Veri Seti Üzerinden Dolandırıcılık Tespiti

Giriş

Veri seti üzerinden dolandırıcılık tespiti için veri madenciliği yapmak istiyorum. Seçtiğim veri seti üzerinden çeşitli algoritmalar uygulayarak bilgi elde etmek ve bu sayede daha başarılı biçimde dolandırıcılık tespiti yapmak hedefimdir.

Makalemin Özeti

Bu çalışmada, veri madenciliği yöntemleri kullanılarak dolandırıcılık tespiti üzerine bir araştırma gerçekleştirilmiştir. Farklı veri madenciliği algoritmalarını deneyerek çeşitli çalışmaları inceleyerek, dolandırıcılık tespitine en uygun algoritmalar ve yollar belirlenmiştir. Araştırma; rastgele ormanlar, destek vektör makineleri (SVM), ve k-en yakın komşu (KNN) gibi güçlü algoritmaları değerlendirmiştir.

Çalışma, dolandırıcılık işlemlerini yüksek doğruluk oranlarıyla tespit edebilecek modeller geliştirmeyi ve bu modellerin performansını doğruluk, kesinlik, duyarlılık, F1 skoru (bir modelin kesinlik ve duyarlılık arasında dengeyi ölçen bir metriktir. Dengesiz veri kümelerinde doğruluk değerlendirmesi için önemlidir ve 1'e yaklaştıkça modelin performansı artar) gibi metriklerle değerlendirmeyi amaçlamıştır. Algoritmaların performanslarını karşılaştırarak dolandırıcılık tespitinde kullanılan mevcut veri madenciliği yöntemlerinin doğruluğu ve hızı test edilmiştir. Bu araştırma, dolandırıcılık tespitinde veri madenciliğini kullanarak finansal güvenliği artırmaya yönelik bir araştırma sunmaktadır.

Veri Madenciliği Dersi için İncelenen İlgili Makalelerin Özeti

Adaptive Machine Learning for Credit Card Fraud Detection

Bu tezde geliştirilen dolandırıcılık tespit sistemi, kredi kartı dolandırıcılığını tespit etmek için çeşitli makine öğrenimi algoritmalarını bir araya getirir ve gerçek dünyada kullanılmaya uygun ve yüksek doğruluğa sahip bir çözüm sunmaktadır. Çalışma, sınıf dengesizliği, concept drift ve araştırmacı geri bildirim entegrasyonu gibi veri madenciliği zorluklarını çözmek için HDDT, Rastgele Ormanlar, SVM, Logit Boost, Sinir Ağları ve SMOTE gibi algoritmaları entegre etmektedir. Bu sayede dolandırıcılık tespitinde güvenilir, ölçeklenebilir ve dinamik bir prototip sistem geliştirilmiştir.

Maximum Independent Set Problem İçin Malatya Merkezlilik Algoritması

Bu çalışmada, bir grafın maksimum bağımsız kümesini bulmak için Malatya merkezlilik algoritması önerilmiştir. Maksimum Bağımsız Küme Problemi (MISP), bir grafın düğümleri arasından birbirine bitişik olmayan en fazla sayıda düğüm seçilmesini hedefler ve bu problem NP-zor olarak sınıflandırılır.

Malatya algoritması, maksimum bağımsız küme problemini çözmek için etkili bir yöntem sunar. Algoritmada, önce tüm düğümler için merkezlilik değerleri hesaplanır. Daha sonra, en düşük merkezlilik değerine sahip düğüm seçilerek bağımsız küme setine eklenir. Seçilen düğüm ve komşuları grafikten çıkarılır ve bu işlem tüm düğümler işlenene kadar tekrarlanır. Yapılan analizler, Malatya algoritmasının örnek grafikler üzerinde etkili çözümler sunduğunu ve diğer yöntemlere kıyasla daha verimli olduğunu göstermektedir.

Minimum Vertex Cover Problem İçin Malatya Merkezlilik Algoritması

Bu çalışmada, Minimum Düğüm Kapsama Problemi (MVCP) için Malatya merkezlilik algoritması önerilmiştir. MVCP, bir grafın tüm kenarlarını kapsayacak en az sayıda düğüm seçmeyi hedefler ve NP-zor bir problemdir.

Banka Ödemelerinde Dolandırıcılığın Çizge Madenciliği ve Makine Öğrenimi Algoritmalarıyla Tespiti

Dicle Üniversitesinin hazırladığı "Banka Ödemelerinde Dolandırıcılığın Çizge Madenciliği ve Makine Öğrenimi Algoritmalarıyla Tespiti" makalesi özetle;

Günümüzdeki şirketler önemli verilerini veri tabanlarında saklarlar. Saldırı olduğunda finansal bilgiler hedef alınmaktadır. Bu saldırı türlerinden biri de dolandırıcılık saldırılarıdır. Grafik veri bilimi, dolandırıcılık tespit yöntemlerini daha doğru ve etkili hale getirir. Bu çalışmada İspanya'daki bir banka ödeme bilgi simülasyonundan oluşturulan BankSim veri kümesi kullanılmıştır.

BankSim üzerinde bulunan normal ödemeler ve sahte veriler sınıflandırılarak dolandırıcılık tespiti gerçekleştirilmesi amaçlanmıştır. Sınıflandırma için Python dilinde Rastgele Ormanlar, Destek Vektör Makineleri, XGBoost (XGB), KNN sınıflandırma algoritmaları kullanılmıştır. Performans değerlendirmeleri için K-katlamalı çapraz doğrulama kullanılmıştır. Çizge madenciliği için Neo4j veritabanı kullanılmış ve Neo4j sorgu dili olarak CypherQL kullanılmıştır. Bu dolandırıcılık tespitinin uygulanması sayesinde daha az saldırı işlemleri ve daha güvenli bir akış elde edilmiştir. Çizge madenciliği sırasında PageRank, Community ve derece gibi algoritmalar ile standart makine öğrenimi yöntemleri kullanılarak elde edilen sonuçlar geliştirilmiştir. Bu açıdan çizge madenciliği ve makine öğrenimi algoritmalarının birlikte kullanılmasının diğer yöntemlere kıyasla doğruluk oranlarının daha yüksek olduğu ve daha hızlı sürede hesaplama yapan bir yöntem olduğu ispatlanmıştır.

Denetimli Makine Öğrenmesi Yöntemleri ile Kredi Kartı Sahteciliğini Tahmin Etme: Karşılaştırmalı Analiz

Güner Altan ve Metin Recep Zafer tarafından İstanbul Üniversitesinin İktisat Politikası Araştırmaları Dergisinde yer alan araştırma makalesi özetle;

Dijital platformlar üzerinden yapılan harcamalar günümüzde oldukça yaygınlaşmıştır. Ödeme işlemlerinin bu kadar hızlanması ve artmasından dolayı güvenliği sağlamak zorlaşmıştır. Bankalar, kredi kartı harcamalarının dolandırıcılık olup olmadığını tespit etmekte kilit bir rol üstlenir. Bu çalışmada, bir kamu bankasının 2023 Ocak ayına ait 13.050 kredi kartı işlemi üzerinde yapılan analizle, dolandırıcılığı tespit etmeye yönelik bir model geliştirilmiştir. Python ile yapılan analizde, Lojistik Regresyon, Rastgele Orman, K-En Yakın Komşu, Karar Ağaçları ve Gradyan Güçlendirme algoritmaları kullanılmıştır. Algoritmaların doğruluk oranları %86.4 ile %93.1 arasında değişmiş; kesinlik, duyarlılık, F1 skoru ve ROC-AUC gibi performanslar incelenmiştir. Çalışma, kredi kartı dolandırıcılığını tespitte bu beş algoritmanın iyi birer seçenek olduğunu göstermektedir.

