DENGESİZ VERİLERDE SENTETİK AZINLIK AŞIRI ÖRNEK TEKNİKLERİNİN (SMOTE) KARŞILAŞTIRILMASI: İNME VERİSİ ÖRNEĞİ

COMPARISON OF SYNTHETIC MINORITY OVERSAMPLING TECHNIQUES (SMOTE) ON IMBALANCED DATA: THE STROKE DATA EXAMPLE

ÖNDER ÖZER

DR. ÖĞR. ÜYESİ ONUR TOKA

Tez Danışmanı

Hacettepe Üniversitesi

Lisansüstü Eğitim-Öğretim ve Sınav Yönetmeliğinin

İstatistik Anabilim Dalı için Öngördüğü

YÜKSEK LİSANS TEZİ olarak hazırlanmıştır.

2025

Canım Anneme…

ÖZET

DENGESİZ VERİLERDE SENTETİK AZINLIK AŞIRI ÖRNEKLEME TEKNİKLERİNİN (SMOTE) KARŞILAŞTIRILMASI: İNME VERİSİ ÖRNEĞİ

Önder ÖZER

Yüksek Lisans, İstatistik Bölümü

Tez Danışmanı: Dr. Öğr. Üyesi Onur TOKA

Haziran 2025, 88 sayfa

Dengesiz sınıflı veri problemi günümüzde sınıflandırma algoritmalarının kullanımını sınırlandıran en büyük problemlerden biridir. Bu çalışma kapsamında aşırı örnekleme tekniği olarak temel SMOTE, Borderline-SMOTE, SVM-SMOTE, SMOTE-ENN, KMeans-SMOTE, SMOTETomek ve ADASYN algoritmaları ve sınıflandırma algoritması olarak lojistik regresyon (LR), rastgele ormanlar (RF), destek vektör makineleri (SVM) ve XGBoost (XGB) algoritmaları kullanılmıştır. Sınıflandırma performansları Kesinlik, Duyarlılık, F1 skoru, ROC eğrisi ve AUC değeri gibi ölçütler üzerinden karşılaştırılmış ve hangi ikilinin sınıflandırmada daha başarılı sonuçlar araştırılmıştır. Çalışmada bahsedilen performans ölçütleri kullanılarak ikili olarak performansları test edilmiş ve son bölümde her aşırı örnekleme algoritması için bir sınıflandırma algoritması seçilecek karşılaştırma yapılmış ve en verimli ikilinin ortaya çıkarılması amaçlanmıştır. Yapılan karşılaştırmaların sonucu olarak en yüksek performansa sahip ikili olarak, SMOTE-ENN aşırı örnekleme algoritması ve sınıflandırma algoritması olarak RF algoritması ikilisinin birlikte kullanımının en başarılı sınıflandırma yapan ikili olduğu gösterilmiştir.

Anahtar Kelimeler: Sınıflandırma, Aşırı örnekleme, Model, Simülasyon, Dengesiz veri.

ABSTRACT

COMPARISON OF SYNTHETIC MINORITY OVERSAMPLING TECHNIQUES (SMOTE) ON IMBALANCED DATA: THE STROKE DATA EXAMPLE

Önder ÖZER

Master of Science, Department of Statistics

Supervisor: Asst. Prof. Onur TOKA

June 2025, 88 pages

The class imbalance problem remains one of the most significant challenges limiting the effectiveness of classification algorithms in contemporary data-driven applications. This study investigates the impact of various oversampling techniques including SMOTE, Borderline-SMOTE, SVM-SMOTE, SMOTE-ENN, KMeans-SMOTE, SMOTE-Tomek, and ADASYN on the performance of classification models. The classifiers employed in this evaluation are Logistic Regression (LR), Random Forest (RF), Support Vector Machines (SVM), and XGBoost (XGB). The models' performances were assessed based on widely used evaluation metrics, including Precision, Recall, F1-score, ROC curve and AUC value. The comparisons were conducted between oversampling techniques and classification algorithms to determine the most effective combinations. In the final phase of the study, a single best-performing classifier was selected for each oversampling method, followed by a comparative analysis to identify the overall most successful pair. The experimental results demonstrate that the combination of the SMOTE-ENN oversampling technique and the RF classifier yields the highest performance across the considered evaluation metrics, indicating it as the most effective pairs for handling imbalanced datasets in this context.

Keywords: Classification, Oversampling, Model, Simulation, Imbalanced data.

TEŞEKKÜR

Lisans eğitimimin ortalarında tanıştığım, tez çalışma sürecimde ve yüksek lisansımın tamamında değerli katkılarıyla çalışmalarıma yön veren sadece eğitim hayatımda değil her konuda desteğini gördüğüm değerli danışmanım sayın Dr. Öğretim Üyesi Onur TOKA'ya bana olan inancı ve destekleri için teşekkür ederim. Ayrıca yüksek lisans tez sınav jürisinde yer alarak gerek sınav öncesinde gerekse sınav anındaki eleştirileri ile bakış açımı genişleten değeri hocalarım Prof. Dr. Nursel KOYUNCU'ya ve Dr. Öğretim Üyesi Esra SATICI'ya ayrı ayrı tüm içtenliğimle teşekkür ederim.

Hayatım boyunca maddi manevi hiçbir durumda desteklerini esirgemeyen başta babam Hasan ÖZER ve annem Ayşe ÖZER olmak üzere bütün aileme teşekkür ederim

Hayatımda her durumda yanımda olan ve ikinci ailem olan bütün arkadaşlarıma teşekkür ederim.

Önder ÖZER

Haziran 2025, Ankara

İÇİNDEKİLER

ŞEKİLLER DİZİNİ

ÇİZELGELER DİZİNİ

SİMGELER VE KISALTMALAR

Kısaltmalar

SMOTE	Synthetic Minority Over-sampling Technique (Sentetik Azınlık Aşırı Önekleme)
RF	Random Forest (Rastgele Orman)
DT	Decision Tree (Karar Ağacı)
SVM	Support Vector Machine (Destek Vektör Makinesi)
XBG	Extreme Gradient Boosting (Aşırı Gradyan Arttırma)
LR	Logistic Regression (Lojistik Regresyon)
CNN	Convolutional Neural networks (Evrişimli Sinir Ağları)
RNN	Recurrent Neural Networks (Tekrarlayan Sinir Ağları)
BERT	Bidirectional Encoder Representations from Transformers (Çift Yönlü Kodlayıcı Gösterimleri Dönüştürücüleri)
LST	Long Short Term Memory (Uzun Kısa Süreli Bellek)
NB	Naive Bayes (Naif Bayes)
KNN	K-Nearest Neighbors (K-En Yakın Komşu)
ENN	Edited Nearest Neighbors (Değiştirilmiş en yakın komşu)
GA	Genetic Algorithm (Genetik Algoritma)
CGMOS	Certainly Guided Minority OverSampling (Güvenirliğe Dayalı Azınlık Sınıfı Aşırı Örneklemesi)
SHAP	Shapley Additive Explanations (Shapley Katkı Açıklamaları)
ROC	Receiver Operating Characteristic (Alıcı Çalışma Karakteristiği Eğrisi)
AUC	Area Under the Curve (Eğrinin Altında Kalan Alan)
ROS	Random Oversampling (Rastgele Aşırı Örnekleme)
GAT	Graph Attention Network (Grafik Dikkat Ağı)

ADASYN	Adaptive Synthetic Sampling Approach for Imbalanced Learning (Dengesiz Öğrenme için Uyarlanabilir Sentetik Örnekleme Yaklaşımı)
TP	True Positive (Doğru Pozitif)
FP	False Positive (Yanlış Pozitif)
TN	True Negative (Doğru Negatif)
FN	False Negative(Yanlış Negatif)
BMI	Body Mass Index (Vücut Kitle İndeksi)

1. GİRİŞ

İnsanlık, sadece neslini devam ettirmek üzere değil, aynı zamanda merak duygusunu tatmin etmek üzere de yaşayagelen bir canlıdır. Bu merak, sadece ne, kim gibi sorularla tanımlayıcı bir süreç için değil aynı zamanda sonraki benzer koşullar durumunda nasıl sorusunun cevabı olan açıklayıcı bir süreci de tetikler. Yani, belirli şartlar altında karar verebilmek için içinde bulunulan durum eski tecrübelerimizin ortaya koyduğu çıktılara göre belirlenmektedir. Günlük hayatın içerisinde yolculuk için otobüs mü, bisiklet mi; yağmur için şemsiye mi yağmurluk mu; derste en ön sıra mı yoksa biraz daha arkalar mı diye seçeneklerin bulunduğu yerlerde maliyetleri çok da yüksek olmayan basit kararlar sürekli verilmektedir. Ancak bu kararların yerine sonuçlarının etkileri daha maliyetli olabilecek kararları vermek hızlıca cevap veremeyeceğimiz süreçleri oluşturabilir. Farklı şartlar altında aynı sonuçları beklemek, aynı davranışta bulunmak mümkün değildir. İlkel bir insanın açken bir hayvanı avlamaya çalışmasındaki isteğiyle tok olduğunda hayvan avlama isteğindeki karar sistemi aynı olmayacaktır. Bu nedenle insanlık, süreç içerisinde hangi şartlar altında hangi kararları alabilirim düşüncesiyle sorulan sorulara verilecek olan cevaplar için evrimsel süreç içerisinde belirli adımları takip etmeye başlamıştır. Bu süreçlerle oluşan ve bir teorinin altında ortaya çıkan bilimsel adımlar, temel olarak elde bulunan veriden bir bilgi oluşturabilme çabasının yolu olarak tanımlanabilir. Günümüzde, teknolojinin gelişimiyle, bilgisayarın hayatımıza girmesiyle, günlük hayatta alınan kararların sistematik olarak üretim süreçlerinde ve yazılım prosedürlerinde örnek alınabilecek sistemler haline getirilebileceği keşfedilmiştir. Bu baş döndürücü gelişim istatiksel analiz ve tahmin süreçlerinde, istatistiksel öğrenmenin, veriden bilgi keşfinin, makine öğrenmesi süreçlerinin ve yapay zekanın hayatımızın ve yaşamımızın her alanında kullanılabilir hale gelmesini yardımcı olmuştur. Veriden öğrenilebilecek gizli birliktelikler, anlamlı değişiklikler, bilgisayarların ve teknolojik diğer aygıtların sistemler üzerinden hızlı karar verebilmesi konusunda çok büyük avantajlar sağlamıştır. Günümüz dünyası, her anın çeşitli şekillerde toplanabilir veri olarak bilgisayarların ya da okyanus altındaki belleklerin içinde, bulut yapıların içinde tutulmakta, bir nevi insanlığın tecrübeleri gibi kararlarda kullanılabilir çıktılar oluşturmak adına kayıt altına alınmaktadır. Bu yapılar, ölçümlerin sıklığının ve üretilen çıktıların sayısını giderek arttırmaktadır. Karar verme süreçlerinde bir sonraki değeri tahmin edebilmek için ilgili değişkenin özelliğine göre çeşitli çıktılar oluşturabilecek modeller, bilgisayarlardaki bu verileri aynı insan belleğinde oluşan alışkanlık kararları gibi uygularlar. İnsanın tecrübesi

ne kadar fazlaysa gelecekteki karar alma süreçleri o kadar filozofça olurken, karar sistemleri için tutulan veri ne kadar fazla ve kaliteliyse o kadar doğru ve tutarlı olacaktır. Veriden öğrenme süreçlerinde, istatistiksel öğrenme modelleri ile ortaya çıkan modeller, makine öğrenmesi süreçlerinden yapay zekaya kadar tüm sistemlerde temel olarak bir karar vermemizi sağlar. Sınıflandırma, elde edilmiş olan veriden, yine verinin içinde bulunan sınıfları baz alarak, gelecekte ortaya çıkan şartlarda alınabilecek kararı en uygun (optimum) şekilde seçebilmek için ya cezayı azaltır (minimizasyon süreci) ya da ödülü arttırır (maksimizasyon süreci). Tüm bu süreç sonucunda ortaya çıkacak olan karar, mevcut durumlar değerlendirildiğinde en olası durumu ortaya koyma amacını taşır. Ancak, eldeki veriden hangi kararın alınacağı şartlara göre değişkenlik göstereceği aşikardır. Bu durum, insanlığı, kararın başarılı olmasını sağlamak için ilgili algoritmaları üretmeye mecbur kılmıştır. Oluşturulacak başarılı bir sınıflandırma algoritması, alınacak kararın en etkin olması için kullanılacaktır. Son yıllarda elde edilen algoritmalar, etkin olmanın yanında veriden öğrenme süreçlerinde de başarılı olmayı hedeflemektedir. Veriden öğrenme süreçleri birçok algoritmayı ortaya çıkartırken bu algoritmaların kullanılacağı verilerde sadece başarı değil aynı zamanda birçok sorun ortaya çıkartmıştır. Bu çalışmada, bu sorunlardan birine, dengesiz sınıflı veri problemine odaklanacağız. Tahmin edilmeye çalışılan sınıf kategorilerinde bir sınıfın diğerine göre çok baskın şekilde gözlem sayısına sahip olması, sınıflandırma algoritmalarında baskın sınıfın daha doğru sınıflandırılmaya çalışılmasına sebep olmuştur. Bu problemin çözümü için birçok alternatif veri ön işleme süreçleri önerilse de son yıllarda çeşitli tartışmalarla birlikte birçok bu tartışmaları gidermek için birçok algoritma da önerilmektedir. Sınıflandırma yöntemleri için elde edilen veri kümesindeki dengesiz sınıf problemi için en çok tartışılan öneri algoritmalarından biri ise temel olarak SMOTE algoritması olarak isimlendirilen, "sentetik azınlık aşırı örnekleme tekniği" (Sampling Minority Oversampling TEchnique-SMOTE)'dir. Bu çalışma, sınıflandırma algoritmaları ve çeşitli SMOTE yöntemlerinin uyumunu farklı performans ölçütleri ile karşılaştırmayı ve ilgili koşullar altında en başarılı algoritmayı elde etmeyi amaçlamaktadır.

Bu kapsamda çalışmanın amacı, mevcut sınıflandırma algoritmaları ve SMOTE algoritmaları üzerine detaylı bir inceleme yaparak, bu algoritmaların birlikte kullanılması durumunda en yüksek verimliliğin nasıl sağlanabileceğini belirlemektir. Çalışma kapsamında, sınıflandırma algoritmalarının önde gelen örnekleri ve SMOTE yöntemleri incelenecektir. Ardından, bu algoritmaların birlikte kullanıldığı bir simülasyon çalışması

gerçekleştirilerek hangi senaryonun en yüksek verimliliği sağladığı belirli performans ölçütleri kullanılarak araştırılacaktır.

Bu çalışmanın bundan sonraki bölümlerinde, sınıflandırma algoritmaları üzerinde öncü nitelikteki makaleler ve bu algoritmaların geliştirilmesine katkı sağlamış önemli çalışmalar kapsamlı bir şekilde gözden geçirilmiştir. Bu bölümde, sınıflandırma algoritmalarının veri kümelerinden daha fazla bilgi çıkarabilmek ve daha doğru sonuçlar elde etmek adına nasıl evrildiği gösterilmiştir. Sonraki bölümde ise, Lojistik Regresyon (Logistic Regression – LR), Karar Ağaçları (Decision Trees – DT), Destek Vektör Makineleri (Support Vector Machines - SVM), K-En Yakın Komşu (k-nearest Neigbor - KNN) ve Rastgele Ormanlar (random forest – RF) olmak üzere çalışmada kullanılan önemli sınıflandırma algoritmaları ve özellikleri özet olarak incelenmiştir. Bununla birlikte SMOTE yönteminin ortaya çıkışı, güncellenmiş ve geliştirilmiş alternatif lgoritmaları ile bu algoritmanın gelişimine katkı sağlamış çalışmalar incelenmiştir. Son bölümde ise ilgili çalışmalarda kullanılan yöntemler ve farklı sınıflandırma algoritmalarının birlikte kullanıldığı farklı senaryolarda performans değerlendirilmesi inme verisi [43] üzerinden yapılmıştır. Elde edilen sonuçlar, tablo ve grafiklerce özetlenmiş ve yorumlanmıştır. Çalışmanın alt bölümlerinde kullanılan veri kümesi, özelliği, karşılaştırma için kullanılan ölçütler hakkındaki bilgiler de alt başlıklarda verilmiştir.

2. SINIFLANDIRMA YÖNTEMLERİ VE SMOTE ALGORİTMALARININ KULLANIMI

Sınıflandırma yöntemleri, veriden öğrenme süreçlerindeki bilinen en eski yöntemleri içermektedir. Kapalı formlara ve hızlı uygulanabilir formüllere sahip olan yöntemlerden, iteratif süreçleri olan yöntemlere, algoritmaların bir araya getirilerek birleştirildiği ve sonuçlara göre sınıflandırıcı fonksiyonlara kadar büyük bir yelpazede sayısız önerinin bir araya geldiği büyük bir çalışma alanıdır. Tahmin süreçlerinde aktarımı ve gösterimleri kolay olmasıyla tanınan algoritmalara sahip olsa da doğası gereği çeşitli dezavantajları olduğu da zaman içerisinde gözlenmiştir. Bu dezavantajları giderebilmek için elde edilen yeni algoritmaların temel amacı da tahmin süreçlerinde daha yüksek başarıların elde edilebilmesi, hata oranının düşürülmesi, genelleştirilebilir olması ve sonraki süreçlerde

de elde edilen başarının korunmasını sağlamaktır. Bu tez kapsamında kullanılan sınıflandırma yöntemleri ile ilgili özet bilgiler, dengesiz veri kümesi, bu durumda sınıflandırma yöntemlerinde karşılaşılan problemler, problemleri çözmek için kullanılan dengeleme algoritmaları incelenecektir. Bu kapsamda yapılan son dönem çalışmaları, amaçları, elde edilen sonuçlar ve algoritmalar aşağıdaki gibi özetlenebilir.

2.1. Literatürde Sınıflandırma Algoritmalarının Başarısına ve SMOTE Algoritmasının Katkısına Bir Bakış

Sınıflandırma yöntemleri için en az hata, genelleştirilebilir olma ve hızlı sonuç verme gibi ölçütler kapsamında birçok karşılaştırmalar yapılmış ve çeşitli algoritmalar önerilmiştir. Bu önerilen algoritmaların bazıları dengesiz veri kümesinde başarısız sonuçlar verdiğinde ise veri önişleme süreçlerinde oldukça yaygın kullanılan çeşitli yeniden örnekleme yöntemleri kullanılmıştır. SMOTE algoritması ise son dönemlerde oldukça fazla kullanılan yeniden örnekleme ve az gözlem sayısına sahip sınıfın gözlem sayılarını sentetik veri ile artırma amacını taşıyan yöntemlerden biridir. Sınıflandırma algoritmalarının karşılaştırılması ve veriden öğrenme süreçlerinin oldukça yaygın kullanıldığı makaleler ve çalışmalar incelendiğinde genel olarak özet bilgiler şu şekilde verilebilir:

Bhatia, Arora, ve Tomar [1], yayınladıkları çalışmada optik disk çapı, lezyon spesifik, görüntü seviyesi gibi farklı retina görüntü işleme algoritmalarının çıktılarından alınan özellikleri kullanarak diyabetik retinopatinin otomatik bilgisayar destekli tespitinde sınıflandırma algoritmalarından faydalanmıştır. DVM, DT, naif bayes (naive bayes - NB), RF ve AdaBoost algoritmaları sınıflandırmada kullandıkları algoritmalar olarak görülmüştür. Sarker I. [2], makine öğrenmesi algoritmalarının çeşitli uygulama alanlarındaki rollerini incelemiş ve sınıflandırma algoritmalarının sağlık, siber güvenlik ve akıllı şehirler gibi alanlardaki etkilerini detaylı olarak ele almıştır. RF algoritmasının birçok gerçek dünya senaryosunda istikrarlı olduğu ve yüksek performans sunduğu gösterilmiştir. Minaee, Nikzad, Chenaghlu ve Gao [3], çalışmalarında, derin öğrenme tabanlı metin sınıflandırma modellerini inceleyerek Evrişimli Sinir Ağları (Convolutional Neural networks -CNN), Tekrarlayan Sinir Ağları (Recurrent Neural Networks – RNN), Uzun Kısa Süreli Bellek (Long Short Term Memory – LST) ve dönüştürücü mimarilerini karşılaştırmıştır. Çift Yönlü Kodlayıcı Gösterimleri Dönüştürücüleri (Bidirectional Encoder Representations from Transformers -BERT) tabanlı modellerin özellikle düşük

veriyle eğitim senaryolarında bile geleneksel sınıflandırma yöntemlerine üstün geldiği gösterilmiştir. Ismail Fawaz [4], zaman serisi sınıflandırmasında derin öğrenme tabanlı yöntemlerin performansını geniş bir veri kümesi üzerinde sistematik olarak analiz etmiştir. CNN temelli modellerin çoğu zaman geleneksel algoritmalara göre daha yüksek doğruluk sağladığı rapor edilmiştir. Qian Li [5], metin sınıflandırmasında yüzeysel yöntemlerden derin öğrenmeye geçiş sürecini kapsamlı şekilde incelemiştir. TF-IDF tabanlı SVM gibi klasik yaklaşımların hâlâ bazı senaryolarda etkili olduğu, ancak çoğu derin öğrenme modelinin genel performansta üstünlük sağladığı bulunmuştur. NB, KNN, RF ve XGB (Extreme Gradien Boosting- XGBoost) karşılaştırılan sınıflandırma algoritmalarının içindedir.

Vakili M. [6], nesnelin interneti ağ trafiği veri kümeleri üzerinde 11 farklı makine ve derin öğrenme algoritmasının sınıflandırma performansını karşılaştırmıştır. XGB ve RF hem doğruluk hem de F1 skoru açısından en istikrarlı sonuçları sunmuştur. Sarker [7], akıllı telefon verileri üzerinden bağlam farkındalığına sahip sistemler için makine öğrenmesi algoritmalarını değerlendirmiştir. DT ve RF, bağlamlı tahminleme görevlerinde yüksek doğruluk oranlarıyla öne çıkmıştır. Kumar ve Vadlamani [8], duygu analizi ve düşünce madenciliğinde kullanılan sınıflandırma algoritmalarını ve bu algoritmaların görev odaklı başarımını analiz etmiştir. NB ve SVM algoritmaları, özellikle kısa metin analizlerinde etkili sonuçlar vermiştir. Sarker, Abushark, Alsolami ve Khan [9], çalışmalarında siber güvenlik saldırılarını sınıflandırmak amacıyla Intrudtree adlı makine öğrenmesi tabanlı model geliştirilmiştir. RF algoritması, saldırı tespitinde en yüksek doğruluk ve en düşük hata oranına sahip algoritma olarak öne çıkmıştır. Sarker ve Salah [10], bağlam farkındalığına sahip mobil uygulama kullanım tahminleri için sınıflandırma algoritmalarını test etmiştir. RF modeli, kullanıcı davranışlarını tahmin etmede en başarılı performansı göstermiştir. Sarker, Kayes, ve Watters [11], kişiselleştirilmiş akıllı telefon kullanım tahmini için çeşitli sınıflandırma modellerinin başarısını analiz etmiştir. ZeroR, NB, DT, RF, SVM, KNN, AdaBoost, RIPPER, RIDOR ve LR gibi on farklı makine öğrenmesi sınıflandırma algoritması değerlendirilmiştir. Ayrıca, derin öğrenme yaklaşımı olarak yapay sinir ağı modeli de karşılaştırmalara dahil edilmiştir. DT ve SVM algoritmalarının kişiselleştirilmiş tahmin görevlerinde yüksek doğruluk ve düşük gecikme ile çalıştığı gözlemlenmiştir. Marbac [12], çalışmasında, advers ilaç reaksiyonlarının tespiti için bayesci model seçimi yaklaşımını LR ile birleştirerek, daha hassas sonuçlar elde etmiştir. Önerilen yöntem, geleneksel aşırılık ölçütlerine göre daha iyi pozitif ve negatif kontrol oranları sağlamıştır. Valle, Lima, Millar, Amratia ve Haque

[13], tanı testlerindeki hataların LR modellerinde önyargıya neden olabileceğini ve bu önyargının pratik düzeltme yaklaşımlarıyla nasıl giderilebileceğini incelemiştir. Simülasyonlar ve saha verileri, tanı testlerindeki hataların lojistik regresyon tahminlerinde sistematik önyargılara yol açabileceğini göstermiştir. Önerilen düzeltme yöntemleri bu önyargıları azaltmada etkili olmuştur. Allam, Nagy, Thoma ve Krauthammer [14], kalp yetmezliği sonrası 30 günlük yeniden hastaneye yatış tahmininde lojistik regresyon ve sinir ağlarını karşılaştırmıştır. LR modeli, AUC skoru açısından en iyi sinir ağı modeliyle benzer performans göstermiştir. Elkouri [15], çalışmasında, Yelp incelemelerinin duygu analizinde lojistik regresyonun etkinliğini değerlendirmiştir. LR modeli, pozitif/negatif sınıflandırmada %92,90 doğruluk oranı ile en iyi performansı göstermiştir.

Aragaw [16], çalışmasında Etiyopya'da evli kadınlar arasında modern kontraseptif kullanımını etkileyen faktörleri belirlemek için LR uygulamıştır. Eğitim düzeyi ve medya erişimi, modern kontraseptif kullanımında önemli belirleyiciler olarak bulunmuştur. Matsui, Cruz ve Tang [17], Çinli kadınlar arasında bilgisayar kullanım süresi ile yüz cilt koşulları arasındaki ilişkiyi LR ile analiz etmiştir. Yüksek bilgisayar kullanımı, belirli cilt koşullarıyla anlamlı şekilde ilişkilendirilmiştir. Kaya, Leite ve Miller [18], polytomous maddelerde farklı işleyiş gösteren maddelerin tespiti için lojistik regresyon modellerini karşılaştırmıştır. Küçük örneklem büyüklükleri ve yetenek dağılımlarının normallikten sapması, model performansını etkilemiştir. Budimir, Atkinson ve Lewis [19], çalışmasında heyelan olasılığı haritalaması için LR kullanan araştırmalar sistematik olarak incelenmiştir. Eğim ve yön gibi değişkenler, heyelan olasılığı tahmininde en sık kullanılan ve anlamlı bulunan değişkenlerdir. Yadav, Bharadwaj ve Pal [20], çalışmalarında öğrencilerin geçmiş performans verileri kullanılarak DT ile başarı tahmini yapılmıştır. DT, öğrenci başarısını tahmin etmede etkili bir yöntem olarak değerlendirilmiştir. Chen ve Lin [21], DT algoritmasının eğitim verisi madenciliğindeki uygulamaları incelenmiştir. DT, öğrencilerin sınıflandırılmasında ve öğrenme deneyimlerinin iyileştirilmesinde kullanılmıştır. Zhang [22], çalışmasında DT algoritmasının iş dünyasındaki uygulamaları incelenmiştir. DT, müşteri segmentasyonu ve pazar stratejilerinde etkili biçimde kullanılmıştır. Al-Sarem [23], akademik danışmanlık süreçlerinde DT algoritmasının kullanımı araştırılmıştır. C4.5 algoritması, öğrencilerin akademik başarılarını tahmin etmede etkili bulunmuştur. Amancio ve diğerleri [24], dokuz farklı sınıflandırma algoritması sistematik olarak karşılaştırılmıştır. Yüksek boyutlu veri kümelerinde, KNN algoritması diğer yöntemlere kıyasla üstün

performans göstermiştir. Sun, Xue, Zhang, Yen ve Lv [25], GA kullanılarak otomatik CNN mimarileri tasarlanmıştır. Otomatik CNN'ler, manuel mimarilere kıyasla daha yüksek doğruluk ve daha düşük hesaplama gereksinimi göstermiştir. Kumar, Sehgal ve Chauhan [26], karar destek sistemleri için çeşitli sınıflandırma algoritmaları karşılaştırılmıştır. GA ve SVM, tahmin doğruluğu açısından en iyi performansı göstermiştir. Hui, Ling, Xiao ve Shan [27], çalışmasında çoklu ortam verileri için yüksek performanslı bir KNN sorgu işleme sistemi geliştirilmiştir. Sistem, büyük veri kümelerinde hızlı ve doğru sınıflandırma sağlamıştır. Zhao, Zhang ve Liu [28], çalışmalarında finansal zaman serisi tahmini için sınıflandırma algoritmalarının hiper-parametre optimizasyonu incelenmiştir. Ağırlıklı hata fonksiyonu ile yapılan grid arama, sınıflandırma doğruluğunu artırmıştır. Jin, De-Lin ve Fen-Xiang [29], çalışmasında ID3 karar ağacı algoritmasının geliştirilmiş bir versiyonu sunulmuştur. Geliştirilmiş algoritma, bilgi kazancı oranını artırarak sınıflandırma doğruluğunu yükseltmiştir. Shao, Zhang, Li ve Chen [30], tolerans tasarımı için bilgi edinmede ID3 algoritmasının uygulanması incelenmiştir. ID3, mühendislik tasarımında bilgi edinme sürecini etkinleştirmiştir.

SMOTE algoritmasının temel amacının baskın olan gözlem sayısına sahip olan sınıfın sınıflandırıcı algoritmalarındaki etkisini baskın olmayan sınıfın lehine olacak şekilde düzenlemektir. Bu kapsamda, Douzas ve Bacao [31], çalışmasında Geometric SMOTE yöntemiyle azınlık sınıflar için yönlendirilmiş örnekleme yöntemi önerilmiş, veri üretimi geometrik bölgelere dayandırılmıştır. Yöntemsel katkı olarak klasik SMOTE'un sınırsız örnekleme yapısına karşı daha yapılandırılmış yaklaşım getirilmiş ve doğru sınıflandırma ölçütü ile test edilmiştir. Dablain, Krawczyk ve Chawla [32], DeepSMOTE algoritması, CNN gibi derin öğrenme yapıları için tasarlanmış özel bir yeniden örnekleme yöntemi sunmuştur. Derin mimarilere uygun gözlem üretimi sağlanmış, doğruluk oranı ölçütünde başarı değerlendirilmiştir. Mansourifar ve Shi [33], çalışmasında DeepSMOTE yöntemi kullanılarak örnekleme sürecindeki rastlantısallığı azaltarak daha kararlı veri artırımı sağlatılmıştır. Denetimli örnekleme mantığı getirilmiş, doğruluk performansı ölçülerek yöntemde geliştirme yapılmıştır. Zhang, Ma, Gan, Jiang ve Agam [34], çalışmalarında güvenirliğe dayalı azınlık sınıf aşırı örneklemesi (Certainty Guided Minority OverSampling – CGMOS) yaklaşımıyla örnekleme süreci gözlemlerin belirsizlik derecesine göre ağırlıklandırılmış, karar sınırına yakın bölgelerde sınıflandırıcıyı etkileyebilecek veri üretimi sağlanmışlardır. Doğruluk ölçütü odaklı analizle SMOTE'un teorik güçlendirmesine katkı sunulmuştur. Joloudari, Marefat, Nematollahi, Oyelere ve

Hussain [35] çalışmalarında, SMOTE ile CNN yapılarını birleştirerek, dengesiz veri kümelerinde örnekleme ve sınıflandırma aynı çatı altında bütünleştirilmiştir. %99,08 genel doğruluk oranına sahip olan yüksek başarılı bir sınıflandırma sonucu elde edilmiştir. Adi Pratama ve Oktora [36] çalışmasında, yoksulluk sınıflandırması için SMOTE uygulanmış, kırsal ve kentsel alanlara özgü duyarlılık ölçümleri yapılmıştır. Başarı ölçütü duyarlılık olarak alınmış ve veri türüne göre farklı sonuçlar raporlanmıştır. Ejiyi [37], çalışmasında polinomsal SHAP (Shapley Additive Explanations – SHAP) kullanılmış ve SMOTE'a alternatif bir veri artırma yöntemi geliştirilmiştir. Tıbbi veri kümelerinde başarı doğrulukla ölçülmüş, açıklanabilirlik katkısı ön plana çıkarılmıştır. Andriyani, Faqih ve Permana [38], çalışmalarında SMOTE kullanılarak SVM doğruluğu %71,41'den %83,89'a çıkarılmıştır. Bu iyileştirme sınıflandırma başarısını ciddi şekilde artırmıştır. Ramezankhani ve diğerleri [39], Tip-2 diyabet verisinde üç farklı sınıflandırıcı üzerinde SMOTE'un etkisi incelenmiş, doğruluk oranı ölçütleriyle başarı değerlendirilmiştir. Yöntem, çoklu model uyumluluğuyla katkı sunmuştur. Seo ve Kim [40], çalışmalarında SMOTE oranları, makine öğrenme yöntemleriyle optimize edilmiş; eğri altı alan (area under curve-AUC) ölçütü üzerinden saldırı tespiti başarıyla iyileştirilmiştir. Hiperparametre ayarı ile örnekleme süreçlerine esneklik kazandırılmıştır. Yang, ve diğerleri [41], çalışmalarında ise SMOTE ve topluluk öğrenmesi birlikte kullanılarak demiryolu sinyal arızalarının sınıflandırılması yapılmıştır. Hata toleranslı sınıflandırma hedeflenmiş ve doğruluk ölçütü üzerinden yöntemler değerlendirilmiştir. Sailasya ve Kumari [42] çalışmasında, Kaggle Stroke Dataset[43] kullanılarak farklı makine öğrenme algoritmalarının doğruluk performansı karşılaştırılmış, veri dengesizliği SMOTE ile giderilmiştir. %94 doğrulukla SMOTE'un sınıflandırma başarısını artırdığı gösterilmiştir. Dubey, ve diğerleri [44], erken inme tespiti için açıklanabilir bir model geliştirilmiş, SMOTE ile veri dengelenmiştir. Doğruluk %92'ye ulaşmış, çalışmanın katkısı da modelin yorumlanabilirliğini artırması olarak ifade edilmiştir. Kitova, Ivanov ve Hooper [45], inme veri kümesi üzerinde derin öğrenme ve klasik sınıflandırıcılar karşılaştırılmış, SMOTE ile dengelenen veriyle %98 doğruluk elde edilmiştir. Derin öğrenme modellerinin diğer yöntemlere göre üstün performansı vurgulanmıştır. Hassan ve diğerleri [46], önemli risk faktörlerini belirlemek için makine öğrenme tabanlı tahmin modelleri geliştirilmiş, SMOTE uygulaması sonrası en iyi model %96 doğruluk sağlamıştır. Yöntemsel katkı, risk faktörü tespiti üzerine odaklanmıştır. Dev ve diğerleri [47], çalışmalarında rasgele aşırı öğrenme (random oversampling – ROS) ile veri dengelenmiş, çeşitli makine öğrenme algoritmaları karşılaştırılmıştır. SVM %99,99 doğrulukla en iyi

sonucu vermiş, klasik dengeleme stratejilerinin etkisi vurgulanmıştır. Jiawei ve diğerleri [48], grafik dikkat ağı (graph attention network – GAT) modeliyle inme sonrası sertlik (spasitise ya da motor disfonksiyon bozukluğu) tahmini yapılmıştır. SMOTE uygulanarak %93 doğruluk elde edilmiş, grafik tabanlı açıklanabilirlik ön plana çıkarılmıştır. Tomita ve diğerleri [49], üç boyutlu evrişmli ağlarla inme sonrası lezyon segmentasyonu gerçekleştirilmiş, SMOTE ile dengelenen veride %90 doğruluk raporlanmıştır. Yöntemsel katkı hacimsel CNN tasarımıdır. Pinto ve diğerleri [50] çalışmalarında, denetimli ve denetimsiz öğrenme teknikleri birleştirilmiş, SMOTE ile dengelenen veriyle %91 doğruluk elde edilmiş, karma öğrenme yapılarının gücü vurgulanmıştır. Biswas ve diğerleri [51], ROS ile veri dengelenmiş, makine öğrenme algoritmaları karşılaştırılmıştır. SVM %99,99 doğrulukla en başarılı model olarak öne çıkmıştır. Geleneksel yöntemlerin dengelenmiş veri kümelerinde güçlü etkisi teyit edilmiştir.

