.Bunlardan ilki, Rapidminer programının kullanıldığı,kümeleme için sadece BORC,ALACAK,FIS TARIHI kolonlarından olusan yeni bir veri seti oluşturuldu.Çok değişkenli analizlerde değişkenlerin standartlaştırılması doğru analiz için önemli bir rol oynar.Farklı ölçeklerde ölçülen değişkenler analize eşit katkıda bulunmadığından dolayı standardizasyon işlemi, yani verilerin aynı ortalama (0) ve aynı standart sapmaya (1) sahip olması için Rapidminer programında “Standardizer” işlemi veri setine uygulandı.Veri seti ölçeklendikten hemen sonra modelde küme sayısı arttırıldığında “avarage cluster” uzaklığı yükseldiği tespit edildi.Bundan dolayı X means yönteminde optimum küme sayısı 8 , K Means yönteminde 6 olarak belirlendi.

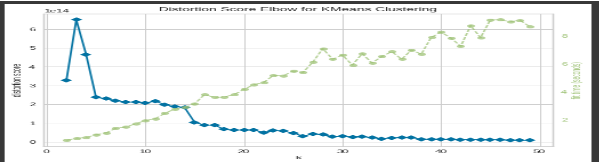
* A ) K-Means yönteminde uzaklık metriği olarak Squared Euclidean Distance yöntemi kullanılıp “avarage cluster” uzaklığı 0.274 olarak bulundu.Davies –Bouldin yani küme içi dağılımın küme arası ayrıma oranının bir fonksiyonu, daha düşük bir değer kümelemenin daha iyi olduğu anlamına geliyor.Her bir kümenin bir en benzer olduğu arasındaki ortalama benzerlik,tüm kümeler arasından ortalamasının alındığı bu yöntemde Davies Bouildin scoru 0.384 olarak bulundu.
* B ) X-Means yönteminde uzaklık metriği olarak Squared Euclidean Distance yöntemi kullanılıp ortalama küme içi uzaklık 0.107 olarak bulundu.Davies Bouldin scoru 03.51 olarak bulundu.



(K-means kümeleme ağacı RapidMiner Programı kullanılarak oluşturulmuştur.)

K-Means kümeleme ağacına baktığımızda Borc kolonun standardize edilerek elde edilen değerlerin 5.037 den büyük veya küçük olmasına göre dallar 2 ayrı parçaya ayrılır.

İkinci olarak Google Colab kullanılarak yapılan çalışmada ise gerekli kütüphaneler import edildikten sonra Borc,Alacak,Borc bakiye,Alacak bakiye kolonları feature olarak seçildi.Sklearn.preprocessing altından Standardscaler metodu çağrılarak verilerin ortalaması 0 standart sapması 1 olarak standardize edildi.Model parametrelerinden n\_clusters parametresi rastgele olarak 7 belirlendi.Cluster\_centers\_ ile kümelenen verilerin merkezi bulunduktan sonra labels\_ komutu ile kümeler bulundu.fit\_predict komutu ile tahmin edilen veriler pandas dataframe haline getirilerek değişken ataması gerçekleştirildi.Matplotlib kütüphanesi aracılığıyla çıkan sonuçlar görselleştirildi.Model oluşturduktan sonra optimum k sayısının belirlenmesi adına yellowbrick kütüphanesi kullanılarak elbow grafiği çizdirildi.Grafik 3,4,5 değerleri arasında kesin bir sonuç vermediğinden dolayı nesnenin diğer kümelere kıyasla kendi kümesine ne kadar benzediğinin değerlendirilmesinin bir ölçütü olan silhouette\_score cağrılarak optimum k değeri 0.997838 ile 3 olarak bulundu.n\_clusters değerine 3 atanarak tekrar model oluşturuldu ve model homogenity\_score ile değerlendirilerek 0.98 bulundu.



(Optimum K komşu sayısı yellowbrick kütüphanesi referans alınarak Elbow Plot yöntemi ile bulunmuştur.)

Optimum K yöntemini bulmak için yellowbrick kütüphanesi yüklenerek elbow grafiği çizdirildi. K değer aralı 1-15 arasında belirlendi.For döngüsüne sokularak belirlenen k değerinde inertia\_ fonksiyonu ile toplam mesafe boş bir listeye aktarılarak hesaplandı.Grafikte 3,4 ve 5 arasında kalınılan değer sklearn kutuphanesinden silhouette score import edilerek çıkan sonuçarlarla k değeri 3 olarak belirlendi. K değeri 3 olarak belirlenerek model oluşturuldu ve kümeleme yapıldı.

