5ML Knime

1.Proje Adı: COVID-19 Günlük Vaka Tahmini

Proje Amacı: COVID-19 Günlük Vaka Tahmini

Bu projenin amacı, COVID-19 pandemisi süresince toplanan günlük vaka ve ölüm sayısı verileri üzerinden, zaman serisi analizine dayalı bir regresyon modeli oluşturarak gelecek günlerdeki COVID-19 vaka sayılarının tahmin edilmesini sağlamaktır.

KNIME Analytics Platformu kullanılarak oluşturulan bu görsel akışta, veri temizleme, zaman bileşenlerinin ayrıştırılması, hareketli ortalama hesaplama gibi veri ön işleme adımlarının ardından, Random Forest regresyon modeli ile öğrenme işlemi gerçekleştirilmiştir. Modelin başarımı ise sayısal değerlendirme metrikleri (RMSE, MAE, R^2) üzerinden analiz edilmiştir.

1. CSV Reader

- Veri setini KNIME'a içeri almak için kullanılır.
- Bu projede Date, Date_YMD, Daily Confirmed, Daily Deceased sütunları olan .csv dosyası okunmuştur.

2. String to Date&Time

- Date veya Date_YMD sütunlarındaki string tarih ifadelerini gerçek tarih veri tipine (DateTime) dönüştürür.
- Zaman serisi analizi yapılabilmesi için gereklidir.

3. Date&Time Part Extractor

- Tarih kolonundan yıl, ay, gün gibi bileşenleri ayırır.
- Örn: "Yıl" değişkeni modelde kullanılabilir hale gelir.

4. Moving Aggregator

- Zaman serisindeki değişkenler için kayan ortalama, toplam, minimum gibi istatistikleri hesaplar.
- Örn: 7 günlük hareketli ortalama, dalgalanmaları yumuşatmak için kullanılabilir.

5. Column Filter

- Analize dahil edilecek veya edilmeyecek kolonları belirler.
- Örn: sadece Daily Confirmed, Daily Deceased, Moving Average gibi sayısal sütunlar bırakılır.

6. Missing Value

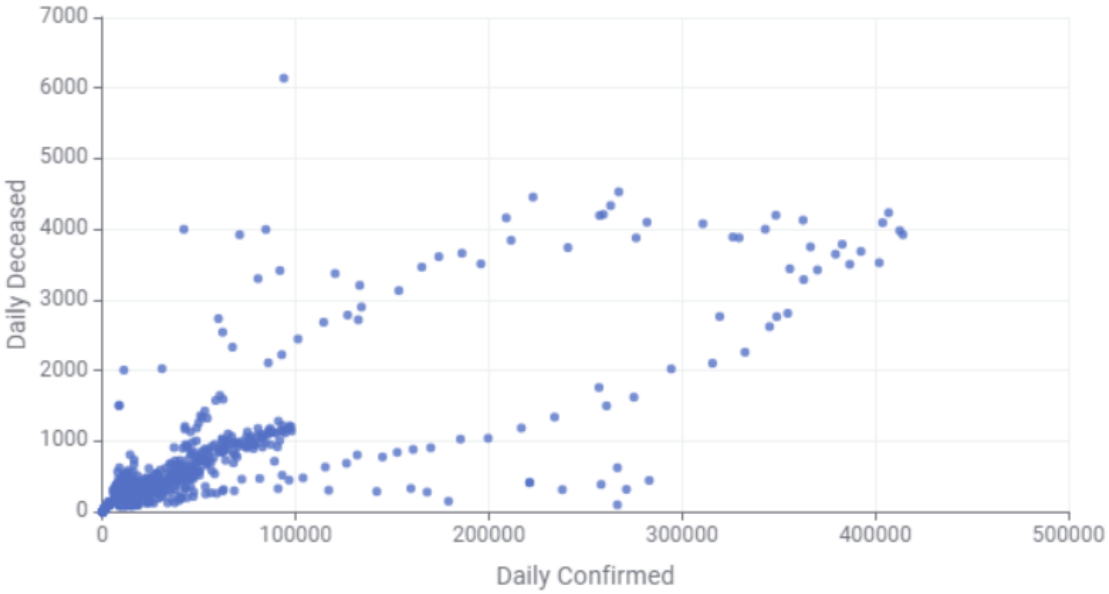
- Eksik verileri doldurmak için kullanılır.
- Bu projede eksik veri yok ama güvenlik için dahil edilmiş olabilir.