Dengesiz veri kümesinde yapılan veri ön işleme süreçlerindeki sentetik veri artırımı sonrasında sınıflardaki gözlem sayılarının dengelenmesi, sınıflandırma algoritmalarında başarıyı arttırdığı aktarılan çalışmalarda da görüldüğü gibi artmaktadır. Bu nedenle SMOTE algoritmasında alternatif düzenlemeler yapılması, yani sentetik veri üretiminin kalitesinin artması, sınıflandırma algoritmalarında da başarıları arttırılmıştır. Bu çalışmadaki amaç ise hangi SMOTE algoritmasıyla hangi sınıflandırma algoritmasının daha başarılı olduğunu göstermektir. Bu nedenle bir sonraki bölümde kullanılan sınıflandırma algoritmaları ile ilgili özet bilgiler aktarılmıştır.

2.2. Sınıflandırma Algoritmaları

Teknolojideki hızlı gelişme, bilgisayar sistemlerinin başarısı ve teorideki bilgilerin öğrenme süreçleriyle karar vericilere kolaylık sağlayacak şekilde öğretilebilmesi, sonrasında ise bu öğrenme sürecini makinenin kendinin yapabilmesi baş döndürücü bir hızı da beraberinde getirmiştir. Bu hız arttıkça, karar mekanizmalarındaki sınıflandırma çözümlerinin daha başarılı, daha az hataya sahip hale getirme merakı giderek artmıştır. Önceleri insanlar tarafından her bir örnek üzerinde yapılan değerlendirmeler, yetersiz ve yavaş olarak eleştirilirken, bilgisayarın gücü, hızı ve istatistiksel modellerin etkinlikleri süreci daha efektif halde gelmiştir. Sınıflandırma algoritmalarının ortaya çıkışını bilgisayar ve verinin bir araya gelmesindeki istatistiksel öğrenme süreci ile açıklayabilsek de kullanım alanları bu alanla sınırlı kalmamıştır. Günümüzde tıp alanında kullanılan görüntüleme sistemlerine entegrasyonu sayesinde doktorlar için bir yardımcı asistan

olması, yüz tanıma sistemlerinde, yazı karakterleri belirlemede, parmak izinde yine görüntü işleme yapabilmesi birçok alanda uygulanabilir olmasını sağlamıştır. Sınıflandırma algoritmaları, denetimli öğrenmenin en yaygın kullanılan yöntemleri arasında yer almakta olup, veri kümesindeki gözlemleri belirli kategorilere veya sınıflara ayırmayı amaçlar. Bu algoritmalar, geçmiş verilerden öğrenerek, yeni verilerin ait olduğu sınıfı tahmin etmeye çalışır. DT, LR, SVM, KNN, RF ve XGB gibi pek çok farklı algoritma, sınıflandırma problemlerinde kullanılmaktadır. Bu yöntemler; sağlık, finans, pazarlama ve doğal dil işleme gibi çok sayıda uygulama alanında başarılı sonuçlar elde etmektedir. Özellikle makine öğrenmesi alanında yapılan çalışmalar, algoritmaların doğruluğunu, genelleme kapasitesini ve işlem maliyetlerini karşılaştırarak farklı senaryolara uygun modellerin seçilmesine katkı sağlamaktadır [52, 53].

2.3. Lojistik Regresyon (LR)

LR'nin temeli, 1830'larda Belçikalı matematikçi Pierre François Verhulst tarafından geliştirilen lojistik büyüme modeline dayanır. Verhulst, bu modeli nüfus artışını sınırlayan faktörleri dikkate alarak tanımlamıştır. 20. yüzyılın ortalarında, Joseph Berkson bu fonksiyonu istatistiksel modelleme amacıyla kullanarak "logit" terimini tanıtmış ve biyolojik doz-cevap analizlerinde uygulamıştır [54]. 1970'lerde ise Daniel McFadden, çoklu logit modelini ayrık seçim teorisiyle ilişkilendirerek ekonometrik modellemelerde önemli bir adım atmıştır [55].

İstatistikte uygulamalarda sıklıkla kullanılan ve en önemli yöntemlerden biri olan LR, denetimli öğrenme yöntemlerinde özellikle ikili sınıflandırma (binary LR), çoklu nominal sınıflandırma (multinomial LR) ve ordinal sınıflandırma (ordinal LR) algoritmaları ile çeşitlenmektedir. Denetimli algoritmalarda, LR bağımsız değişkenler (özellikler) ile bağımlı değişken (hedef değişken) arasındaki ilişkiyi modellemek için istatistiksel bir yaklaşımdır. LR, bağımsız değişkenlerin doğrusal bir kombinasyonunu kullanarak hedef değişkenin belirli bir sınıfa ait olma olasılığını tahmin eder. Model, sigmoid (lojistik) fonksiyonu sayesinde çıktıların 0 ile 1 arasında bir değer almasını sağlar, bu da sınıflandırma problemlerinde karar verme sürecini kolaylaştırır. Ayrıca, LR, yorumlanabilirliği yüksek olan her bir bağımsız değişkenin sınıflandırma üzerindeki etkisini anlamaya olanak sağlayan önemli bir yöntem olup bu özellikleriyle açıklanabilir makine öğrenme ya da yapay zeka için kullanımı uygun olan bir algoritmadır. LR, bu özellikleriyle hem akademik çalışmalarda hem de endüstriyel uygulamalarda önemli bir

yer edinmiştir. Özellikle tıbbi teşhis, finansal risk analizi ve müşteri davranışları gibi ikili sonuçların elde edilmesi gereken alanlarda etkili bir şekilde uygulanmaktadır.

2.4. Karar Ağaçları (DT)

Karar ağaçları, denetimli öğrenme süreçlerinde hedef değişkenin kategorik ya da sürekli olması durumu için bile alternatif algoritmalara sahip olan dolayısıyla hem sınıflandırma hem de regresyon problemleri için uygulanabilir. Bu algoritmalar, veriyi değişkenlerinin aldığı değerler için anlamlı kırılımları tespit edecek şekilde dallara ayırarak, sonuçları bir kökten başlayıp sırasıyla çeşitli dallara ve sonunda ayrılmama kısıtına uygun olacak şekilde tekrar bölünmeyen yaprak düğümlere kadar indirgeme özelliğine sahiptir. İlk kez Morgan ve Sonquist tarafından sosyal bilimler alanında önerilen bu yaklaşım daha sonra Quinlan tarafından geliştirilen ID3 (Iterative Dichotomiser 3) algoritması ile makine öğrenmesi literatüründe önemli bir yer edinmiştir. Quinlan'ın bu çalışması, karar ağaçlarının bilgi kazancı kavramı ile daha etkili hale gelmesini sağlamıştır. Karar ağaçları, yorumlanabilirliği yüksek modeller üretmeleri ve veri ön işleme gereksinimlerinin düşük olması sayesinde günümüzde tıp, finans ve pazarlama gibi birçok alanda sıklıkla tercih edilmektedir[56,57].

Karar ağaçları, veri kümelerini sınıflandırmak veya regresyon analizi yapmak için hiyerarşik bir ağaç yapısı kullanan esnek ve güçlü makine öğrenmesi algoritmalarıdır. Bu yöntem, veri kümesindeki her bir örneği belirli özellikler üzerinden parçalara ayırarak sınıflandırma işlemi gerçekleştirir. Ağaç yapısında her düğüm, belirli bir özellik üzerinde bir karar noktasıdır ve bu noktada veriler, belirlenen kurala göre alt dallara ayrılır. Yaprak düğümler, sınıflandırmanın nihai sonucunu temsil eder. Karar ağaçlarının en büyük avantajlarından biri, modelin kolay anlaşılır ve görselleştirilebilir olmasıdır. Bu sayede, modelin nasıl bir karar verdiği açık bir şekilde yorumlanabilir. Ayrıca, karar ağaçları eksik verilerle çalışabilme yeteneğine sahip olup hem kategorik hem de sayısal veri türleriyle etkili bir şekilde işlem yapabilir. Bu özellikleri, karar ağaçlarını veri madenciliği, tıbbi teşhis ve müşteri segmentasyonu gibi çeşitli alanlarda tercih edilen bir yöntem haline getirmektedir [58].

2.5. XGBoost (XGB- Extreme Gradient Boosting)

XGBoost, 2016 yılında Tianqi Chen ve Carlos Guestrin tarafından "XGBoost: A Scalable Tree Boosting System" başlıklı çalışmada tanıtılmıştır [59]. Bu algoritma, özellikle büyük veri kümelerinde yüksek performans ve hız sağlamak amacıyla geliştirilmiş, eksik verilerle başa çıkabilen ve paralel işlemeye uygun bir yapı sunmuştur.

XGB, dallanma optimizasyonu, düzenlileştirme (regularization) ve paralelleştirilmiş hesaplama gibi iyileştirmelerle aşırı öğrenme (overfitting) riskini azaltır ve büyük veri kümelerinde yüksek performans gösterir. Ayrıca, dengesiz veri kümelerinde başarı gösterdiği için SMOTE gibi aşırı örnekleme yöntemleriyle birlikte kullanılarak modelin doğruluğunu artırmak mümkündür. XGB'nin hisse senedi tahmini, dolandırıcılık tespiti, görüntü sınıflandırma ve tıbbi teşhis gibi birçok alanda başarılı sonuçlar verdiği çalışmalarda gösterilmiştir [60, 61].

2.6. Destek Vektör Makineleri (SVM- support vector machines)

SVM algoritması, 1963 yılında Vladimir Vapnik ve Alexey Chervonenkis tarafından doğrusal sınıflandırma problemleri için geliştirilmiştir. 1992'de Bernhard Boser, Isabelle Guyon ve Vapnik [62] çekirdek (kernel) yöntemiyle SVM'yi doğrusal olmayan sınıflandırmalar için geliştirmişlerdir. 1995'te ise Corinna Cortes ve Vapnik, "soft margin" yaklaşımını tanıtarak SVM'nin pratik uygulamalarını kolaylaştırmışlardır [63].

SVM, verileri en iyi şekilde ayıran bir hiper düzlem bularak sınıflandırma ve regresyon problemlerinde etkili sonuçlar elde eden güçlü bir makine öğrenmesi algoritmasıdır. Bu yöntem, verileri farklı sınıflara ayırmak için mümkün olan en geniş marjı sağlayan hiper düzlemi belirlemeye çalışır. SVM, özellikle yüksek boyutlu veri kümelerinde etkili olup, doğrusal olarak ayrılabilir olmayan veriler için, çekirdek fonksiyon yöntemleri kullanılarak verileri daha yüksek boyutlu uzaylarda izlemleyebilir. Bu sayede karmaşık sınıflandırma problemlerinde güçlü performans sergiler. Model, yalnızca marj üzerinde yer alan destek vektörleri ile çalıştığı için diğer veri noktalarının aykırılıklarından etkilenme olasılığı düşüktür. SVM'nin bu yapısal özellikleri, özellikle yüksek boyutlu veri kümelerinde sınıflandırma performansının korunması, yalnızca destek vektörlerine dayanarak öğrenme yapılabilmesi ve çekirdek fonksiyonları aracılığıyla doğrusal

olmayan ayrımların modellenebilmesi açısından güçlü avantajlar sunmaktadır. Bu nitelikler, Guyon, Weston, Barnhill ve Vapnik tarafından yapılan bir çalışmada, gen ifadesi verileri üzerinde gerçekleştirdikleri sınıflandırma deneyleri ile somut biçimde ortaya konmuş; SVM'nin hem yüksek doğruluk oranı hem de etkili değişken seçimi açısından üstün performans sergilediği gösterilmiştir [64].

2.7. Rastgele Ormanlar (RF-Random Forest)

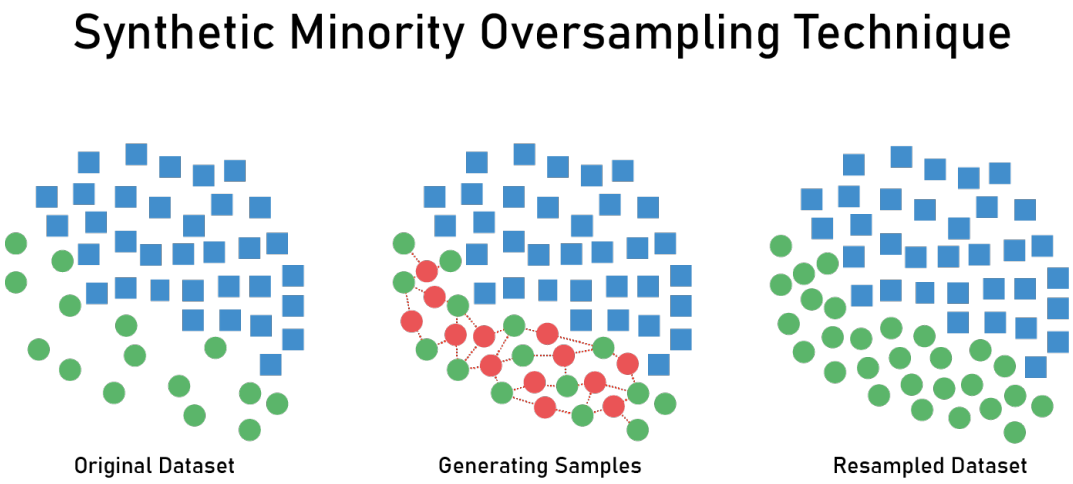
Random Forest algoritması, ilk olarak 1995 yılında Tin Kam Ho tarafından "Random Decision Forests" başlıklı çalışmasında tanıtılmıştır [65]. Ho, bu yöntemle karar ağaçlarının rastgele alt uzaylarda oluşturulmasını önermiştir. 2001 yılında Leo Breiman, bu yaklaşımı "bagging" yöntemiyle birleştirerek Random Forest algoritmasını geliştirmiş ve geniş çapta bilinen bir algoritma haline getirmiştir [66].

RF, çok sayıda karar ağacını bir araya getirerek sınıflandırma ve regresyon problemlerinde yüksek doğruluk sağlayan güçlü bir makine öğrenmesi algoritmasıdır. Bu yöntem, her bir ağacın farklı veri alt kümeleri ve özelliklerle eğitilmesi sayesinde modelin genelleme kabiliyetini artırır. Nihai tahmin, tüm ağaçların tahminlerinin ortalaması alınarak (regresyon için) veya çoğunluk oyu kullanılarak (sınıflandırma için) elde edilir. RF, karar ağaçlarının aşırı öğrenme (overfitting) eğilimine karşı oldukça dirençlidir ve veri kümesindeki gürültüden minimum düzeyde etkilenir. Bu algoritma, yüksek boyutlu veri kümeleriyle etkili bir şekilde çalışabilir ve eksik verilerle başa çıkma konusunda da güçlüdür. RF algoritmasının bu özellikleri Statnikov ve diğerleri tarafından yapılan biyomedikal sınıflandırma çalışmalarında kapsamlı biçimde gösterilmiştir. Araştırmada, RF algoritmasının diğer birçok sınıflayıcıya kıyasla daha yüksek genelleme başarısı sergilediği ve özellikle biyolojik veri kümelerinde istikrarlı sonuçlar ürettiği vurgulanmıştır [67].

Kullanılan bu sınıflandırma algoritmaları, amaç fonksiyonlarının özelliklerinden dolayı dengesiz veri kümesinde azınlık olan kümeyi sınıflandırmada başarısız olmaktadır. Bu nedenle, veri önişleme süreçlerinde veri kümesindeki dengesizliğin giderilmesi için azınlık olan sınıfın gözlem sayısında sentetik bir artış ile başarısızlık giderilmeye çalışılmaktadır. Bir sonraki bölüm sentetik veri artırımı için önerilen SMOTE ve bu algoritma üzerinden geliştirilen SMOTE tabanlı algoritmaları özetlemektedir.

2.8. SMOTE Algoritması ve Geliştirilmiş SMOTE Tabanlı Algoritmalar

2002 yılında Nitesh V. Chawla ve çalışma arkadaşları tarafından önerilmiş olan SMOTE (Synthetic Minority Over-sampling Technique), algoritmasının ortaya çıkışı ve gelişimi de bu çalışmanın odak noktalarından biridir. SMOTE algoritmasının ortaya çıkmasının temel nedeni, dengesiz veri kümelerinde klasik makine öğrenmesi algoritmalarının genellikle çoğunluk sınıfı lehine çalışarak azınlık sınıfın doğru sınıflandırılmasını zorlaştırmasıdır. SMOTE, azınlık sınıfından sentetik örnekler oluşturarak veri kümesinin dengesini iyileştiren bir aşırı örnekleme yöntemidir [68]. Bu özelliği sayesinde, özellikle tıbbi teşhis, sahtecilik tespiti ve arıza tahmini gibi kritik alanlarda yanlış sınıflandırmayı büyük ölçüde önleyebilecek bir yöntemdir.



Şekil 1: SMOTE Tekniğinin Veri Kümesi Üzerindeki Etkisini Gösteren Bir Görselleştirme Çalışması (Orellana) [69]

Bu yaklaşım, rasgele tekrarlar yerine, azınlık sınıfındaki veri noktalarının komşuluk ilişkilerine dayalı olarak yeni örnekler üretir ve böylece sınıflar arası karar sınırlarının daha sağlıklı öğrenilmesini sağlar. Zamanla bu temel yaklaşıma gelen eleştiriler veya iyileştirmeler ile SMOTE tabanlı birçok algoritma da geliştirilmiştir. Bunlardan başlıcası, Borderline-SMOTE, SMOTE-NC, ADASYN, SMOTE-ENN ve SMOTE-Tomek gibi algoritmalardır. Bu algoritmalar, SMOTE algoritmasının eksikliklerini gidermeyi ve daha etkili örnekleme stratejileri sunmayı hedeflemiştir. Bu gelişmeler, dengesiz veri

kümeleriyle çalışmanın yaygın olduğu sağlık, güvenlik, finans gibi kritik alanlarda sınıflandırma performansının artırılmasına katkı sağlamıştır [70].

2.9. Borderline-SMOTE

Borderline-SMOTE algoritması, Han ve arkadaşları tarafından 2005 yılında önerilmiştir [71]. Eğitim verisi T, azınlık sınıfı P ve baskın sınıf N olmak üzere;

$$P = \{p_1 , p_2 , \dots , p_{pnum}\} , N = \{n_1 , n_2 , \dots , n'_{nnum}\}$$

Borderline-SMOTE algoritması aşağıdaki gibidir;

1- Komşuluk-Sınıf Analizi: Eğitim kümesindeki her bir azınlık sınıfı örneği p_i(i=1, 2, ..., pnum) için belirli bir öklidyen mesafe içerisindeki m en yakın komşusu belirlenir. Bu komşular üzerinde sınıf analizi yapılarak örneğin komşularının çoğunluğu baskın sınıfa ait ise bu örnek "tehlikeli örnek" (dangerous instance) olarak belirlenir sentetik örnek üretimi sadece bu örnekler üzerinden gerçekleştirilecektir. Bunun dışındaki örnekler güvenli (safe) ve aykırı değer (outlier) olarak değerlendirilir.

2- Sentetik Örnek Üretimi için Örnek Seçimi: İlk adımda tehlikeli olarak belirlenmiş azınlık sınıfı örnekleri üzerinden sentetik örnek seçimi için bir analiz gerçekleştirilir. Eğer bir azınlık örneğin etrafındaki bütün komşuları baskın sınıfa ait olarak belirlenmişse bu örnek gürültü olarak kabul edilir ve bu adımda işleme dahil edilmez. Ayrıca ilk adımda bahsedilen güvenli ve aykırı değer olarak değerlendirdiğimiz örnekler de sonraki adımlarda işlemlere dahil edilmezler.

3- Tehlikeli Olarak Belirlenmiş Örnekler için En Yakın Komşuları Hesaplama:

$$DANGER = \{p'_1, p'_2, \dots , p'_{dnum}\} , 0 \leq dnum \leq pnum$$

İlk 2 adımda tehlikeli olarak belirlenmiş örnekler için azınlık sınıfındaki k en yakın komşusu hesaplanır.

4- Sentetik Örnekleri Üretme:

$$SYNTHETIC_j = p'_i + r_j \times dif_j , j = 1,2, \dots , s$$

Bu adımda tehlikeli örnekler için $s \times dnum$ tane sentetik örnek üretilir. Her bir tehlikeli p' örneği için 1 ile k arasında bir s sayısı olsun. Her bir p' örneği için azınlık sınıfından seçtiğimiz k en yakın komşu içinden s en yakın komşu

seçilir. p' örneğinin en yakın s komşusu ile farkları dif_j (j= 1, 2, … , s) hesaplanır. Sonra dif_j değeri 0 ile 1 arasında rastgele bir sayı olan r_j ile çarpılarak p_i' örneğinin yakınında s adet sentetik örnek üretilmiş olur.

Bu adımlar tehlikeli olarak belirlenmiş bütün azınlık sınıfı örnekleri için tekrarlanır.

Temel SMOTE yöntemine ek olarak, yalnızca sınıf sınırına yakın, yani sınıflandırılması daha zor olan azınlık örneklerine odaklanarak sentetik veri üretir. Bu yaklaşım, karar sınırlarının daha iyi öğrenilmesini sağlayarak sınıflandırma performansını artırmayı hedeflemektedir. Yöntem, özellikle sınıf ayrımının net olmadığı veri kümelerinde etkili bir şekilde kullanılmaktadır. Yöntemin, sınıf sınırlarına odaklanarak sınıflandırma algoritmasının karar sınırlarını daha iyi öğrenmesine olanak sağlamak, temel SMOTE algoritmasına göre daha az sentetik örnek oluşturarak aşırı öğrenme ihtimalini azaltmak, doğrusal olmayan ve sınıflar arası sınıfların belirsiz olduğu veri kümelerinde daha etkili sonuçlar vermek gibi avantajları vardır. Diğer taraftan ise, sınıf sınırlarına yakın değerler daha fazla gürültü içerebileceğinden oluşturulan sentetik örneklerin de gürültülü olma riski, azınlık sınıfı için üretilen sentetik değerler sınıf sınırlarına odaklanarak üretileceğinden üretilen değerler baskın sınıf ile karışması riski bulunmaktadır. Bu da sınıflandırma algoritmasının çalışmasını olumsuz etkileyebilecek bir durumdur. Modelin karar sınırında karmaşıklaşmaya ve aşırı öğrenmeye yol açabilir

2.10. SVM-SMOTE

SVM-SMOTE algoritması, SMOTE yönteminin destek vektör makineleri (SVM) ile birleştirilmiş versiyonudur ve 2011 yılında önerilmiştir [72]. Algoritma genel olarak girdi, çıktı ve değişken mantığıyla aşağıdaki gibi çalışır:

Girdi:

- X:Orijinal eğitim kümesi, X = $\{x_1, x_2, x_3, \dots, x_n\}$
- N: Örnekleme Seviyesi (%100, %200, %300, …)
- k: En yakın komşuların sayısı
- m: Aşırı örnekleme tipine karar vermek için en yakın komşuların sayısı (enterpolasyon- interpolation ya da extrapolasyon- extrapolation)

- p: 0 ile 1 arasında tekdüze dağılımlı rasgele bir sayı

Çıktı:

- X_{new}: Aşırı örneklenmiş eğitim seti

Değişkenler:

- SV^+: Pozitif destek vektörleri kümesi, $SV^+ = \{sv_1^+, sv_2^+, sv_3^+, \dots, , sv_{N_{SV}}^+\}$
- T: Oluşturulacak sentetik örneklerin sayısı
- amount: Her bir SV^+'e denk gelen sentetik örneklerin sayısını içeren bir seri
- nn: Her bir SV^+'e denk gelen azınlık sınıftan k en yakın komşu sayısını içeren bir seri olmak üzere SVM-SMOTE algoritması adımları aşağıdaki gibidir:
  1- Orijinal eğitim kümesi X üzerinde standart bir SVM sınıflandırıcısı eğitilir. Eğitim sonucu, azınlık sınıfına ait destek vektörleri kümesi SV^+ elde edilir. Bu kümedeki destek vektörleri, azınlık sınıfının karar sınırına en yakın ve kritik örnekleri temsil eder. Buna bağlı olarak, yeni sentetik örnekler yalnızca bu SV^+ çevresinde üretilecektir.

  2- Azınlık sınıfı vektörü kümesindeki her bir sv_i^+için m sayıda en yakın komşusu belirlenerek bu komşuların sınıf dağılımları incelenir. Bu komşulardan azınlık sınıfına ait olanların sayısı n_{minor} ve baskın sınıfa ait olanların sayısı n_{major} olsun
     - Eğer $n_{major} < \frac{m}{2}$ ise, karar sınırının azınlık sınıfı lehine genişletileceği extrapolasyon yöntemi seçilir.
     - Eğer $n_{major} \geq \frac{m}{2}$ ise, mevcut karar sınırını güçlendirmek amacıyla interpolasyon yöntemi uygulanır.

  3- Her bir destek vektörü sv_i^+ için en yakın k komşu ($\{nn_{i1}^+, nn_{i2}^+, \dots, nn_{ik}^+\}$) tespit edilir. Bu komşularla destek vektörü arasındaki fark hesaplanarak d_{ij} olarak ifade edilir.

$$d_{ij} = nn_{i1}^+ - sv_i^+ \ , \qquad j = 1,2, \dots, k$$

Burada nn_{i1}^+ sv_i^+'nın j-inci en yakın azınlık sınıfı komşusudur.

- Sentetik örneklerim üretimi için örnekleme yöntemi seçimine göre aşağıdaki formüllerden biri kullanılır.
- Extrapolasyon (Karar sınırını genişletme) yöntemi seçildi ise:

Burada azınlık sınıfı destek vektörünün ve en yakın azınlık sınıfı komşusunun arasındaki farkın tam tersi yönünde bir örnek oluşturularak azınlık sınıfın sınırları baskın sınıfa doğru genişletilmiş olur.

$$x_{new}^{+} = sv_i^{+} + p\left(sv_i^{+} - nn_{ij}^{+}\right)$$

- İnterpolasyon (Karar sınırını güçlendirme) yöntemi seçildi ise:

  Burada destek vektörleri ve azınlık sınıfı komşuları arasında rastgele bir örnek üreterek karar sınırının güçlenmesi sağlanır.

$$x_{new}^{+} = sv_i^{+} + p\left(nn_{ij}^{+} - sv_i^{+}\right)$$

Borderline-SMOTE algoritması ile benzer olarak bu yöntemde de sınır değerler üzerinden aşırı örnekleme yapılır ancak bu yöntemde farklı olarak sınır değerlerin belirlenmesinde SVM algoritması kullanılır ve sentetik örnekler yalnızca karar sınırına yakın bölgelerde, yani sınıflandırmanın daha zor olduğu alanlarda üretilir. Bu sayede, modelin sınıflar arası ayrım gücü artırılarak genel doğruluk ve F1 skor gibi ölçütlerde iyileştirme sağlanabilir. Özellikle sınır sınıflarındaki örneklerin sınıflandırılmasının önemli olduğu durumlarda tercih edilmektedir.

2.11. SMOTE-ENN

Batista, Prati ve Monard [73] tarafından "A study of the behavior of several methods for balancing machine learning training data" isimli çalışmalarında önerilen SMOTE-ENN yöntemi, dengesiz veri kümeleriyle başa çıkmak için iki aşamalı bir yaklaşım sunar. İlk olarak, SMOTE algoritmasıyla azınlık sınıfına ait sentetik örnekler üretilerek sınıf dengesi sağlanır. Ardından, ENN (Edited Nearest Neighbors) yöntemi uygulanarak sınıf uyumsuzluğu gösteren örnekler veri kümesinden çıkarılır. Bu yöntem, hem sınıf dengesini geliştirir hem de gürültülü verilerin etkisini azaltarak sınıflandırma performansını artırır.

SMOTE-ENN algoritmasının adımları aşağıdaki gibidir:

Burada x_i azınlık sınıfından bir örnek, $x_i^{(k)}$, x_i'nin azınlık sınıfı içerisinden seçilen rasgele bir komşusu ve λ, 0 ile 1 arasında rasgele bir sayıdır.

1- Verilen bilgiler kapsamında ilk adım olarak SMOTE yöntemi ile aşağıdaki formül kullanılarak azınlık sınıfı için sentetik değerler üretilir:

$$x_{new} = x_i + \lambda(x_i^{(k)} - x_i)$$

Bu adım sonucunda SMOTE uygulanmış bir veri kümesi oluşur.

2- İkinci adımda ilk adımda oluşan SMOTE ile aşırı örneklenmiş veri kümesi üzerinde ENN algoritması uygulanır. Bu yöntemde veri kümesi tamamıyla ele alınır ve her bir örnek m en yakın komşusuna göre değerlendirilir. x, veri kümesinde bir örnek ve m onun komşuları olmak üzere eğer x değerinin sınıfı m tane komşusunun çoğunluğunun sınıfından farklı ise x değeri gürültü olarak tanımlanır ve veri kümesinden çıkarılır.

Bu yöntemde aykırı ve gürültülü örneklerin veri kümesinden çıkarılması, sınıflandırma yapılırken aşırı öğrenme (overfitting) probleminin engellenmesinde rol oynar. ENN algoritmasının filtreleme etkisiyle karar sınırlarında sınıfların birbirinden ayrılmasını sağlar. Bu sayede sınırları netleştirerek sınıflandırma algoritmalarının performansına olumlu yönde etki eder. Ayrıca, yalnızca sınıf sayısını artırmakla kalmayıp, aynı zamanda veri kümesinin tutarlılığını da iyileştirerek model eğitimine uygun bir yapı oluşturur. Bu avantajlara karşılık aykırı değer olarak veri kümesinden çıkarmalarda bilgi kaybının olması, kullanılacak komu sayısına aşırı duyarlı olması, SMOTE ve ENN algoritmalarının birlikte kullanımı gerektiren bir yöntem olması nedeniyle büyük veri kümelerinde hesaplama maliyeti ve zaman açısından sınırlayıcı olabilir.

2.12. K-Means SMOTE

KMeans-SMOTE algoritması, SMOTE yönteminin kümeleme temelli bir genişletmesidir ve Douzas ve Bacao tarafından 2018 yılında önerilmiştir [74]. Bu yöntemde, ilk olarak K-means algoritması kullanılarak azınlık sınıfı örnekleri anlamlı alt gruplara ayrılır. Daha sonra her bir küme için SMOTE uygulanarak sentetik örnekler üretilir. Bu yapı sayesinde, verinin iç yapısına daha duyarlı bir şekilde örnekleme yapılır ve daha dengeli bir veri kümesi oluşturulması sağlanır. KMeans-SMOTE, klasik SMOTE yöntemine kıyasla daha iyi genelleme performansı sunabilir.

K-Means SMOTE algoritmasının adımları genel olarak aşağıdaki gibidir:

1- Orijinal veri kümesinden azınlık sınıfı örnekleri ayrılarak X_{min} kümesi oluşturulur. Aşırı örnekleme işlemleri bu küme üzerinden yapılacaktır.

2- Azınlık sınıfa ait kümeler üzerinden K-Means kümeleme yapılır ve bu örnek kümesi k adet kümeye ayrılır ve her kümenin yoğunluk değeri p_i, örnek sayısı ve örneklerin merkez etrafında gruplanma düzeyine göre hesaplanır:

$$C = \{C_1, C_2, \dots, C_k\}$$

3- Her bir azınlık sınıfı kümesi C_1 için p_i değeri ve azınlık saflığı oranı π_i aşağıdaki formüller kullanılarak hesaplanır:

Bu hesaplamalar sonucunda elde edilen sonuçlar kullanılarak yoğunluk olarak önceden belirlenmiş p_{th} değerinin altında kalan ve saflık değeri $\pi_i \approx 1$ olan kümeler örnekleme işlemi için seçilir. Bu işlem SMOTE'un sınıf karar sınırlarında aykırı değer üretmesini engellemede etkilidir.

4- Bir önceki adımda yapılan eleme sonucunda örnekleme için seçilen uygun her küme C_i^* üzerinde SMOTE uygulanır. Bu kümelerin içerisindeki her bir azınlık örneği x için k en yakın komşu belirlenir. Bu komşulardan rasgele bir tanesi seçilir ve bu komşu ile interpolasyon yapılarak yeni bir örnek üretilir:

$$\mathrm{x_{new}} = \mathrm{x} + \lambda(\mathrm{x}_{nn} - \mathrm{x}), \qquad \lambda \sim \mathrm{U}(0,1)$$

Bu formüldeki x_{nn}, x'in k en yakın azınlık komşusundan biridir ve λ, 0 ile 1 aralığındaki rasgele bir sayıdır.

5- Son adım olarak tüm uygun kümeler için örnekleme yapılarak elde edilmiş sentetik örnekler kümesi (X_{syn}) orijinal veri kümesi ile birleştirilerek aşırı örneklenmiş veri kümesi (X') elde edilir;

$$X' = X \cup X_{syn}$$

Bu yöntem sayesinde aykırı alanlarda ve seyrek alanlarda örnekleme yapmamak üzere filtrelemeler ve koşullar uygulanıldığından sınıf sınırı belirsizliklerini ve aşırı öğrenmeyi azaltır. Örnekleme, kümelerin büyüklüklerine göre dağıtılır. Bu sayede sentetik örneklerin verideki doğal dağılıma uygun şekilde daha uyumlu yerleştirilmesini sağlayarak veri kümesinin kalitesini arttırır. Ancak, uygun küme sayısı değerinin (k) belirlenmesinde sorun yaşanırsa ya aşırı ayrıştırılmış ya da fazla genelleştirilmiş kümeler oluşabilir. Bu da örnekleme kalitesini olumsuz yönde etkileyecektir. Veri kümesi üzerinde kümeleme yaparken karşılaşılabilecek dengesiz küme yoğunluğu, çok küçük kümeler bulunması ve başarısız kümeleme gibi sorunlarda öğrenme sürecini amacının tam tersine olacak şekilde olumsuz etkileyebilir. Bu durumlar da yöntemin bilinen ve belirtilen dezavantajlarıdır.

2.13. SMOTETomek

SMOTE-Tomek algoritması ilk olarak Batista, Prati ve Monard [73] tarafından önerilmiştir. Azınlık sınıfı örneklerini çoğaltmak için SMOTE kkullanır ardından Tomek bağlantılarının temizlenmesi prensibine dayanır. Farklı sınıflara ait ve birbirine en yakın olan örnek çiftleri olan Tomek bağlantıları, genellikle sınıf sınırlarını bulanıklaştıran örneklerdir. Bu çiftlerin veri kümesinden çıkarılmasıyla daha net karar sınırları elde edilir. Böylece hem sınıf dengesi sağlanır hem de veri kümesi daha homojen bir hale getirilir.

SMOTETomek algoritmasının işleyişi aşağıdaki gibidir:

1- Klasik SMOTE kullanılır, burada k adet en yakın komşu bulunur ve azınlık sınıfına ait her bir örnek için rasgele seçilen bir komşu (x_i^{nn}) ile örnek (x_i) arasında doğrusal bir ara nokta rasgele seçilerek sentetik örnek üretilir:

$$x_{new} = x_i + \lambda(x_i^{nn} - x_i), \qquad \lambda \sim \mathrm{U}(0,1)$$

Bu işlem sınıf dağılımını dengelese de sınıf sınırlarında örtüşmeler olması muhtemeldir.

2- SMOTE işlemi sonrasında Tomek bağlantılarının tespiti bu adımda yapılır. Bir örnek çifti seçilir (x_i, x_j). Bu örnekler aşağıdaki koşulu sağlıyorsa Tomek bağlantısı olarak belirlenir:

$$x_i \in C_1, x_i \in C_2, C_1 \neq C_2$$

$$\mathrm{d}(x_i, x_j) = \min\{\mathrm{d}(x_i, x_k)\} \vee \mathrm{d}(x_i, x_j) = \min\{\mathrm{d}(x_j, x_k)\}$$

Bu koşula göre iki örnek hem birbirine en yakın örnekler hem de farklı sınıfların örnekleri ise, bu örnek çifti tomek bağlantısı olarak tanımlanır.