FİNANS SEKTÖRÜNDE DOLANDIRICILIK TESPİTİ ÜZERİNE MELEZ SINIFLANDIRMA VE REGRESYON AĞACI UYGULAMASI

DergiPark ın Yönetim Bilişim Sistemleri Dergisi Cilt:5 Sayı:2 deki bu çalışma, veri madenciliğinin finansal işlemlerde dolandırıcılık tespit etmekte nasıl kullanıldığını ele alıyor. Özellikle, PaySim adlı bir simülatörle oluşturulan sahte bir veri seti üzerinde sınıflandırma yapılmış. Araştırmada, öncelikle klasik CART algoritması kullanılmış, ardından bu algoritma Genetik Algoritma (GA) ile optimize edilerek GA-CART modeli oluşturulmuş. Sonuçlar, GA-CART modelinin klasik CART'a göre %37 daha iyi performans sergilediğini gösteriyor. Bu sayede, finansal illegal işlemlerin tespitinde veri madenciliği tekniklerinin faydalarına dikkat çekilmiştir.

Finansal Tablolarda Hile Riskinin Tespit Edilmesinde Veri Madenciliği Yöntemlerinin Kullanılmasına Yönelik Bir Araştırma

Yaşar Üniversitesi Dergisindeki bu makalede, 2015-2019 yılları arasında Borsa İstanbul'da işlem gören tekstil, giyim eşyası ve deri sektöründeki şirketlerin finansal tablolarındaki hile riskinin veri madenciliği yöntemleriyle tespiti amaçlanmıştır. Çalışmada hileli finansal raporlamayı tespit etmek için 127 finansal tablo incelenmiş ve 12 farklı finansal oran ile veri madenciliği teknikleri kullanılarak analiz yapılmıştır. Bu oranlar arasında brüt kâr/aktif toplamı, çalışma sermayesi/aktif toplamı ve borç/öz sermaye oranı gibi ölçütler bulunur.

Kullanılan veri madenciliği algoritmaları ise Derin Öğrenme, Yapay Sinir Ağı, Karar Ağacı, J48, Rastgele Orman, Destek Vektör Makineleri, K-En Yakın Komşu, AdaBoost, Navie Bayes ve Lojistik Regresyon'dur. Çalışma sonucunda en başarılı yöntemlerin J48 ve Derin Öğrenme olduğu görülmüş, bu iki algoritmanın %84.25 başarı oranıyla en yüksek performansı gösterdiği tespit edilmiştir.

Kaggle - Fraud Transaction Detection

Bu projede, finansal veri seti kullanılarak sahtekar işlemleri tespit etmek için Karar Ağaçları, Rastgele Ormanlar ve KNN veri madenciliği algoritmaları kullanılmıştır. Proje, sahtekar işlemleri tespit etmek için gerekli özellikleri belirlemek, verileri temizlemek ve model oluşturmak gibi adımları içerir.

Karar Ağacı Destekli Hile Tespiti ve Bir Uygulama

Alanya Akademik İncelemeler Dergisindeki bu çalışmada, hileli finansal işlemlerin tespit edilmesi için makine öğrenimi tekniklerinden biri olan karar ağacı algoritması kullanılmıştır. Sertifikalı Hile Denetçileri Birliği'nin (ACFE) hile ağacındaki öne çıkan hileli ödemeler örnek alınarak, özellikle de bankacılık sektöründe hileli işlem riskini azaltmayı amaçlayan bir model geliştirilmiştir. Çalışma, hileli ve normal işlemleri içeren bir yapay veri setiyle gerçekleştirilmiş ve Python kullanılarak uygulamaya geçirilmiştir.

Karar ağacı tekniği, %97.1 doğruluk, %98.4 F1-skor, %98.9 kesinlik ve %98 duyarlılık oranlarıyla oldukça yüksek bir performans sergilemiştir. Finans işlemlerindeki kontrolleri hızlandırıp, şüpheli işlemleri daha çabuk bulmak hedeflenmiştir. Modelin karar ağacı tabanlı yapısı sayesinde yüksek hacimli verilerle çalışabilme, eksik verilere duyarlılığı az olma ve karmaşık veri yapılarında tutarlı sonuçlar verme gibi avantajlara sahiptir.

Çalışmada kullanılan veri madenciliği algoritması olan karar ağacı, veriyi sınıflandırarak hileli işlemlerin tespit edilmesini kolaylaştırır. Yapılan analizlerde, modelin sonuçları incelenmiş ve karar ağacının hileli işlemleri doğru şekilde tahmin edebilme oranının yüksek olduğu görülmüştür.

MAKİNE ÖĞRENİMİ ALGORİTMALARI İLE KREDİ KARTI İŞLEMLERİNDE DOLANDIRICILIK TESPİTİ YÜKSEK LİSANS TEZİ

Bu tezde, kredi kartı işlemlerinde dolandırıcılığı tespit etmek için **Rastgele Orman (Random Forest)**, **Gradyan Artırma Ağaçları (Gradient Boosting Trees)**, **Yapay Sinir Ağları (Artificial Neural Networks)** ve **Karar Ağaçları (Decision Trees)** gibi makine öğrenimi algoritmaları analiz edilmiştir. Amaç, dolandırıcılık riskini öngörmek ve işlemlerin güvenliğini artırmaktır.

Veri seti, gerçek işlemleri yansıtan bir simülasyon modeli ile oluşturulmuş ve algoritmalar şüpheli işlemleri tespit etmeye odaklanmıştır. Karar ağacı tabanlı yöntemlerin daha yüksek doğruluk sağladığı görülmüştür. Çalışma, finansal dolandırıcılığın tespitinde güvenilir ve etkili çözümler sunmaktadır.

Tekdüzen kaynak bulucu yoluyla kimlik avı tespiti için makine öğrenmesi algoritmalarının özellik tabanlı performans karşılaştırması

Bu makalede, makine öğrenimi algoritmaları kullanılarak URL tabanlı kimlik avı (phishing) saldırılarının tespiti incelenmiştir. Amaç, URL adreslerini zararlı veya güvenli olarak sınıflandırarak kimlik avı saldırılarını önlemektir. Destek Vektör Makineleri (SVM), Rastgele Orman, Gaussian Naive Bayes (GNB), Lojistik Regresyon, KNN, Karar Ağaçları, MLP ve XGBoost.algoritmaları kullanılmıştır. Veri seti USOM, Alexa ve PhishTank gibi kaynaklardan 11.935 veri toplanmış, URL uzunluğu, IP adresi içerme ve SSL sertifikası gibi 12 özellik çıkarılmıştır. Sonuç olarak, en yüksek doğruluk %99.8 ile Rastgele Orman algoritmasında elde edilmiştir, GNB algoritması en düşük performansı (%73.4) göstermiştir, Karmaşıklık matrisleri ile doğru ve yanlış sınıflandırmalar analiz edilmiştir.

Çalışma, detaylı özellik çıkarımı ve güncel veri seti kullanımıyla makine öğrenimi yöntemlerinin kimlik avı tespitinde etkili olduğunu göstermiştir.

Veri Seti - Sınıflandırma İlişkisinde Performansa Etki Eden Faktörlerin Değerlendirilmesi

Fırat Üniversitesi Müh. Bil. Dergisindeki bu çalışmada veri madenciliği yöntemlerinden sınıflandırma üzerinde durulmuştur ve sınıflandırmanın başarımına etki eden faktörler detaylı olarak incelenmiştir. Çalışmada k-en yakın komşu , karar ağaçları , rastgele orman , Naive bayes, lojistik regresyon, destek vektör makineleri, gradyan artırma, Adaboost ve yapay sinir ağları kullanılmıştır.