7. Extract Table Spec

- Veri setinin yapısını (sütun adları, veri tipleri) çıkarır. Genellikle denetleme ve görselleştirme öncesi kullanılır.

8. Scatter Plot

- Hedef değişken ile diğer değişkenler arasındaki ilişkiyi grafikle inceler.
- Örn: Daily Confirmed vs Moving Average ilişkisini görselleştirir.



9. Partitioning

- Veri setini eğitim (%70) ve test (%30) olmak üzere ikiye böler.
- Modelin hem öğrenmesi hem de test edilmesi için kullanılır.

10. Random Forest Learner (Regression)

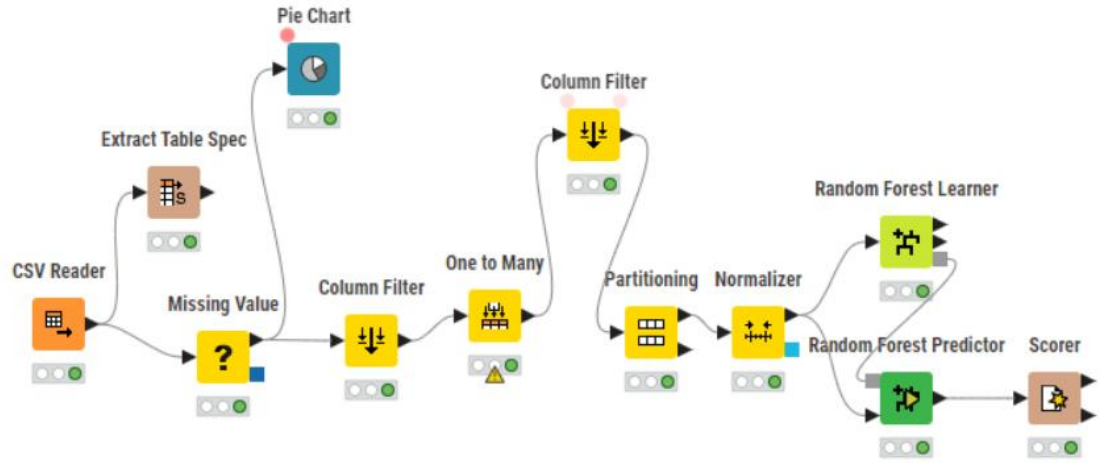
- Regresyon problemi çözmek için rastgele karar ağaçlarından oluşan bir model kurar.
- Bu projede Daily Confirmed gibi bir hedef değişkeni tahmin etmek üzere eğitilmiştir.
- Test verisine model uygulanır ve tahminler elde edilir.
- Örn: Gelecek günlerdeki Daily Confirmed değerleri tahmin edilir.

11. Numeric Scorer

R²:	0,915
Mean absolute error:	93,761
Mean squared error:	75.152,224
Root mean squared error:	274,139
Mean signed difference:	-2,089
Mean absolute percentage error:	NaN
Adjusted R²:	0,915

- Gerçek ve tahmin değerleri karşılaştırarak RMSE, MAE, R^2 gibi regresyon başarı metriklerini hesaplar.
- Modelin başarısı bu node üzerinden yorumlanır.