3- Bu adımda tomek bağlantısı olarak belirlenmiş örneklerin veri kümesinden çıkarılma işleminin nasıl yapılacağı seçilir;

- Sadece baskın sınıfa ait örneklerin veri kümesinden çıkarılması (çalışmada önerilen kullanım)
- Bağlantıyı oluşturan iki örneğin de veri kümesinden çıkarılması (agresif filtreleme)

4- Tomek bağlantısı olarak belirlenmiş örnekleri veri kümesinden çıkartma işlemi aşağıdaki gibidir:

$$X_{new} = X_{SMOTE} - \{x_j \in C_{maj} | (x_i, x_j)$$

Formüle göre örneğin veri kümesinden çıkarılması örneğin baskın sınıfta (C_{maj}) olup olmadığına bağlıdır. Çalışmada önerilen çalışma sistemine bağlı olarak bu formül değişiklik gösterebilecektir.

Bu yöntem, SMOTE adımı ile azınlık örnekleri arttırırken Tomek bağlantılarını temizleme adımı ile baskın sınıftaki gürültü oluşturabilecek örnekleri temizlerken kullanır. Kullanılan iki yöntem de kavramsal olarak kolay adımlar içermektedir. Ancak, tomek bağlantılarının temizlenmesinde her iki sınıftaki örneğin de veri kümesinden çıkarılması kararı verilirse bilgi kaybına yol açabilir. SMOTE uygulanması ve ardından her bir örnek için çift taraflı en yakın komşu hesaplamaları ile Tomek bağlantılarının elde edilmesi, özellikle büyük veri kümelerinde işlem maliyetini arttıracaktır. Belki de SMOTE aşırı örneklemesi ile sağlanan sınıf dengesi, Tomek bağlantılarının temizlenmesi adımında aşırı fazla çoğunluk örneği veri kümesinden çıkarılması halinde tekrar bozulabilir. Bu dezavantajlar giderilerek yöntemin başarısı arttırılabilir.

2.14. ADASYN

ADASYN algoritması, He ve meslektaşları tarafından 2008 yılında geliştirilmiştir [75]. Bu yöntem, sentetik örnek üretim sürecini adaptif bir yapıda gerçekleştirir; yani öğrenilmesi zor olan örneklere daha fazla sentetik veri üretilirken, kolay örneklerde daha az üretim yapılır. Böylece sınıflandırma algoritmaları, karar sınırına yakın karmaşık bölgelerde daha iyi genelleme yapabilir hale gelir. ADASYN, özellikle çok dengesiz veri kümelerinde etkili sonuçlar vermektedir.

ADASYN Algoritması aşağıdaki gibi çalışır:

D_{tr} , m örnekli bir veri kümesi olsun

m_s: Azınlık sınıfın örnek sayısı

m_l: Çoğunluk sınıfın örnek sayısı buna bağlı olarak

$m_s \leq m_l$ ve $m_s + m_l = m$

$\{x_i, y_i\}, i = 1, 2, \ldots, m$ olarak tanımlansın ve x_i , n boyutlu bir X örnek uzayının içerisinde bir örnek ve

$y_i \in Y = \{1, -1\}$ ise x_i ile bağlantılı bir sınıf etiketi olmak üzere,

1- Veri kümesindeki sınıf dengesizliği (d) tanımlanarak başlanır:

$$d = \frac{m_s}{m_l}$$

Yukarıdaki formül ile hesaplanan sınıf dengesizliği önceden belirlenmiş eşik değerin (d_{th}) altında ise işlemlere devam edilir.

$$d \leq d_{th}$$

2- Azınlık sınıfı için üretilecek sentetik örnek sayısının hesaplanması aşağıdaki gibi yapılır:

$$G = (m_l - m_s) \times \beta$$

Burada $\beta \in [0,1]$, değeri sentetik örnek üretimi sonrası istenen sınıf dengesini belirtmek için kullanılır. $\beta = 1$ örnek üretimi işlemi sonrasında tamamen dengeli bir veri elde edileceği anlamına gelmektedir.

3- Azınlık sınıfı örneklerine ait sınıflandırma zorluk derecesi (r_i) hesaplama:

$$r_i = \frac{\Delta_i}{K}, i = 1,2,3, \dots, m_s$$

Burada Δ_i , azınlık sınıfı örneği olan x_i'ye ait K en yakın komşudan baskın sınıfa ait olanların sayısıdır buna bağlı olarak $r_i \in [0,1]$ zorluk derecesi değerlerini normalleştirir:

$$\widehat{r_i} = {r_i}\Big/{\sum_{i=1}^{m_s} r_i}$$

Buradaki kısıt ise aşağıdaki gibidir:

$$\sum_i \widehat{r_i} = 1$$

4- Normalleştirilmiş zorluk derecesi ($\widehat{r_i}$) üzerinden her bir azınlık sınıf örneği için üretilmesi gereken sentetik örnek sayısının hesaplanması aşağıdaki gibi yapılır:

$$g_i = \widehat{r_i} \times G$$

Burada G 2. Adımda hesaplanan azınlık sınıf için toplam üretilmesi gereken sentetik örnek sayısıdır.

5- Sentetik örnek hesaplama adımı:

Aşağıdaki adımlar her bir azınlık sınıfı örneği x_i için uygulanarak azınlık sınıf için sentetik örnek üretilecektir:

- x_i, azınlık sınıfı örneğinin K en yakın komşuları arasından rastgele bir azınlık sınıfı örneği x_{zi} seçilir.
- Aşağıdaki formül kullanılarak sentetik örnek üretilir:

$$s_i = x_i + (x_{zi} - x_i) \times \lambda$$

Burada $(x_{zi} - x_i)$ n boyutlu uzayda bir fark vektörüdür ve λ ise 0 ile 1 arasında bir rasgele sayıdır.

5. adımdaki sentetik örnek üretim adımları her bir örnek x_i için g_i kez tekrarlanarak bütün azınlık sınıf örnekleri için gerekli sayıda örnek üretilecektir.

Bu yöntemde, klasik SMOTE tekniğine göre daha dengeli ve orijinal verinin yapısına uygun aşırı örnekleme sağlar. Her azınlık örneği için üretilecek sentetik veri miktarı, o örneğin sınıflandırılma zorluğuna göre belirlenir. Buna bağlı olarak öğrenmesi kolay örneklere daha az odaklanarak verimlilik sağlar ve dengesizliğe daha hassas çözümler sunarak aşırı öğrenmenin önüne geçmeyi amaçlar. Diğer taraftan ise, veri kümesinin aykırılıkları, sınıf sınırı olarak algılanabilir ve örnek oluşturulurken yanlış yere odaklanılmasına sebep olabilir. Ek olarak, üretilecek örnek sayısını doğrudan etkileyen β ve komşu sayısı K gibi parametrelerin belirlenilmesinde hata yapılmasına bağlı olarak karar sınırlarının bozulması ve sınıflarda yığılmaya yol açılabilme gibi dezavantajlara sahiptir.

2.15. Algoritmaların Performansları için Kullanılan Karşılaştırma Ölçütleri

Sınıflandırma problemlerinin değerlendirilmesinde kullanılan performans ölçütleri, istatistiksel karar kuramı ve bilgi alma (information retrieval) alanlarındaki erken dönem çalışmalara dayanmaktadır. Bu ölçütler, sınıfların gerçek ve tahminlerinin çapraz tablosu üzerinden elde edilen değerlerle hesaplanabilir. Genel bir sınıflandırmada performans ölçümü için iki sınıflı durum için Çizelge 1'deki tablo örnek olarak verilebilir. Bu örnekdeki TP, TN, FP ve FN ölçümleri sırasıyla;

- TP(doğru pozitif): Tahmin sonucu ve gerçekleşen değerin pozitif olduğu örneklerin sayısı
- TN (Doğru negatif): Tahmin sonucu ve gerçekleşen değerin negatif olduğu örneklerin sayısı

- FP (Yanlış pozitif): Tahmin değerinin pozitif olduğu ancak gerçekleşen değerin negatif olduğu örneklerin sayısı
- FN (Yanlış negatif): Tahmin edilen değerin negatif ancak gerçekleşen değerin pozitif olduğu örneklerin sayısı

Çizelge 1: Performans Değerlendirmesi için Karmaşıklık Matrisi

	Tahmin edilen değer	
Gerçekleşen değer	TP (True Positive)	TN (True Negative)
	FP (False Positive)	FN (False Negative)

2.15.1. Doğruluk (Accuracy)

Doğruluk (Accuracy), sınıflandırma modellerinin genel doğruluğunu değerlendirmek amacıyla kullanılan en temel performans ölçütlerinden biridir. Modelin doğru tahmin ettiği örneklerin, toplam örnek sayısına oranı olarak hesaplanır:

$$Doğruluk = \frac{TN + TP}{TP + TN + FP + FN}$$

Özellikle dengeli veri kümelerinde etkili bir ölçüt olarak kullanılsa da dengesiz sınıf dağılımlarında yanıltıcı sonuçlar verebilir. İlk olarak istatistiksel karar kuramı çerçevesinde tanımlanan bu ölçüt, makine öğrenmesi algoritmalarının değerlendirilmesinde uzun süredir standart bir ölçüt olarak kullanılmaktadır.

2.15.2. Kesinlik (Precision)

Kesinlik (Precision), bir modelin pozitif olarak sınıflandırdığı örnekler arasında gerçekten pozitif olanların oranını gösterir. Özellikle yanlış pozitif sonuçların maliyetinin yüksek olduğu alanlarda (örneğin, hastalık teşhisi veya sahtekârlık tespiti) kritik bir ölçüt olarak ön plana çıkar. Kesinlik ölçütü, ilk olarak bilgi erişim alanındaki çalışmalarla tanımlanmış ve daha sonra sınıflandırma sistemlerine entegre edilmiştir.

$$Kesinlik = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FP}}$$

2.15.3. Duyarlılık (Recall)

Duyarlılık (Recall), gerçek pozitif örneklerin ne kadarının doğru şekilde pozitif olarak sınıflandırıldığını gösteren bir ölçüttür. Duyarlılık özellikle kaçırılmaması gereken sınıfların olduğu uygulamalarda (örneğin, kanserli hastaların saptanması) önemli bir role sahiptir. Kesinlik gibi, duyarlılık da bilgi erişim alanından sınıflandırma problemlerine uyarlanmıştır.

$$Duyarlılık = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FN}}$$

2.15.4. F1 Skoru

F1 skoru, kesinlik ve duyarlılık arasındaki dengeyi sağlamak için geliştirilmiş harmonik ortalama tabanlı bir ölçüttür. Kesinlik ve duyarlılık değerlerinden biri düşükse F1 skoru da düşük olur. Van Rijsbergen [76] tarafından bilgi erişim performansının değerlendirilmesinde önerilen bu ölçüt, özellikle dengesiz veri kümelerinde yaygın olarak kullanılmaktadır. Günümüzde sınıflandırma modellerinin genel performansını özetlemek için sıklıkla tercih edilmektedir.

$$F1 = 2 \times \frac{\mathrm{Kesinlik} \times \mathrm{Duyarlılık}}{\mathrm{Kesinlik} + \mathrm{Duyarlılık}}$$

2.15.5. AUC-ROC Eğrisi

ROC eğrisi, bir modelin farklı eşik değerlerinde duyarlılık (True Positive Rate) ve özgüllük (False Positive Rate) arasındaki ilişkiyi gösterir. Bu eğrinin altındaki alan (AUC), modelin genel ayrım gücünü ölçer. Fawcett [77] tarafından yapılan kapsamlı analizlerle AUC metriği, özellikle dengesiz sınıflarda model karşılaştırmaları için önerilmiştir. AUC, eşik bağımsız olması sayesinde birçok uygulamada güvenilir karşılaştırma sağlar.

2.15.6. Karmaşıklık Matrisi (Confusion Matrix)

Karmaşıklık Matrisi, modelin tahmin performansını detaylı bir şekilde gösteren bir tablodur. TP, FP, TN ve FN gibi sınıflama sonuçlarını ayrı ayrı sunar. Bu matris, modellerin hangi tür hataları yaptığını analiz etmede oldukça faydalıdır. İstatistik ve makine öğrenmesi literatüründe erken dönemden beri kullanılmakta olup günümüzde model analizi için temel araçlardan biridir.

3. YÖNTEM

3.1. Araştırma Yöntemi ve Tasarımı

Bu çalışmada kullanılan veri kümesi üzerinden sınıflandırma algoritması ve SMOTE tekniği ikililerinden hangi ikilinin daha verimli olduğunu araştırabilmek amacıyla aynı veri kümesi üzerinde 4 farklı sınıflandırma algoritmasının test edildiği bir alan oluşturulmuş ve bu alan sabit tutularak sadece SMOTE tabanlı yöntemler kullanılarak sınıflandırma ölçütleri incelenmiştir.

3.2. Veri Kümesi

Bu çalışmada kullanılan veri kümesi, Kaggle platformunda " Stroke Pridiction Dataset" adıyla yayımlanmış ve Samuel Taiwo Grace tarafından paylaşılmıştır [43]. Veri kümesi, bireylerin demografik, medikal ve yaşam tarzı bilgilerinden yola çıkarak inme (stroke) riskini tahmin etmeye yönelik bir sınıflandırma problemi içermektedir.

Veri kümesi, toplam 5110 bireye ait gözlem içermektedir. Bu gözlemlerde, 11 adet bağımsız değişken ve 1 adet bağımlı değişken (hedef değişken) yer almaktadır. Bağımlı değişken olan `stroke`, bireyin geçmişte inme geçirip geçirmediğini belirtmekte olup 0 (hayır) ve 1 (evet) olmak üzere iki sınıftan oluşan bir ikili sınıflandırma problemini tanımlamaktadır. Veri kümesinde kullanılan bazı temel özellikler aşağıda özetlenmiştir:

- gender: Bireyin cinsiyeti (Male, Female, Other)
- age: Yaş bilgisi
- hypertension: Hipertansiyon durumu (0: Yok, 1: Var)
- heart_disease: Kalp hastalığı geçmişi (0: Yok, 1: Var)

- ever_married: Hiç evlenip evlenmediği (Yes/No)
- work_type: Meslek türü (Private, Self-employed, children, etc.)
- Residence_type: Yaşanılan bölge tipi (Urban/Rural)
- avg_glucose_level: Ortalama glikoz seviyesi
- bmi: Beden kitle indeksi
- smoking_status: Sigara içme durumu (never smoked, formerly smoked, smokes, unknown)
- stroke: Bağımlı değişken, bireyin inme geçmişi (0: Yok, 1:Var)

Veri kümesinde bazı eksik değerler bulunmaktadır özellikle `bmi` değişkeninde yer almakta olan bu eksik değerler veri önişleme süreçlerinde çoklu imputasyon ile doldurulmuştur. Ayrıca, sınıflar arasında ciddi bir dengesizlik gözlenmiştir. Örneğin, `stroke=1` (inme geçiren birey) sınıfı toplam örneklerin yaklaşık %5'ini oluşturmaktadır. Bu nedenle, sınıflandırma algoritmalarının farklı SMOTE teknikleri ile kullanıldığında etkisinin gözlemlenmesi için uygun bir veri kümesidir.

Bu veri kümesi hem tıbbi veri analizi hem de dengesiz sınıflandırma problemleri açısından literatürde yaygın olarak kullanılan bir örnek olup, çalışmanın sınıflandırma algoritmalarının ve SMOTE tekniklerinin birlikte kullanımının değerlendirilmesine uygun bir alan sağlamaktadır.

3.3. Kodlama ve Karşılaştırma Süreci

Bu çalışma kapsamında kodlama yapılırken Python kodlama dili kullanılmış ve Visual Studio Code üzerinden Jupyter notebook eklentisinden faydalanılmıştır. Kodlama yapılırken ikili senaryolara ait verimlilik üzerinde SMOTE tekniği, sınıflandırma algoritmaları ve çalışma kapsamında incelenen algoritmaların doğası gereği bulunan rasgelelik etkeni dışında bir etken olmayacaktır.

4. BULGULAR VE ANALİZLER

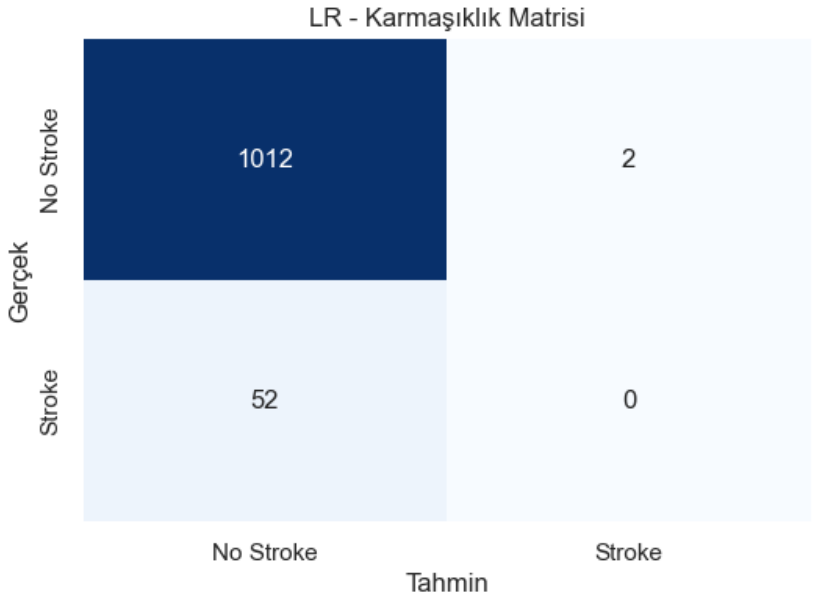
Çalışmada bahsedilen sınıflandırma algoritmaları, SMOTE teknikleri ve performans ölçütleri kullanılarak bu kısımda ikililer karşılaştırılacak ve hangi ikilinin daha başarılı sınıflandırma yaptığı tartışılacaktır. Dengesiz veri problemi ile ortaya çıkan tüm sorunlar ve bu sorunlara karşılık gelen öneri algoritmaların karşılaştırılması için izlenen yol, öncelikle dengesiz sınıf problemi ile ilgili hiçbir iyileştirme yapılmadan sınıflandırma yöntemleriyle elde edilen sonuçların incelenmesidir. Elde edilen bu sonuçlarla dengesiz veri kümesinde sınıflandırma algoritmalarının başarısızlık düzeyleri incelenerek yorumlanmıştır. Daha sonra da dengesiz veri problemi giderilerek aynı algoritmaların sonuçları incelenerek elde edilen gelişme ve diğer sınıflandırma algoritmalarıyla elde edilen sonuçlar değerlendirilmiştir. Ayrıca tüm bu ikili durumlar için genel değerlendirme ile bulgu ve analiz bölümü tamamlanmıştır. Bu süreçte aynı veri kümesi için farklı yöntemlerle oluşturulan tasarımda toplam 32 farklı sınıflandırma uygulamış ve elde edilen sonuçlar test kümesi üzerinden yorumlanmıştır. Tüm sonuçların incelenmesi eğitim kümesinde elde edilen modelin test kümesinde değerlendirilmesi sonucu oluşan karmaşıklık matrisi ve diğer ölçütlerle yapılmıştır.

4.1. Sınıflandırma Algoritmalarının Orijinal Veri Kümesi Üzerindeki Performans Sonuçları

Çalışmanın bu kısmında sınıflandırma algoritmalarının orijinal veri üzerindeki performansları incelenecek ve yukarıda bahsedilen performans ölçütleri kullanılarak dengesiz veri üzerindeki başarısız sonuçları eleştirilecektir. Bu sayede daha sonraki bölümlerde kullanacağımız aşırı örnekleme algoritmalarının etkilerinin daha iyi gözlemlenebilmesi amaçlanmaktadır.

Orijinal veri kümesi üzerindeki sınıflandırma performansları üzerinde değerlendirme yapılırken göz önünde bulundurulmalıdır ki performans ölçütleri incelendiğinde bazı değerler 0 olarak hesaplanmıştır. Performans ölçütlerinin formüllerinden kaynaklanan bu sonuca modelin tahmin çıktısında Doğru Pozitif tahmin sayısının sıfır olması sebep olmaktadır. Yani dengesiz veri kümesinde gözlem sayısı çok olan sınıfı doğru sınıflandırmayı genel doğruluk değeri için daha öncelikli gören optimizasyon problemleri, gözlem sayısı az olan azınlık sınıfını hiç doğru sınıflandıramayarak bazı ölçütlerin sıfır hesaplanmasına neden olmaktadır.

LR kullanılarak yapılan deneme sonucunda elde edilen performans ölçütleri için karmaşıklık matrisi ve ölçüt sonuçları sırasıyla Şekil 2 ve Çizelge 2'de verilmiştir. Lojistik regresyon modelinin orijinal veri üzerinde, İnme geçmişi olmayan bireyleri %95 kesinlik ile tahmin edebildiği görülmüştür. Buna rağmen çalışmanın ve veri kümesinin doğrudan ilgi alanında olan azınlık sınıfını tahmin etmede başarısız olduğundan modelin bu veri kümesi üzerinde tek başına başarısız olduğunu söylemek mümkündür.



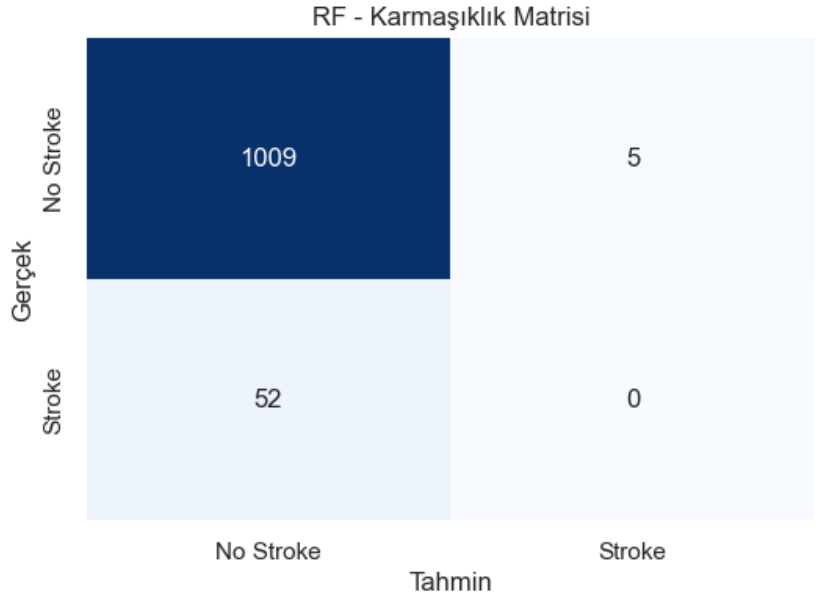
Şekil 2: Orijinal Veri Kümesi Üzerinde Yapılan LR Modellemesine Ait Karmaşıklık Matrisi.

Çizelge 2: Orijinal Veri Kümesi Üzerinde Yapılan LR Modellemesine Ait Sınıflandırma Raporu

Sınıf	Doğruluk (Accuracy)	Kesinlik (Precision)	Duyarlılık (Recall)	F1 Skoru	Gerçek Örnek Sayısı (support)
0		0.95	1.00	0.97	1014
1		0.00	0.00	0.00	52
Model	0.949		0.0	0.0	

RF modelinin de LR modelinde olduğu gibi orijinal veri üzerinde inme geçmişi olmayan bireyleri %95 kesinlik ile tahmin edebildiği görülmüştür. Buna rağmen çalışmanın ve veri kümesinin doğrudan ilgi alanında olan azınlık sınıfını tahmin etmede başarısız

olduğundan modelin bu veri kümesi üzerinde tek başına başarısız olduğunu söylemek mümkündür. İlgili matris ve ölçütler sırasıyla Şekil 3 ve Çizelge 3'te verilmiştir.

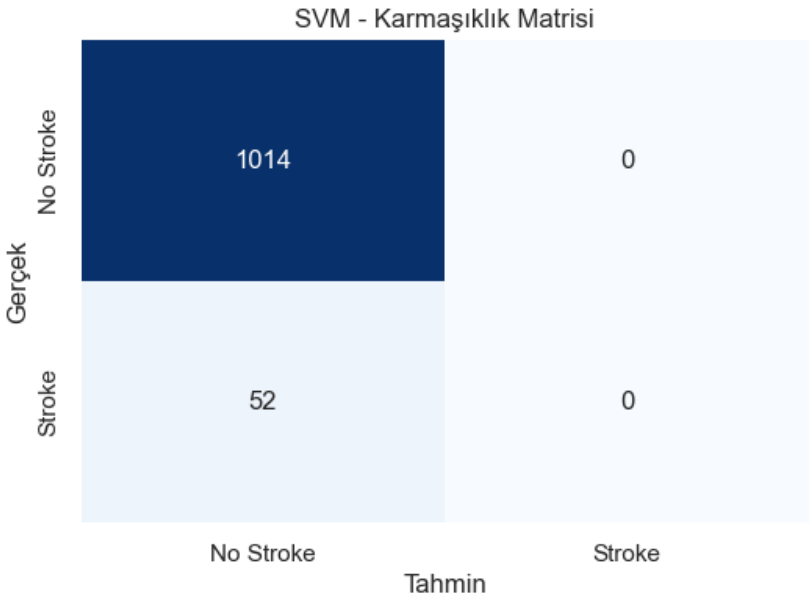


Şekil 3: Orijinal Veri Kümesi Üzerinde Yapılan RF Modellemesine Ait Karmaşıklık Matrisi.

Çizelge 3: Orijinal Veri Kümesi Üzerinde Yapılan RF Modellemesine Ait Sınıflandırma Raporu

Sınıf	Doğruluk (Accuracy)	Kesinlik (Precision)	Duyarlılık (Recall)	F1 Skoru	Gerçek Örnek Sayısı (support)
0		0.95	1.00	0.97	1014
1		0.00	0.00	0.00	52
Model	0.946		0.0	0.0	

SVM modelinin de diğer modellerde olduğu gibi orijinal veri üzerinde inme geçmişi olmayan bireyleri %95 kesinlik ile tahmin edebildiği görülmüştür. Buna rağmen çalışmanın ve veri kümesinin doğrudan ilgi alanında olan azınlık sınıfını tahmin etmede başarısız olduğundan modelin bu veri kümesi üzerinde tek başına başarısız olduğunu söylemek mümkündür. İlgili bilgiler Şekil 4 ve Çizelge 4'tedir.

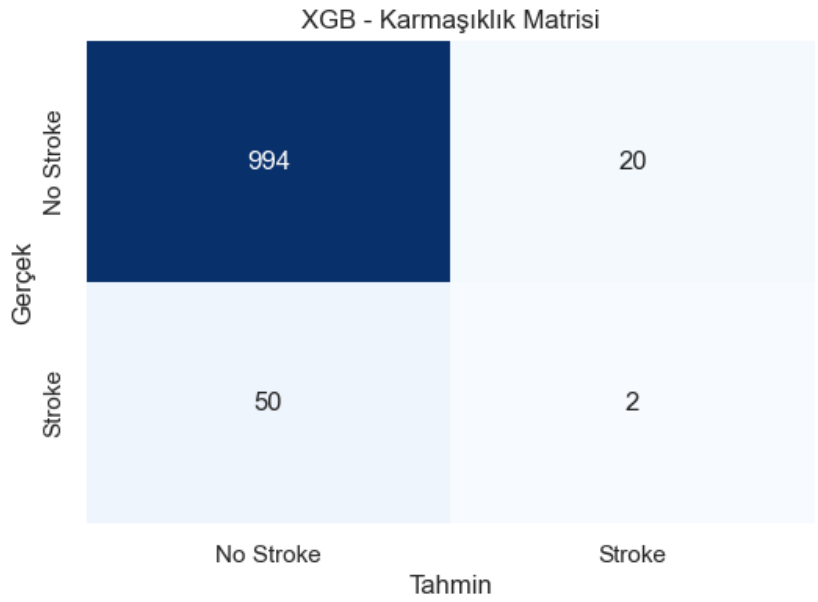


Şekil 4: Orijinal Veri Kümesi Üzerinde Yapılan SVM Modellemesine Ait Karmaşıklık Matrisi.

Çizelge 4: Orijinal Veri Kümesi Üzerinde Yapılan SVM Modellemesine Ait Sınıflandırma Raporu

Sınıf	Doğruluk (Accuracy)	Kesinlik (Precision)	Duyarlılık (Recall)	F1 Skoru	Gerçek Örnek Sayısı (support)
0		0.95	1.00	0.97	1014
1		0.00	0.00	0.00	52
Model	0.951		0.0	0.0	

XGB modelinin de diğer modellerde olduğu gibi orijinal veri üzerinde inme geçmişi olmayan bireyleri %95 kesinlik ile tahmin edebildiği görülmüştür. Bununla beraber XGB modeli tahminlemede diğer modellerden farklı olarak pozitif değer tahminlemesini doğru yapabilmiştir. Buna rağmen çalışmanın ve veri kümesinin doğrudan ilgi alanında olan azınlık sınıfını tahmin etmede başarısız olduğundan modelin bu veri kümesi üzerinde tek başına başarısız olduğunu söylemek mümkündür. XGB karmaşıklık matrisi ve ölçüt bilgileri sırasıyla Şekil 5 ve Çizelge 5'te verilmiştir.



Şekil 5: Orijinal Veri Kümesi Üzerinde Yapılan XGB Modellemesine Ait Karmaşıklık Matrisi.

Çizelge 5: Orijinal Veri Kümesi Üzerinde Yapılan XGB Modellemesine Ait Sınıflandırma Raporu

Sınıf	Doğruluk (Accuracy)	Kesinlik (Precision)	Duyarlılık (Recall)	F1 Skoru	Gerçek Örnek Sayısı (support)
0		0.95	1.00	0.97	1014
1		0.09	0.04	0.05	52
Model	0.934		0.038	0.054	

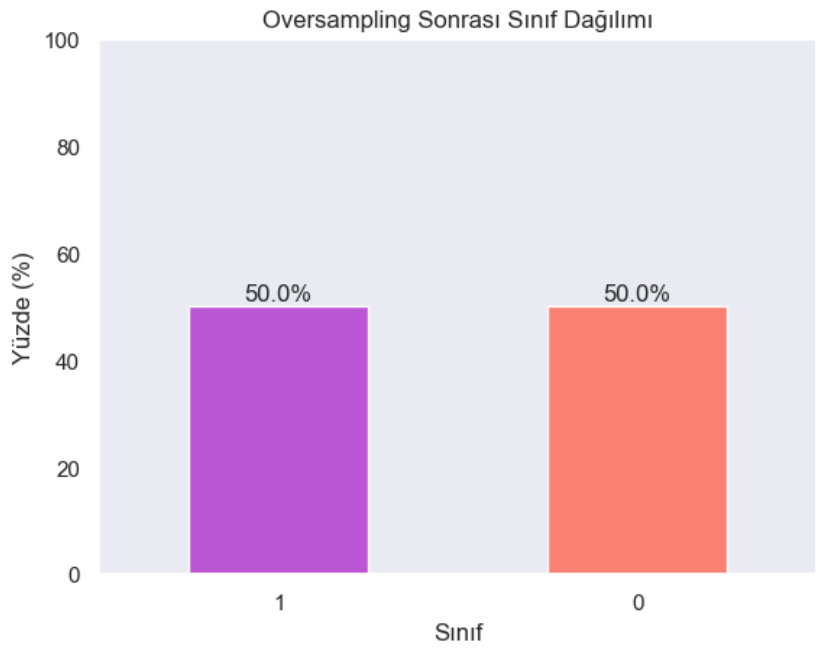
Orijinal veri seti üzerinde yapılan modellemelerin performans ölçütleri incelendiğinde; incelenen dört uygulamadan üçünün hiç TP değer vermediğini ve bir modelin ise yalnızca 2 TP değeri verdiği göze çarpmaktadır. Buna dayanarak modellerin tek başında azınlık sınıfı tahminlemede başarısız olduğu söylenebilir. Bu sonuçlar göz önünde bulundurulduğunda, modellerin tahminlemede kullanılmasının bir çözüm olmayacağını söylemek mümkündür. Buradan yola çıkarak ilgili veri kümesi üzerinde sınıflandırma modellemesi yapılmak isteniyor ise bir aşırı örnekleme yöntemine ihtiyaç duyulduğunu söylemek mümkündür.

4.2. Aşırı Örnekleme Algoritmaları ile Elde Edilen Sonuçlar

Orijinal veri kümesi üzerinde performansını gördüğümüz sınıflandırma algoritmalarının çalışmada bahsedilen aşırı öğrenme algoritmaları ile kullanıldığındaki performansları her bir SMOTE tabanlı algoritma için ayrı ayrı incelenmiş ve daha sonra genel olarak yorumlanmıştır.

4.3. Sınıflandırma Algoritmalarının Temel SMOTE ile Dengelenmiş Veri Kümesi Üzerindeki Performans Sonuçları

Önceki bölümde orijinal veri kümesi üzerinde Temel SMOTE algoritması uygulanarak sınıf eşitlemesi yapıldığında sınıf dağılımları Şekil 6'daki gibi olmuştur. Sınıflandırma algoritmalarının performansları Çizelge 6'daki gibidir. Temel SMOTE ile bağımlı değişkene ait sınıfları tekrar düzenlenmiş veri kümesi üzerinde yapılan modelleme sonucu ortaya çıkan performans ölçütleri incelendiğinde, XGB algoritması ile yapılan sınıflandırmanın diğerlerinden daha başarılı olduğunu söylemek mümkündür. Bunun yanında modellere ait sınıflandırma raporu incelendiğinde RF algoritmasının sınıflandırmada XGB algoritmasına alternatif oluşturabileceğini söylemek mümkündür.