Çalışma, farklı veri setlerinde sınıflandırma algoritmalarının başarısını değerlendirerek test tekniklerinin ve veri seti özelliklerinin performansa etkisini incelemeyi amaçlamaktadır. Literatürde yaygın olarak kullanılan 32 veri seti üzerinde deneyler gerçekleştirilmiş ve kullanılan dokuz farklı sınıflandırma algoritmasıyla başarım kıyaslaması yapılmıştır. Elde edilen bulgular, test yöntemlerinin (Hold-out ve 10 katlı çapraz doğrulama) ve veri seti parametrelerinin sınıflandırma başarımına etkilerini gözler önüne sermiştir. Çalışma, yalnızca doğruluk metriğine bağlı kalmadan F1 ölçütü, Matthews korelasyon katsayısı (MCC) ve alıcı işlem karakteristik eğrisi (AUC) gibi metriklerin önemine dikkat çekerek, doğru sınıflandırıcı modelin seçimi için bir yol haritası sunmuştur.

Kendi düşüncem

Makaleleri inceledim, her biri iyi algoritmalar seçmiş ve bazıları yöntemleri karşılaştırma yapmış. Rastgele Orman (Random Forest), Destek Vektör Makineleri (SVM),ve K-En Yakın Komşu (K-Nearest Neighbors, KNN) veri madenciliği algoritmalarının kendi çalışmam için en uygun olabilecek yöntemler olduğuna karar verdim. Sırası ile çalışmam için seçtiğim yöntemler;

Veri Setim Hakkında

Veri seti, **6362620 satır** ve **11 sütun** içermektedir. Bu veriler, her biri 1 saati temsil eden zaman dilimleri üzerinden toplanmıştır ve 30 günlük bir simülasyona dayanmaktadır. Veriler, her bir işlemin özelliklerini ve dolandırıcılık durumu hakkında önemli bilgileri içermektedir.

Veri Setimdeki Sütunlar:

1. **step**: Bu sütun, her bir işlemi temsil eden zaman dilimini belirtir. Her bir adım (step) 1 saatlik bir süreyi ifade eder. Bu sütun toplamda 744 adımı (30 günlük bir simülasyonu) kapsar.
2. **type**: Bu sütun, işlem türünü belirtir. Aşağıdaki türler bulunur:
    - **CASH-IN**: Hesaba para yatırma işlemi
    - **CASH-OUT**: Hesaptan para çekme işlemi
    - **DEBIT**: Hesaptan yapılan ödeme işlemi
    - **PAYMENT**: Ödemeler
    - **TRANSFER**: Para transferleri
3. **amount**: Bu sütun, işlemin yerel para birimindeki tutarını gösterir.
4. **nameOrig**: Bu sütun, işlemi başlatan müşteri veya hesap sahibinin adını belirtir.
5. **oldbalanceOrg**: İşlemden önce, işlem başlatan (sender) hesabın bakiyesi.
6. **newbalanceOrig**: İşlemden sonra, işlem başlatan (sender) hesabın bakiyesi.
7. **nameDest**: Bu sütun, işlemde parayı alan (receiver) müşteri veya hesabın adını belirtir. Ayrıca, bu sütunda "M" ile başlayan müşteri isimleri, ticaret yapan müşteri hesaplarıdır ve bunlara ait bilgiler bulunmamaktadır.
8. **oldbalanceDest**: Alıcı hesabın işlem öncesi bakiyesi. Bu sütun "M" ile başlayan alıcı hesapları için mevcut değildir.
9. **newbalanceDest**: Alıcı hesabın işlem sonrası bakiyesi. "M" ile başlayan hesaplar için bu bilgi de bulunmamaktadır.
10. **isFraud**: Bu sütun, işlemde dolandırıcılık olup olmadığını gösteren ikili bir değişkendir. 1, dolandırıcılık olduğu anlamına gelirken 0, normal bir işlem olduğunu belirtir.
11. **isFlaggedFraud**: Bu sütun, bir işlemde 200.000 veya daha fazla para transferi yapılması durumunda "flag" (işaret) durumunun aktive olduğunu belirtir. Eğer bu değer 1 ise, işlem şüpheli ve potansiyel olarak yasa dışıdır.

Sırasıyla Veri Seti Üzerinde Kullandığım Algoritmalar ve Yapılan İşlemler

Model performansı;

- **Doğruluk (Accuracy)**: Tüm tahminler içinde doğru sınıflandırma oranıdır. Ancak dengesiz veri dağılımlarında yanıltıcı olabilir.
- **Kesinlik (Precision)**: Dolandırıcılık tahminlerinin doğruluğunu gösterir. Yanlış alarmları (False Positive) azaltmada önemlidir.
- **Duyarlılık (Recall)**: Gerçek dolandırıcılık durumlarının tespit başarısını ölçer. Gerçek pozitiflerin (True Positive) kaçırılmaması açısından kritiktir.
- **F1 Skoru**: Kesinlik ve duyarlılık arasındaki dengeyi ölçer.

$$\boldsymbol{F1\ Skoru = 2\ \times (Kesinlik\ \times Duyarlılık)/(Kesinlik + Duyarlılık)}$$

Bu denklem ile F1 Skoru ölçülür.

- **Karışıklık Matrisi (Confusion Matrix)**: Modelin tahminlerini ve gerçek değerleri karşılaştırarak doğru (TP, TN) ve yanlış (FP, FN) sınıflandırmaları gösterir:
    - **TP (True Positive)**: Doğru olarak dolandırıcılık olarak tahmin edilenler.
    - **FP (False Positive)**: Yanlış dolandırıcılık olarak tahmin edilenler.
    - **TN (True Negative)**: Doğru olarak dolandırıcılık değil diye tahmin edilenler.
    - **FN (False Negative)**: Yanlış dolandırıcılık değil diye tahmin edilenler.

İle değerlendirilmiştir.

Rastgele Orman Algoritması

Nedir?

Rastgele Ormanlar (Random Forest), bir dizi bağımsız karar ağacından (decision trees) oluşan ansamble bir öğrenme algoritmasıdır. Her bir karar ağacı, verilerin bir alt kümesi üzerinde eğitim alır ve bu ağaçlardan elde edilen tahminlerin çoğunluk oyu veya ortalama değeri kullanılarak nihai tahmin yapılır. Bu yöntem, özellikle sınıflandırma ve regresyon problemleri için etkili bir çözüm sunar.

Nasıl Çalışır?

Rastgele Ormanlar algoritması, aşağıdaki adımları izleyerek çalışır:

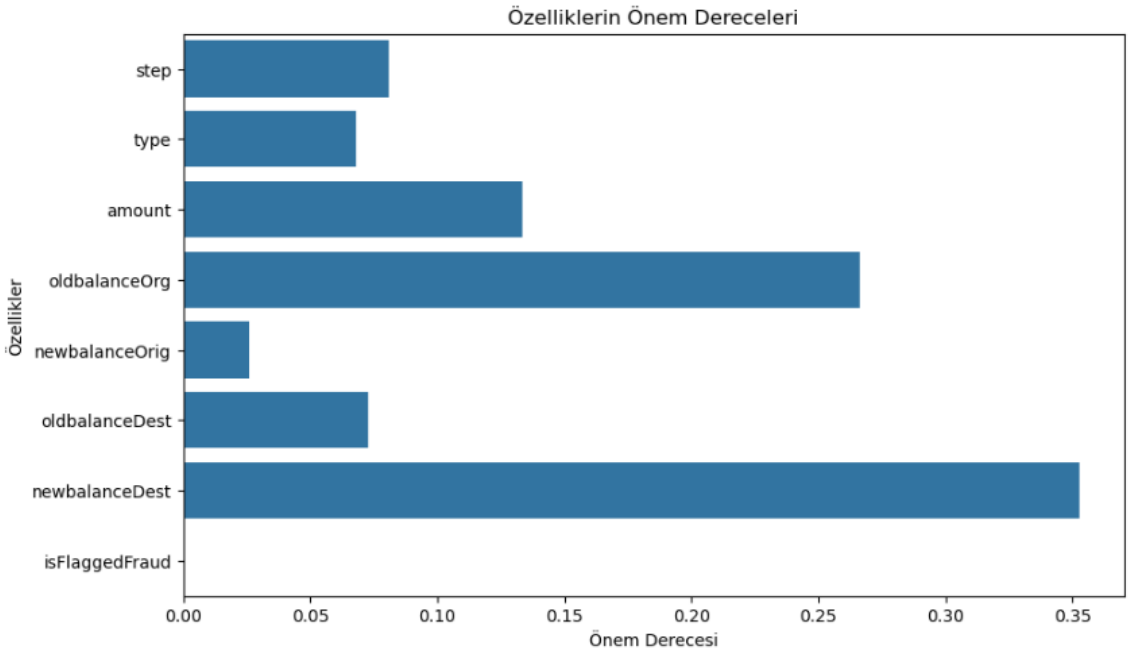
1. **Veri Alt Kümesi Seçimi (Bootstrap)**: Veri setinden rastgele örnekler seçilir. Bu örnekler, her bir karar ağacını eğitmek için kullanılır. Bu işlem, **bootstrap örneklemesi** olarak bilinir.
2. **Karar Ağaçları Oluşturma**: Her ağaç, seçilen alt küme üzerinde eğitim alır. Her ağaçta her düğümde, veri setindeki özelliklerden rastgele bir alt küme seçilir ve en iyi özelliği belirlemek için bu alt küme kullanılır.
3. **Tahmin Yapma**: Yeni bir örnekle karşılaşıldığında, her ağaç tahmin yapar ve bu tahminlerin çoğunluk oyuyla nihai tahmin yapılır (sınıflandırma için) ya da ağaçlardan gelen tahminlerin ortalaması alınır (regresyon için).

Teorik Formül:

1. **Bootstrap Örneklemesi:** Veri setinden $x_1, x_2, \ldots, x_n$ örnekleri rastgele seçilir.
2. **Ağaçlarda Karar Verme:** Her bir düğümde, veri kümesindeki rastgele seçilmiş özellik kümesi üzerinden karar verilir.
3. **Ortalama Tahmin:** Her bir ağaç $f_i(x)$ tahmini yapar ve tahminler birleştirilir:

$$f_{\text{final}}(x) = \frac{1}{T}\sum_{i=1}^{T} f_i(x)$$

Burada $\mathbf{T}$ toplam ağaç sayısını ifade eder.



Yukarıdaki grafik Rastgele Ormanlar yöntemini uygularken ağaç oluşunda hangi sütunun ne kadar önemli olduğunu önem derecesi olarak belirterek gösteren grafiktir.

Destek Vektör Makineleri (SVM)

Nedir?

Destek Vektör Makineleri (SVM), denetimli öğrenme algoritmalarından biridir ve genellikle sınıflandırma problemleri için kullanılır. SVM, verileri en iyi şekilde ayıran bir hiper düzlem (hyperplane) bulmayı amaçlar. Bu hiper düzlem, iki sınıf arasındaki sınırı tanımlar ve doğrusal olmayan sınıflandırmalar için de kernel fonksiyonları kullanılarak daha karmaşık sınırlar oluşturulabilir.

Nasıl Çalışır?

1. **Doğrusal Sınıflandırma**:
   - SVM, verileri iki sınıfa ayıran doğrusal bir hiper düzlem arar. Hedef, bu düzlemi veriye en iyi şekilde yerleştirmek ve iki sınıf arasındaki mesafeyi maksimize etmektir.
   - En iyi sınır, sınıfların arasındaki mesafeyi (margin) maksimize eden düzlemdir.
2. **Kernel Fonksiyonları**:
   - Veriler doğrusal olarak ayıramıyorsa, kernel fonksiyonları kullanılarak veriler daha yüksek boyutlu bir uzaya taşınır ve burada doğrusal ayırma yapılır.
   - Yaygın kernel fonksiyonları arasında **lineer**, **polinomial**, **radial basis function (RBF)** ve **sigmoid kernel** bulunur.
3. **Destek Vektörleri**:
   - En önemli veri noktaları, doğru sınıflandırmanın sağlanmasında en etkili olan verilerdir. Bu veri noktalarına **destek vektörleri** denir.
   - Bu vektörler, hiper düzlemin etrafında "destek" sağlar.

Teorik Formül:

Veri seti $\{(x_1, y_1), (x_2, y_2), \ldots, (x_n, y_n)\}$ olsun. Burada x_i veriyi ve y_i ise sınıf etiketini temsil eder.

Hedef, şu doğrusal sınırı bulmaktır:

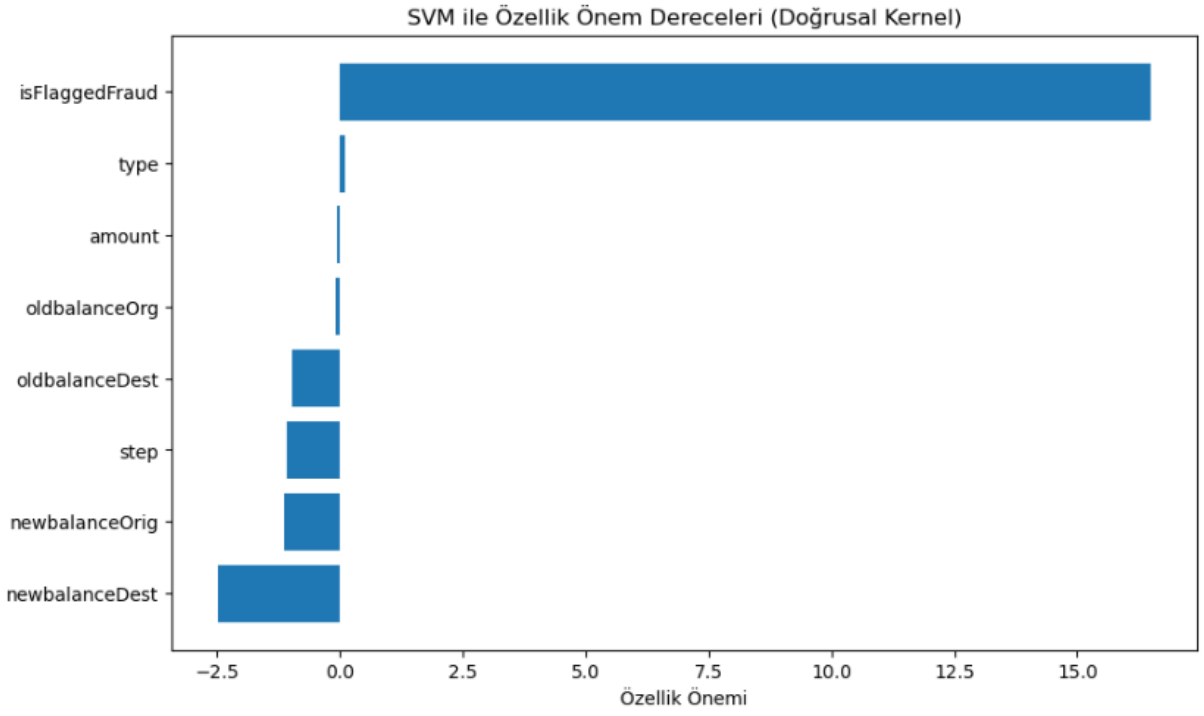
$$\text{ɯ} \cdot x + b = 0$$

Burada ɯ normal vektörü ve b bias terimidir. Sınıflandırma, y_i etiketine göre yapılır:

- $y_i = 1$ için, $\text{ɯ} \cdot x_i + b \geq 1$
- $y_i = -1$ için, $\text{ɯ} \cdot x_i + b \leq -1$

SVM, bu sınıfları ayıran en büyük marjini bulmayı hedefler. Eğer veriler doğrusal olarak ayrılabilir değilse, kernel fonksiyonları kullanılarak veri yüksek boyutlu bir uzaya taşınır.