2.Proje Adı: KNIME ile meme Kanseri Verisi Üzerinden Sınıflandırma Modeli Geliştirme

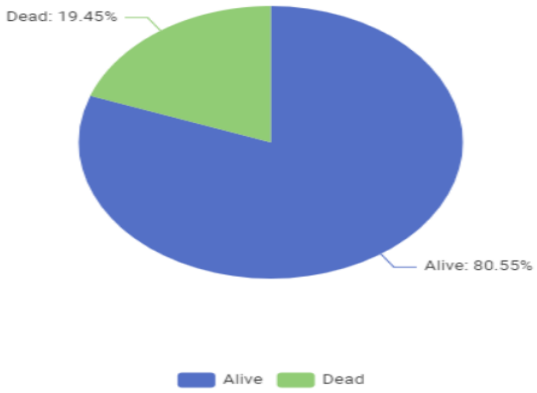


Proje Amacı:

Bu proje kapsamında, öğrencilerin çeşitli özelliklerine (çalışma süresi, uyku alışkanlıkları, sosyal medya kullanımı vb.) göre Final Grade (son not) değişkenini tahmin edebilecek bir sınıflandırma modeli geliştirilmiştir. Süreç, KNIME Analytics Platform üzerinde sadece node'lar kullanılarak görsel akış ile gerçekleştirilmiştir.

1. Veri Yükleme ve İlk Keşif (Exploratory Data Analysis)

- Node'lar:
    - CSV Reader veya File Reader: Veri seti KNIME'a yüklendi.
    - Pie chart ile target gözlemlendi.



2. Eksik Verilerin Doldurulması

- Node: Missing Value
    - Sayısal Değişkenler: "Replace with median" seçeneği ile dolduruldu.
    - Kategorik Değişkenler: "Replace with mode" kullanılarak en sık görülen kategoriyle dolduruldu.

3. Kategorik Değişkenlerin Sayısala Dönüştürülmesi

- Node'lar:
    - One to Many: Kategorik değişkenler dummy (one-hot) formatına dönüştürüldü.
    - Alternatif olarak:
        - Ordinal Encoder: Sıralı kategoriler için.
        - Category to Number: Label encoding yapmak için.

4. Eğitim ve Test Verisine Ayırma

- Node: Partitioning
    - Seçenekler:
        - %70 Eğitim, %30 Test
        - Stratified sampling: Hedef değişkenin sınıf dağılımını korumak için etkinleştirildi.

5. Özellik Ölçeklendirme (Scaling)

- Node: Normalizer
    - Sayısal değişkenler normalize edildi.

6. Model Eğitimi

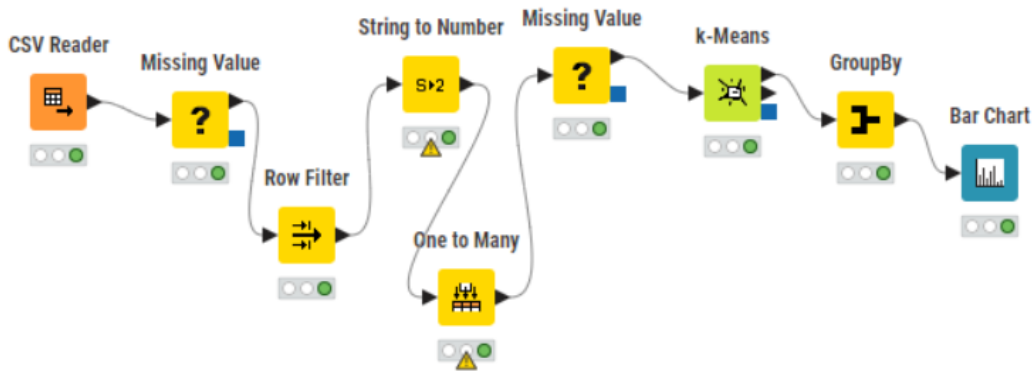
- Kullanılan Node: Random Forest Learner
    - Hedef değişken: Paitent Status
    - Giriş değişkenleri: Sayısallaştırılmış ve temizlenmiş tüm özellikler
    - Parametre ayarları: Tree sayısı, maksimum derinlik vb. node içinde tanımlandı.

7. Model Değerlendirme

Recall ↓ Number (dou...	Precision Number (dou...	Sensitivity Number (dou...	Specificity Number (dou...	F-measure Number (dou...
1	0.995	1	0.976	0.997
0.976	1	0.976	1	0.988

- Node: Scorer
    - Doğruluk (Accuracy)
    - Precision
    - Recall
    - F1 Score gibi sınıflandırma başarı metrikleri hesaplandı.