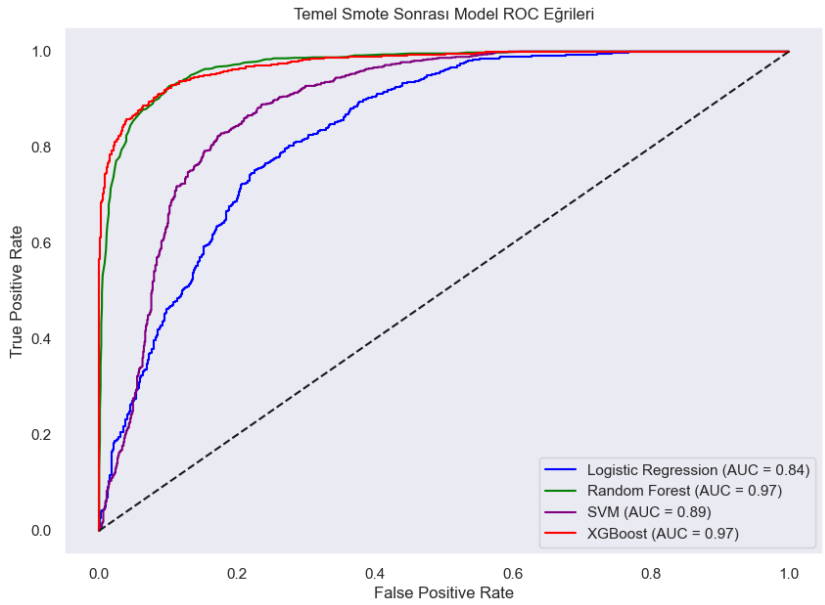
Şekil 6: Temel SMOTE Uygulaması Sonrası Bağımlı Değişkende Sınıfların Dağılımı

Çizelge 6: Temel SMOTE Uygulaması Sonrası Yapılan LR, RF, SVM ve XGB Modellerine Ait Sınıflandırma Raporu

	Sınıf	Doğruluk (Accuracy)	Kesinlik (Precision)	Duyarlılık (Recall)	F1 Skoru	Gerçek Örnek Sayısı (support)
LR	0		0.79	0.73	0.76	1020
	1		0.76	0.80	0.77	990
	Model	0.763		0.800	0.769	
	Sınıf	Doğruluk (Accuracy)	Kesinlik (Precision)	Duyarlılık (Recall)	F1 Skoru	Gerçek Örnek Sayısı (support)
RF	0		0.95	0.86	0.90	1020
	1		0.87	0.95	0.91	990
	Model	0.907		0.954	0.910	
	Sınıf	Doğruluk (Accuracy)	Kesinlik (Precision)	Duyarlılık (Recall)	F1 Skoru	Gerçek Örnek Sayısı (support)
SVM	0		0.89	0.74	0.81	1020
	1		0.77	0.91	0.83	990
	Model	0.819		0.905	0.832	
	Sınıf	Doğruluk (Accuracy)	Kesinlik (Precision)	Duyarlılık (Recall)	F1 Skoru	Gerçek Örnek Sayısı (support)
XGB	0		0.93	0.89GB	0.91	1020
	1		0.89	0.93	0.91	990
	Model	0.910		0.932	0.911	

Şekil 7'de verilen ROC eğrileri ve AUC değerleri incelendiğinde de sınıflandırma raporları ile benzer sonuçlar elde edilmiştir. SMOTE ile dengelenmiş veri üzerinde performansları test edilen 4 algoritma içerisinden XGB ve RF algoritmalarının, ilgili veri kümesi üzerinde en başarılı sınıflandırma algoritmalar olduğu gözlemlenmiştir. Bunun

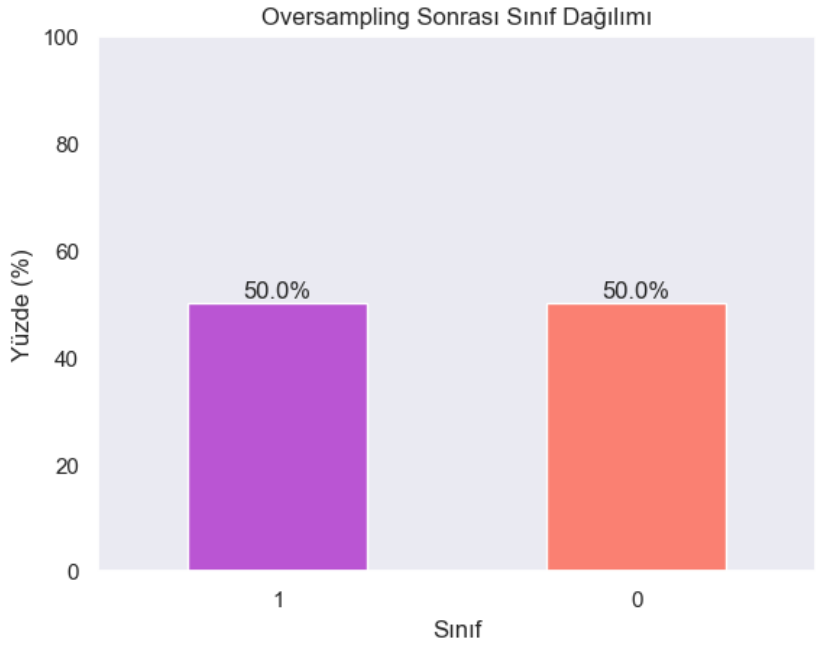
yanında bir seçim yapılması gerekirse XGB algoritmasının seçilmesi gerektiği ROC eğrisinde öne çıkmasıyla da görülebilmektedir.



Şekil 7: Temel SMOTE Uygulanmış Veri Kümesi Kullanılarak Kurulan Modellerin ROC ve AUC Değerleri

4.4. Sınıflandırma Algoritmalarının Borderline-SMOTE ile Dengelenmiş Veri Kümesi Üzerindeki Performans Sonuçları

Orijinal veri kümesi üzerinde Borderline SMOTE algoritması uygulanarak sınıf eşitlemesi yapıldığında sınıf dağılımları Şekil 7'dedir ve sınıflandırma algoritmalarının performansları Çizelge 7'de verilmiştir. Borderline-SMOTE ile bağımlı değişkene ait sınıfları tekrar düzenlenmiş veri kümesi üzerinde yapılan modelleme sonucu XGB algoritması ile yapılan sınıflandırmanın diğerlerinden daha başarılı olduğunu söylemek F1 skoruna ve doğruluk değerine bakarak mümkündür. Buna rağmen RF algoritması burada da öne çıkarak diğer değerlerde verdiği yakın değerler ve duyarlılık değerinde XGB algoritmasından daha iyi sonuçlar verdiği görülmüştür. Modellere ait sınıflandırma raporu incelendiğinde Rastgele ormanlar algoritmasının sınıflandırmada XGB algoritmasına alternatif oluşturabileceğini orijinal verideki azınlık sınıf olan 1 sınıfı üzerindeki duyarlılık performansına bakarak söylemek mümkündür.



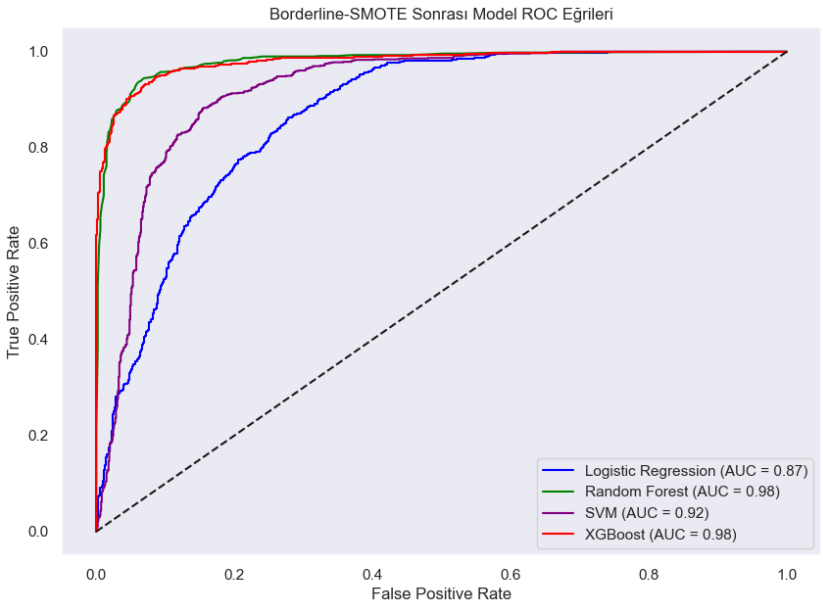
Şekil 8: Borderline-SMOTE Uygulaması Sonrası Bağımlı Değişkende Sınıfların Dağılımı

Çizelge 7: Borderline-SMOTE Uygulaması Sonrası Yapılan LR,RF,SVM ve XGB Modellerine Ait Sınıflandırma Raporu

	Sınıf	Doğruluk (Accuracy)	Kesinlik (Precision)	Duyarlılık (Recall)	F1 Skoru	Gerçek Örnek Sayısı (support)
LR	0		0.82	0.75	0.78	1020
	1		0.76	0.83	0.79	990
	Model	0.787		0.829	0.793	
	Sınıf	Doğruluk (Accuracy)	Kesinlik (Precision)	Duyarlılık (Recall)	F1 Skoru	Gerçek Örnek Sayısı (support)
RF	0		0.96	0.89	0.92	1020
	1		0.89	0.96	0.93	990
	Model	0.924		0.954	0.910	

	Sınıf	Doğruluk (Accuracy)	Kesinlik (Precision)	Duyarlılık (Recall)	F1 Skoru	Gerçek Örnek Sayısı (support)
SVM	0		0.91	0.79	0.84	1020
	1		0.81	0.92	0.86	990
	Model	0.851		0.917	0.858	
	Sınıf	Doğruluk (Accuracy)	Kesinlik (Precision)	Duyarlılık (Recall)	F1 Skoru	Gerçek Örnek Sayısı (support)
XGB	0		0.95	0.91	0.93	1020
	1		0.91	0.95	0.93	990
	Model	0.930		0.945	0.930	

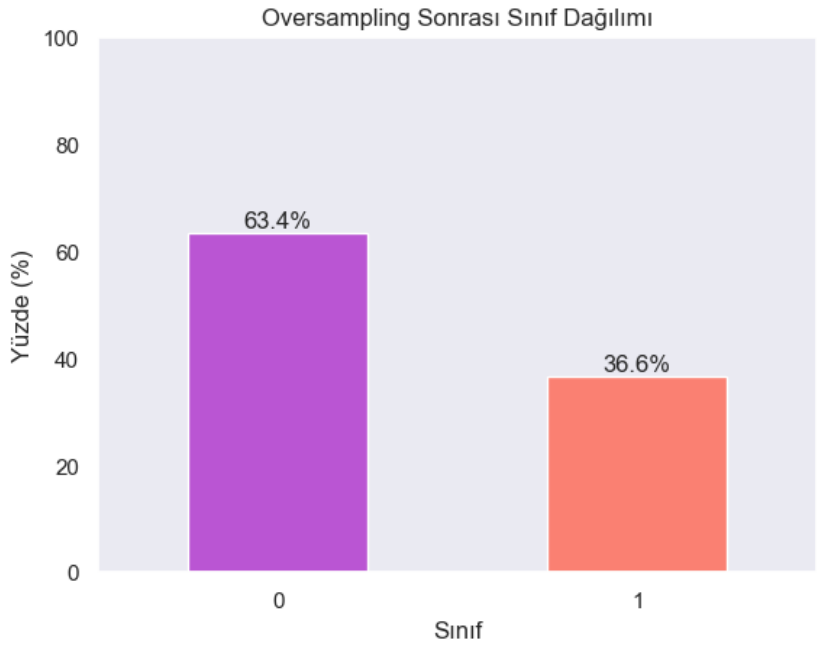
Şekil 9'da incelenen ROC eğrileri ve AUC değerleri yorumlandığında sınıflandırma raporları ile beklenildiği gibi benzer sonuçlar vermiş ve Borderline-SMOTE ile dengelenmiş veri üzerinde performansları test edilen 4 algoritma içerisinden XGB ve RF algoritmalarının, ilgili veri kümesi üzerinde en başarılı sınıflandırma algoritmalar olduğu gözlemlenmiştir.



Şekil 9: Borderline-SMOTE Uygulanmış Veri Kümesi Kullanılarak Kurulan Modellerin ROC ve AUC Değerleri

4.5. Sınıflandırma Algoritmalarının SVM-SMOTE ile Dengelenmiş Veri Kümesi Üzerindeki Performans Sonuçları

Orijinal veri kümesi üzerinde SVM-SMOTE algoritması uygulanarak sınıf eşitlemesi yapıldığında sınıf dağılımları Şekil 10'daki gibi elde edilmiştir ve sınıflandırma algoritmalarının performansları Çizelge 8'de verilmiştir. Diğer SMOTE tekniklerine göre farklı olarak burada sınıf dağılım oranlarının değiştiğini görmekteyiz. SVM-SMOTE ile bağımlı değişkene ait sınıfları tekrar düzenlenmiş veri kümesi üzerinde yapılan modelleme sonucu ortaya çıkan performans ölçütleri sonucunda RF algoritması ile yapılan sınıflandırmanın diğerlerinden daha başarılı olduğunu iki tablodaki değerlerin çoğunda en büyük değer vermesine bakarak söylemek mümkündür. Bunun yanında modellere ait sınıflandırma raporu incelendiğinde RF algoritmasının bir alternatifi olarak XGB algoritmasına alternatif oluşturabileceğini söylenebilir.

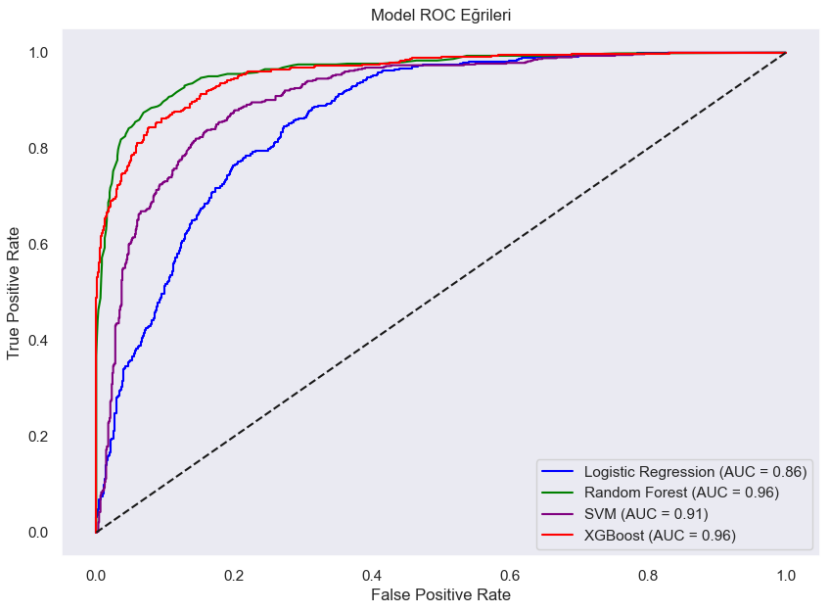


Şekil 10: SVM-SMOTE Uygulaması Sonrası Bağımlı Değişkende Sınıfların Dağılımı

Çizelge 8: SVM-SMOTE Uygulaması Sonrası Yapılan LR,RF,SVM ve XGB Modellerine Ait Sınıflandırma Raporu

	Sınıf	Doğruluk (Accuracy)	Kesinlik (Precision)	Duyarlılık (Recall)	F1 Skoru	Gerçek Örnek Sayısı (support)
LR	0		0.82	0.83	0.83	977
	1		0.72	0.72	0.72	608
	Model	0.785		0.717	0.719	
	Sınıf	Doğruluk (Accuracy)	Kesinlik (Precision)	Duyarlılık (Recall)	F1 Skoru	Gerçek Örnek Sayısı (support)
RF	0		0.93	0.91	0.92	977
	1		0.86	0.89	0.87	608
	Model	0.902		0.888	0.874	
	Sınıf	Doğruluk (Accuracy)	Kesinlik (Precision)	Duyarlılık (Recall)	F1 Skoru	Gerçek Örnek Sayısı (support)
SVM	0		0.89	0.84	0.87	977
	1		0.77	0.83	0.80	608
	Model	0.839		0.832	0.799	
	Sınıf	Doğruluk (Accuracy)	Kesinlik (Precision)	Duyarlılık (Recall)	F1 Skoru	Gerçek Örnek Sayısı (support)
XGB	0		0.91	0.92	0.91	977
	1		0.87	0.85	0.86	608
	Model	0.893		0.845	0.859	

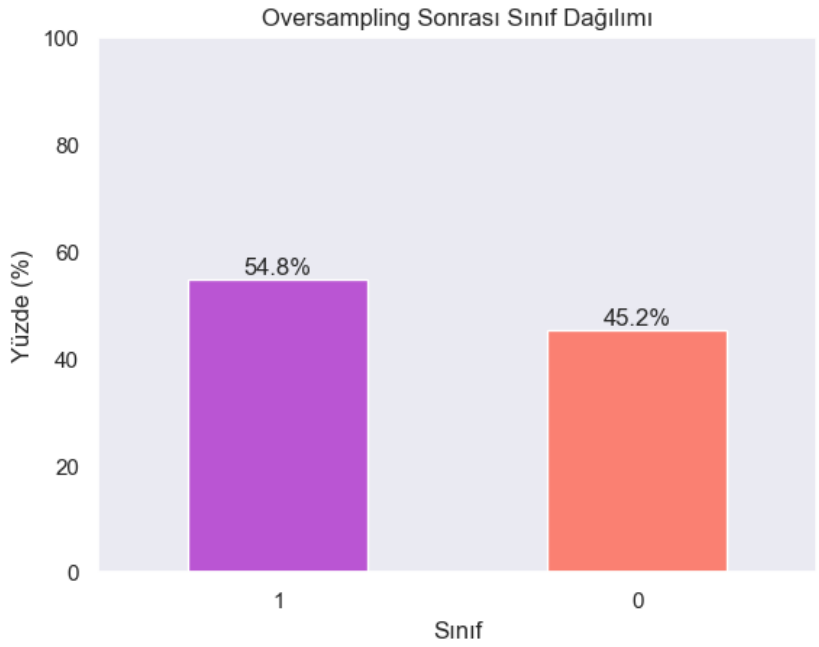
ROC eğrileri ve AUC değerleri incelendiğinde Şekil 11'de gösterildiği gibi SVM-SMOTE ile dengelenmiş veri üzerinde performansları test edilen 4 algoritma içerisinden RF algoritmasının, ilgili veri kümesi üzerinde en başarılı sınıflandırma algoritması olduğu gözlemlenmiştir.



Şekil 11: SVM-SMOTE Uygulanmış Veri Kümesi Kullanılarak Kurulan Modellerin ROC ve AUC Değerleri

4.6. Sınıflandırma Algoritmalarının SMOTE-ENN ile Dengelenmiş Veri Kümesi Üzerindeki Performans Sonuçları

Orijinal veri kümesi üzerinde SMOTE-ENN algoritması uygulanarak sınıf eşitlemesi yapıldığında sınıf dağılımları Şekil 12'de ve sınıflandırma algoritmalarının performansları Çizelge 9'da verilmiştir. Diğer SMOTE tekniklerine göre farklı olarak SMOTE-ENN algoritması uygulanmış veri kümesinin sınıf dağılım oranlarını incelediğimizde orijinal veride azınlık olan sınıfın burada baskın sınıf olduğu görülmektedir. SMOTE-ENN ile bağımlı değişkene ait sınıfları tekrar düzenlenmiş veri kümesi üzerinde yapılan modelleme sonucu ortaya çıkan performans ölçütlerine göre, RF algoritması ile yapılan sınıflandırmanın diğerlerinden daha başarılı olduğunu iki tablodaki değerle dayandırarak söylemek mümkündür. Bunun yanında modellere ait sınıflandırma raporu incelendiğinde RF algoritmasının sınıflandırmada diğer algoritmalardan daha başarılı olduğunu söylemek mümkündür. Ek olarak XGB algoritmasının da yine bu veri kümesinde de alternatif oluşturduğunu söylemek mümkündür.



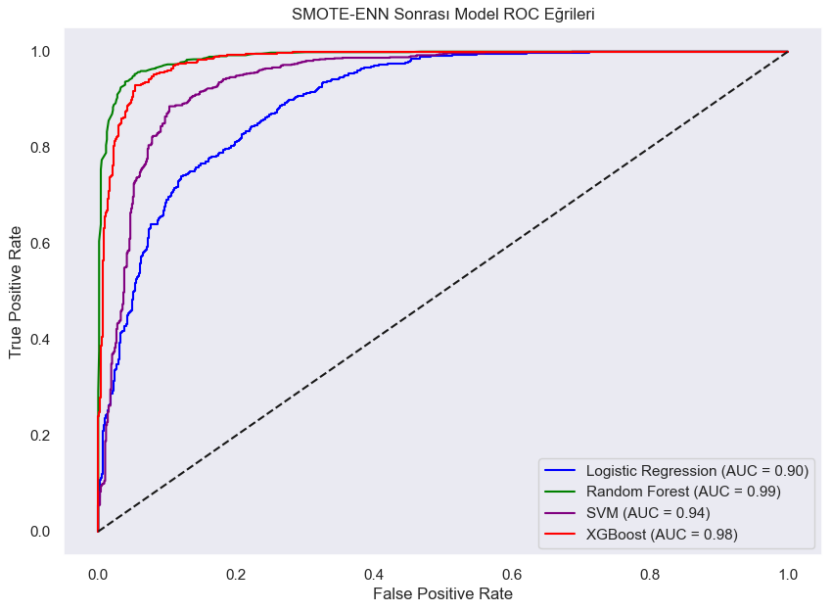
Şekil 12 SMOTE-ENN Uygulaması Sonrası Bağımlı Değişkende Sınıfların Dağılımı

Çizelge 9: SMOTE-ENN Uygulaması Sonrası Yapılan LR,RF,SVM ve XGB Modellerine Ait Sınıflandırma Raporu

	Sınıf	Doğruluk (Accuracy)	Kesinlik (Precision)	Duyarlılık (Recall)	F1 Skoru	Gerçek Örnek Sayısı (support)
LR	0		0.81	0.77	0.79	795
	1		0.81	0.85	0.83	939
	Model	0.812		0.850	0.830	
	Sınıf	Doğruluk (Accuracy)	Kesinlik (Precision)	Duyarlılık (Recall)	F1 Skoru	Gerçek Örnek Sayısı (support)
RF	0		0.96	0.91	0.93	795
	1		0.92	0.97	0.95	939
	Model	0.941		0.971	0.947	
	Sınıf	Doğruluk (Accuracy)	Kesinlik (Precision)	Duyarlılık (Recall)	F1 Skoru	Gerçek Örnek Sayısı (support)
SVM	0		0.91	0.83	0.87	795

	1		0.86	0.93	0.90	939
	Model	0.883		0.897	0.799	
XGB	Sınıf	Doğruluk (Accuracy)	Kesinlik (Precision)	Duyarlılık (Recall)	F1 Skoru	Gerçek Örnek Sayısı (support)
	0		0.95	0.90	0.93	795
	1		0.92	0.96	0.94	939
	Model	0.934		0.960	0.940	

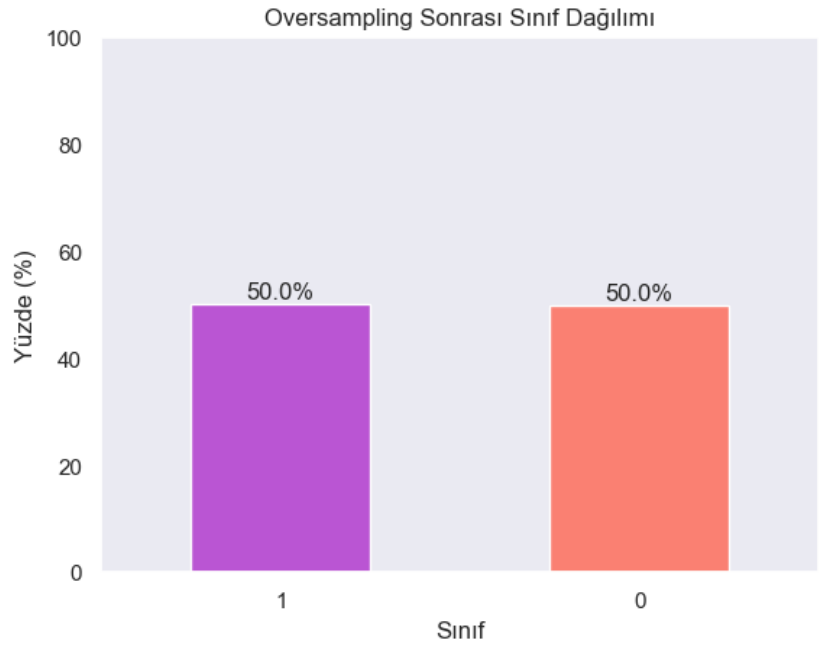
Şekil 13'te ROC eğrileri ve AUC değerleri incelendiğinde SMOTE-ENN ile dengelenmiş veri üzerinde performansları test edilen 4 algoritma içerisinden rastgele ormanlar algoritmasının, ilgili veri kümesi üzerinde en başarılı sınıflandırma algoritması olarak öne çıktığı gözlenmiştir gözlemlenmiştir.



Şekil 13: SMOTE-ENN Uygulanmış Veri Kümesi Kullanılarak Kurulan Modellerin ROC ve AUC Değerleri

4.7. Sınıflandırma Algoritmalarının K-Means SMOTE ile Dengelenmiş Veri Kümesi Üzerindeki Performans Sonuçları

Orijinal veri kümesi üzerinde K-Means SMOTE algoritması uygulanarak sınıf eşitlemesi yapıldığında sınıf dağılımları Şekil 14'te ve performans sonuçları ise Çizelge 10'da verilmiştir. K-Means SMOTE tekniği ile bağımlı değişkene ait sınıfları tekrar düzenlenmiş veri kümesi üzerinde yapılan modelleme sonucu ortaya çıkan performans sonucunda RF algoritması ile yapılan sınıflandırmanın diğerlerinden daha başarılı olduğunu iki tablodaki değerle dayandırarak söylenebilir. Bunun yanında modellere ait sınıflandırma raporu incelendiğinde RF algoritmasına ek olarak XGB algoritmasının da yine bu veri kümesinde de alternatif oluşturduğunu söylemek mümkündür.

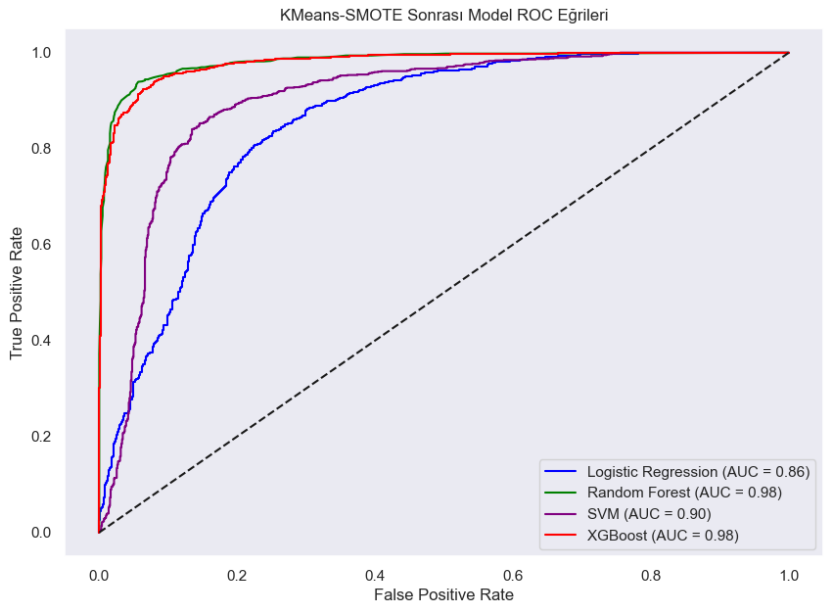


Şekil 14:K-Means SMOTE Uygulaması Sonrası Bağımlı Değişkende Sınıfların Dağılımı

Şekil 15'te ROC eğrileri ve AUC değerleri incelendiğinde K-Means SMOTE ile dengelenmiş veri üzerinde performansları test edilen 4 algoritma içerisinden RF algoritmasının, ilgili veri kümesi üzerinde en başarılı sınıflandırma algoritması olarak XGB algoritmasına göre az farkla öne çıktığı gözlemlenmiştir.

Çizelge 10: K-Means SMOTE Uygulaması Sonrası Yapılan LR,RF,SVM ve XGB Modellerine Ait Sınıflandırma Raporu

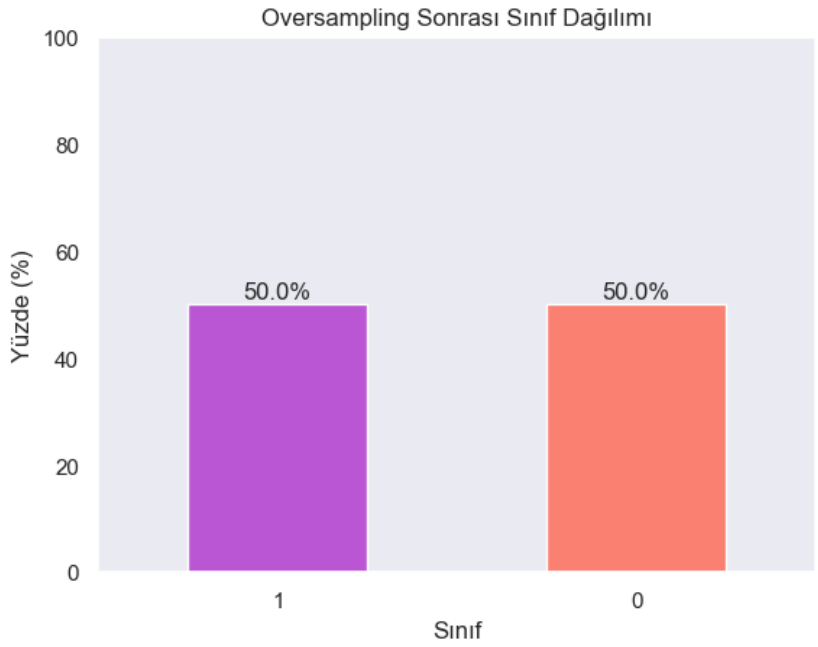
	Sınıf	Doğruluk (Accuracy)	Kesinlik (Precision)	Duyarlılık (Recall)	F1 Skoru	Gerçek Örnek Sayısı (support)
LR	0		0.83	0.74	0.78	1026
	1		0.76	0.84	0.80	985
	Model	0.788		0.840	0.795	
	Sınıf	Doğruluk (Accuracy)	Kesinlik (Precision)	Duyarlılık (Recall)	F1 Skoru	Gerçek Örnek Sayısı (support)
RF	0		0.95	0.92	0.93	1026
	1		0.91	0.95	0.93	985
	Model	0.932		0.950	0.932	
	Sınıf	Doğruluk (Accuracy)	Kesinlik (Precision)	Duyarlılık (Recall)	F1 Skoru	Gerçek Örnek Sayısı (support)
SVM	0		0.90	0.77	0.83	1026
	1		0.79	0.91	0.85	985
	Model	0.838		0.906	0.845	
	Sınıf	Doğruluk (Accuracy)	Kesinlik (Precision)	Duyarlılık (Recall)	F1 Skoru	Gerçek Örnek Sayısı (support)
XGB	0		0.93	0.92	0.93	1026
	1		0.92	0.93	0.93	985
	Model	0.927		0.930	0.926	



Şekil 15: K-Means SMOTE Uygulanmış Veri Kümesi Kullanılarak Kurulan Modellerin ROC ve AUC Değerleri

4.8. Sınıflandırma Algoritmalarının SMOTETomek ile Dengelenmiş Veri Kümesi Üzerindeki Performans Sonuçları

Orijinal veri kümesi üzerinde SMOTETomek algoritması uygulanarak sınıf eşitlemesi yapıldığında sınıf dağılımları Şekil 16'da ve sınıflandırma algoritmalarının performansları Çizelge 11'de verilmiştir. SMOTETomek tekniği ile bağımlı değişkene ait sınıfları tekrar düzenlenmiş veri kümesi üzerinde yapılan modelleme XGB ve RF algoritmalarının yine yakın sonuçlar verdiği görülmektedir. Bu iki algoritma ile yapılan sınıflandırmanın diğerlerinden daha başarılı olduğunu iki tablodaki değerle dayandırarak söylemek mümkündür. Buna rağmen bir algoritma seçimi yapılması gerekirse sınıflandırma raporundaki ölçütleri de göz önünde bulundurarak seçimi XGB algoritması yönünde yapmak daha mantıklı olacaktır.



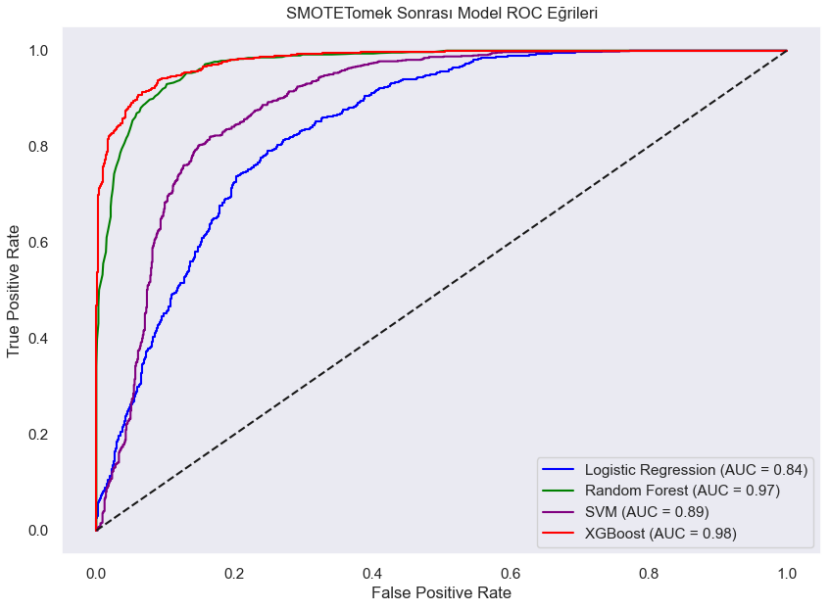
Şekil 16: SMOTETomek Uygulaması Sonrası Bağımlı Değişkende Sınıfların Dağılımı

Çizelge 11: SMOTETomek Uygulaması Sonrası Yapılan LR, RF, SVM ve XGB Modellerine Ait Sınıflandırma Raporu

	Sınıf	Doğruluk (Accuracy)	Kesinlik (Precision)	Duyarlılık (Recall)	F1 Skoru	Gerçek Örnek Sayısı (support)
LR	0		0.79	0.74	0.76	1010
	1		0.75	0.80	0.77	989
	Model	0.768		0.799	0.773	
	Sınıf	Doğruluk (Accuracy)	Kesinlik (Precision)	Duyarlılık (Recall)	F1 Skoru	Gerçek Örnek Sayısı (support)
RF	0		0.95	0.87	0.91	1010
	1		0.88	0.95	0.91	989
	Model	0.910		0.948	0.912	
	Sınıf	Doğruluk (Accuracy)	Kesinlik (Precision)	Duyarlılık (Recall)	F1 Skoru	Gerçek Örnek Sayısı (support)
SVM	0		0.88	0.74	0.81	1010

	1		0.77	0.90	0.83	989
	Model	0.818		0.906	0.845	
XGB	Sınıf	Doğruluk (Accuracy)	Kesinlik (Precision)	Duyarlılık (Recall)	F1 Skoru	Gerçek Örnek Sayısı (support)
	0		0.94	0.90	0.92	1010
	1		0.90	0.94	0.92	989
	Model	0.919		0.942	0.920	

ROC eğrileri ve AUC değerleri Şekil 17'de SMOTETomek ile dengelenmiş veri üzerinde performansları test edilen 4 algoritma içerisinden XGB algoritmasının, ilgili veri kümesi üzerinde en başarılı sınıflandırma algoritması olarak RF algoritmasına göre öne çıktığı gözlemlenmiştir.