Fraud Veri Seti ile SVM Uygulaması



Yandaki grafik SVM yöntemini uygularken Destek vektörünü oluştururken hangi sütunun ne kadar önemli olduğunu önem derecesi olarak belirterek gösteren grafiktir.

K-En Yakın Komşu (K-Nearest Neighbors, KNN)

Nedir?

K-En Yakın Komşu (KNN), denetimli öğrenme kategorisinde yer alan basit ve etkili bir makine öğrenimi algoritmasıdır. Hem sınıflandırma hem de regresyon problemleri için kullanılabilir. KNN, yeni bir veri noktasının sınıfını tahmin etmek için, bu noktaya en yakın kkk veri noktasını (komşularını) analiz eder ve çoğunluğa göre karar verir.

Nasıl Çalışır?

1. Yeni bir veri noktası için en yakın **k** komşu seçilir.
2. Komşuluk, genellikle **Öklid Uzaklığı** veya **Manhattan Uzaklığı** gibi bir mesafe metriğiyle hesaplanır.
3. Sınıflandırma için: Komşular arasında en sık görülen sınıf tahmin edilir (çoğunluk oylaması).
   Regresyon için: Komşuların ortalama değeri kullanılır.
4. **k** parametresi, algoritmanın esnekliğini ve doğruluğunu etkiler.
   - Çok küçük **k**: Model gürültüye duyarlı olabilir.
   - Çok büyük **k**: Model fazla genelleştirilebilir.

Teorik Formül

1. Mesafe Fonksiyonu:

KNN algoritması, veriler arasındaki mesafeyi ölçmek için genellikle **Öklid Mesafesi** kullanır. Öklid mesafesi formülü:

$$d(x, y) = \sqrt{\sum_{i=1}^{n} (x_i - y_i)^2}$$

Burada:

- x ve y: Karşılaştırılan iki veri noktası,
- n: Özelliklerin (boyutların) sayısı,
- x_i, y_i: i-inci özelliğin değerleri.

Diğer mesafe metrikleri (Manhattan, Minkowski vb.) de kullanılabilir.

---

2. K Komşularının Belirlenmesi:

Her bir test noktası x_{test} için:

- Tüm eğitim verisi noktalarına olan mesafeler hesaplanır.
- En kısa mesafeye sahip k komşu seçilir.

---

3. Sınıflandırma Kararı:

Eğer problem **sınıflandırma** ise:

- k komşuların sınıf etiketlerinin çoğunluk oyu alınır. Sınıf etiketi şu şekilde belirlenir:

$$y_{öngörülen} = argmax_c \sum_{i=1}^{k} \delta(y_i, c)$$

Burada:

- y_i : i-inci komşunun sınıf etiketi
- c: Bir sınıf
- $\delta(y_i, c)$: Eğer $y_i = c$ ise 1, aksi takdirde 0.

---

4. Regresyon Kararı:

Eğer problem **regresyon** ise:

- k komşuların hedef değerlerinin ortalaması alınır:

$$y_{öngörülen} = \frac{1}{k}\sum_{i=1}^{k} y_i$$

Burada y_i, i-inci komşunun hedef değeridir.

---

5. K Değeri Seçimi:

- k, genellikle küçük tek sayılar (örneğin 3, 5, 7) olarak seçilir.
- **Dikkat:** Çok küçük k değerleri aşırı öğrenmeye (overfitting), çok büyük k değerleri ise yetersiz öğrenmeye (underfitting) neden olabilir.

Bu formüller, KNN algoritmasının temel teorik temelini oluşturur. Hem sınıflandırma hem de regresyon problemlerinde uygulanabilirliği açısından oldukça esnektir.

Malatya Merkezlilik

Malatya merkezlilik, bir grafın düğümleri arasındaki ilişkilerin etkinliğini ölçen bir algoritmadır. Bu algoritma, graf teorisindeki Minimum Vertex Cover (MVCP) ve Maximum Independent Set (MISP) gibi NP-zor problemleri çözmek için optimize edilmiştir.

Malatya Merkezlilik değeri $\psi(V)$, bir düğümün derecesini, komşu düğümlerinin derecelerine göre normalize ederek hesaplar. Bir düğümün merkezlilik değeri şu formülle ifade edilir:

$$\psi(V_i) = \sum_{V_j \in N(V_i)} \frac{d(V_i)}{d(V_j)}$$

Burada:

- V_i: Grafın bir düğümü.
- $N(V_i)$: V_i düğümünün komşuları.
- $d(V_i)$: V_i düğümünün derecesi (komşu düğüm sayısı).

MVCP:
En yüksek merkezlilik değerine sahip düğümler, grafın tüm kenarlarını kapsayacak şekilde seçilir.

MISP:
En düşük merkezlilik değerine sahip düğümler, birbirine bitişik olmayacak şekilde seçilir.

Örnek Veri Seti ve Çözüm

Düğüm	Komşuluk Bilgileri	Merkezlilik Hesaplaması ψ(v)
v_1	$\{v_2, v_3, v_4\}$	$\frac{3}{2}+\frac{3}{3}+\frac{3}{1}=5.5$
v_2	$\{v_1, v_5\}$	$\frac{2}{3}+\frac{2}{2}=1.\overline{6}$
v_3	$\{v_1, v_5\}$	$\frac{3}{3}+\frac{3}{2}=2.5$
v_4	$\{v_1\}$	$\frac{1}{3}=0.\overline{3}$
v_5	$\{v_2, v_3\}$	$\frac{2}{2}+\frac{2}{3}=1.\overline{6}$

- **MIS için**:
  - Minimum merkezlilik: $v_4(0.\overline{3})$ seçilir ve grafdan kaldırılır.
  - Grafı güncelleyin ve tekrarlayın.
- **MVC için**:
  - Maksimum merkezlilik: $v_1(5.5)$, seçilir ve grafdan kaldırılır.
  - Grafı güncelleyin ve tekrarlayın.

Malatya Merkezlilik Hesaplama Kod Örneği

Minimum Düğüm Kapsama (MVCP) kümesini belirleme kodu

```
def find_min_vertex_cover(graph):
    """Minimum Düğüm Kapsama (MVCP) kümesini belirler."""
    vertex_cover = set()
    while graph.number_of_edges() > 0:
        centrality_values = malatya_centrality(graph)
        max_node = max(centrality_values, key=centrality_values.get)
        vertex_cover.add(max_node)
        graph.remove_node(max_node)  # Düğümü ve bağlantılarını kaldır
    return vertex_cover
```