**3.5 Model ön hazırlık**

Bir veri kümesi, bir nokta anormalliği algılama problemini temsil edecek şekilde önceden işlendiğinde,

denetimsiz anormallik algılama algoritmasının uygulanabilmesinden önceki son adım standardizasyon işlemidir.

Standardizasyon işlemi , her özelliğin ortalama sıfır ve standart sapması bir olacak şekilde dönüştürülmesi işlemidir.

Bu çalışma veri setindeki aykırı değer tespiti üzerine odaklandığından aykırı değerlere duyarlı ve aykırı değerlere duyarlı olmayan yöntemler,MinMaxScaler ve RobustScaler olmak üzere 2 farklı standardizasyon yöntemi kullanıldı.Bu açıklanabilir bir görselleştirme oluştuırmak adına çok önemlidir.

MinMaxscaler, verinin 0 ile 1 arasında değerler aldığı bir durumdur.Buarada dağılım,verinin dağılımı ile benzerdir ve “outlier” yani dışta kalan verilere karşı hassasiyet durumu vardır.

RobustScaler, aykırı değerlere sahip verilerde daha iyi sonuç verir.Veri dağılımı ile benzerlik gösterir ancak aykırı değerler dışarıda kalır.Medyan değeri sonradan kullanılamak üzere elenir ve değerler 1.ve 3.çeyrekler aralığına oturturulur.

Scikit learn : Veri seti minmaxscaler / robustscaler ile standardize edildi.

Pyod : Veri seti standardize fonksiyonu ile standardize edildi

Pycaret : Robustscaler ile standardize edildi.

**4.Model kurulması**

Bu çalışmada model 3 farklı kütüphane üzerinden kurulmuştur.

Pyod kütüphanesinde 2 farklı çalışma yapılmıştır.İlk çalışmada feature olarak “BORC “ ve “ALACAK” değerlerinin birbirinden çıkarılıp mutlak değerinin alınmasıyla oluşan “BAKIYE” kolonu seçilmiştir.İkinici yapılan çalışmada ise,”Borc”, “ALACAK” , “BORC BAKIYE”, “ALACAK BAKIYE” kolonları feature olarak seçilmiştir ve çıkan sonuçlar karşılaştırıldığında “BAKIYE” kolonunu daha üstün performans gösterdiği saptanmıştır.

Sckit-learn kütüphanesinde feature olarak “BAKIYE” kolonu seçilmiştir.

Pycaret kütüphanesinde 2 farklı çalışma yapılmıştır.İlk çalışmada feature olarak “BORC” , “ALACAK” kolonları seçilmiştir.İkinci yapılan çalışmada ise “BORC” , “ALACAK” , “BORC BAKIYE”, “ALACAK BAKIYE” kolonları seçilmiştir.Borc ve alacak kolonuyla oluşturulan modelin daha üstün performans gösterdiği saptanmıştır.

**4.1 Pycaret**

Bu bölümde pycaret kütüphanesi kullanılarak 2 farklı çalışma yapılıyor.İlk olarak “BORC” , “ALACAK” , “BORC BAKIYE” , “ALACAK BAKIYE” kolonları kullanılıp, kolonlar standardize edilmeden model kuruluyor.İkinci olarak “BORC” ve “ALACAK” kolonları RobustScaler yöntemi ile standardize edilerek, modek kuruluyor.

Veriler veri temizleme aşamasında temizlendikten sonra, pycaret kütüphanesi yükleniliyor.Pycaret kütüphanesi altından pycaret.anomaly import ediliyor.İlk aşama da etup fonksiyonu çağrılarak içine hiperparemetre olarak , feature değerlerimiz giriliyor.Normalize parametresi = True olarak , normalize\_method yöntemi ‘robust’ olarak , loglarımızın çıktısını almak adına da log\_experiment parametresi True olarak belirleniyor.2. aşamada ise create\_model fonksiyonu çağrılarak 8 farklı algoritma (PCA, SVM, LOF, HISTOGRAM, CLUSTER, IFOREST, ABOD, KNN) üzerinden model oluşturuluyor.3. aşamada ise assign\_model() fonksiyonu kullanılarak veriler anormal/anormal değil olarak etiketleniyor.Value\_counts() fonksiyonu yardımıyla modeldeki anormal veri sayısı belirleniyor.4.aşamada ise sonuçlar evaluate\_model() fonksiyonu kullanılarak değerlendiriliyor.5.aşamada ise veri setinde olmayan yeni veriler