3.Proje Başlığı: KNIME ile Öğrenci Performans Verisi Üzerinde Kümeleme Analizi



Projenin Amacı:

Bu projenin temel amacı, öğrencilerin başarılarını etkileyen faktörleri analiz ederek benzer özelliklere sahip öğrenci gruplarını (kümeleri) tespit etmektir. Bu sayede eğitim stratejileri geliştirilerek öğrenci başarısı artırılabilir. KNIME Analytics Platformu kullanılarak, görsel veri akışı üzerinden veri temizleme, dönüştürme, kümeleme ve analiz adımları gerçekleştirilmiştir.

1. Veri Yükleme

- Kullanılan Node'lar:
    - File Reader veya CSV Reader: CSV formatındaki veriler KNIME'a yüklendi.
    - Data Explorer / Statistics: Kolon isimleri, veri tipleri, eksik değerler ve temel istatistikler gözlemlendi.
- Amaç:
    - Veriyi analiz öncesi incelemek, veri tipi problemleri ve eksik veri durumlarını belirlemek.

2. Veri Temizliği ve Eksik Değerlerin İşlenmesi

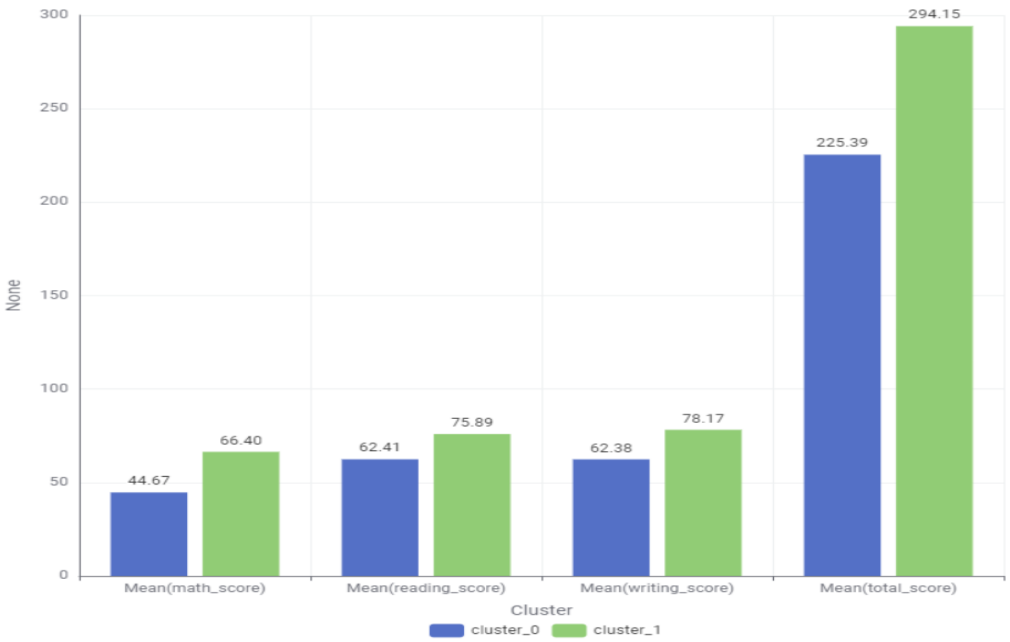
- Kullanılan Node'lar:
    - Missing Value:
        - Sayısal sütunlar için ortalama (mean) ile dolduruldu.
        - Kategorik sütunlar için mod (mode) kullanıldı.
    - Row Filter: ID sütununda (örneğin roll_no) eksik değer varsa, ilgili satırlar çıkarıldı.
    - Column Filter: Analize dahil edilmeyecek kolonlar çıkarıldı.
- Amaç:
    - Eksik verilerin istatistiksel yöntemlerle doldurulması ve analiz kalitesinin artırılması.