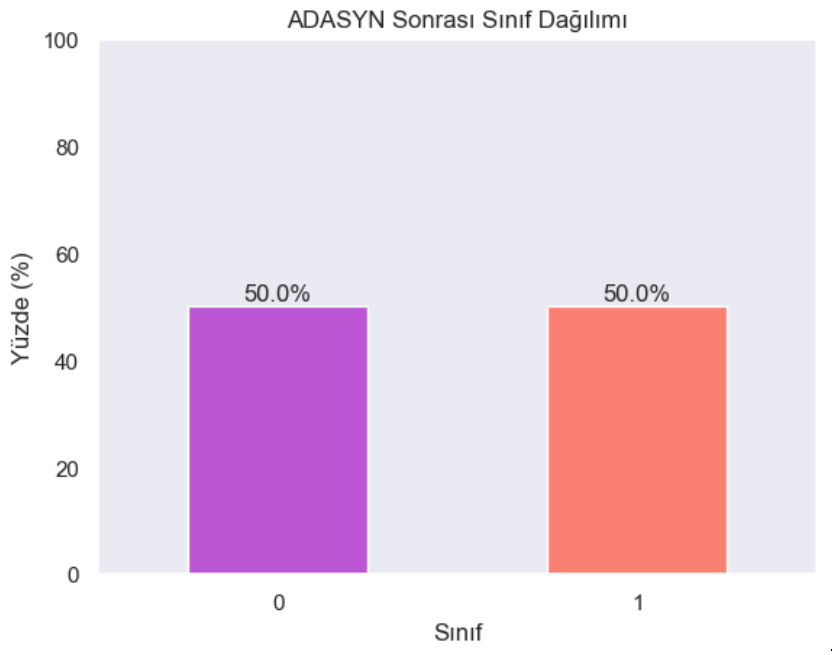


Şekil 17: SMOTETomek Uygulanmış Veri Kümesi Kullanılarak Kurulan Modellerin ROC ve AUC Değerleri

4.9. Sınıflandırma Algoritmalarının ADASYN ile Dengelenmiş Veri Kümesi Üzerindeki Performans Sonuçları

Orijinal veri kümesi üzerinde ADASYN algoritması uygulanarak sınıf eşitlemesi yapıldığında sınıf dağılımları Şekil 18'de ve sınıflandırma algoritmalarının

performansları Çizelge 12'de verilmiştir. Modelleme sonucu elde edilen sonuçlar incelendiğinde XGB ve RF algoritmalarının bu veri kümesi üzerinde de yakın sonuçlar verdiği görülmektedir. Bu iki algoritma ile yapılan sınıflandırmanın diğerlerinden daha başarılı olduğunu iki tablodaki değerle dayandırarak söylemek mümkündür. Buna rağmen bir algoritma seçimi yapılması gerekirse sınıflandırma raporundaki ölçütleri de göz önünde bulundurarak seçimi XGB algoritması yönünde yapılabilir.



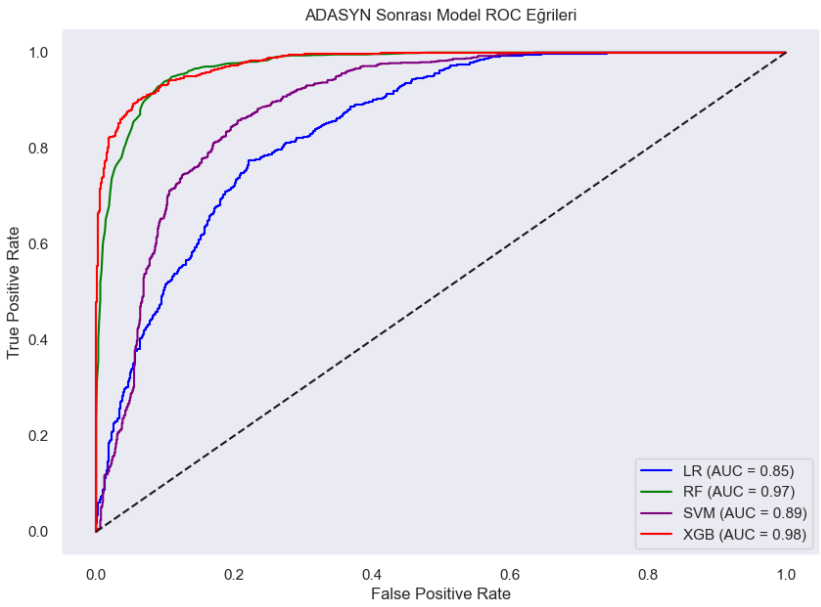
Şekil 18: ADASYN Uygulaması Sonrası Bağımlı Değişkende Sınıfların Dağılımı

Çizelge 12: ADASYN Uygulaması Sonrası Yapılan LR, RF, SVM ve XGB Modellerine Ait Sınıflandırma Raporu

	Sınıf	Doğruluk (Accuracy)	Kesinlik (Precision)	Duyarlılık (Recall)	F1 Skoru	Gerçek Örnek Sayısı (support)
LR	0		0.80	0.72	0.75	1020
	1		0.74	0.81	0.77	990
	Model	0.763		0.813	0.772	
	Sınıf	Doğruluk (Accuracy)	Kesinlik (Precision)	Duyarlılık (Recall)	F1 Skoru	Gerçek Örnek Sayısı (support)
RF						

	0		0.96	0.86	0.91	1020
	1		0.87	0.97	0.91	990
	Model	0.910		0.967	0.914	
SVM	Sınıf	Doğruluk (Accuracy)	Kesinlik (Precision)	Duyarlılık (Recall)	F1 Skoru	Gerçek Örnek Sayısı (support)
	0		0.88	0.73	0.80	1020
	1		0.77	0.90	0.83	990
	Model	0.815		0.899	0.827	
XGB	Sınıf	Doğruluk (Accuracy)	Kesinlik (Precision)	Duyarlılık (Recall)	F1 Skoru	Gerçek Örnek Sayısı (support)
	0		0.93	0.91	0.92	1020
	1		0.91	0.93	0.92	990
	Model	0.917		0.925	0.916	

ROC eğrileri ve AUC değerleri Şekil 19'da takip edildiğinde ise, ADASYN ile dengelenmiş veri üzerinde performansları test edilen 4 algoritma içerisinden RF ve XGB algoritmalarının, benzer şekilde ilgili veri kümesi üzerinde en başarılı sınıflandırma algoritması olarak diğer iki sınıflandırma algoritmasına göre öne çıktığı gözlemlenmiştir.



Şekil 19: ADASYN Uygulanmış Veri Kümesi Kullanılarak Kurulan Modellerin ROC ve AUC Değerleri

4.10. Senaryo Karşılaştırmaları ve İstatistiksel Analizler

Önceki bölümlerde her bir aşırı örnekleme yöntemi için kendi içinde sınıflandırma algoritmalarının performansları karşılaştırılmıştır. Bu bölümde her bir aşırı örnekleme yöntemi için en başarılı sınıflandırma algoritması alınarak birbirleri ile karşılaştırılmaları sağlanacaktır. Algoritmalardan en başarılı olanları bu bölümde birlikte karşılaştırılacak ve en verimli ikili belirlenecektir. Her sürecin içinde seçilen en iyi modeller bir araya getirilip karşılaştırıldığında elde edilen sonuçlar Çizelge 13'te özetlenmiştir. Çizelge incelendiğinde yıldız (*) ile işaretlenmiş performans ölçütlerinin en yüksek değerleri verdiği görülmektedir. Buradan yola çıkarak SMOTE-ENN aşırı örnekleme tekniği kullanılarak dengelenmiş veri kümesi ve RF algoritması ikilisinin en yüksek performans ölçütlerini verdiğini gözlemlemek dolayısıyla bu ikilinin en verimli ikili olduğunu söylemek mümkündür. Bu ikilinin test kümesinde %94,1 doğruluk ile SMOTE-ENN ve RF algoritmaları olduğu görülmüştür.

Çizelge 13: Aşırı Örnekleme Yöntemleri için En Başarılı Sınıflandırma Algoritmaları Seçilerek Oluşturulmuş Sınıflandırma Raporu

	Sınıf	AUC Skoru	Doğruluk (Accuracy)	Kesinlik (Precision)	Duyarlılık (Recall)	F1 Skoru	Gerçek Örnek Sayısı (support)
SMOTE-XGB	0			0.93	0.89	0.91	1020
	1			0.89	0.93	0.91	990
	Model	0.97	0.910		0.932	0.911	
Borderline-SMOTE-XGB	Sınıf	AUC Skoru	Doğruluk (Accuracy)	Kesinlik (Precision)	Duyarlılık (Recall)	F1 Skoru	Gerçek Örnek Sayısı (support)
	0			0.95	0.91	0.93*	1020
	1			0.91	0.95	0.93	990
	Model	0.98	0.930		0.945	0.930	
SVM-SMOTE-RF	Sınıf	AUC Skoru	Doğruluk (Accuracy)	Kesinlik (Precision)	Duyarlılık (Recall)	F1 Skoru	Gerçek Örnek Sayısı (support)
	0			0.93	0.91	0.92	977
	1			0.86	0.89	0.87	608
	Model	0.96	0.902		0.888	0.874	
SMOTE-ENN-RF*	Sınıf	AUC Skoru	Doğruluk (Accuracy)	Kesinlik (Precision)	Duyarlılık (Recall)	F1 Skoru	Gerçek Örnek Sayısı (support)
	0			0.96*	0.91	0.93*	795
	1			0.92*	0.97*	0.95*	939
	Model	0.99*	0.941*		0.971*	0.947*	
KMeans-SMOTE-RF	Sınıf	AUC Skoru	Doğruluk (Accuracy)	Kesinlik (Precision)	Duyarlılık (Recall)	F1 Skoru	Gerçek Örnek Sayısı (support)
	0			0.95	0.92*	0.93*	1026
	1			0.91	0.95	0.93	985
	Model	0.98	0.932		0.950	0.932	
SMOTETomek-XGB	Sınıf	AUC Skoru	Doğruluk (Accuracy)	Kesinlik (Precision)	Duyarlılık (Recall)	F1 Skoru	Gerçek Örnek Sayısı (support)
	0			0.94	0.90	0.92	1010
	1			0.90	0.94	0.92	989
	Model	0.98	0.919		0.942	0.920	

	Sınıf	AUC Skoru	Doğruluk (Accuracy)	Kesinlik (Precision)	Duyarlılık (Recall)	F1 Skoru	Gerçek Örnek Sayısı (support)
ADASYN-XGB	0			0.93	0.91	0.92	1020
	1			0.91	0.93	0.92	990
	Model	0.98	0.917		0.925	0.916	

5. SONUÇ VE ÖNERİLER

Bu araştırma kapsamında aşırı örnekleme tekniği olarak temel SMOTE, Borderline-SMOTE, SVM-SMOTE, SMOTE-ENN, KMeans-SMOTE, SMOTETomek ve ADASYN algoritmaları ile bağımlı değişkenine ait sınıfları dengelenmiş olan ve sınıflandırma algoritmalarından LR, RF, SVM ve XGB algoritmaları sınıflandırma yapılarak çalışmada bahsedilen performans ölçütleri kullanılarak ikili olarak performansları test edilmiş ve son bölümde her aşırı örnekleme algoritması için bir sınıflandırma algoritması seçilecek karşılaştırma yapılmış ve en verimli ikilinin belirlenmiştir. Elde edilen bilgiler özetlenecek olursa, tek bir veri kümesi üzerinde çalışılmasının yanıltıcı olabileceği göz önünde bulundurulması gerektiğini not ederek; aşırı örnekleme algoritmalarının performanslarını tek başına incelendiğinde çalışmada yer verilen bütün sınıflandırma algoritmalarında benzer şekilde yüksek performans vermesi ile SMOTE-ENN algoritmasının öne çıkmıştır. Çalışmanın amacı kapsamında en yüksek performansa sahip ikiliyi seçmemiz gerektiğinde bu ikilinin aşırı örnekleme algoritmaları arasından en başarılısı olarak öne çıkan SMOTE-ENN algoritması ve çalışmadaki performans tablolarının çoğunluğunda başarılı sınıflandırma algoritmalarından biri olarak öne çıkan RF algoritması ikilisinin birlikte kullanılmasıdır.

Bu çalışmada kullanılan performans ölçütleri, dengesiz veri kümesinde F1 skorunun neden çok incelenen ölçütlerden biri olduğunu da ortaya konulmuştur. Sadece kesinlik ve duyarlılık üzerinden yapılan incelemelerde azınlık veri kümesindeki başarının az olmasından dolayı yanıltıcılığın olabileceği model doğruluğunun %94.6 değerini gösterirken F1 skorunun 0 değerini göstermesi sonucu görülmüştür. Bu nedenle performans ölçütlerinde iki değerin harmonik bir ortalamasından elde edilen F1-skorunun

kullanılması bir zorunluluktur denebilir. Ayrıca, dengesiz veri kümelerinde mutlaka veri önişleme sürecinde sınıf dengeleme sürecinin gerekliliğini de F1 skorunun 0'dan 0.947'ye yükselmesi göstermiştir. SMOTE-ENN yönteminin diğer yöntemlere göre başarısı sınıf dengeleme sonucunda orijinal veri kümesinde azınlık olan pozitif sınıfın işlem sonucunda negatif sınıftan daha çok örneğe sahip olmasıyla görülmüştür. Bu nedenle hızlı ve etkili bir sınıflandırma sürecinde RF algoritmasının aşırı örnekleme süreciyle dengeli hale getirilmiş veri kümesi üzerinde yüksek sınıflandırma performansı gösterdiği görülmüştür. Bu çalışmanın devamında sınıflandırma algoritmalarının sayısının arttırılması rekabetin daha da artması ve buna bağlı olarak yeni veri kümeleri ile denemelerin yapılması ihtiyacı görülmektedir. Buna bağlı olarak çalışmadaki algoritmaların farklı dengesiz veri kümeleri üzerinde test edilerek sonuçların sağlamlaştırılması veya farklı özellikteki veri kümeleri ve değişkenler ile hangi ikilinin daha verimli olacağının testinin yapılması önemli olacaktır. Veriden öğrenme sürecinde veriyi daha iyi öğrenebilen yapının hangisinin olduğunu görmek ve ortaya çıkartabilmek bir süreç olarak işletilmelidir. Aynı şekilde, benzer çalışmaların yapılması ve ilgili sonuçların benzer veri kümelerindeki farklılıklarının incelenmesi ve yorumlanması önemli olacaktır.

6. KAYNAKÇA

[1] Bhatia, K., Arora, S., & Tomar, R. (2016). Diagnosis of diabetic retinopathy using machine learning classification algorithm. 2016 2nd International Conference on Next Generation Computing Technologies (NGCT). Dehradun, India: IEEE.

[2] Sarker, I. (2021). Machine Learning: Algorithms, Real-World Applications and Research Directions. SN COMPUT. SCI. 2, 160.

[3] Minaee, S. a., Nikzad, N., Chenaghlu, M., & Gao, J. (2021). Deep Learning-based Text Classification: A Comprehensive Review. ACM Computing Surveys (CSUR), 1-40.

[4] Ismail Fawaz, H. F. (2019). Deep learning for time series classification: a review. Data Min Knowl Disc, 917-963.

[5] Qian Li, H. P. (2021). A Survey on Text Classification: From Shallow to Deep Learning. arXiv preprint, arXiv:2008.00364.

[6] Vakili M., G. M. (2020). Performance analysis and comparison of machine and deep learning algorithms for IoT data classification. . arXiv preprint, arXiv:2001.09636.

[7] Sarker, I. H. (2019). Context-aware rule learning from smartphone data: Review, challenges and future directions. Journal of Big Data, 1-28.

[8] Kumar, R., & Vadlamani, R. (2015). A survey on opinion mining and sentiment analysis: Tasks, approaches and applications. Knowledge-Based Systems, 14-46.

[9] Sarker, I. H., Abushark, Y. B., Alsolami, F., & Khan, A. I. (2020). Intrudtree: A machine learning based cyber security intrusion detection model. SN Computer Science, 754.

[10] Sarker, I. H., & Salah, K. (2019). Appspred: A context-aware mobile app prediction model using random forest learning. Journal of Network and Computer Applications, 35-45.

[11] Sarker, I. H., Kayes, A. S., & Watters, P. (2019). Effectiveness analysis of machine learning classification models for predicting personalized context-aware smartphone usage. Journal of Big Data 6, 57.

[12] Marbac, M., Tubert-Bitter, P., & Sedki, M. (2016). Bayesian model selection in logistic regression for the detection of adverse drug reactions. Biometrical Journal, 58(6), 1376-1389.

[13] Valle, D., Lima, J. M., Millar, J., Amratia, P., & Haque, U. (2015). Bias in logistic regression due to outcome misclassification: Illustration in the context of malaria risk prediction. Malaria Journal 14(1), 434.

[14] Allam, A., Nagy, M., Thoma, G., & Krauthammer, M. (2019). Neural networks versus Logistic regression for 30 days all-cause readmission prediction. Scientific reports, 9(1), 9277.

[15] Elkouri, A. (2015). Predicting the sentiment polarity and rating of yelp reviews. arXiv preprint, arXiv:1512.06303.

[16] Aragaw, K. A. (2015). Application of Logistic Regression in Determining the Factors Influencing the Use of Modern Contraceptive Among Married Women in Ethiopia. American Journal of Theoretical and Applied Statistics, 4(3),, 156-162.

[17] Matsui, M., Cruz, J., Tang, J., & al., e. (2017). Logistic regression analysis differentiates high from low computer users by facial skin conditions in a population of Chinese women. Applied Informatics, 4, 4.

[18] Kaya, Y., Leite, W. L., & Miller, M. D. (2016). A comparison of logistic regression models for detecting differential item functioning in polytomous items. International Journal of Assessment Tools in Education, 3(1),, 22-38.

[19] Budimir, M., Atkinson, P., & Lewis, H. (2015). A systematic review of landslide probability mapping using logistic regression. Landslides, 12, 419-436.

[20] Yadav, S. K., Bharadwaj, B., & Pal, S. (2012). Data mining applications: A comparative study for predicting student's performance. arXiv preprint, arXiv:1202.4815.

[21] Chen, S., & Lin, X. (2023). Application of Decision Tree Algorithm in Educational Data Mining. Curriculum and Teaching Methodology, 6(8),, 120-127.

[22] Zhang, Z. (2021). Applications of the Decision Tree in Business Field. In Proceedings of the 11th International Conference on E-business, Management and Economics (ICEMCI 2021) (s. 199-203). Atlantis Press.

[23] Al-Sarem, M. (2015). A decision tree based approach to academic advising in higher education. arXiv preprint, arXiv:1511.04026.

[24] Amancio, D. R., Comin, C. H., Casanova, D., Travieso, G., Bruno, O. M., Rodrigues, F. A., & da Fontoura Costa, L. (2014). A Systematic Comparison of Supervised Classifiers. PLoS ONE 9(4), e94137.

[25] Sun, Y., Xue, B., Zhang, M., Yen, G. G., & Lv, J. (2020). Automatically Designing CNN Architectures Using the Genetic Algorithm for Image Classification. IEEE transactions on cybernetics 50(9), 3840-3854.

[26] Kumar, P., Sehgal, V. K., & Chauhan, D. S. (2012). A benchmark to select data mining based classification algorithms for business intelligence and decision support systems. arXiv preprint, arXiv:1210.3139.

[27] Hui, L., Ling, L., Xiao, Z., & Shan, W. (2015). Hike: A High Performance kNN Query Processing System for Multimedia Data. 2015 IEEE Conference on Collaboration and Internet Computing (CIC) (s. 296-303). Hangzhou, China: IEEE.

[28] Zhao, Y., Zhang, W., & Liu, X. (2024). Grid search with a weighted error function: Hyper-parameter optimization for financial time series forecasting. Applied Soft Computing, 154, 111362.

[29] Jin, C., De-Lin, L., & Fen-Xiang, M. (2009). An improved ID3 decision tree algorithm. 2009 4th International Conference on Computer Science & Education (s. 127-130). Nanning: IEEE.

[30] Shao, X., Zhang, G., Li, P., & Chen, Y. (2001). Application of ID3 algorithm in knowledge acquisition for tolerance design. Journal of materials processing technology, 117(1-2), 66-74.

[31] Douzas, G., & Bacao, F. (2019). Geometric SMOTE a geometrically enhanced drop-in replacement for SMOTE. Information sciences, 501, 118-135.

[32] Dablain, D., Krawczyk, B., & Chawla, N. V. (2022). DeepSMOTE: Fusing deep learning and SMOTE for imbalanced data. IEEE transactions on neural networks and learning systems, 34(9), 6390-6404.

[33] Mansourifar, H., & Shi, W. (2020). Deep synthetic minority over-sampling technique. arXiv preprint , arXiv:2003.09788.

[34] Zhang, X., Ma, D., Gan, L., Jiang, S., & Agam, G. (2016). CGMOS: Certainty Guided Minority OverSampling. Proceedings of the 25th ACM International on Conference on Information and Knowledge Management (s. 1623–1631). New York: Association for Computing Machinery.

[35] Joloudari, J. H., Marefat, A., Nematollahi, M. A., Oyelere, S. S., & Hussain, S. (2023). Effective class-imbalance learning based on SMOTE and convolutional neural networks. Applied Sciences,13(6), 4006.

[36] Adi Pratama, F. R., & Oktora, S. I. (2023). Synthetic Minority Over-sampling Technique (SMOTE) for handling imbalanced data in poverty classification. Statistical Journal of the IAOS, 39(1), 233-239.

[37] Ejiyi, C. J. (2025). Polynomial-SHAP as a SMOTE alternative in conglomerate neural networks for realistic data augmentation in cardiovascular and breast cancer diagnosis. Journal of Big Data, 12(1), 97.

[38] Andriyani, D., Faqih, A., & Permana, S. E. (2025). The Effect of SMOTE Application on Support Vector Machine Performance in Sentiment Classification on Imbalanced Datasets. Journal of Artificial Intelligence and Engineering Applications (JAIEA), 4(2), 752.

[39] Ramezankhani, A., Pournik, O., Shahrabi, J., Azizi, F., Hadaegh, F., & Khalili, D. (2016). The impact of oversampling with SMOTE on the performance of 3 classifiers in prediction of type 2 diabetes. Medical decision making, 36(1), 137-144.

[40] Seo, J. H., & Kim, Y. H. (2018). Machine-learning approach to optimize smote ratio in class imbalance dataset for intrusion detection. Computational intelligence and neuroscience, 9704672.

[41] Yang, L., Li, P., Xue, R., Ma, X., Li, X., & Wang, Z. (2018). Intelligent classification model for railway signal equipment fault based on SMOTE and ensemble learning. In IOP Conference Series: Materials Science and Engineering, 012042.

[42] Sailasya, G., & Kumari, G. L. (2021). Analyzing the performance of stroke prediction using ML classification algorithms. International Journal of Advanced Computer Science and Applications, 12(6)., 539-545.

[43] GRACE, S. T. (30, 06 2024). Stroke Pridiction Dataset. Kaggle: https://www.kaggle.com/datasets/samueltaiwograce/stroke-dataset/data adresinden alındı

[44] Dubey, Y., Tarte, Y., Talatule, N., Damahe, K., Palsodkar, P., & Fulzele, P. (2024). Explainable and Interpretable Model for the Early Detection of Brain Stroke Using Optimized Boosting Algorithms. . Diagnostics, 14(22), 2514.

[45] Kitova, K., Ivanov, I., & Hooper, V. (2024). Stroke Dataset Modeling: Comparative Study of Machine Learning Classification Methods. Algorithms, 17(12), 571.

[46] Hassan, A., Gulzar Ahmad, S., Ullah Munir, E., Ali Khan, I., & Ramzan, N. (2024). Predictive modelling and identification of key risk factors for stroke using machine learning. Scientific Reports 14, 11498.

[47] Dev, S., Wang, H., Nwosu, C. S., Jain, N., Veeravalli, B., & John, D. (2022). A predictive analytics approach for stroke prediction using machine learning and neural networks. . Healthcare Analytics, 2, 100032.

[48] Jiawei, X., Yonggeon, L., Anthony, E. Y., Eunjin, Y., Tinglin, H., Tianjian, G., . . . Ying, D. (2025). Beyond Feature Importance: Feature Interactions in Predicting Post-Stroke Rigidity with Graph Explainable AI. arXiv preprint , arXiv:2504.08150.

[49] Tomita, N., Jiang, S., Maeder, M. E., & Hassanpour, S. (2020). Automatic post-stroke lesion segmentation on MR images using 3D residual convolutional neural network. . NeuroImage: clinical, 27, 102276.

[50] Pinto, A., Pereira, S., Meier, R., Wiest, R., Alves, V., Reyes, M., & Silva, C. A. (2021). Combining unsupervised and supervised learning for predicting the final stroke lesion. Medical image analysis, 69, 101888.

[51] Biswas, N., Uddin, K. M., Rikta, S. T., & Dey, S. K. (2022). A comparative analysis of machine learning classifiers for stroke prediction: A predictive analytics approach. Healthcare Analytics, 2, 100116.

[52] Kotsiantis, S. B., Zaharakis, I., & Pintelas, P. (2007). Supervised machine learning: A review of classification techniques. Emerging artificial intelligence applications in computer engineering, 160(1), 3-24.

[53] Duda, R. O., Hart, P. E., & Stork, D. G. (2000). Pattern classification. 2nd edn Wiley. New York, 153.

[54] Berkson, J. (1944). Application of the Logistic Function to Bio-Assay. Journal of the American Statistical Association, 39(227), 357–365.

[55] McFadden, D. (1972). Conditional logit analysis of qualitative choice behavior.

[56] Morgan, J. N., & Sonquist, J. A. (1963). Problems in the analysis of survey data, and a proposal, 58(302),. Journal of the American statistical association, 415-434.

[57] Quinlan, J. R. (1986). Induction of decision trees. Machine learning, 1,, 81-106.

[58] Jena, M., & Dehuri, S. (2020). DecisionTree for classification and regression: a state-of-the art review. Informatica, 44(4).

[59] Chen, T., & Guestrin, C. (2016). XGBoost: A Scalable Tree Boosting System. Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (s. 785–794). New York, NY, USA: Association for Computing Machinery.

[60] Zhang, X., Zhang, Y., Wang, S., Yao, Y., Fang, B., & Yu, P. S. (2018).Improving stock market prediction via heterogeneous information fusion. Knowledge-Based Systems, 143, 236-247

[61] Wang, Maoguang; Yu, Jiayu; and Ji, Zijian, "Credit Fraud Risk Detection Based on XGBoost-LR Hybrid Model" (2018). ICEB 2018 Proceedings (Guilin, China). 68.

[62] Boser, B. E., Guyon, I. M., & Vapnik, V. N. (1992). A training algorithm for optimal margin classifiers. Proceedings of the Fifth Annual Workshop on Computational Learning Theory (s. 144–152). Pittsburgh, Pennsylvania, USA: Association for Computing Machinery.

[63] Cortes, C., & Vapnik, V. (1995). Support-vector networks. Machine learning, 20,, 273-297.

[64] Guyon, I., Weston, J. B., & Vapnik, V. (2002). Gene selection for cancer classification using support vector machines. Machine learning, 46,, 389-422.

[65] Ho, T. K. (1995). Random decision forests. Proceedings of 3rd International Conference on Document Analysis and Recognition (s. 278-282). Montreal, QC, Canada: IEEE.

[66] Breiman, L. (2001). Random forests. Machine learning, 45,, 5-32.

[67] Statnikov, A., Aliferis, C. F., Tsamardinos, I., Hardin, D., & Levy, S. (2005). A comprehensive evaluation of multicategory classification methods for microarray gene expression cancer diagnosis. . Bioinformatics, 21(5),, 631-643.

[68] Chawla, N. V., Bowyer, K. W., Hall, L. O., & Kegelmeyer, W. P. (2002). SMOTE: synthetic minority over-sampling technique. Journal of artificial intelligence research, 16,, 321-357.

[69] Orellana, E. (2, 06 2025). SMOTE. Medium: https://emilia-orellana44.medium.com/smote-2acd5dd09948 adresinden alındı

[70] He, H., & Garcia, E. A. (2009). Learning from imbalanced data. IEEE Transactions on knowledge and data engineering, 21(9), 1263-1284.

[71] Han, H., Wang, W.-Y., & Mao, B.-H. (2005). Borderline-SMOTE: A New Over-Sampling Method in Imbalanced Data Sets Learning", booktitle="Advances in Intelligent Computing. Advances in Intelligent Computing (s. 878-887). Berlin, Heidelberg: Springer Berlin Heidelberg.

[72] Nguyen, H. M., Cooper, E. W., & Kamei, K. (2011). Borderline over-sampling for imbalanced data classification. International Journal of Knowledge Engineering and Soft Data Paradigms, 3(1), 4-21.

[73] Batista, G. E., Prati, R. C., & Monard, M. C. (2004). A study of the behavior of several methods for balancing machine learning training data. . ACM SIGKDD explorations newsletter, 6(1), 20-29.

[74] Douzas, G., Bacao, F., & Last, F. (2018). Improving imbalanced learning through a heuristic oversampling method based on k-means and SMOTE. Information sciences, 465, 1-20.

[75] He, H., Bai, Y., Garcia, E. A., & Li, S. (2008). ADASYN: Adaptive synthetic sampling approach for imbalanced learning. In 2008 IEEE international joint conference on neural networks . IEEE world congress on computational intelligence (s. (pp. 1322-1328)). Hong Kong: IEEE.

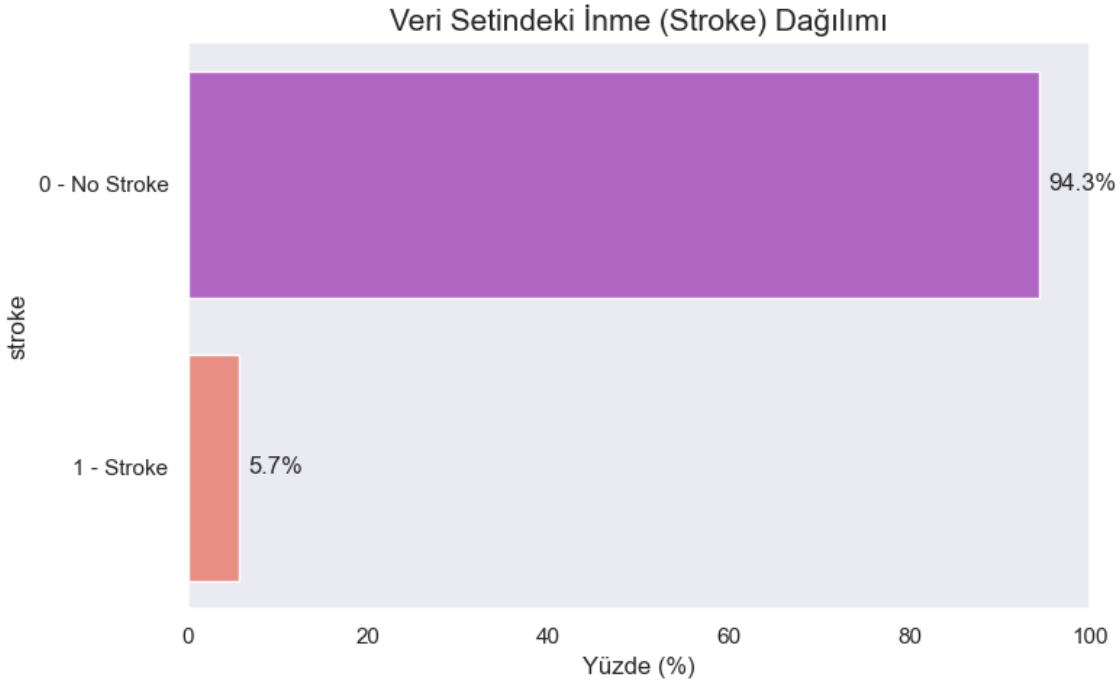
[76] Van Rijsbergen, C. J. (1979). Information retrieval (2nd ed.). Butterworths.

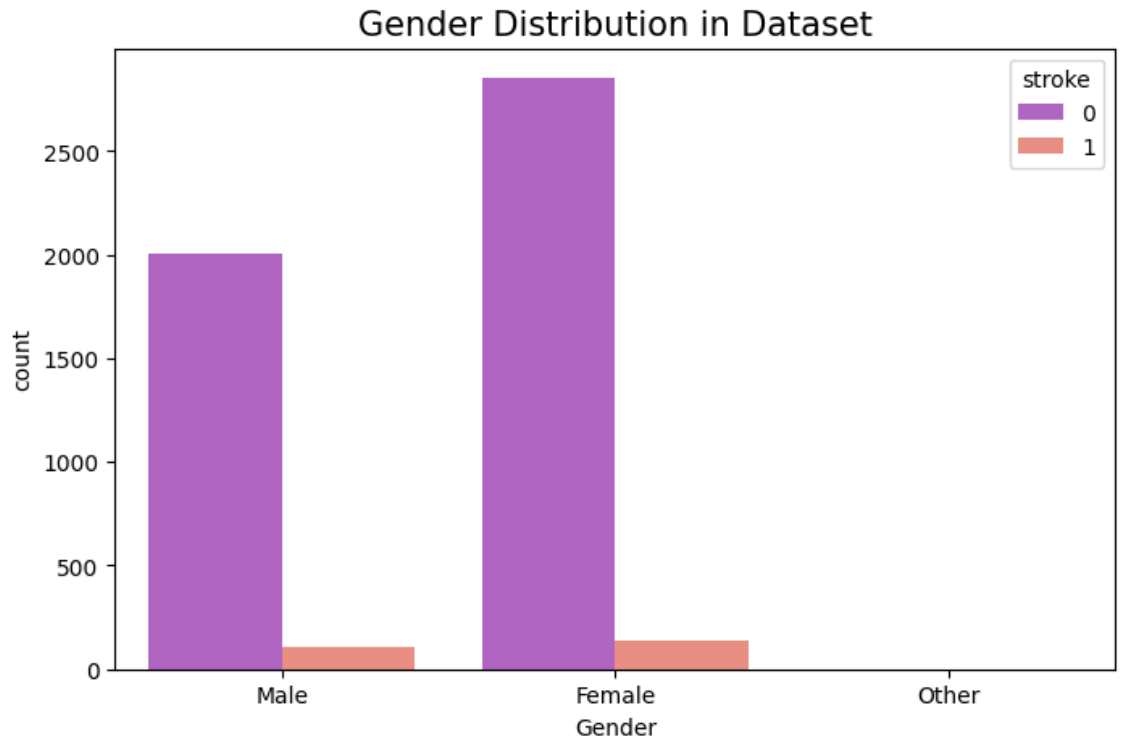
[77] Fawcett, T. (2006). An introduction to ROC analysis. Pattern recognition letters, 27(8),, 861-874.