predict\_model() fonksiyonu kullanılarak tahmin ediliyor.Son olarak save\_model() fonksiyonu kullanarak model bir daha kullanılmak üzere kaydediliyor.Mflow ui kütüphanesi kullanılarak yapılan tüm işlemler csv dosyası olarak kaydediliyor.

**4.2 Sckit-learn**

Veriler 3.2 de belirtildiği gibi temizlenerek modele hazır hale getiriliyor.Bu bölümde Sckit-learn kütüphanesi kullanılarak yapılan anomali tespitinde feature olarak “BORC” ve “ALACAK” kolonlarının farkının mutlak değer içine alınmasıyla oluşan “BAKIYE” kolonu kullanılarak model oluşturuluyor.

BAKIYE kolonu sklearn.preprocessing altında bulunan 2 farklı yöntem kullanılarak(MinMaxScaler ve RobustScaler yöntemleri) BAKIYE kolonu scaling ediliyor.İlk olarak veriler MinMaxScaler yöntemi ile scale ediliyor.Bakiye kolonunun ilk 70000 satırı eğitim için,30000 satır da test için kullanılıyor.Bu bölümde Scikit-learn kütüphanesinden Isolation Forest, Local Outlier Factor, Elliptic Envelope , One Class SVM kullanılarak anormallik tespiti yapılıyor.4 farklı algoritma ile model kuruluyor.Kurulan modellerin anormallik skoru, decision\_function() yardımıyla bulunuyor.Daha sonra predicit() fonksiyonu kullanılarak verilerin anormal olup olmadığı tahmin ediliyor.Sonuçlar pandas dataframe haline getiriliyor.Decision\_function() yardımıyla belirlenen anormallik skoru “Anamoly Score” adlı değişkene, predict() fonksiyonu ile belirlenen verilerin anormal mi değil mi olduğunu gösteren değer (-1 : Outlier / 1: İnlier) “Anamoly” değişkenine atanıyor.Pandas dataframe haline getirilen Anormallik skorunun iqr (çeyrekler açıklığı) değeri fonksiyon yazılarak, oluşturulan her model için bulunuyor.

Anamoly kolonu groupby() fonksiyonu ile gruplanarak describe() fonksiyonu yardımıyla da “BAKIYE” ve “ANAMOLY SCORE” kolonunun betimsel istatistiklerine bakılıyor.Çıkan sonuçlar gözlemleniyor ve tablo haline getiriliyor.Kurulan modellerde Novelty parametresi “True” olarak ayarlanıp, veya veriler numpy array formuna dönüştürülüp (reshape(-1,1) yeni veriler predict() fonksiyonu ile tahmin ediliyor.

Yapılan işlemler RobustScaler yöntemi kullanarak tekrarlanıyor.MinMaxscaler yöntemi kullanılarak yapılan ölçekleme işleminde,”outlier” verilere karşı hassasiyetlik gösterdiği tespit ediliyor.Robust Scaler yöntemi aykırı değerlere sahip verilerde daha iyi sonuç gösteriyor çünkü bu yöntemde aykırı değerler dışarıda kalıyor ve medyan değeri sonradan kullanılmak üzere 1. ve 3. Kartil aralığına oturtuluyor.