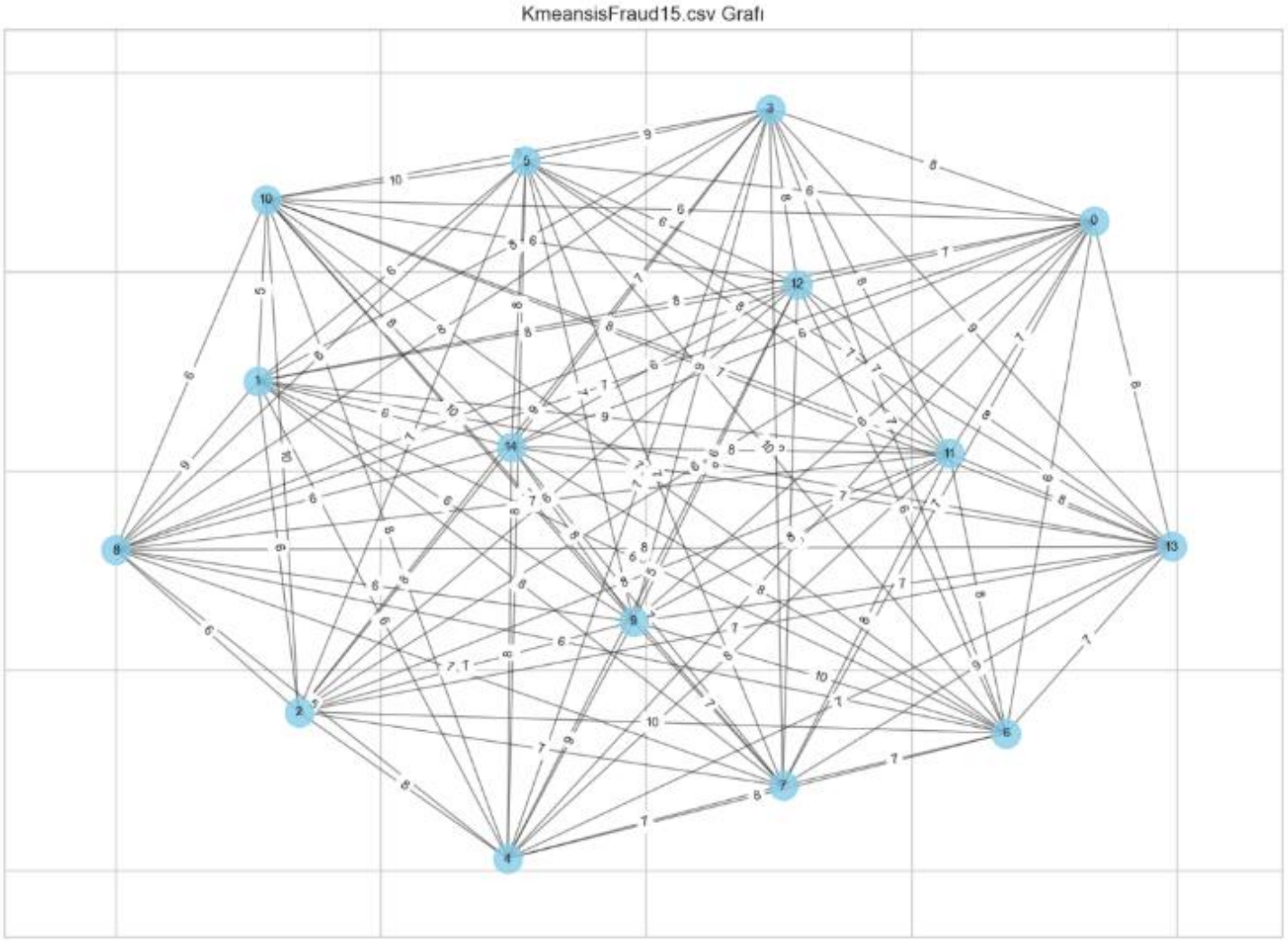
Maksimum Bağımsız Küme (MISP) kümesini belirleme kodu

```
def find_max_independent_set(graph):
    """Maksimum Bağımsız Küme (MISP) kümesini belirler."""
    independent_set = set()
    while graph.number_of_edges() > 0:
        centrality_values = malatya_centrality(graph)
        min_node = min(centrality_values, key=centrality_values.get)
        independent_set.add(min_node)
        neighbors = list(graph.neighbors(min_node))
        graph.remove_node(min_node)  # Düğümü kaldır
        graph.remove_nodes_from(neighbors)  # Komşularını kaldır
    return independent_set
```

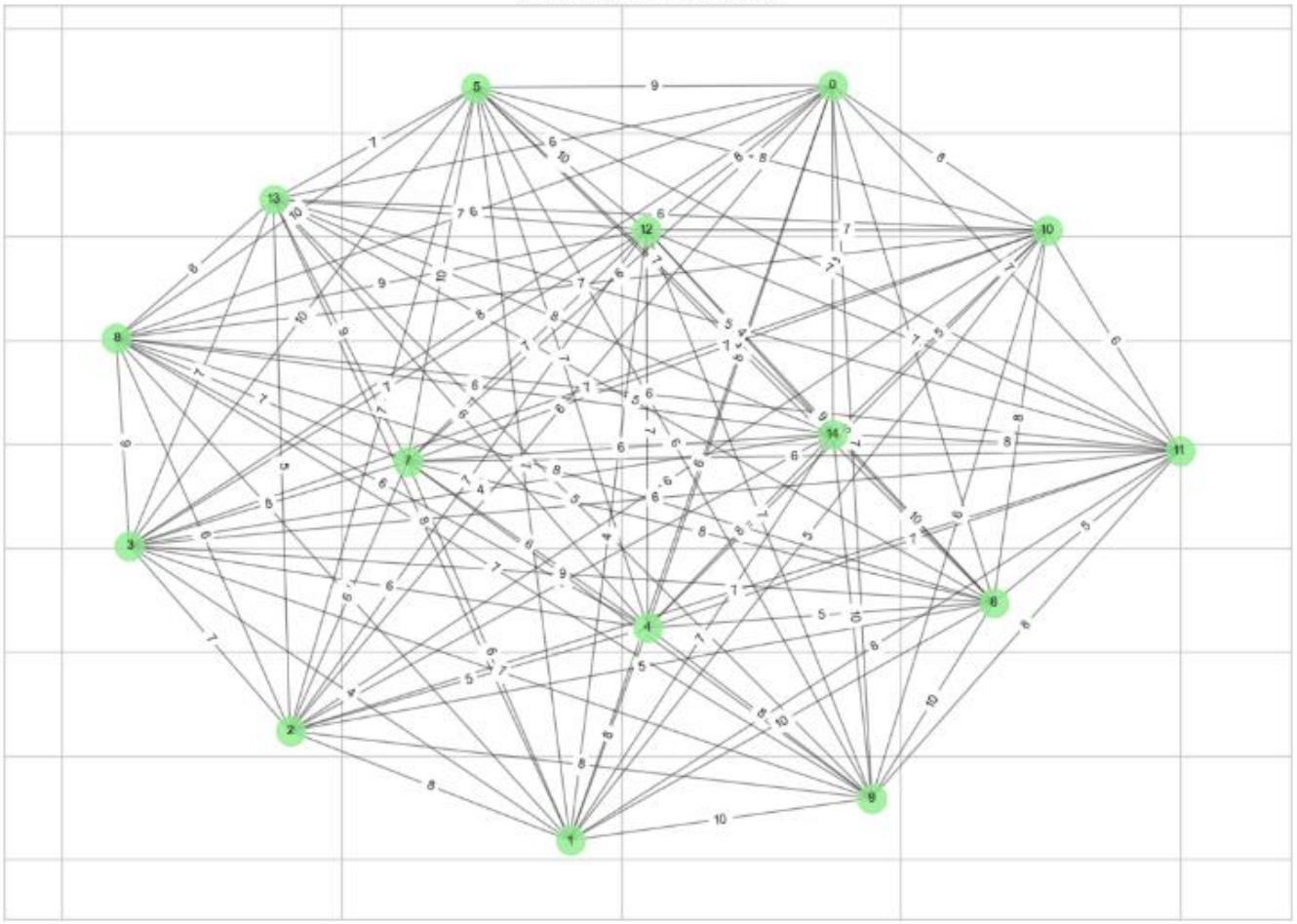
Malatya Merkezlilik Hesaplama kodu

```
def malatya_centrality(graph):
    """Graf düğümlerinin Malatya merkezlilik değerlerini hesaplar."""
    centrality_values = {}
    for node in graph.nodes:
        neighbors = list(graph.neighbors(node))
        degree_node = graph.degree[node]
        centrality = sum(degree_node / graph.degree[neighbor] for neighbor in neighbors)
        centrality_values[node] = centrality
    return centrality_values
```