3. Tip Dönüşümleri (Veri Dönüştürme)

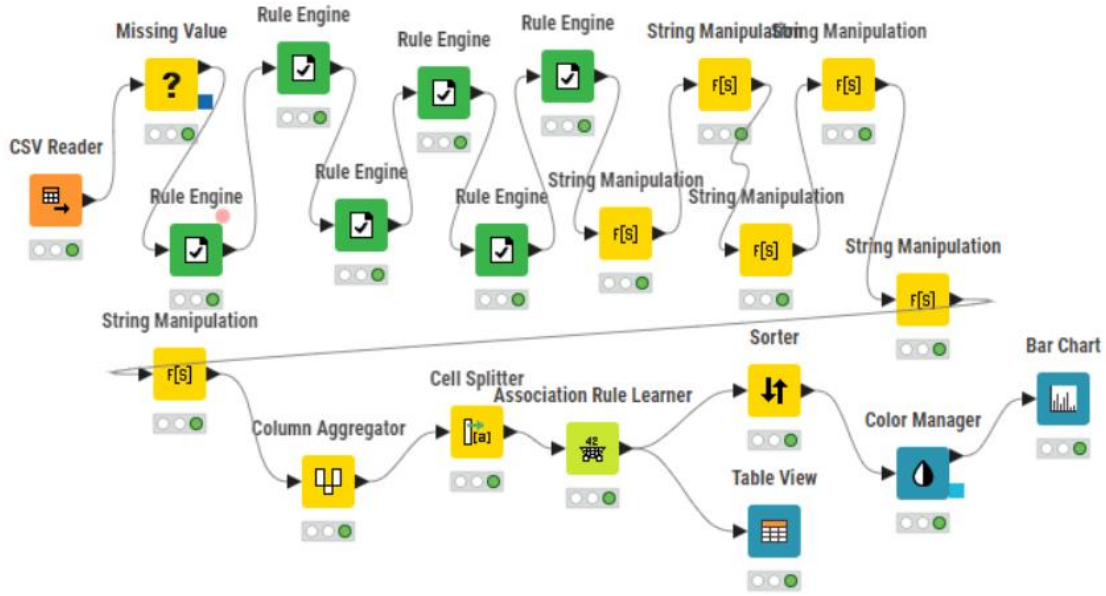
- Kullanılan Node'lar:
    - String to Number: math_score gibi yanlışlıkla string gelen sayısal sütunlar sayıya dönüştürüldü.
    - One to Many (One-Hot Encoding): gender, race_ethnicity, grade gibi kategorik değişkenler sayısallaştırıldı.
- Amaç:
    - Makine öğrenmesi algoritmalarının anlayabileceği şekilde sayısal veri formatı elde etmek.

4. Kümeleme (Clustering)

- Kullanılan Node'lar:
    - K-Means: Öğrenciler belirlenen sayıda kümeye ayrıldı. k değeri elle girilerek farklı senaryolar test edildi.
    - GroupBy: Küme etiketine göre gruplandırma yapılarak her kümedeki öğrencilerin ortalama başarı skorları hesaplandı.
- GroupBy Node Ayarları:
    - Group Column: kmeans_cluster
    - Aggregation Columns:
        - math_score: Mean
        - reading_score: Mean
        - writing_score: Mean
        - total_score:Mean



4.Proje Adı: Öğrenci Performansı Üzerindeki Faktörlerin İncelenmesi ve İlişkilendirme Analizi



Proje Amacı:

Bu projenin amacı, öğrencilerin sınav performanslarını etkileyen faktörler arasındaki ilişkisel örüntüleri keşfetmek ve bu örüntülerden anlamlı birliktelik kurallarını çıkarmaktır. Proje kapsamında Association Rule Mining (İlişkilendirme Kuralı Madenciliği) yöntemi kullanılmıştır.

Veri ön işlemeler sonrası kategorik hale getirilen değişkenlerle öğrenci davranışları, alışkanlıkları ve çevresel faktörler arasında sık rastlanan kurallar elde edilmiştir.

1. Veri Yükleme ve Temel İnceleme

Kullanılan Node'lar:

- CSV Reader: Veri dosyası KNIME'a yüklendi.
- Data Explorer veya Statistics: Kolon isimleri, veri tipleri, eksik değerler ve temel istatistikler incelendi.