**4.3 Pyod**

Veriler veri temizleme aşamasında temizlendikten sonra 20 den farklı algoritmayı destekleyen pyod kütüphanesini kullanııyoruz.Bu çalışmada pyod kütüphanesinden 13 farklı algoritma kullanılmıştır.2 farklı ipynb dosyası üzerinde çalışılmıştır.İlk çalışma dosyasında feature olarak borc,alacak,borc bakıye,alacak bakiye kolonları seçilmiştir.Generate data ile 70000 adet train datası,30000 adet test datası oluşturulup train\_only parametresi True seçilerek test datası sentetik data ile üretilmiştir.Sonrasında sklearn kütüphanesinden model\_selection adı altından train\_test\_split fonksiyonu çağrılarak featurelar %70 eğitim %30 test olarak ikiye bölünmüştür.X\_train ve X\_test datası pyod kütüphanesinin standardizer fonksiyonu ile standardize edilerek modele hazır hale getirilmiştir.Model için gerekli tüm kütüphaneler ve değerlendirme skorları import edilmiştir.n\_ite parametresi 10 olarak n\_classifiers parametresi 10 olarak outliers\_fraction parametresi y değikenindeki 0 olmayan değerlerin toplam değerlere oranından belirlenmiştir.Modeller sözlük yapısı kullanılarak oluşturulmuştur.Time ve os yani sistem kütüphaneleri import edilerek algoritmaların çalışma süresi alınmıştır.Predict fonksiyonu ile verilerin anormal olup olmadığı, decision\_function ile anormallik skoru tespit edilmiştir. Pyod kütüphanesin visualize komutu ile kullanılan her algoritma için sonuçlar görselleştirilmiş , evaluate\_print fonksiyonu ile sonuçlar değerlendirilmiştir.Tespit edilen anormallik skoru bir değişkene atanıp sonrasında pyod kütüphanesinin standardizer modülü ile standardize edilip avarege,maximization,median fonksiyonları kullanılarak değerlendirilme yapılmıştır.Değerlendirme sonuçları tablo olarak oluşturulmuştur.İkinci çalışmada ise feature olarak borc ve bakiye kolonları birbirinden çıkarılıp mutlak değeri alınarak bakiye kolonuna atanmıştır.Aynı işemler tekrarlanarak birbirleriyle karşılaştırılmıştır.

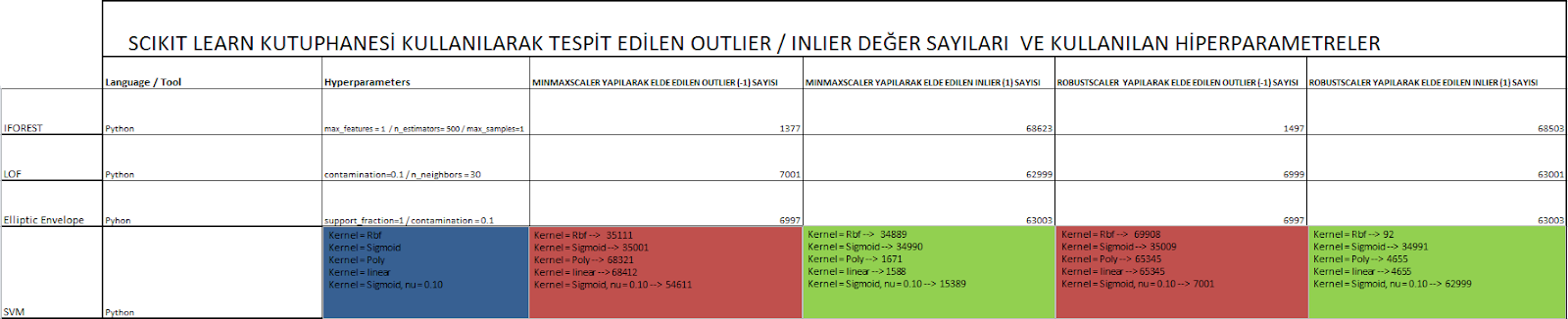
**4.3.1 SENTETİK DEĞER ÜRETİLMESİ**

Pyod kütüphanesi altından generate\_data fonksiyonu kullanarak 70000 train verisi 30000 test verisi sentetik data olarak üretilmiştir. Normal veriler, çok değişkenli bir Gauss dağılımı ile üretilir ve aykırı değerler tek tip bir dağılımla oluşturulur.

**5.Değerlendirme**

KNN,ABOD,IFOREST,Cluster Based,Histogram Based,LOF,SVM,PCA,CBLOF,LODA yöntemi kullanılarak anomali tespiti yapıldı.ve ABOD % 0 algılama ile en düşük algılamaya sahiptir.%0.083 ile Histogram based en düşük 2.algılamaya sahip algoritmadır.Knn,Cluster based,LOF,SVM algoritması kullanılarak veri setinin 5,264 ‘ ü,IF algoritmasını kullanarak veri setinin 4,999’u, PCA algoritmasını kullanarak veri setinin %3.382 ‘si anomali olarak tespit edilmiştir.Pyod kütüphanesi referans alınarak yapılan anomli tespitinde ise HBOS algoritması veri setinin %9.871’ini tespit ederken , contamination parametresi diğer algoritmalar için 0.1 olarak tanımlandığından veri setinin %10 ‘ u aykırı değer olarak tespit edilmiştir.