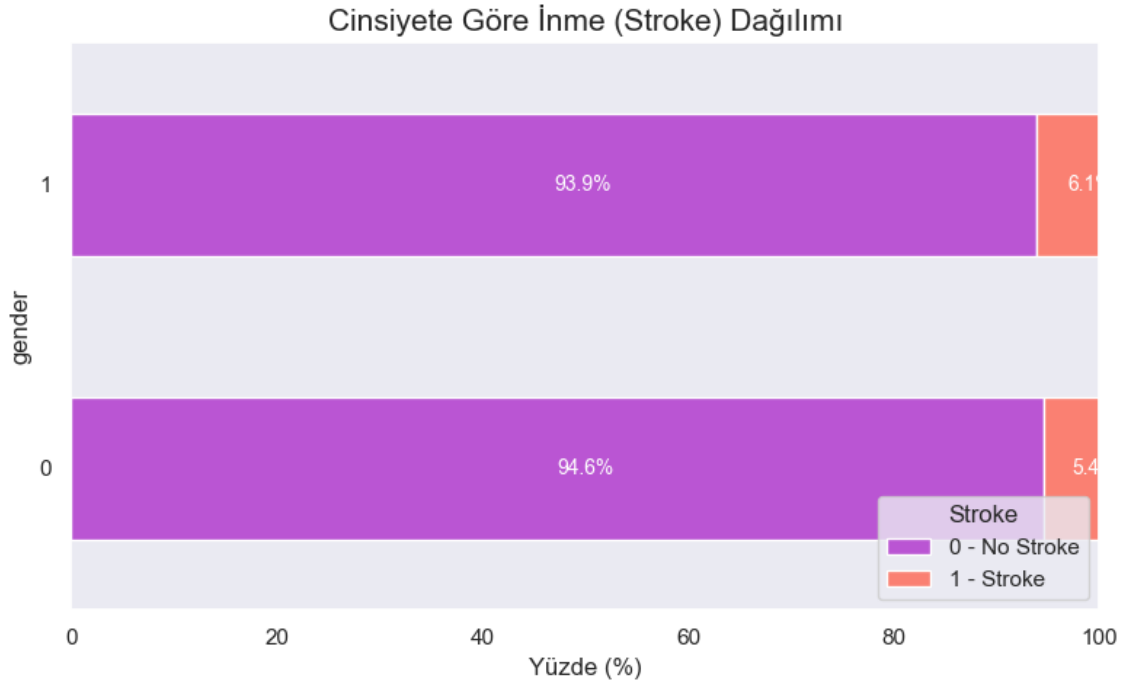
7. EKLER

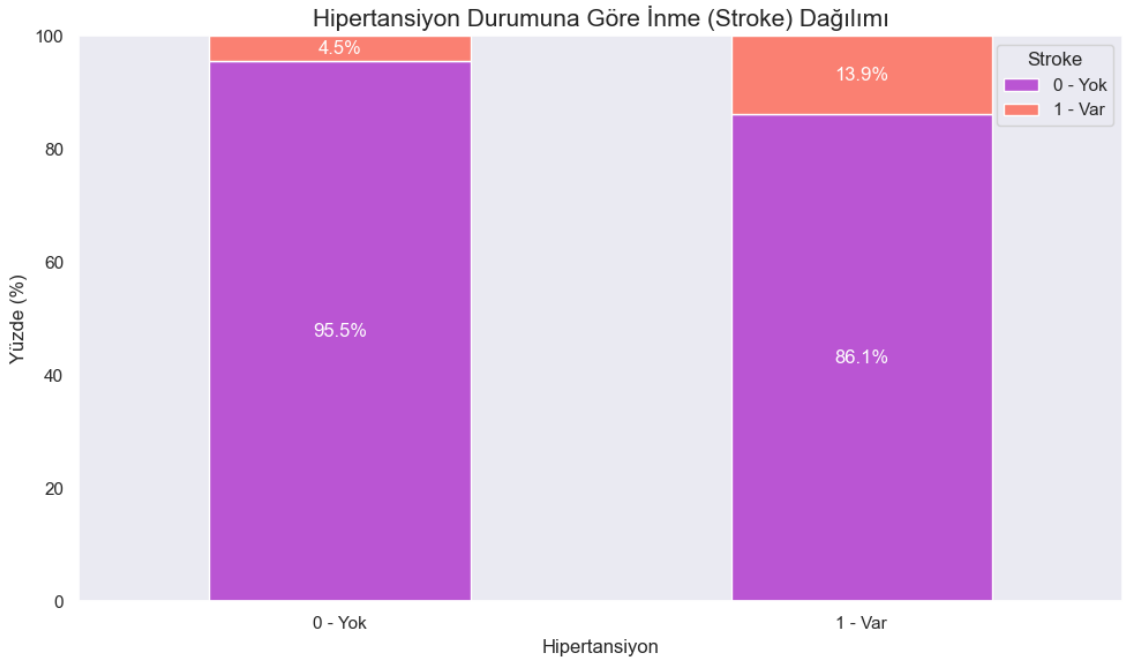
EK-1 Grafikler

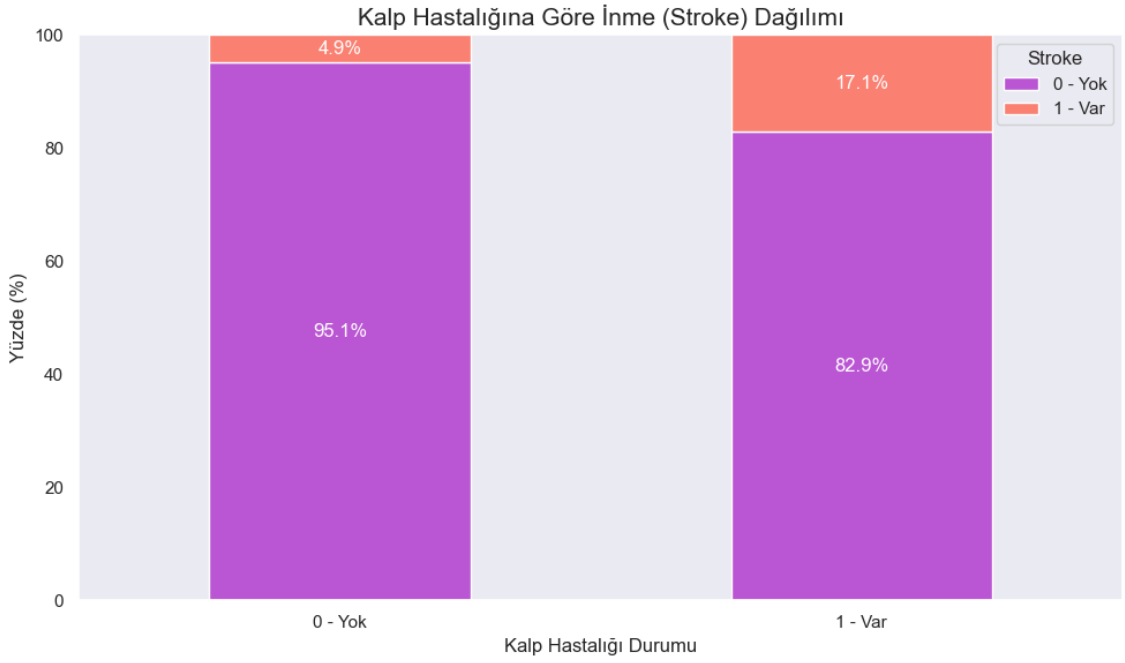
Kullanılan Veri kümesine Ait Grafikler

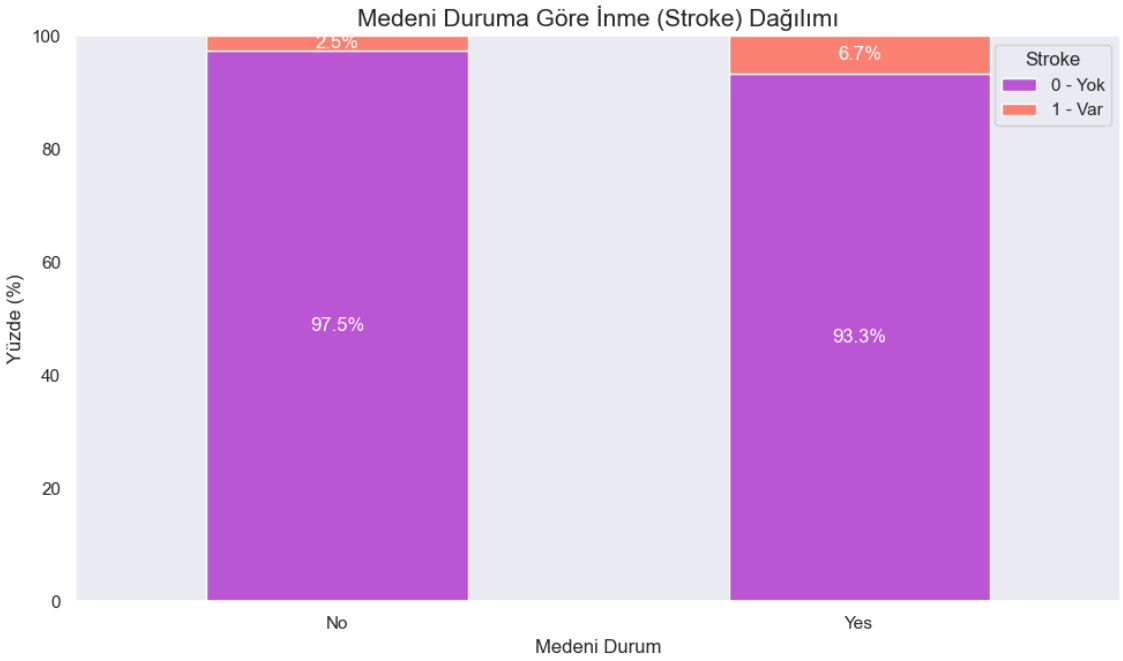


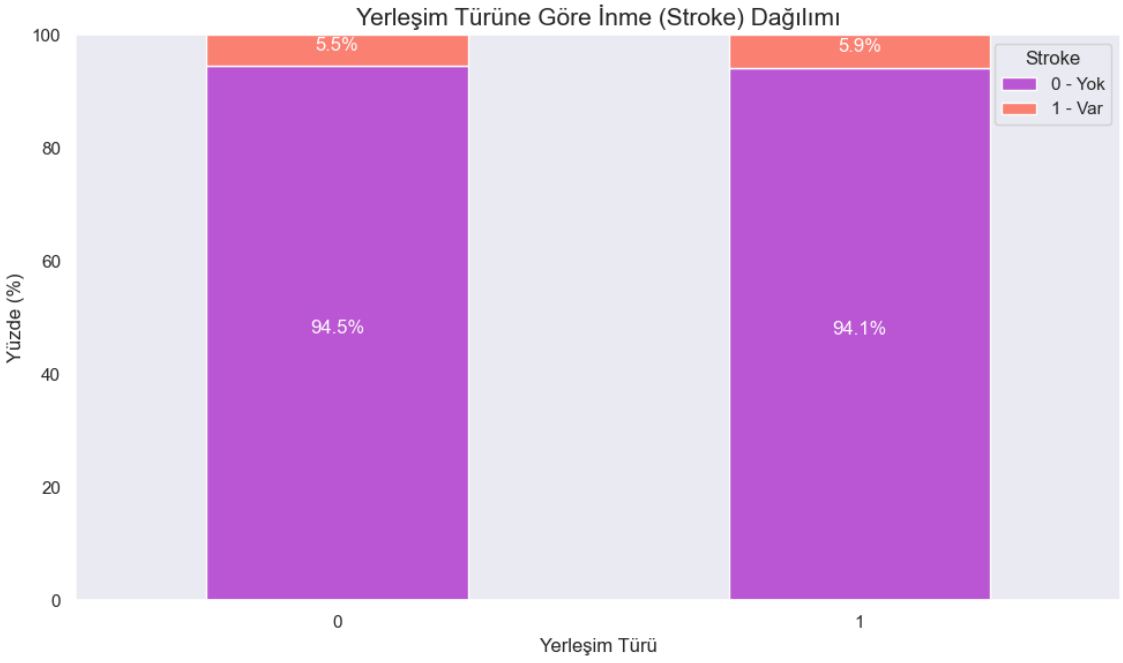

Gender Distribution in Dataset
stroke
0
1
count
2500
2000
1500
1000
500
0
Male
Female
Other
Gender

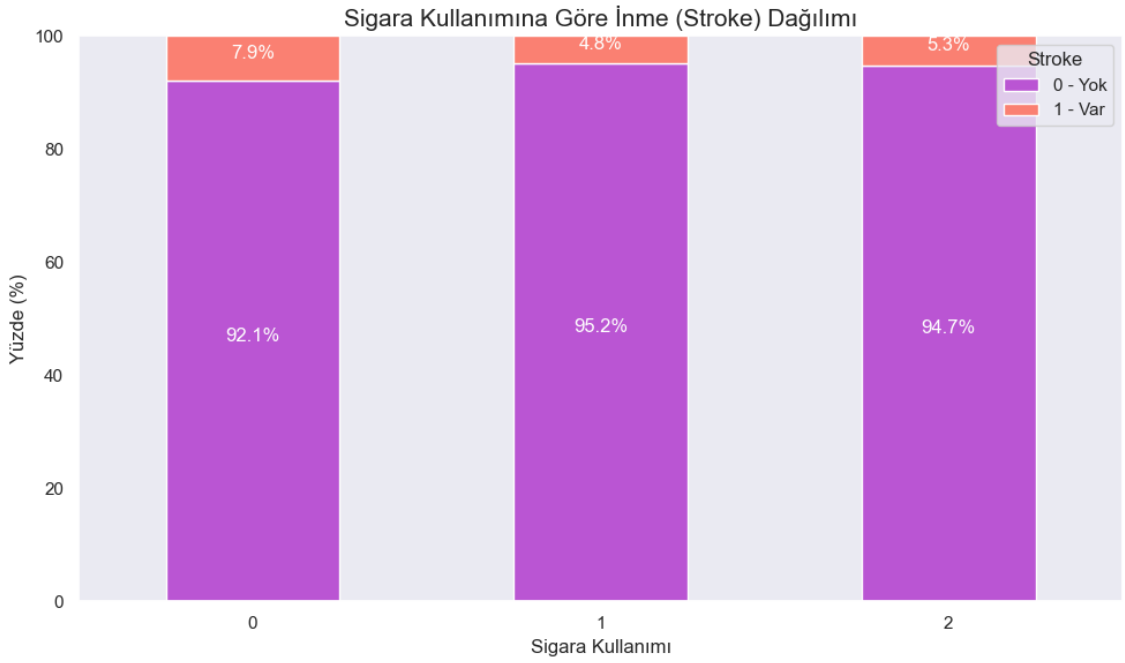

Cinsiyete Göre İnme (Stroke) Dağılımı
gender
1
0
93.9%
94.6%
Stroke
0 - No Stroke
1 - Stroke
0
20
40
60
80
100
Yüzde (%)

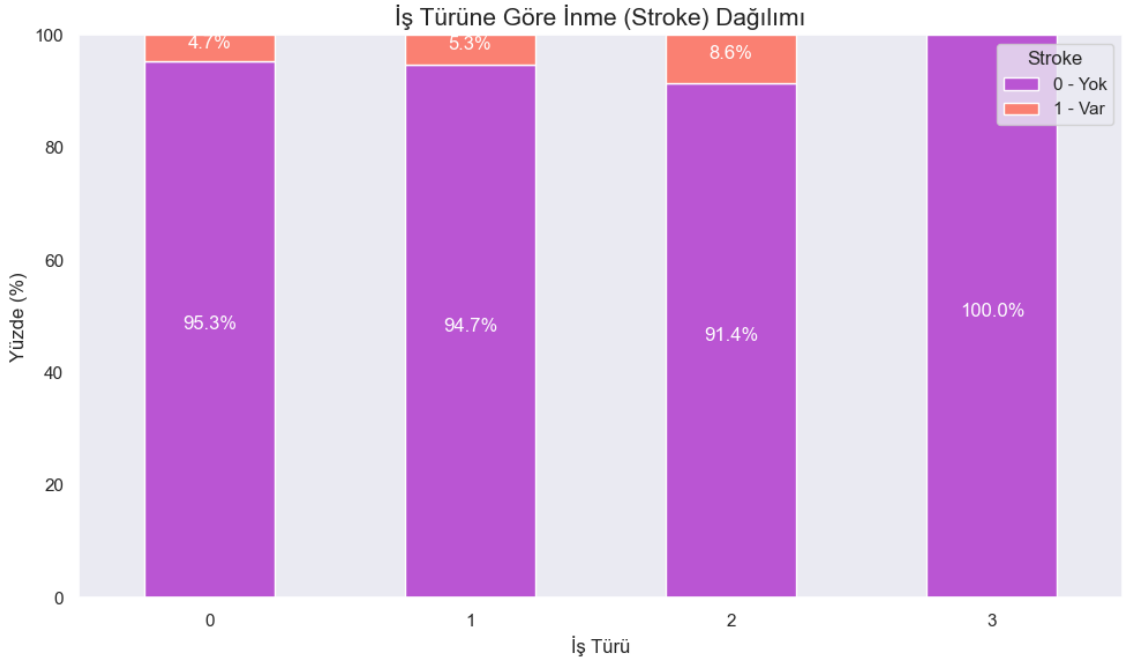

Hipertansiyon Durumuna Göre İnme (Stroke) Dağılımı
Yüzde (%)
100
80
60
40
20
0
4.5%
95.5%
13.9%
86.1%
Stroke
0 - Yok
1 - Var
0 - Yok
1 - Var
Hipertansiyon

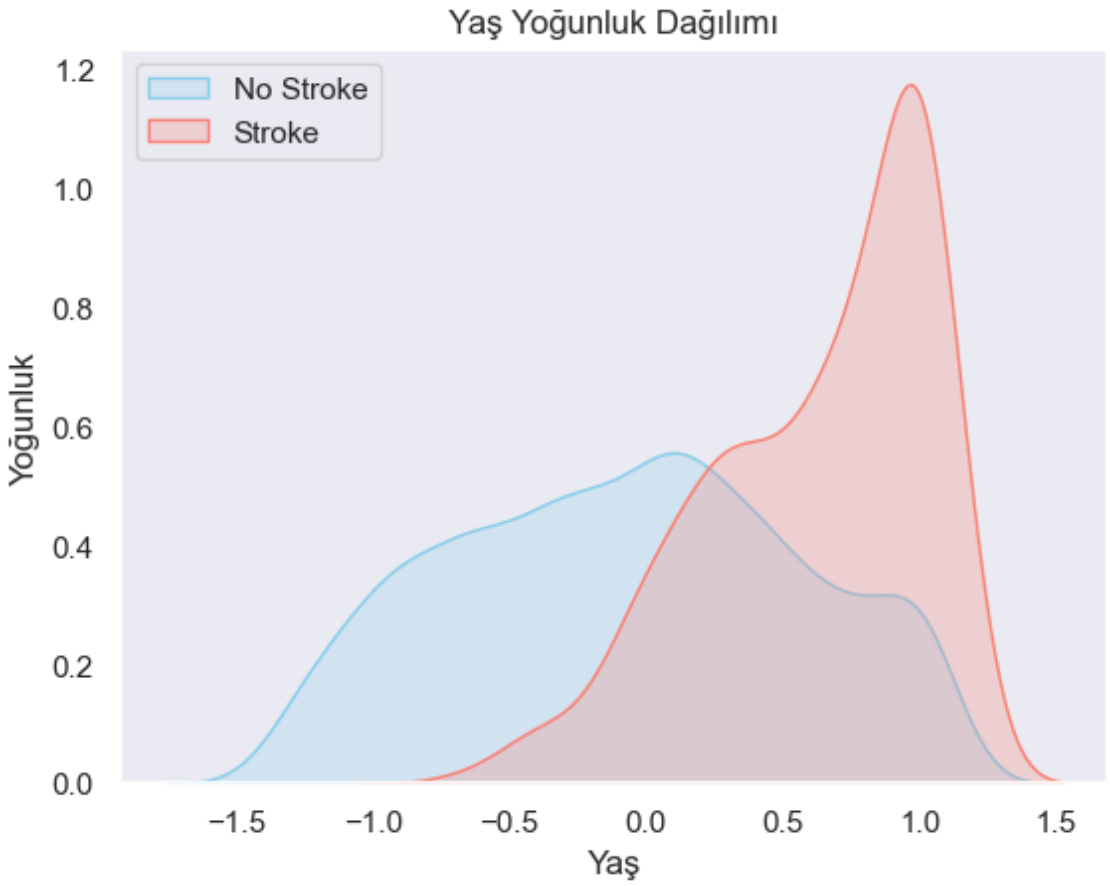

Kalp Hastalığına Göre İnme (Stroke) Dağılımı
Yüzde (%)
100
80
60
40
20
0
4.9%
95.1%
17.1%
82.9%
Stroke
0 - Yok
1 - Var
0 - Yok
1 - Var
Kalp Hastalığı Durumu

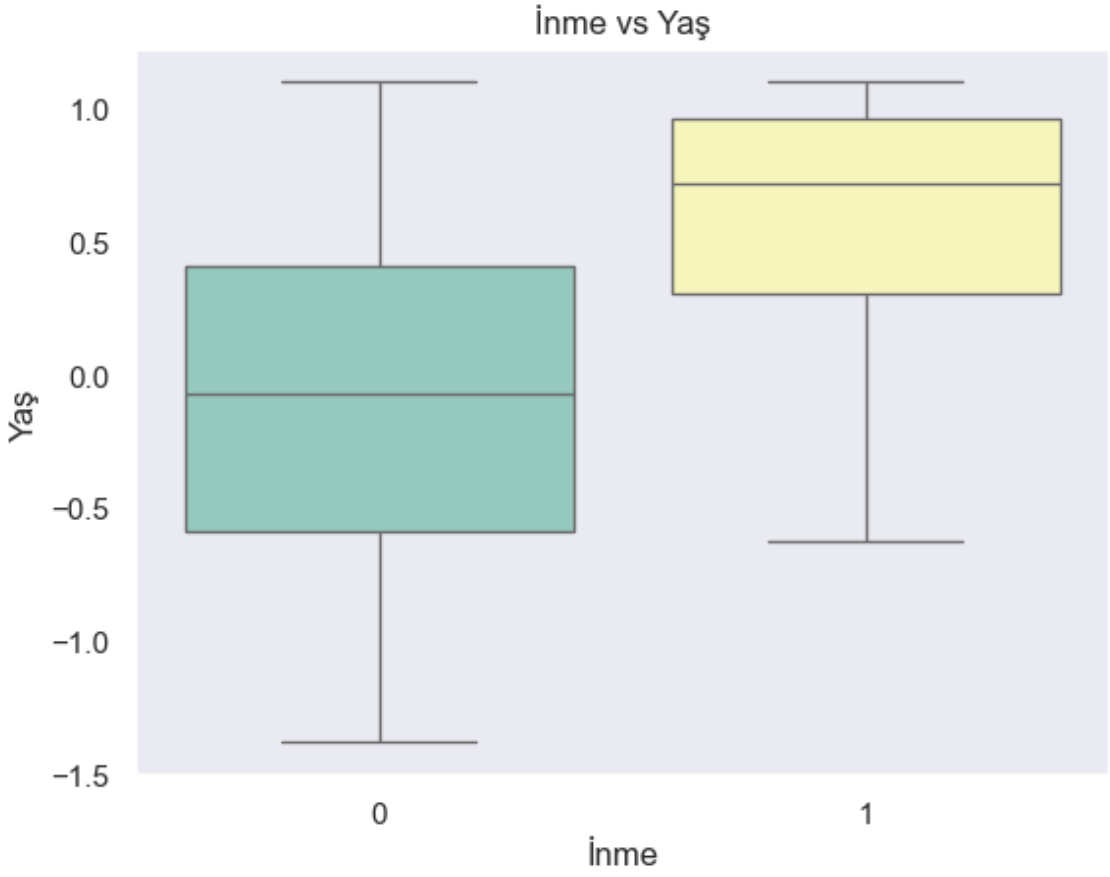

Medeni Duruma Göre İnme (Stroke) Dağılımı
Yüzde (%)
100
80
60
40
20
0
2.5%
6.7%
97.5%
93.3%
Stroke
0 - Yok
1 - Var
No
Yes
Medeni Durum

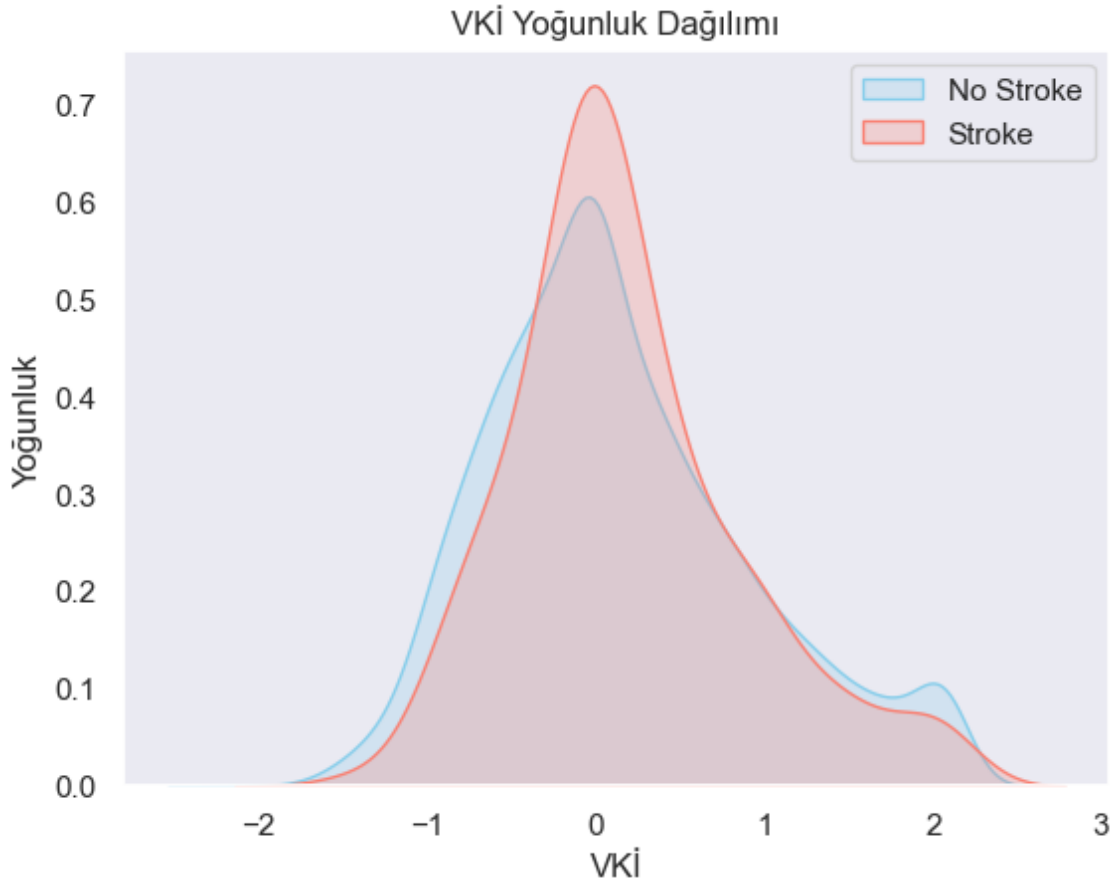

Yerleşim Türüne Göre İnme (Stroke) Dağılımı
Yüzde (%)
100
80
60
40
20
0
5.5%
5.9%
94.5%
94.1%
Stroke
0 - Yok
1 - Var
0
1
Yerleşim Türü

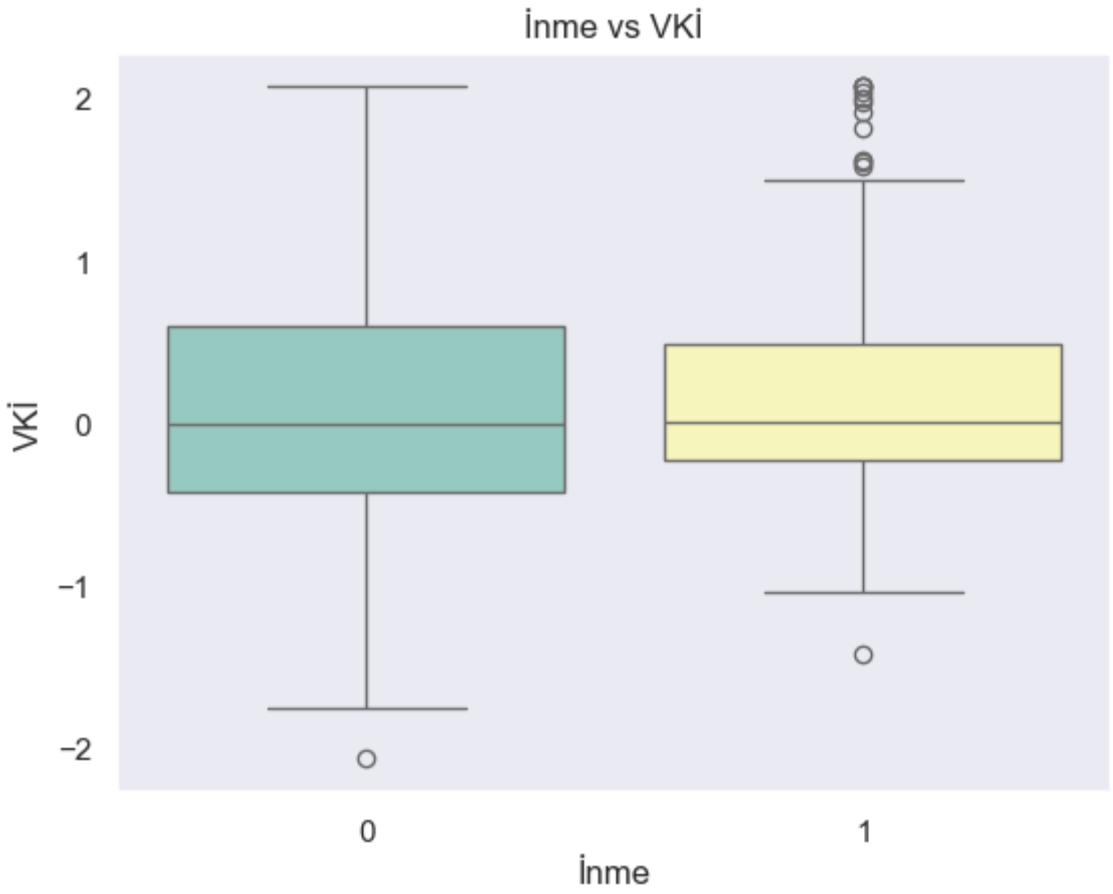

Sigara Kullanımına Göre İnme (Stroke) Dağılımı
Yüzde (%)
100
80
60
40
20
0
7.9%
4.8%
5.3%
92.1%
95.2%
94.7%
Stroke
0 - Yok
1 - Var
0
1
2
Sigara Kullanımı


İş Türüne Göre İnme (Stroke) Dağılımı
Yüzde (%)
100
80
60
40
20
0
4.7%
5.3%
8.6%
95.3%
94.7%
91.4%
100.0%
Stroke
0 - Yok
1 - Var
0
1
2
3
İş Türü

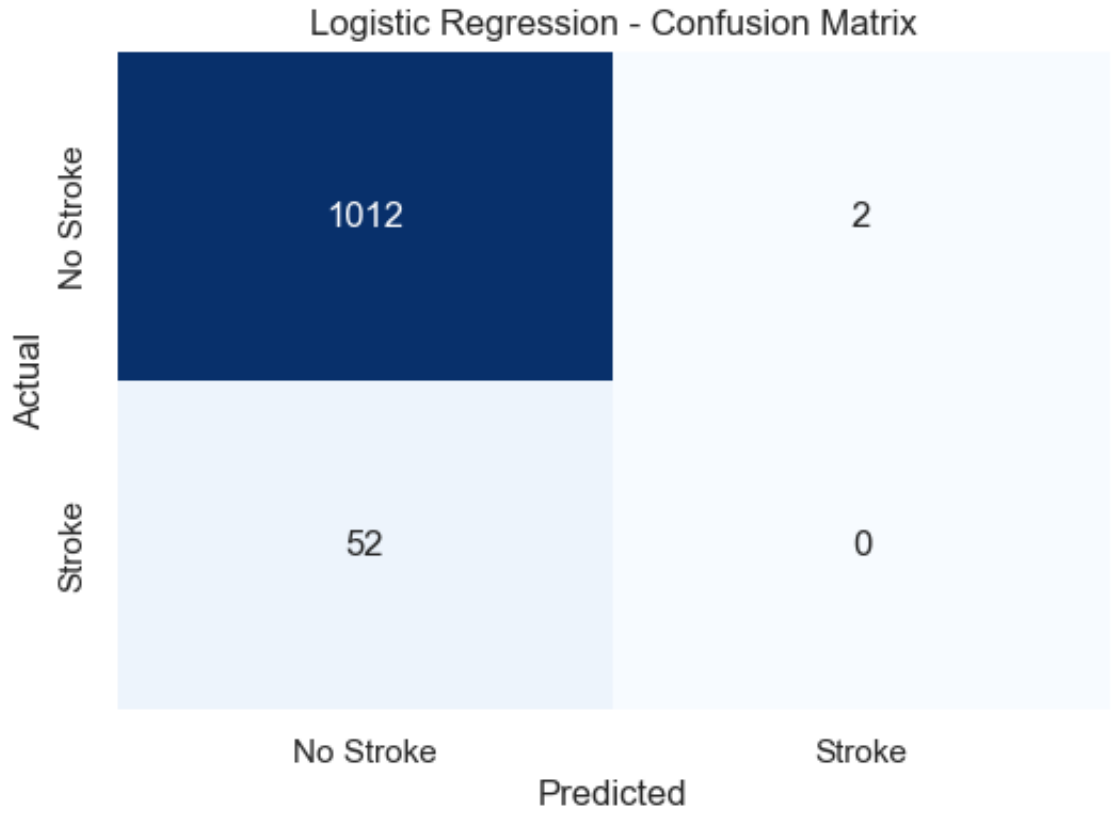

Yaş Yoğunluk Dağılımı
No Stroke
Stroke
Yoğunluk
1.2
1.0
0.8
0.6
0.4
0.2
0.0
−1.5
−1.0
−0.5
0.0
0.5
1.0
1.5
Yaş

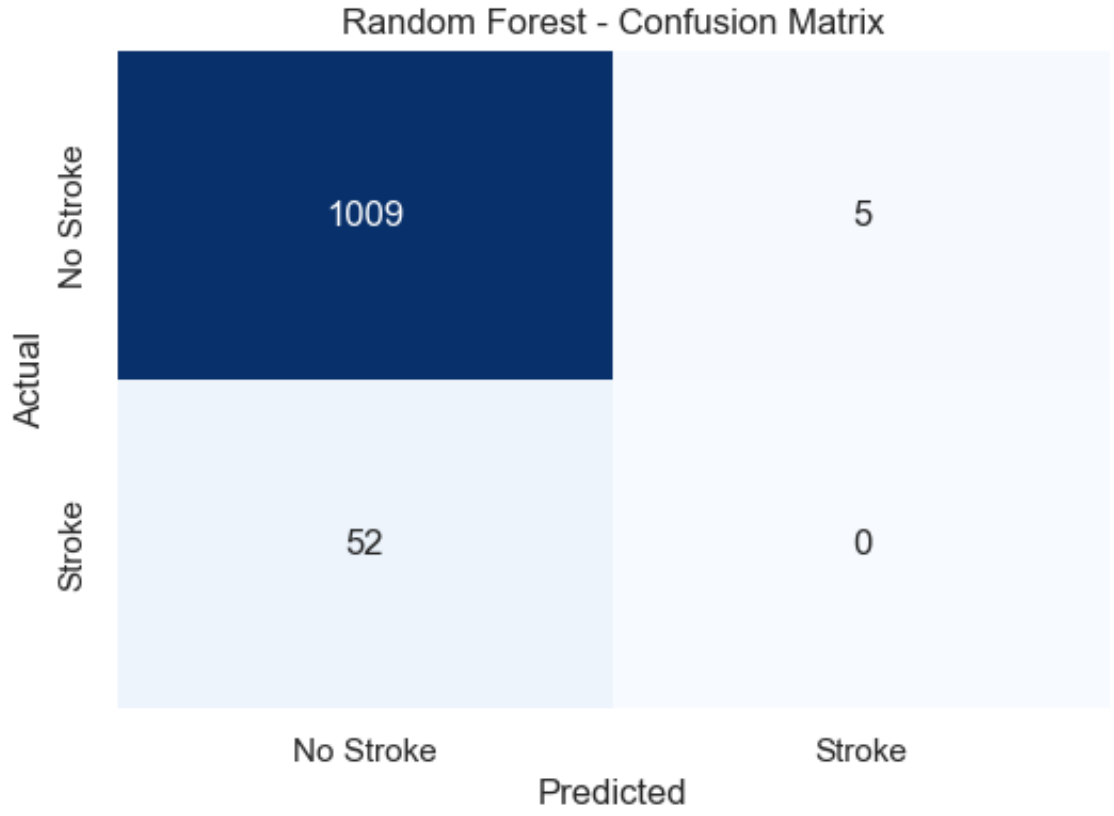

İnme vs Yaş
Yaş
1.0
0.5
0.0
−0.5
−1.0
−1.5
0
1
İnme

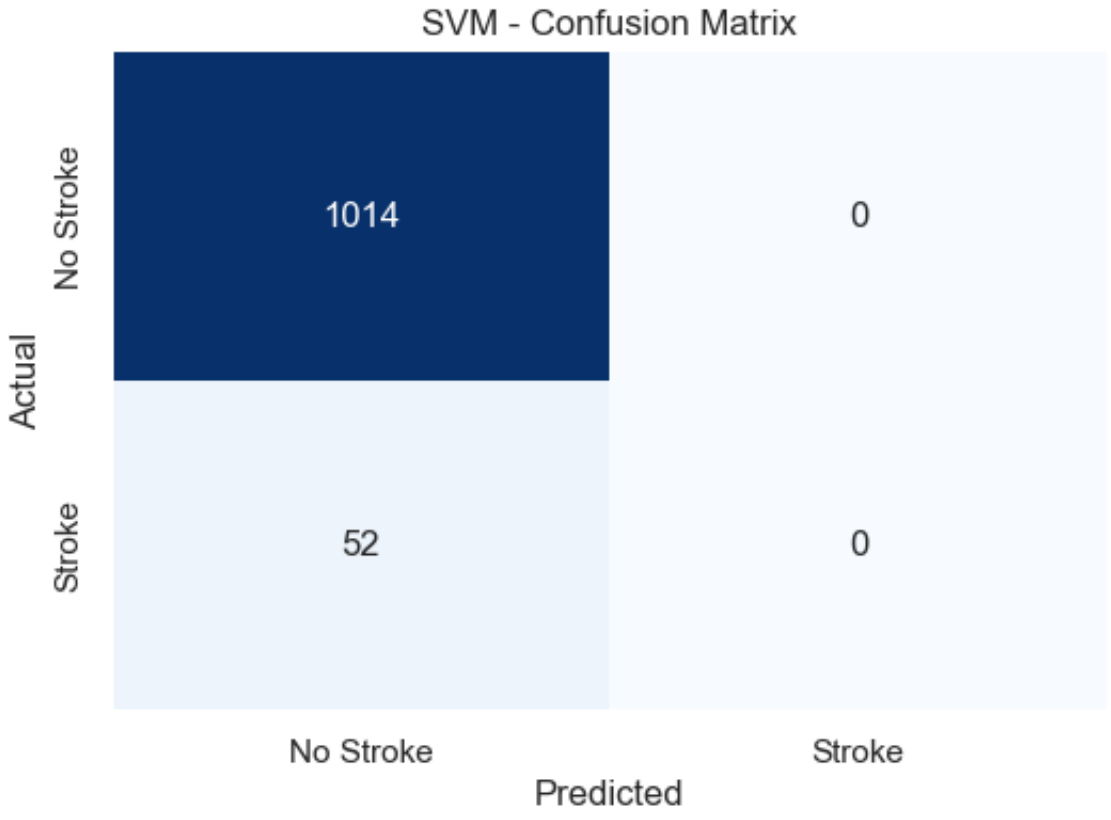

VKİ Yoğunluk Dağılımı
No Stroke
Stroke
Yoğunluk
0.0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
−2
−1
0
1
2
3
VKİ