2. Veri Temizliği ve Eksik Değerlerin İşlenmesi

Kullanılan Node'lar:

- Missing Value:
    - Sayısal sütunlar için *ortalama (mean)* veya *medyan* ile doldurma.
    - Kategorik sütunlar için *mod (en sık değer)* ile doldurma.
- Rule Engine: Belirli kurallarla boş hücreleri veya özel durumları işaretleme.
- Column Filter: Analize dahil edilmeyecek kolonları dışarıda bırakma.

3. Değişkenlerin Dönüştürülmesi (Ölçekleme & Sınıflama)

Kullanılan Node'lar:

- Rule Engine: Sürekli sayısal değişkenleri kategorik sınıflara ayırmak için.
- String Manipulation: Değer formatlarını değiştirme, birleştirme.
- Normalizer veya Standard Scaler (gerekirse): Model eğitimi öncesi ölçekleme işlemi.

4. Özellikleri Birleştirme ve Hazırlama (Sepet Mantığı)

Kullanılan Node'lar:

- Column Aggregator: Belirli sütunlardaki verileri tek bir kolonda birleştirerek sepet yapısı oluşturmak.
- String Manipulation: Birleştirilmiş verileri düzenlemek.
- Cell Splitter: Bir hücredeki liste halindeki verileri ayrı öğelere ayırmak.

Amaç:

- Association Rule Mining için her öğrenciyi/sepeti hazırlamak.

5. İlişkilendirme Kurallarının Öğrenilmesi

Kullanılan Node'lar:

- Association Rule Learner: Apriori algoritması ile sık görülen örüntüleri ve ilişki kurallarını çıkarmak.
- Sorter: Kuralları support, confidence veya lift değerine göre sıralamak.

Amaç:

- Öğrencilerin özellikleri arasında sık görülen ilişkilendirme kurallarını elde etmek.

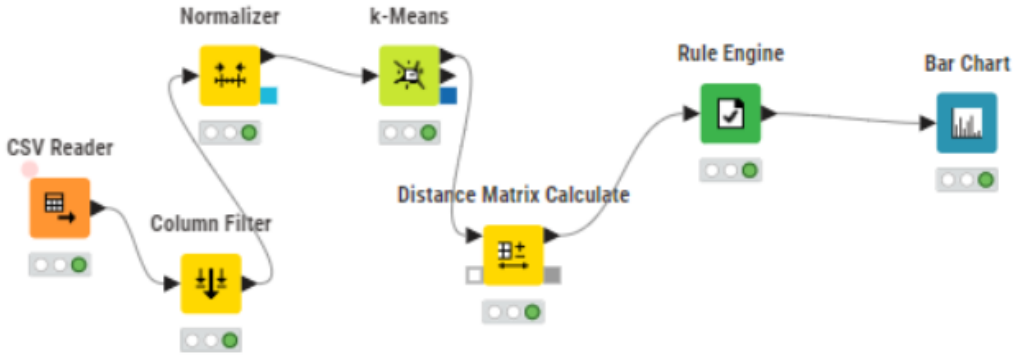
RowID	Support *Number (double)*	Confidence *Number (double)*	Lift *Number (double)*	Consequent *String*	implies *String*	Items *Set*
rule96	0.126	0.429	1.061	Previous_Scores=Medium	<---	[Physical_Activity=Medium]
rule97	0.126	0.311	1.061	Physical_Activity=Medium	<---	[Previous_Scores=Medium]
rule29	0.109	0.278	1.058	Attendance=Low	<---	[Previous_Scores=High]
rule30	0.109	0.413	1.058	Previous_Scores=High	<---	[Attendance=Low]
rule0	0.1	0.524	1.054	Attendance=Medium	<---	[Tutoring_Sessions=Frequent]
rule1	0.1	0.202	1.054	Tutoring_Sessions=Frequent	<---	[Attendance=Medium]
rule8	0.101	0.251	1.043	Attendance=High	<---	[Previous_Scores=Medium]
rule7	0.101	0.421	1.043	Previous_Scores=Medium	<---	[Attendance=High]
rule31	0.109	0.483	1.042	Hours_Studied=High	<---	[Tutoring_Sessions=Few,Previous_Sc

6. Sonuçların Görselleştirilmesi

Kullanılan Node'lar:

- Table View: Üretilen kuralları tablo halinde göstermek.
- Bar Chart: Seçilen kuralları görsel olarak ifade etmek.
- Color Manager: Görselleştirme sırasında renk atamaları yapmak.