Scikit-learn kütüphanesi kullanılarak yapılan anomali tespitinde MimaxScaler,RobustScaler olmak üzere 2 farklı scaling yöntemi karşılaştırılmıştır.



(Scikit-Learn kütüphanesi kullanılarak tespit edilen outlier/inlier değer sayıları tablo olarak gösterilmiştir.)

Tabloya bakılarak Support Vector Machine algoritmasında kernel parametresi sigmoid olarak , nu parametresi 0.10 olarak seçildiğinde aykırı değerleri tespit etmede en iyi performansa sahiptir.

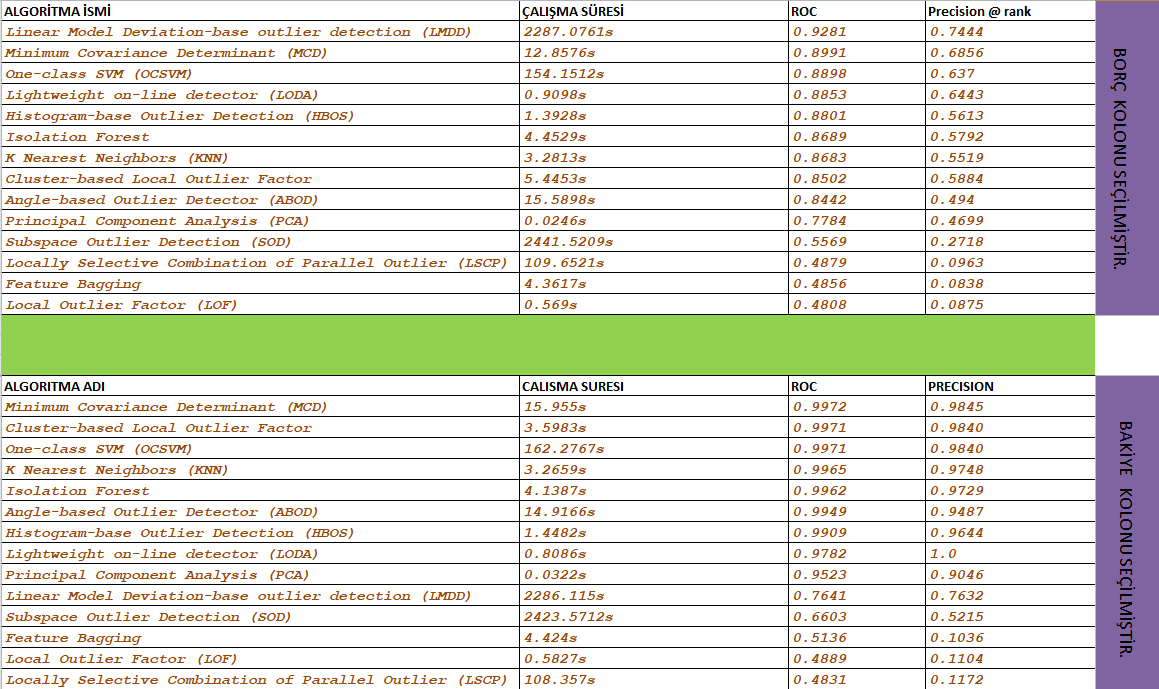
Robustscaler yöntemi kullanılarak yapılan aykırı değer tespitinde Isolation Forest algoritmasıyla 1497 değer aykırı olarak tespit edilmiştir.Minmaxscaler yöntemi kullanılarak yapılan aykırı değer tespitinde Isolation Forest algoritmasıyla 1377 değer aykırı olarak tespit edilmiştir.

SVM algoritması kullanılarak Minmaxscaler yöntemi ile 54611 değer , Robustscaler yöntemi ile 7001 değer tespit edildiği saptanmıştır.Bu durum SVM algoritmasının scaling yöntemine göre hassas olduğunu göstermiştir.