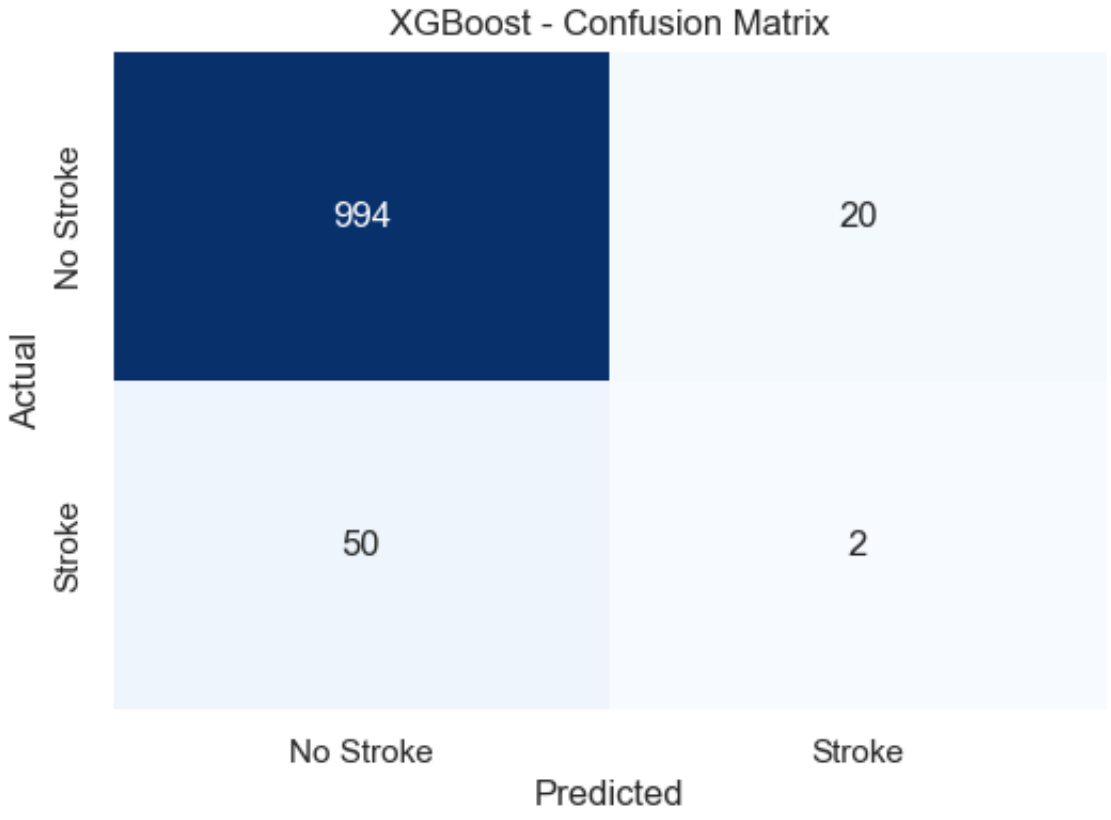

İnme vs VKİ
2
1
0
−1
−2
VKİ
0
1
İnme

Orijinal Veri kümesi

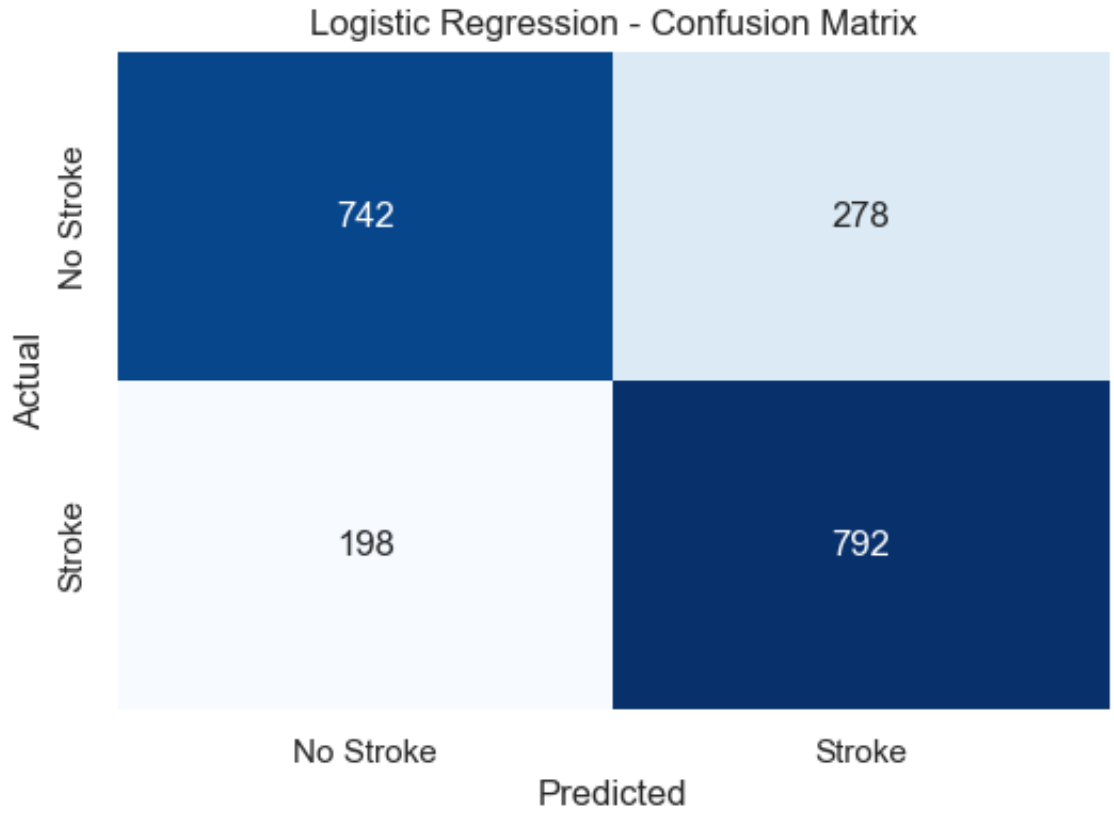


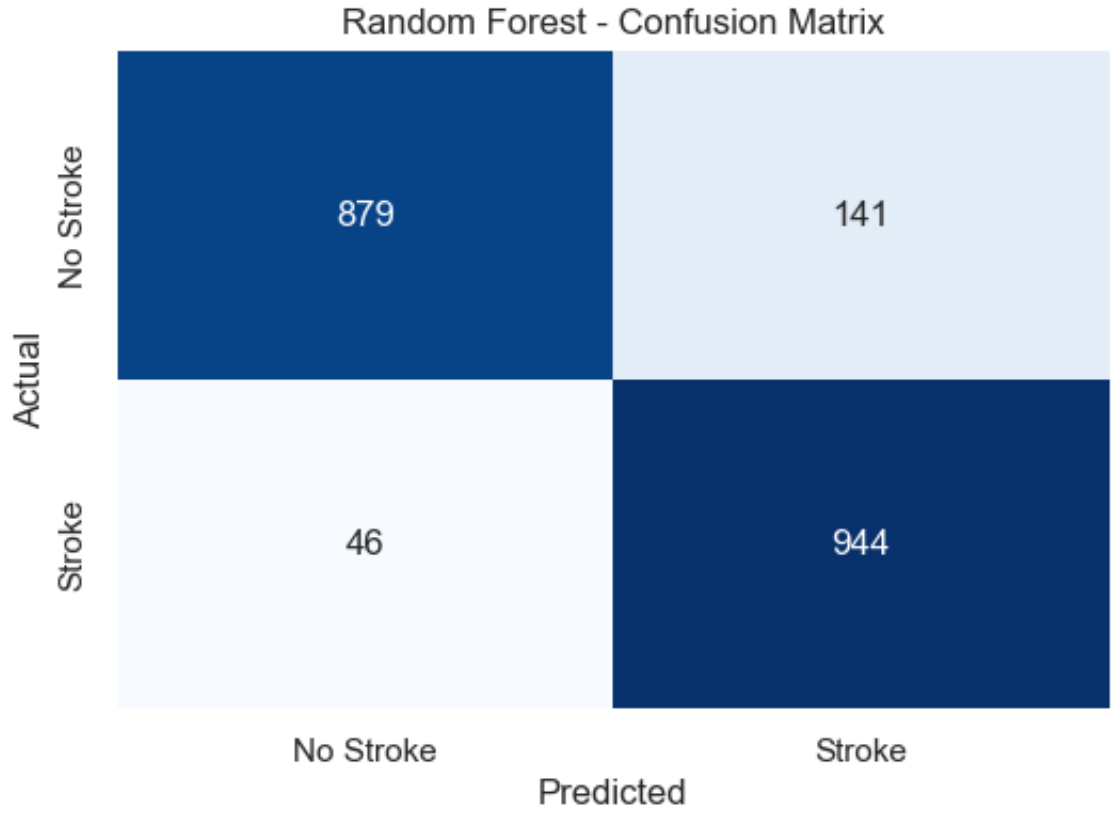


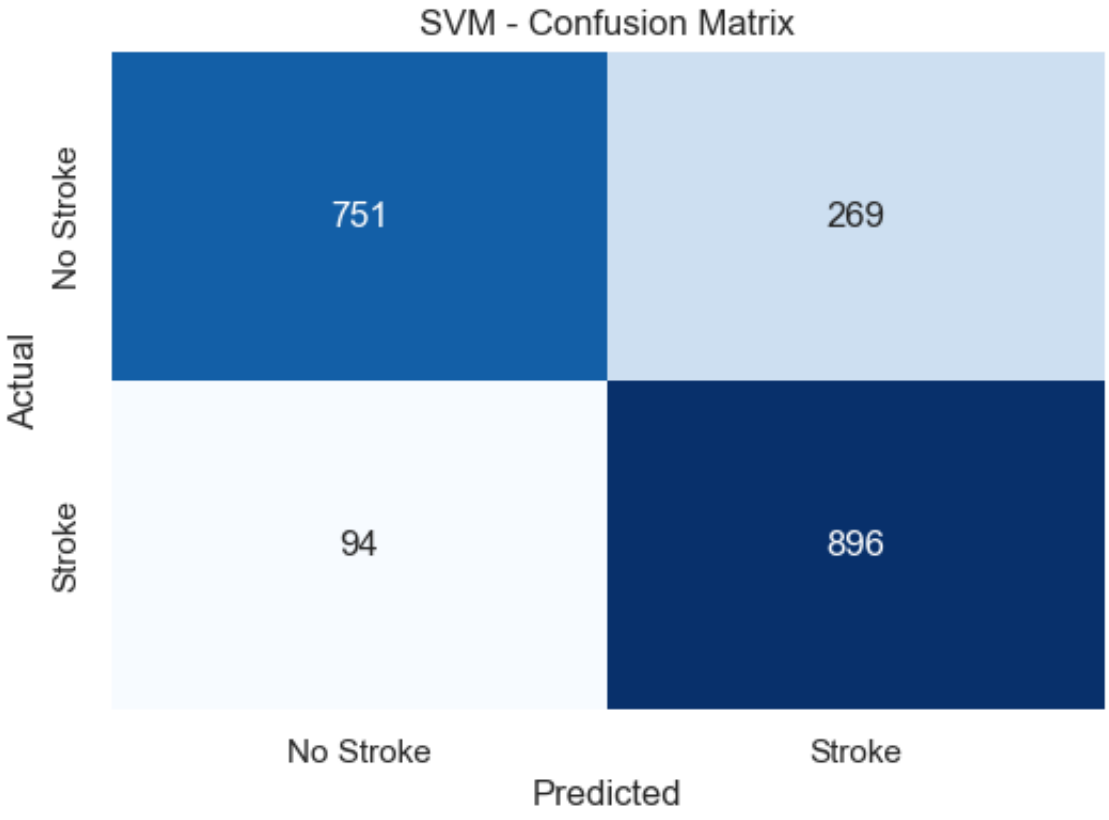

SVM - Confusion Matrix
Actual
No Stroke
Stroke
1014
0
52
0
No Stroke
Stroke
Predicted

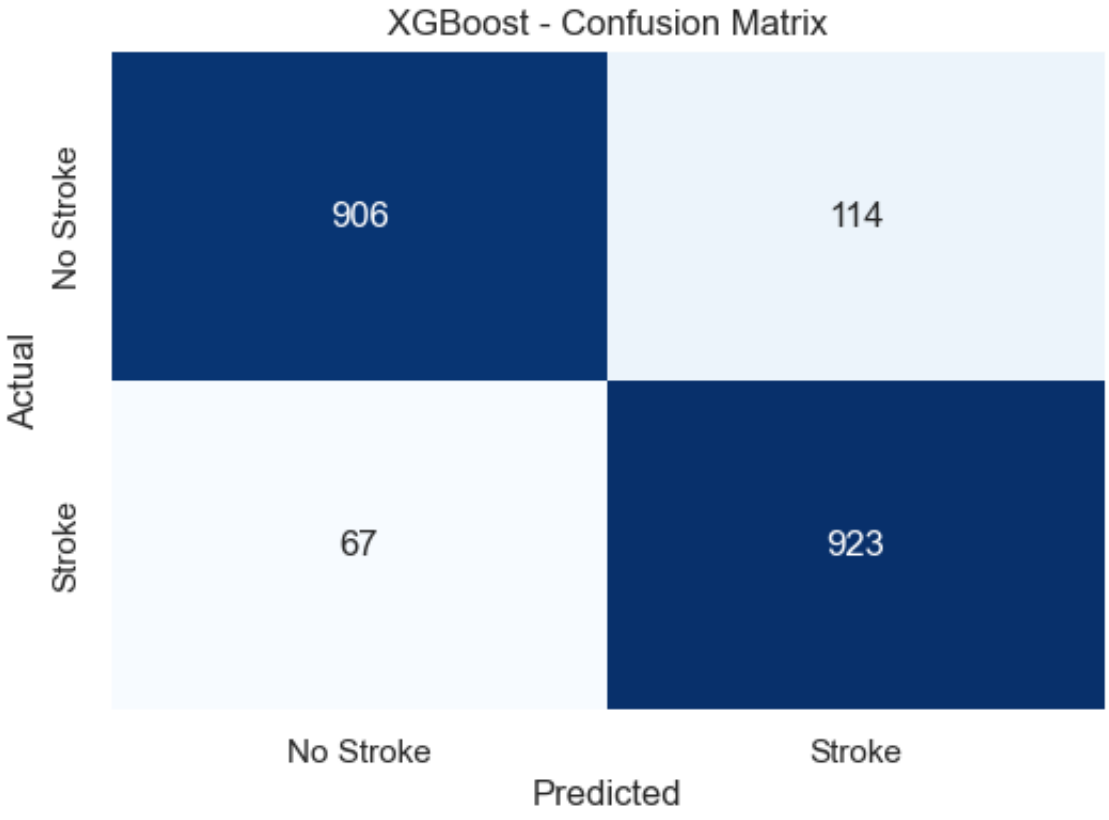

XGBoost - Confusion Matrix
Actual
No Stroke
Stroke
994
20
50
2
No Stroke
Stroke
Predicted

Temel SMOTE

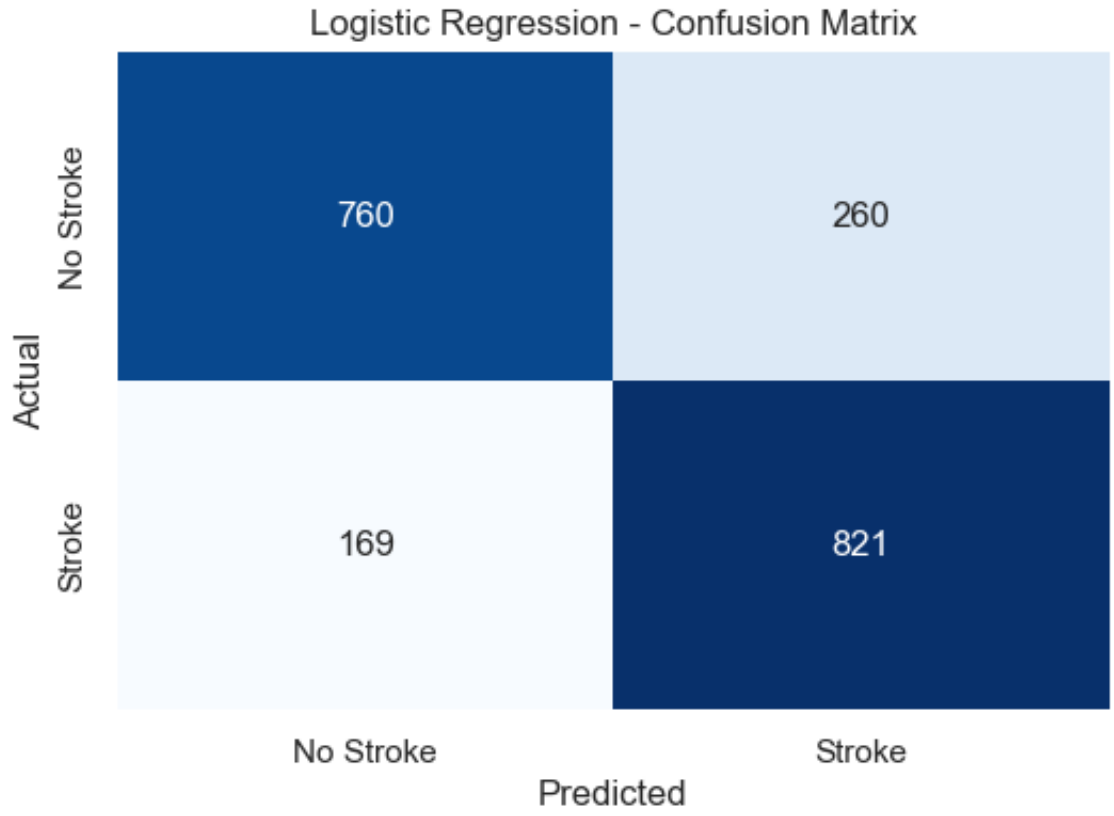

Logistic Regression - Confusion Matrix
742
278
198
792
No Stroke
Stroke
Actual
No Stroke
Stroke
Predicted

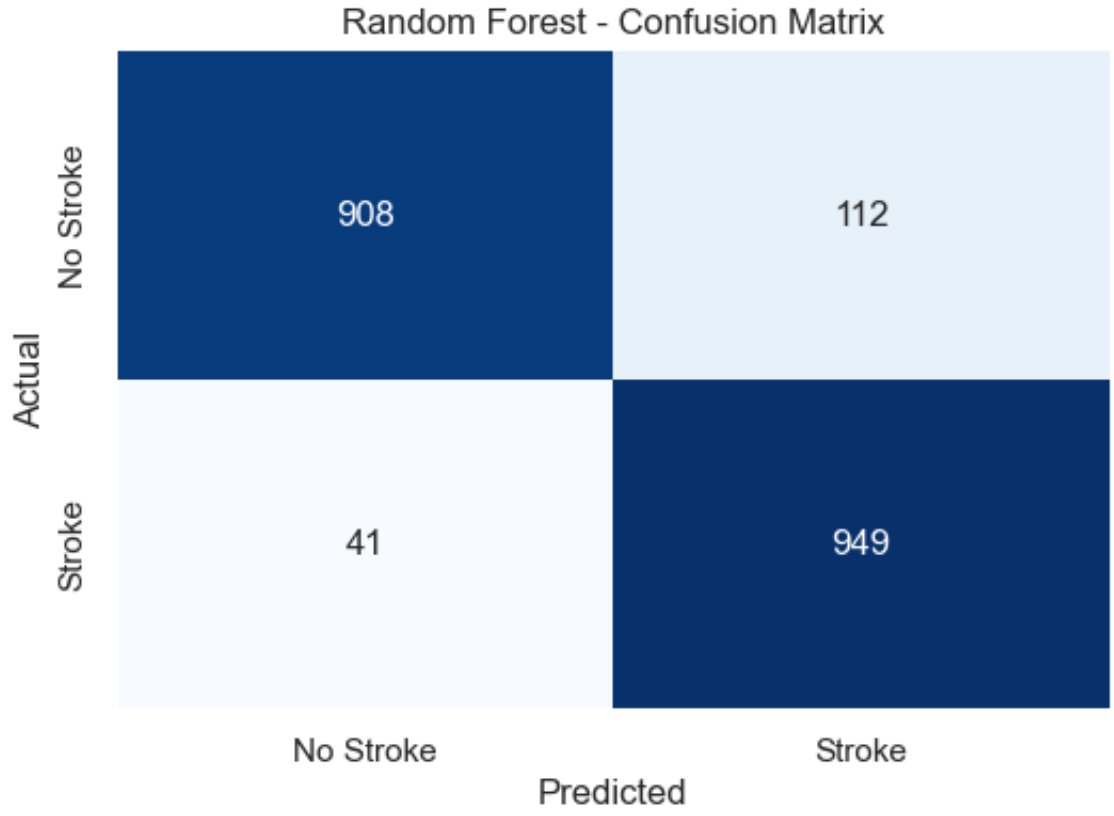

Random Forest - Confusion Matrix
879
141
46
944
No Stroke
Stroke
Actual
No Stroke
Stroke
Predicted

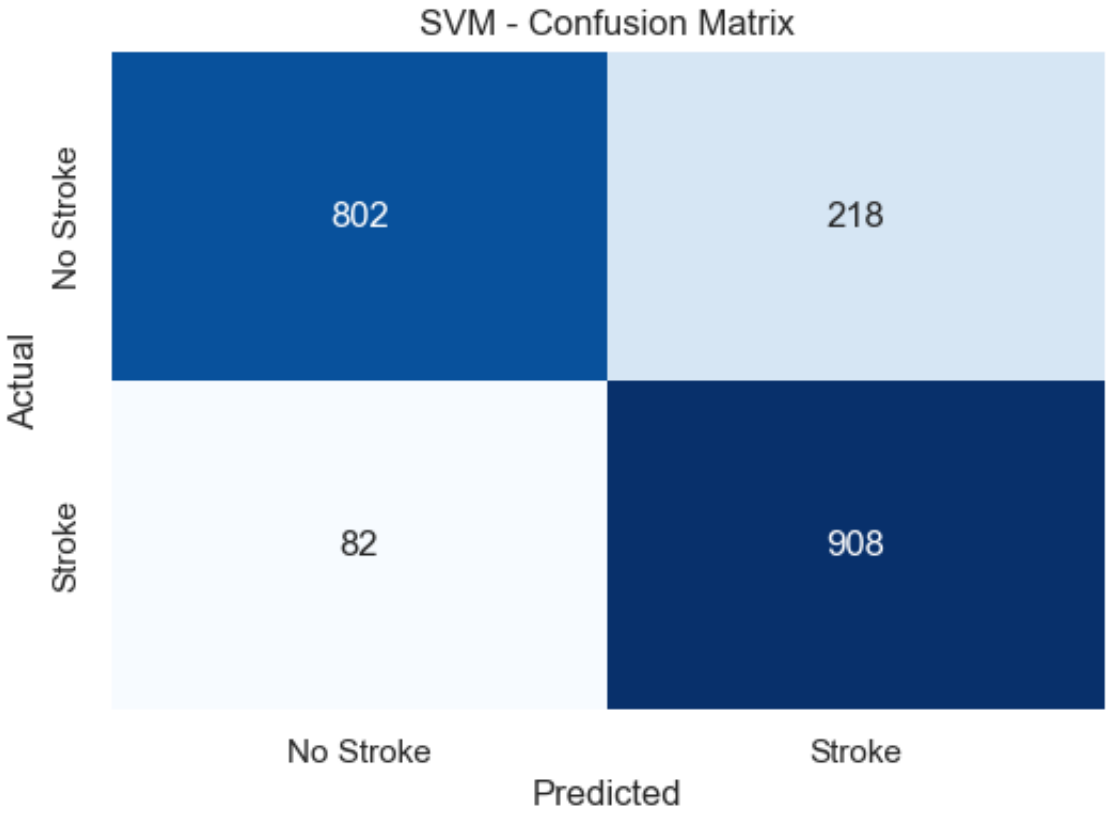

SVM - Confusion Matrix
751
269
94
896
No Stroke
Stroke
Actual
No Stroke
Stroke
Predicted

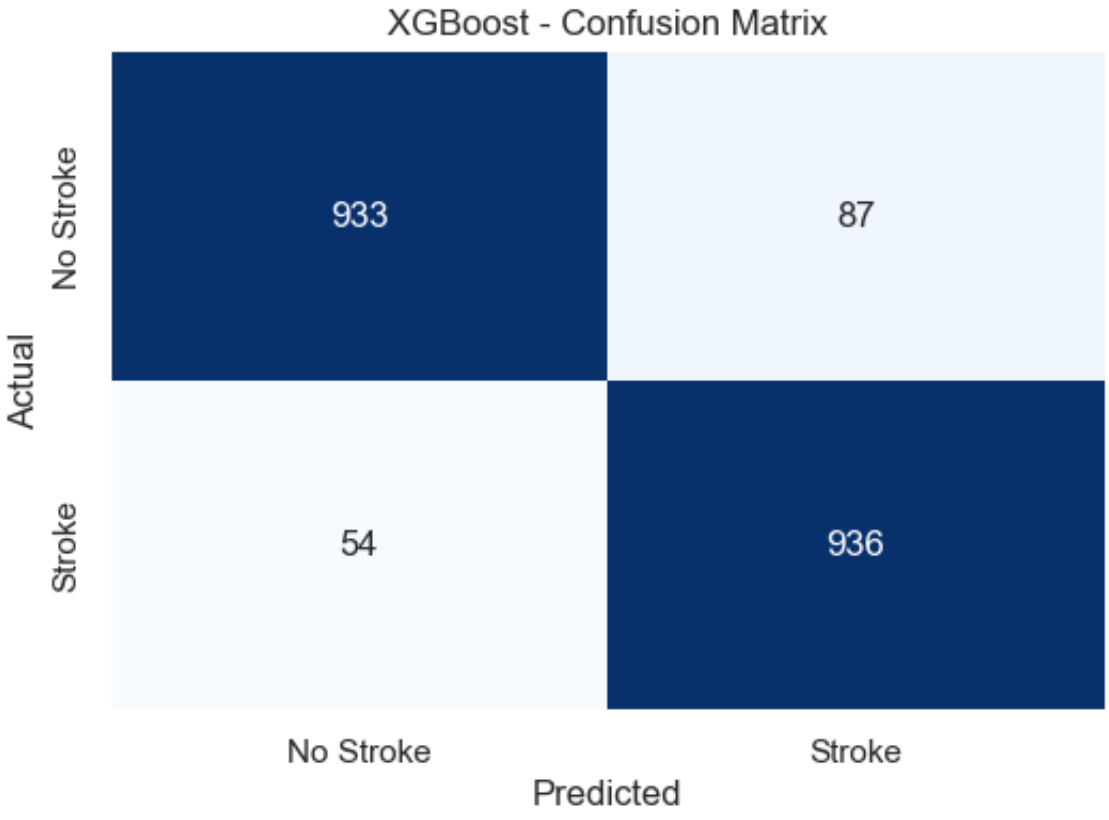

XGBoost - Confusion Matrix
906
114
67
923
No Stroke
Stroke
Actual
No Stroke
Stroke
Predicted

Borderline-SMOTE

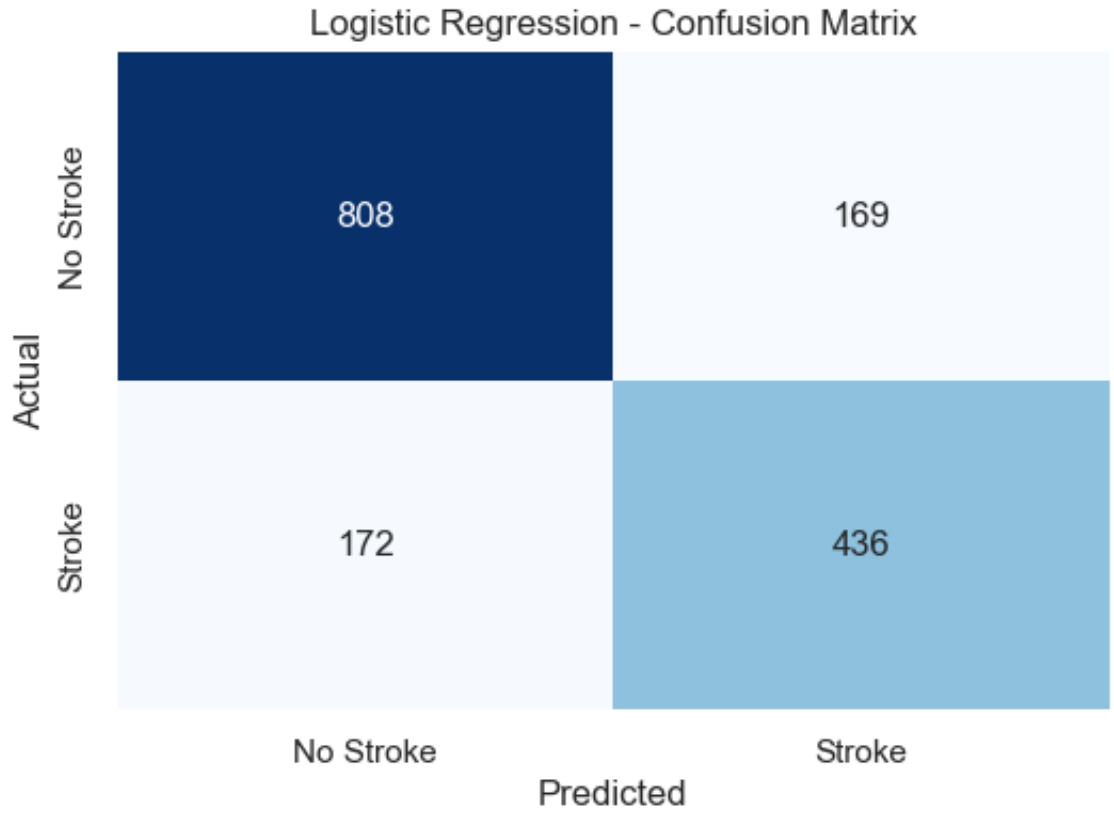


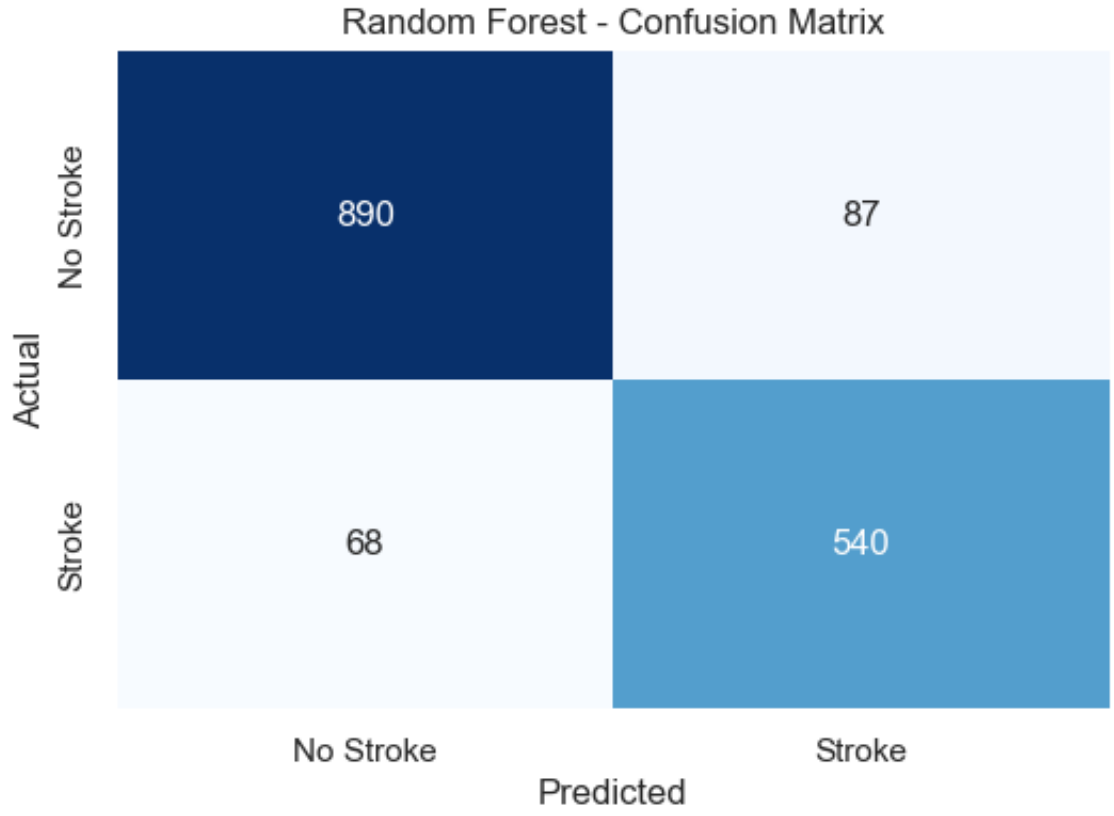


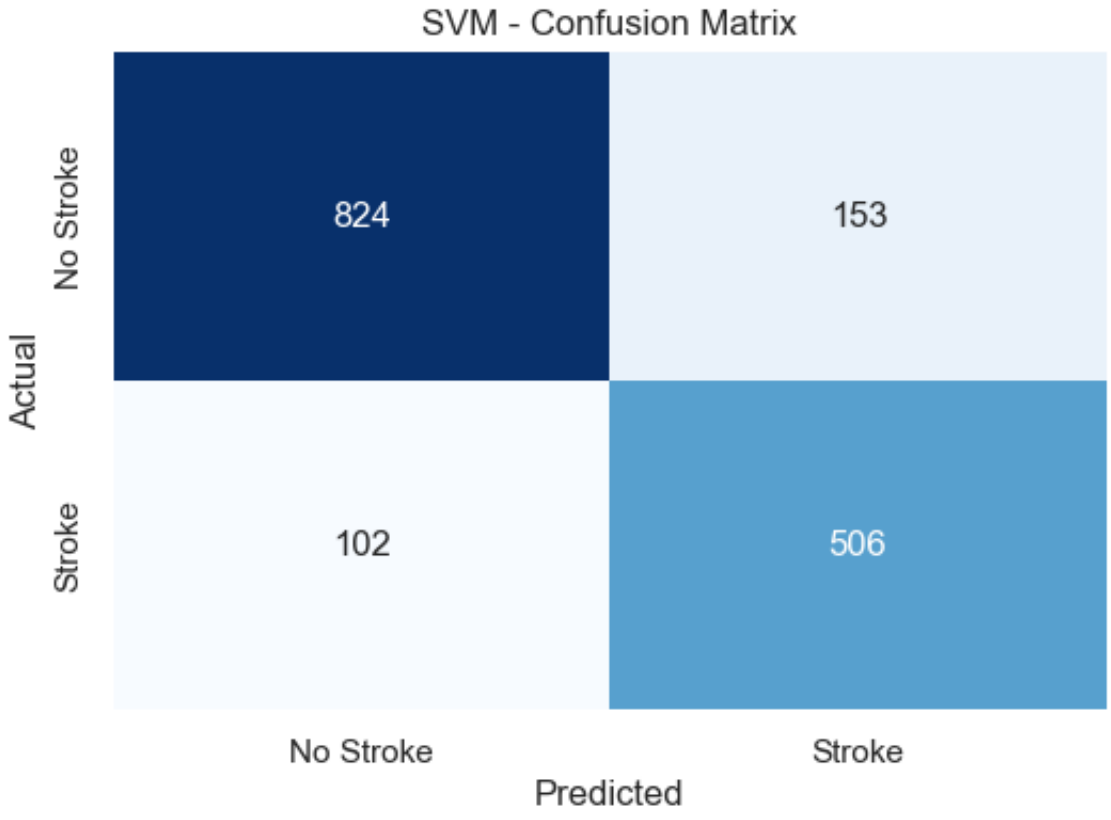

SVM - Confusion Matrix
802
218
82
908
No Stroke
Stroke
Actual
No Stroke
Stroke
Predicted