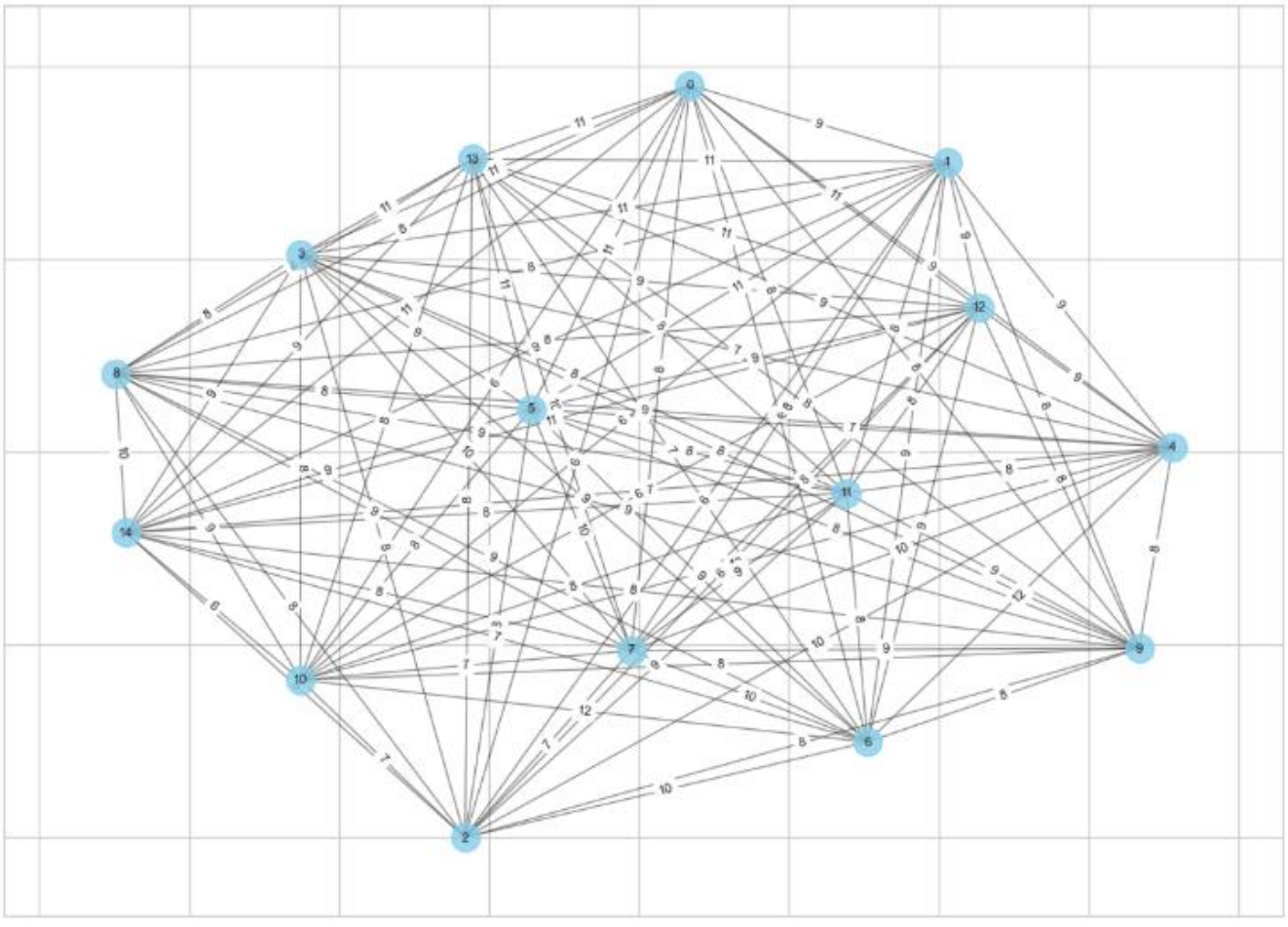
Yöntemimde sırasıyla

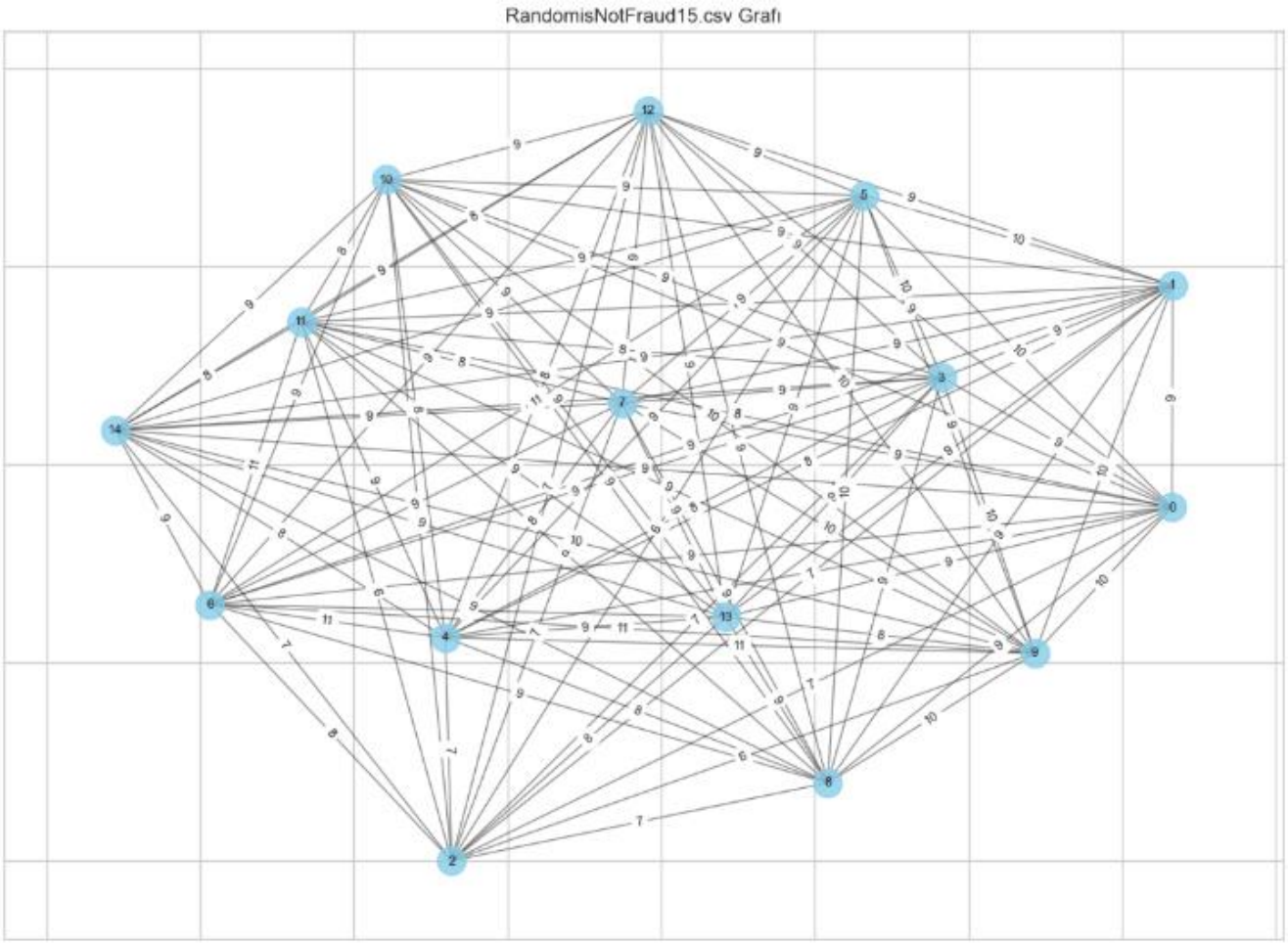


KmeanisNotFraud15.csv Grafı



RandomisFraud15.csv Grafı





Grafları oluştu ve Yöntemde, finansal dolandırıcılık (fraud detection) problemini çözmek için bir makine öğrenmesi modeli geliştirmek amacıyla bir dizi işlem yapmaktadır. İlk olarak, Malatya Merkezlik (Malatya Centrality) hesaplama fonksiyonu, bir grafikteki her bir düğümün merkezliğini hesaplar. Bu merkezlik, düğümün komşularının derecelerinin toplamına oranlanarak belirlenir ve düğümün diğer düğümlerle olan bağlantı gücünü temsil eder. Daha sonra, test verisini kullanarak dolandırıcılık olup olmadığını tahmin eden bir fonksiyon tanımlanır. Bu fonksiyon, test verisinin her bir satırını bir düğüm olarak grafiğe ekler, komşularla olan benzerliklere göre kenarlar oluşturur ve merkezlik hesaplayarak her iki grafikteki merkezlik değerlerini karşılaştırır. Eğer fraud grafiği daha yüksek merkezlik değerine sahipse, veri dolandırıcılık olarak etiketlenir, aksi takdirde dolandırıcılık olmadığına karar verilir. Son olarak, modelin doğruluğu, kesinliği, duyarlılığı, F1 skoru gibi metriklerle performansı değerlendirilir ve karışıklık matrisi ile detaylı dağılım yazdırılır. Bu yaklaşım, grafik tabanlı bir yöntemle dolandırıcılık tespiti yapılmasına olanak tanır ve sonuçlar detaylı bir şekilde raporlanır. Train veri setindeki verilerimi random olarak da kmeans ile de ayrı ayrı seçtim ve yöntemin ortalamasını alıp karşılaştırma tablomda verdim.

Karşılaştırma

Yöntem	Doğruluk	Kesinlik	Duyarlılık	F1 Skoru
Rastgele Ormanlar	0.9996	0.9939	0.7111	0.8291
Destek Vektör Makineleri (SVM)	0.9992	0.9548	0.3784	0.5420
K-En Yakın Komşu (KNN)	0.9989	0.7500	0.2056	0.3227
Malatya Merkezlilik	0.9004	0.7483	0.6459	0.6942

1. **Rastgele Ormanlar (Random Forests):**

    - Genel olarak en başarılı yöntemlerden biri. F1 skoru (%82.91) oldukça yüksek, kesinlik (%99.39) ve doğruluk (%99.96) da çok iyi seviyede. Ancak duyarlılığı (%71.11) biraz daha artırılabilir.

2. **Destek Vektör Makineleri (SVM):**

    - Kesinlik oranı (%95.48) fena değil ama duyarlılık (%37.84) ve F1 skoru (%54.20) düşük kaldı. Bu, modelin sınıf dengesizliğiyle iyi başa çıkamadığını gösteriyor.

3. **K-En Yakın Komşu (KNN):**

    - Bu yöntem veri setiniz için pek uygun değil gibi görünüyor. Kesinlik (%75.00) düşük, duyarlılık (%20.56) çok düşük. F1 skoru da (%32.27) vasat.

4. **Malatya Merkezlilik:**

    - Bu yöntem, model performansı açısından ilginç bir durum sergiliyor. Duyarlılık (%64.59) düşük olsa da, kesinlik (%74.83) ve F1 skoru (%69.42) daha dengeli. Model, özellikle duyarlılığın kritik olduğu durumlar için uygun olabilir, ancak kesinlik ve yanlış pozitifleri (False Positives) azaltmak da önemli. Veri setim için uygun bir seçenek olsa da, performansın sınırlı olduğunu unutmamalısınız.

Sonuç Değerlendirme:

- Rastgele Ormanlar kullanmak mantıklı görünüyor, çünkü yüksek doğruluk ve kesinlik sağlarken, duyarlılık biraz düşük olsa da dengeyi iyi kuruyor. Verilerdeki dengesizliği düzeltmek için SMOTE gibi yöntemlerle "Fraud" sınıfını artırabilirsiniz. Diğer modellerdeki düşük performansı iyileştirmek için hiperparametre optimizasyonu yapabilirsiniz. Genel olarak, model seçimi, ihtiyaçlarınıza göre duyarlılık mı yoksa kesinlik mi öncelikli olduğuna bağlıdır. Ancak, Rastgele Ormanlar yöntemi, hem yüksek doğruluğu hem de dengeli performansı ile veri setim için en uygun seçenek gibi görünüyor.

Kaynakça

1. Adaptive Machine Learning for Credit Card Fraud Detection
   Université Libre de Bruxelles, Bilgisayar Bilimi Makine Öğrenimi Grubu. Uyarlanabilir makine öğrenimi algoritmalarını kullanarak dolandırıcılık tespitine yönelik zorlukları ele alan bir tez.
2. Banka Ödemelerinde Dolandırıcılığın Çizge Madenciliği ve Makine Öğrenimi Algoritmalarıyla Tespiti
   Dicle Üniversitesi, BankSim veri seti üzerinde dolandırıcılık tespiti için grafik veri bilimi ve makine öğrenimi yöntemlerini inceleyen çalışma.
3. Denetimli Makine Öğrenmesi Yöntemleri ile Kredi Kartı Sahteciliğini Tahmin Etme: Karşılaştırmalı Analiz
   Altan G. ve Zafer M.R., İstanbul Üniversitesi. Kredi kartı dolandırıcılığı tespiti üzerine algoritmaların karşılaştırmasını içeren araştırma.
4. Finans Sektöründe Dolandırıcılık Tespiti Üzerine Melez Sınıflandırma ve Regresyon Ağacı Uygulaması
   Yönetim Bilişim Sistemleri Dergisi, Genetik Algoritmalar ile optimize edilmiş sınıflandırma modelleri.
5. Finansal Tablolarda Hile Riskinin Tespit Edilmesinde Veri Madenciliği Yöntemlerinin Kullanılması
   Yaşar Üniversitesi, 2015-2019 yıllarında hileli finansal raporlamayı inceleyen veri madenciliği çalışması.
6. Kaggle - Fraud Transaction Detection
   Karar Ağaçları, Rastgele Ormanlar ve KNN algoritmalarını kullanarak finansal dolandırıcılık tespit çalışması.
7. Karar Ağacı Destekli Hile Tespiti ve Bir Uygulama
   Alanya Akademik İncelemeler Dergisi, karar ağacı tabanlı finansal hile tespitine yönelik model geliştirme.
8. Makine Öğrenimi Algoritmaları ile Kredi Kartı İşlemlerinde Dolandırıcılık Tespiti
   Hitit Üniversitesi, kredi kartı işlemlerini analiz eden yüksek lisans tezi.
9. Tekdüzen Kaynak Bulucu Yoluyla Kimlik Avı Tespiti için Makine Öğrenmesi Algoritmalarının Performans Karşılaştırması
   URL tabanlı kimlik avı saldırılarını sınıflandırmaya yönelik makine öğrenimi analizi.
10. Veri Seti - Sınıflandırma İlişkisinde Performansa Etki Eden Faktörlerin Değerlendirilmesi
    Fırat Üniversitesi, veri sınıflandırmasında algoritma başarılarını karşılaştıran çalışma.
11. https://dergipark.org.tr/tr/download/article-file/2734758 A New Approach Based on Centrality Value in Solving the Minimum Vertex Cover Problem: Malatya Centrality Algorithm
12. https://dergipark.org.tr/en/download/article-file/2853600 A New Approach Based on Centrality Value in Solving the Maximum Independent Set Problem: Malatya Centrality Algorithm