Pyod kütüphanesi kullanılarak , seçilen 2 farklı feature üzerinden,”BORC” ve “BAKIYE” kolonları seçilerek kurulan 2 farklı model birbirleriyle karşılaştırılmıştır.

BORC kolonu ile oluşturulan modelde en uzun süre eğitilen algoritma , LMDD algoritması olup model 2287.0761 saniyede eğitilmiştir. ROC: 09281 Precision Rank : 0.7444 olarak bulunmuştur.BORC kolonu ile oluşturulan modelde en başarılı algoritma ,eğitim süresi en uzun olan LMDD algoritmasıdır.2. olarak MCD algoritması ROC : 0.8991 PRECISION Rank : 0.6856 olarak bulunmuştur.

BAKIYE kolonu ile oluşturulan modelde en uzun süre eğitilen algoritma LMDD algoritması olup model 2286.115 saniyede eğitilmiştir.ROC : 07641 , Precision Rank : 0.7632 olarak bulunmuştur.BAKIYE kolonu ile oluşturulan modelde en başarılı algoritma, MCD algoritması olup ROC : 0.9972 Precision Rank : 0.9845 olarak bulunmuştur.2. olarak One Class Support Vector Machine algoritması olup ROC : 0.9971 Precision Rank : 0.9840 olarak bulunmuştur.



(PYOD kütüphanesi kullanılarak BORC ve BAKIYE kolonlarının değerlendirilmesinin karşılaştırılma tablosu)

2 farklı feature olarak kurulan modeller karşılaştırıldığında BAKIYE kolonu kullanılarak oluşturulan modelin daha üstün performans gösterdiği saptanmıştır.

KNN çalışması , veri setlerinde aykırı değerleri tespit etmede IF'e göre daha avantajlı görünmektedir.

LOF ,diğer denetimsiz öğrenme yöntemlerine kıyasla en kötü algılama performansına sahiptir.

Rapor edilen sonuçlara göre HBOS araştırması, en iyi aykırı tespit tekniği olmuştur.

Pycaret kutuphanesi kullanılarak yapılan çalışma T-SNE ve Umap Dimensionality üzerinden değerlendirilmiştir.T-SNE algoritması, yüksek boyutlu uzayda ve düşük boyutlu uzayda örnek çiftleri arasında bir benzerlik ölçüsü hesaplar. Daha sonra bir maliyet fonksiyonu kullanarak bu iki benzerlik ölçüsünü optimize etmeye çalışır..ABOD,HISTOGRAM algoritması diğer denetimsiz öğrenme yöntemlerine kıyaslaen kötü algılama performansına sahiptir.

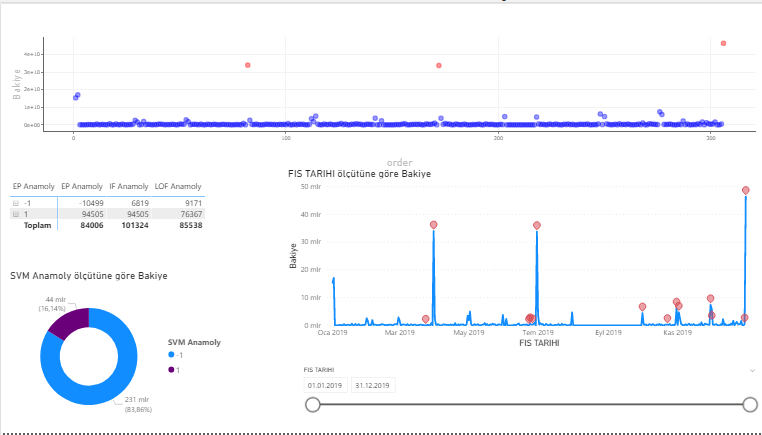
**6.Power BI Implementation**

Pycaret ve scikit-learn kütüphaneleri kullanılarak Power BI programi üzerinden 2 farklı çalışma gerçekleştirilmiştir.Power Bı içerisinde python betiği yazılarak çalıştırılmıştır.

Pycaret kutuphanesi ile yapılan çalışma “BORC” ve “ALACAK” kolonlarındaki anormal değerleri tespit etmeyi amaçlamıştır.