5.Proje Adı: Öğrenci Performansı Üzerindeki Anomali Tespiti



Proje Amacı

Bu projenin amacı, öğrenci performans verileri üzerinde anomali tespiti yaparak, olağan dışı davranış veya sonuçları ortaya çıkarmaktır. Öğrencilerin sınav notları, çalışma alışkanlıkları ve demografik özellikler gibi çok boyutlu veriler kullanılarak KMeans kümeleme algoritması ile öğrenciler belirli gruplara ayrılmıştır. Küme merkezlerinden uzak olan, yani kendi grubuna ait genel davranıştan farklılaşan veriler anomali olarak işaretlenmiştir.

Bu sayede, eğitim süreçlerinde dikkate alınması gereken sıra dışı öğrenci performansları, veri odaklı ve sistematik bir şekilde tespit edilerek, eğitim kalitesinin artırılması ve bireysel desteklerin sağlanması hedeflenmiştir.

1. Veri Yükleme ve Hazırlama

CSV Reader

Veri seti, CSV Reader node'u ile KNIME'a aktarılmıştır.
Yalnızca sayısal özelliklerin anomali tespitinde kullanılması gerektiğinden, Column Filter node'u ile sayısal olmayan sütunlar filtrelenmiştir. Bu sayede bazı kolonlar analiz dışı bırakılmıştır.

2. Veri Normalizasyonu

Özelliklerin farklı ölçeklerde olması, kümeleme algoritmasının performansını olumsuz etkileyebileceğinden, Normalizer node'u kullanılmıştır..

3. Kümeleme: K-Means

Normalized veri, K-Means node'una verilerek veri seti belirlenen küme sayısına göre (3 test edilmiştir) kümelenmiştir.

4. Küme Merkezlerine Uzaklıkların Hesaplanması

- Veri noktaları ve ilgili küme merkezi bilgisi, küme numarası üzerinden birleştirilmiştir.

Birleştirilen tabloda, veri noktalarının küme merkezlerine olan uzaklıkları Distance Matrix Calculate node'u ile Öklidyen mesafe olarak hesaplanmıştır.

5. Anomali Belirleme

Hesaplanan uzaklıkların dağılımı Rule Engine node'u kullanılarak değerlendirilmiştir.

- Uzaklıkların belirli bir eşik değerin üzerinde olan veri noktaları "Anomali" olarak etiketlenmiştir.
- Eşik değeri olarak mesafelerin ortalaması ile standart sapmanın belirli bir katsayı (örneğin 2) çarpımının toplamı baz alınmıştır.

#	RowID	Distance *Distance vector*	anomaly_prediction *String*
1	Row0	0 []	Anomaly
2	Row1	1 [0.9272359370796288]	Anomaly
3	Row2	2 [0.9727137731528096, 0.789971288897204]	Normal
4	Row3	3 [1.340200292035928, 1.2408013421234052, 1.2247887319927993]	Normal
5	Row4	4 [1.1549981217114704, 0.8625772346675737, 0.8068039975990337, 1.237(	Normal
6	Row5	5 [1.0681760154581266, 0.9286597370696986, 0.6190138488654644, 0.701-	Normal
7	Row6	6 [0.5658080515049622, 0.7854441404099439, 0.8451886733033384, 1.267'	Normal
8	Row7	7 [1.027259944498998, 1.066280276091094, 1.0219697755913442, 0.95619:	Normal
9	Row8	8 [0.9964569926034722, 0.8928089925361145, 0.757598181486313, 1.1414:	Normal