Borc ve Alacak kolonlarının Fıs Tarihine göre indekslenmesiyle oluşan zaman serilerinde anomali tespiti Power Bı Anomaly Detection Tool kullanılarak yapılmıştır.Burada beklenen değer,beklenen en küçük değer ve beklenen en büyük değer,anomaly kolonları çıktı olarak oluşmuştur. Çıkan sonuçlar analiz edilmek üzere .csv olarak kayıt edilmiştir.

SCIKIT-LEARN kütüphanesi bakiye kolonundaki anormal değerleri tespit etmeyi amaçlamıştır.Uygulama üzerinde python betiği çalıştırılmıştır.Çıktılar ile dashboard oluşturulmuştur.Çıkan sonuçlar csv olarak kayıt edilmiştir.Oluşan dashboard ile belirli zaman aralıklarına göre 4 farklı algoritma kullanılarak (SVM,EP,LOF-IF) “BAKIYE” kolonunda anamoli tespiti yapılabilir ve çıkan sonuçlar görselleştirilebilir.



(Sckit-learn kütüphanesi kullanılarak Power Bı programı üzerinde dashboard oluşturulmuştur.)

**7.Sonuç ve Öneriler**

Local outlier algoritmasını kullanılarak yapılan anomali tespitinde en fazla anomalinin Elektrik motoru kategorisinde yapıldığı saptanmıştır.Elliptic envelope yöntemi kullanılarak yapılan anomali tespitinde bakiye kolonundaki anomali değerlerin ortalamasının 263142.929 olduğu normal değerlerin ise 13411.638 olduğu belirlenmiştir.Bu çalışmada autoencoder ve neural networklar kullanılmamıştır.Kullanılırsa daha iyi sonuçlar verebilir.Aykırı değer tespit için model öncesinde verilerinin standardize edilmesi için kullanılan RobustScaler yönteminin Minmaxscaler yöntemine göre aykırı değerleri tespit etmede daha üstün performans göstermektedir.Angle-based outlier detection yöntemi aykırı değer tespiti içerisinde diğer denetimsiz öğrenme algoritmalarına kıyasla en başarısız algoritmadır.Çalışma süresi bakımından en uzun algoritma LMDD algoritmasıdır.Model oluşturulurken kullanılan hiperparametreler mlflow kütüphanesi kullanılarak csv dosyası olarak kaydedilmiştir.

Yapılan literatür araştırmalarına bağlı kalarak , Autoencoder Neural Networks büyük veri kümelerinde anormal yevmiye kayıtlarını tespit etmede çok başarılı olduğu bilinmektedir.Deep learning yöntemleri kullanılarak Anormal yevmiye kaydı tespiti yapılabilir.ABOD ve HISTOGRAM algoritmalarının diğer denetimsiz aykırı değer tespit algoritmalarına oranla daha başarısız performans gösterdiğinden dolayı parametreleri değiştirilebilir.

**8.Tartışma**

Ek olarak, bu çalışmanın sonuçları, karşılaştırılan algoritmalara göre IF'nin en iyisi olduğunu göstermektedir.

özellik seçimi ve hiperparametre ayarı açısından çok tutarlı performans gösterdiğinden bunların birden çok kombinasyonu verildiğinde. HBOS'un yanı sıra ,algılama performansı açısından, bazı durumlarda IF'den biraz daha iyi performans gösterir

Autoencoder Neural Networks büyük veri kümelerinde anormal yevmiye kayıtlarını tespit etmede çok başarılı olduğu bildirilmektedir

KNN'nin algılama için çok uygun bir aykırı değer algılama algoritması olduğu sonucuna varıldı.Hesaplama süresinin önem düzeyine göre HBOS en uygun tekniktir

Belirli bir eşik değerine göre işlemler Aykırı değerin eşikten yüksek olması durumunda bir 'kırmızı bayrak' ile gösterilir.

IF genel olarak en iyi algılamaya sahip algoritmadır. Küresel sentetik dış yevmiye kayıtlarının şu şekilde açıklanması gerekir: çok anormal günlük kayıtları ve hatta oldukça gerçekçi olmayacak kadar anormal. Bir denetlenmemiş aykırı değer tespit algoritmasının bir katma değer olmadığını ileri sürebilir

Özelliklerin seçimi ve ek özelliklerin oluşturulması, oluşturmak için daha fazla araştırma gerektiren bir alandır.