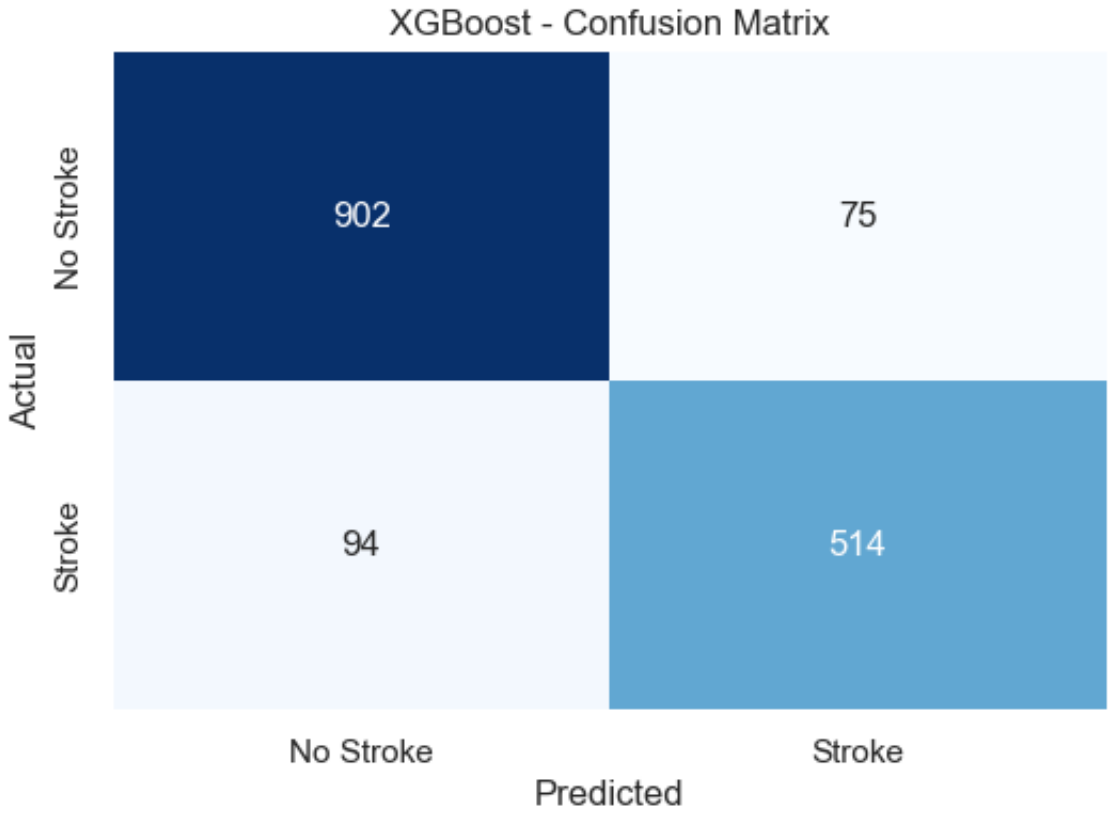

XGBoost - Confusion Matrix
933
87
54
936
No Stroke
Stroke
Actual
No Stroke
Stroke
Predicted

SVM-SMOTE

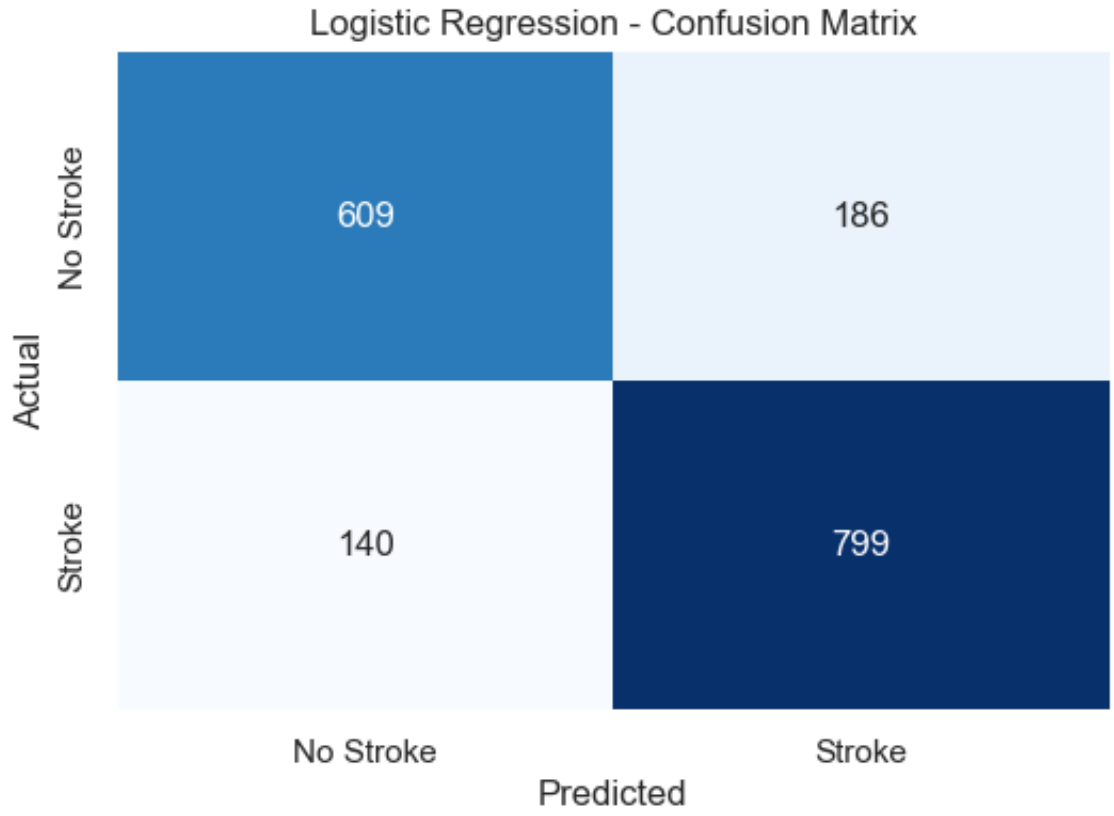

Logistic Regression - Confusion Matrix
808
169
172
436
No Stroke
Stroke
Actual
No Stroke
Stroke
Predicted

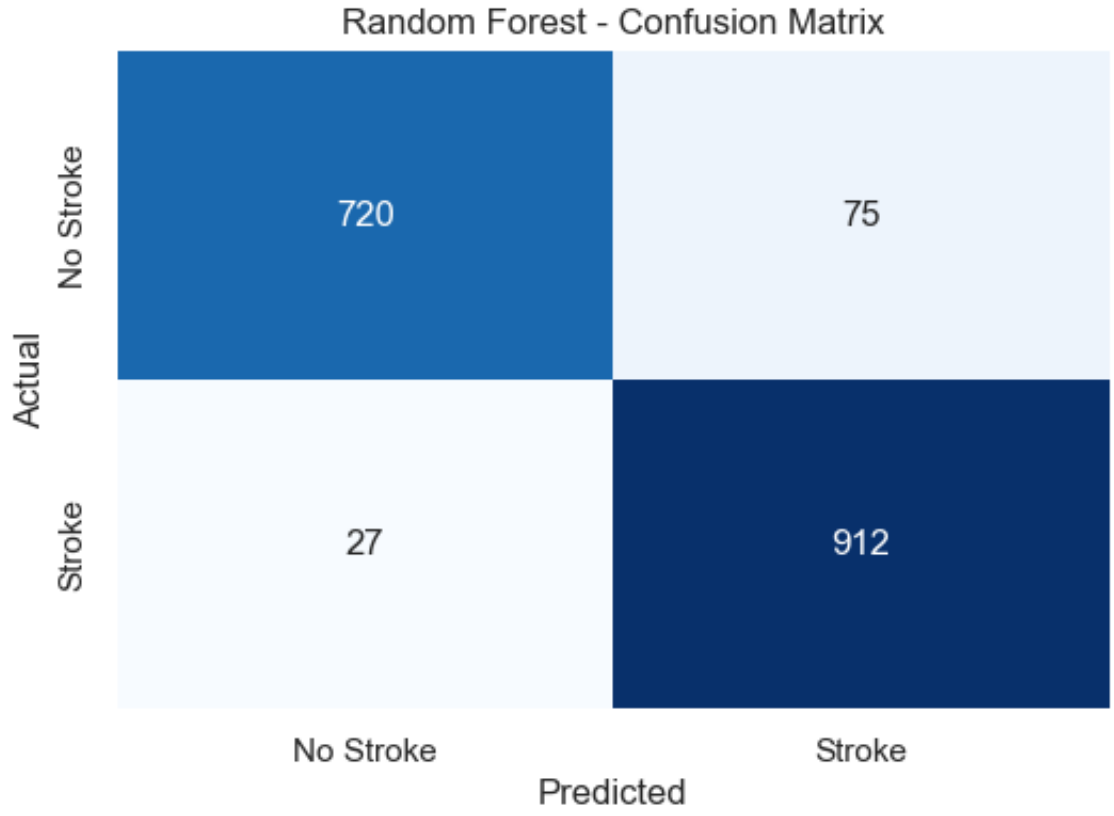

Random Forest - Confusion Matrix
890
87
68
540
No Stroke
Stroke
Actual
No Stroke
Stroke
Predicted

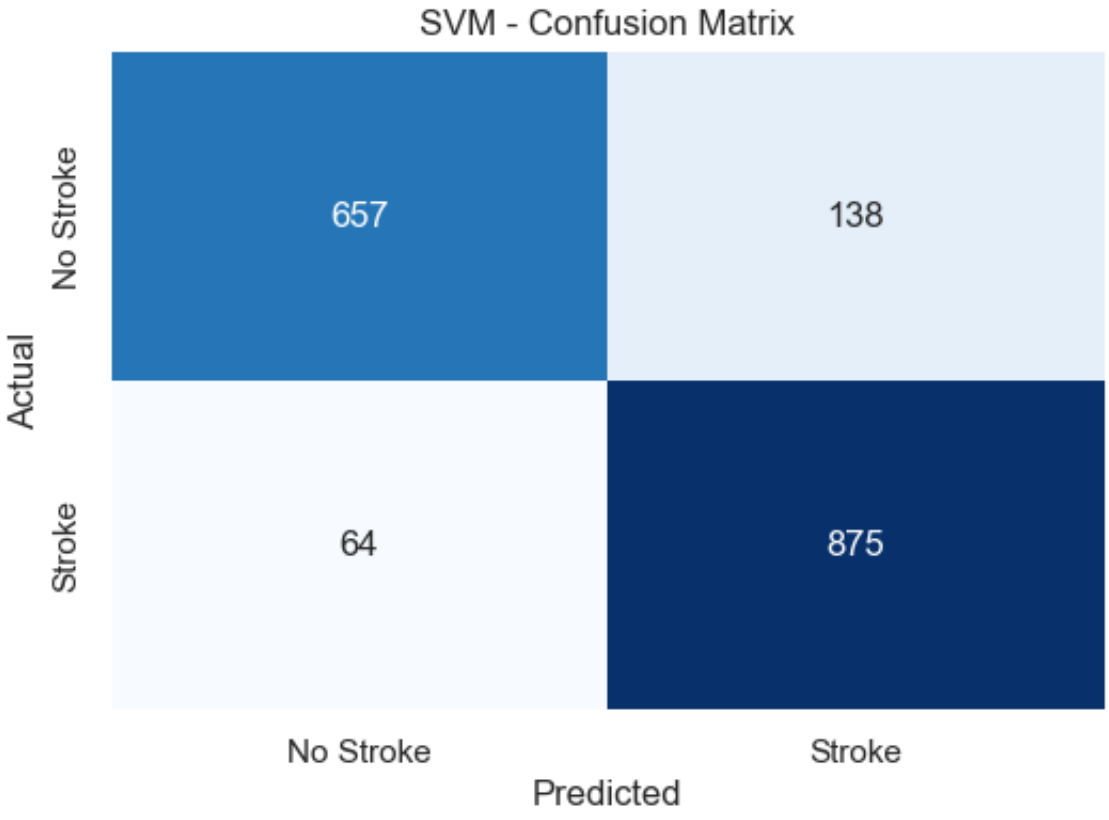

SVM - Confusion Matrix
824
153
102
506
No Stroke
Stroke
Actual
No Stroke
Stroke
Predicted

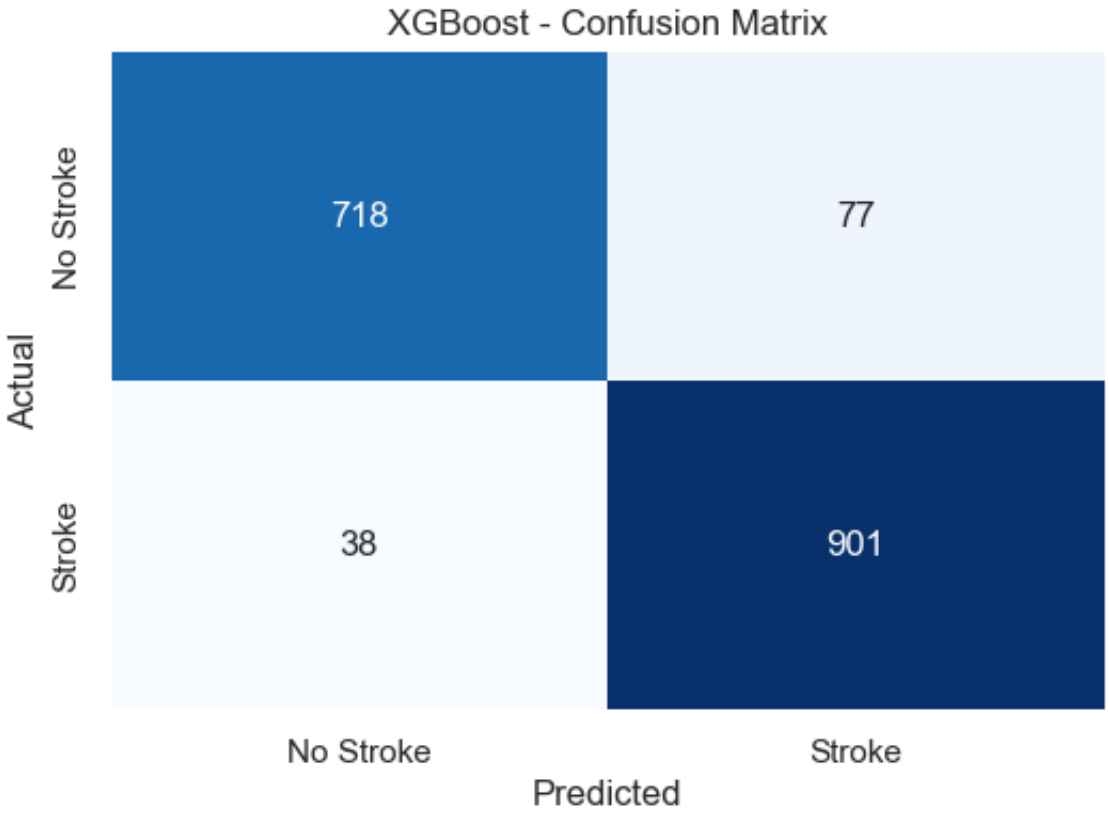

XGBoost - Confusion Matrix
902
75
94
514
No Stroke
Stroke
Actual
No Stroke
Stroke
Predicted

SMOTE-ENN

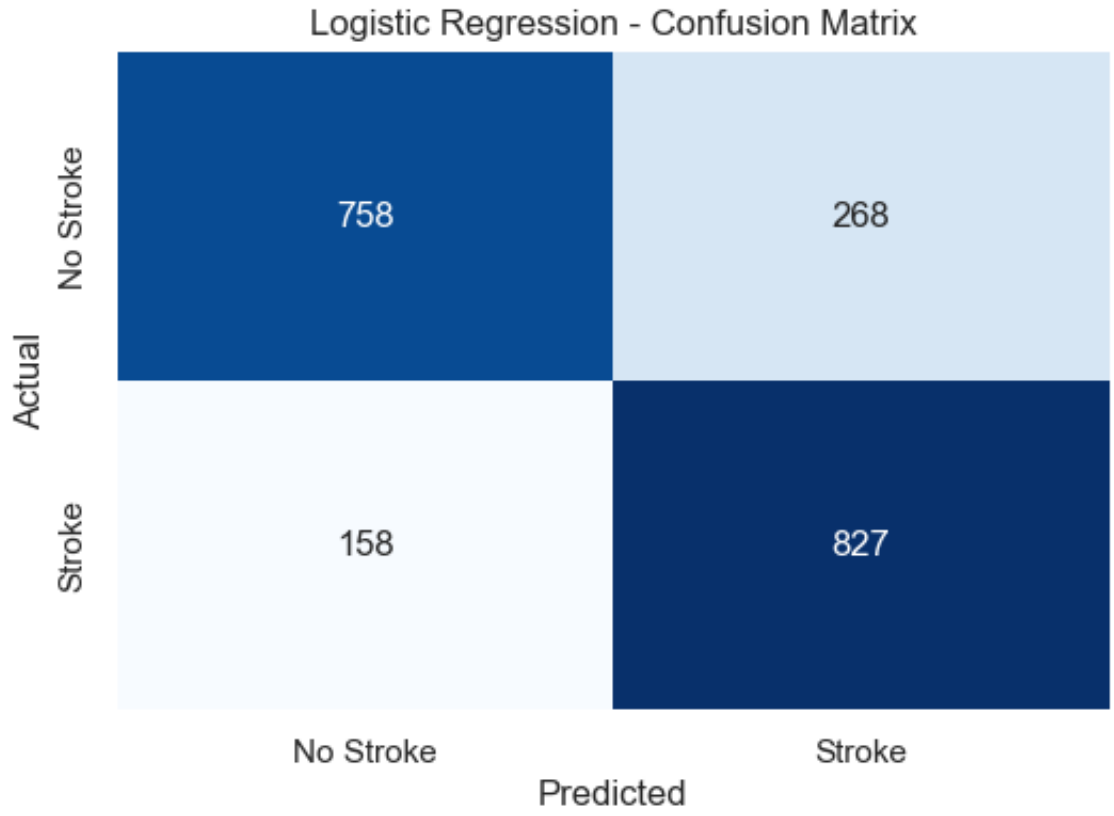

Logistic Regression - Confusion Matrix
609
186
140
799
No Stroke
Stroke
Actual
No Stroke
Stroke
Predicted

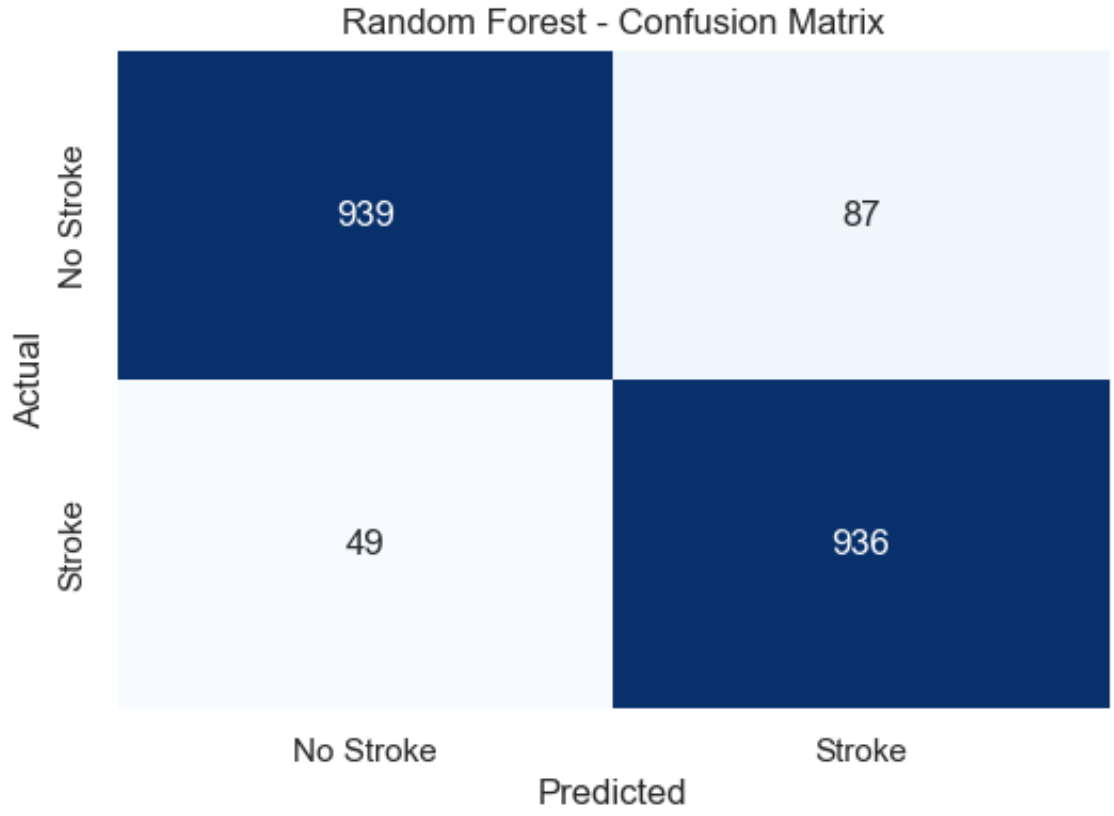

Random Forest - Confusion Matrix
720
75
27
912
No Stroke
Stroke
Actual
No Stroke
Stroke
Predicted

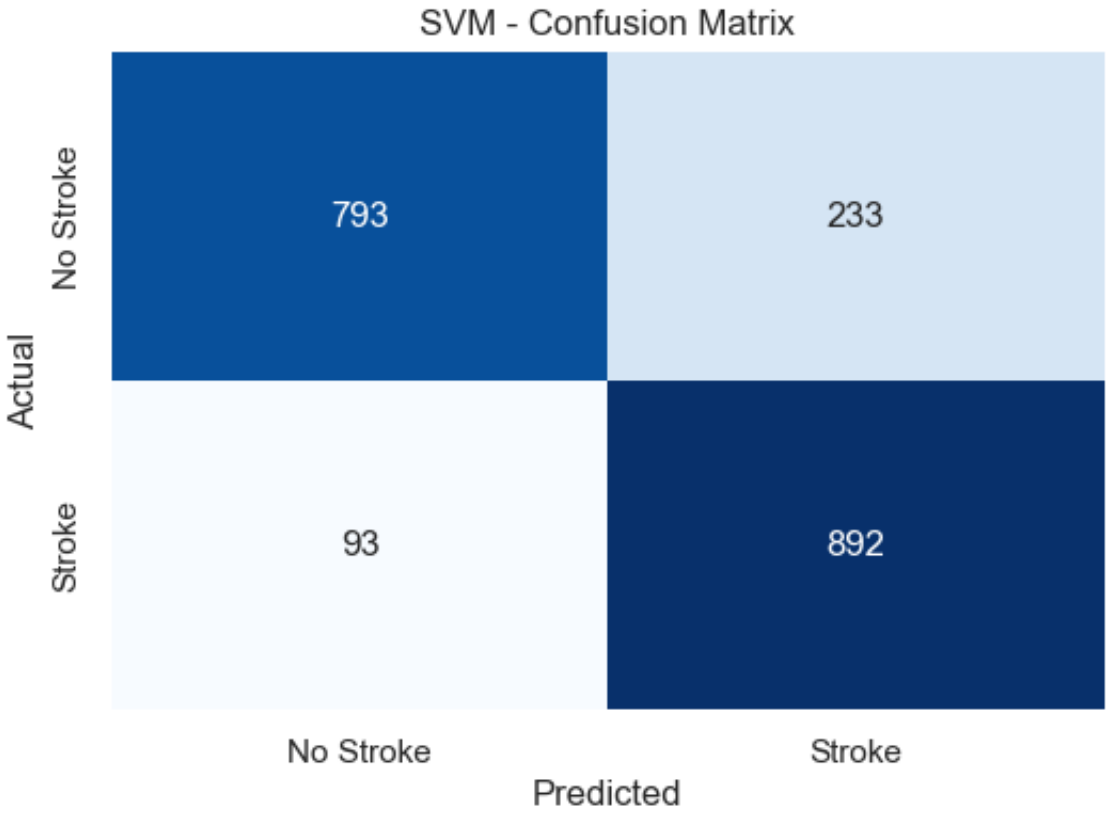

SVM - Confusion Matrix
657
138
64
875
No Stroke
Stroke
Actual
No Stroke
Stroke
Predicted

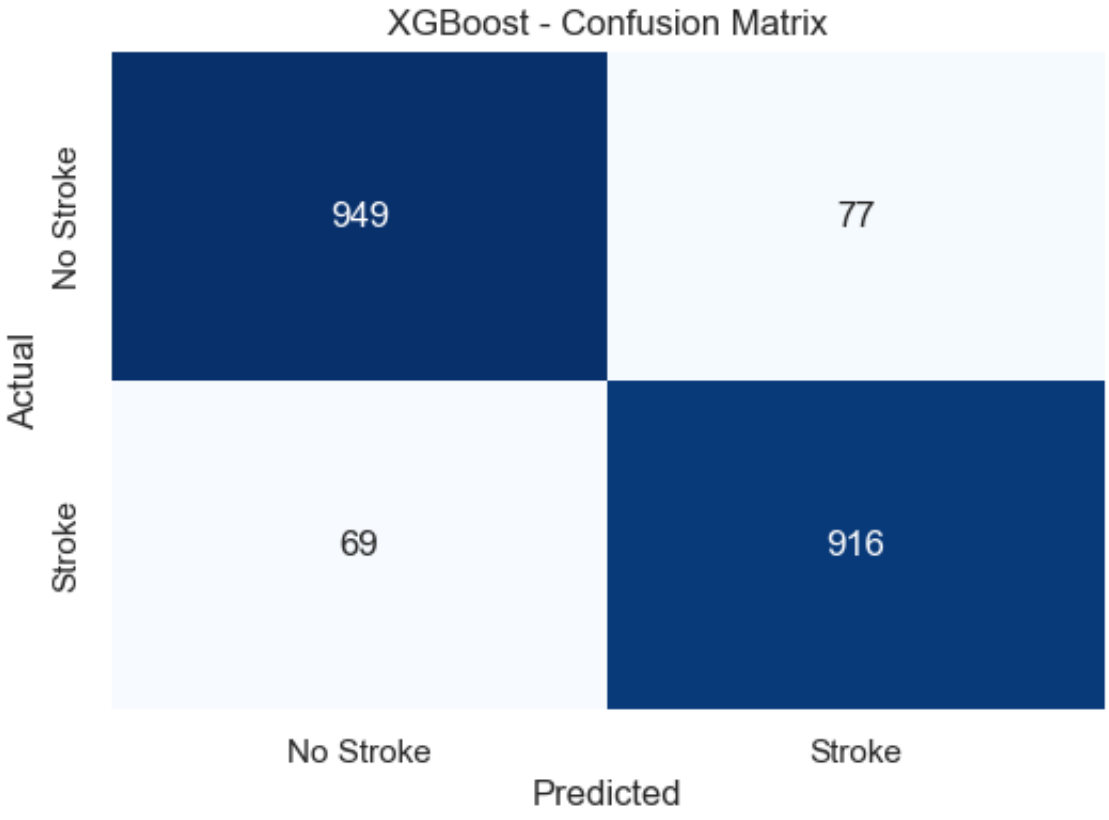

XGBoost - Confusion Matrix
718
77
38
901
No Stroke
Stroke
Actual
No Stroke
Stroke
Predicted

K-Means-SMOTE

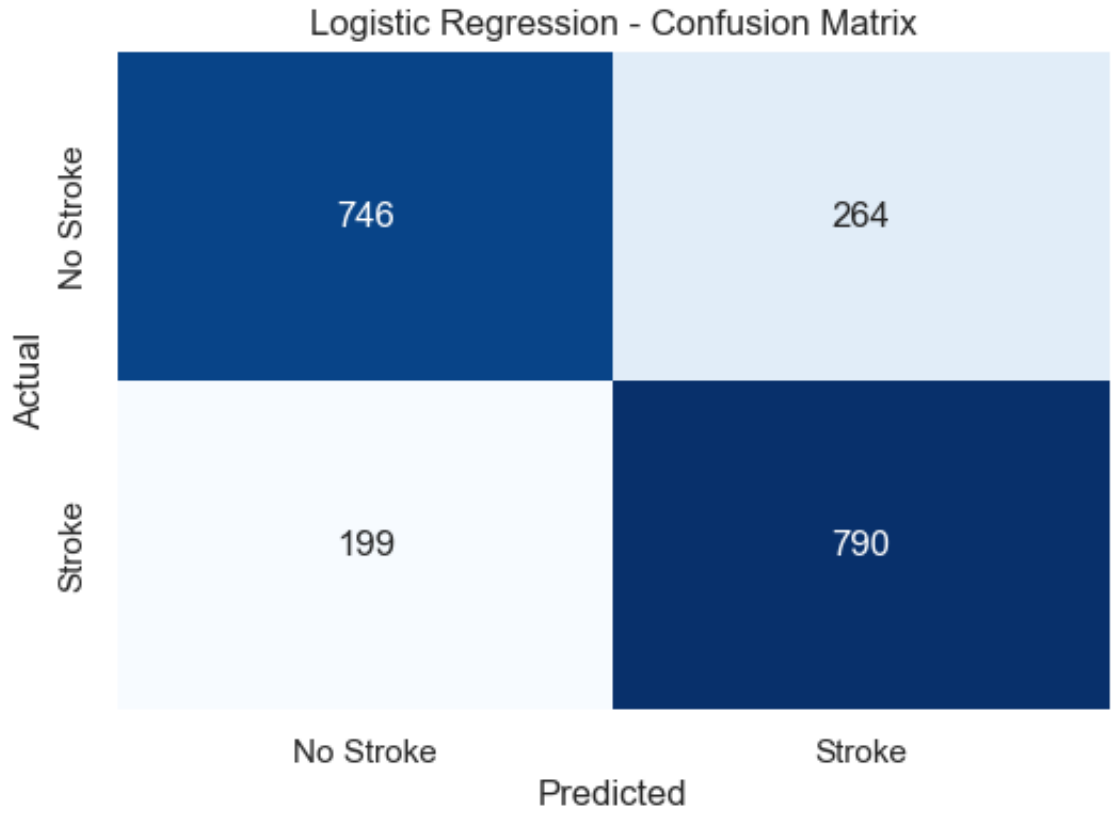


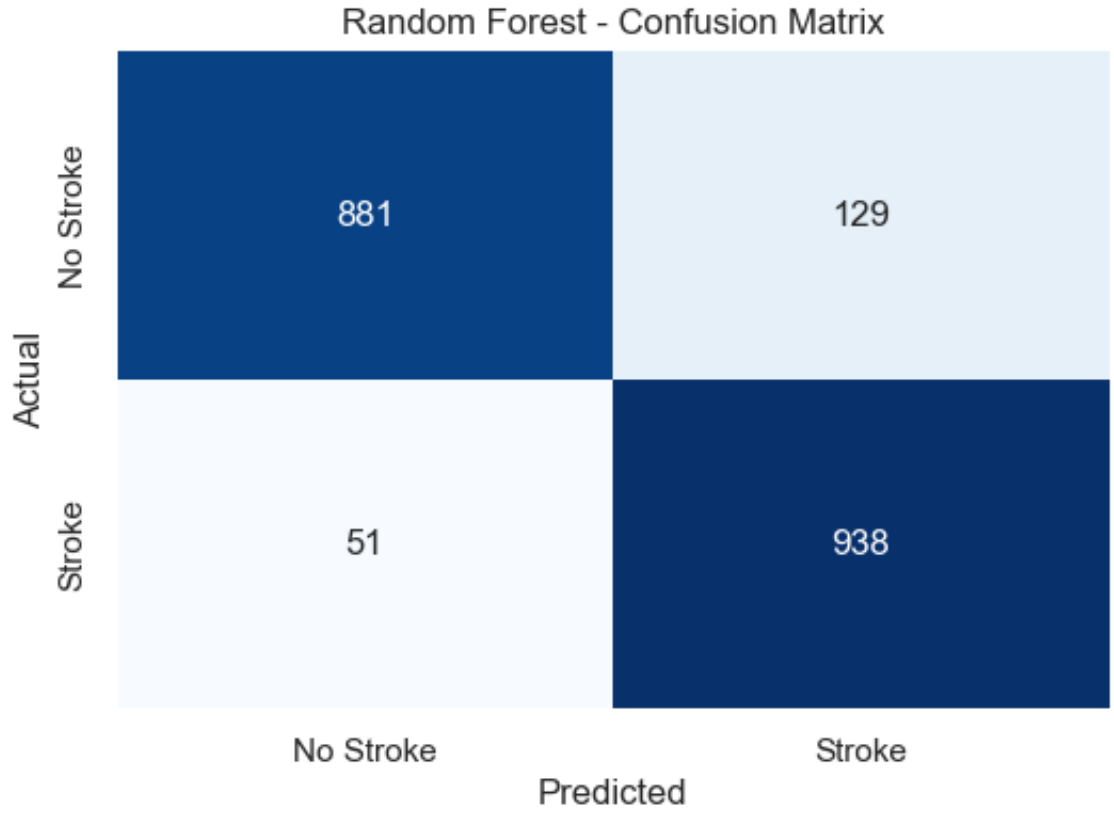


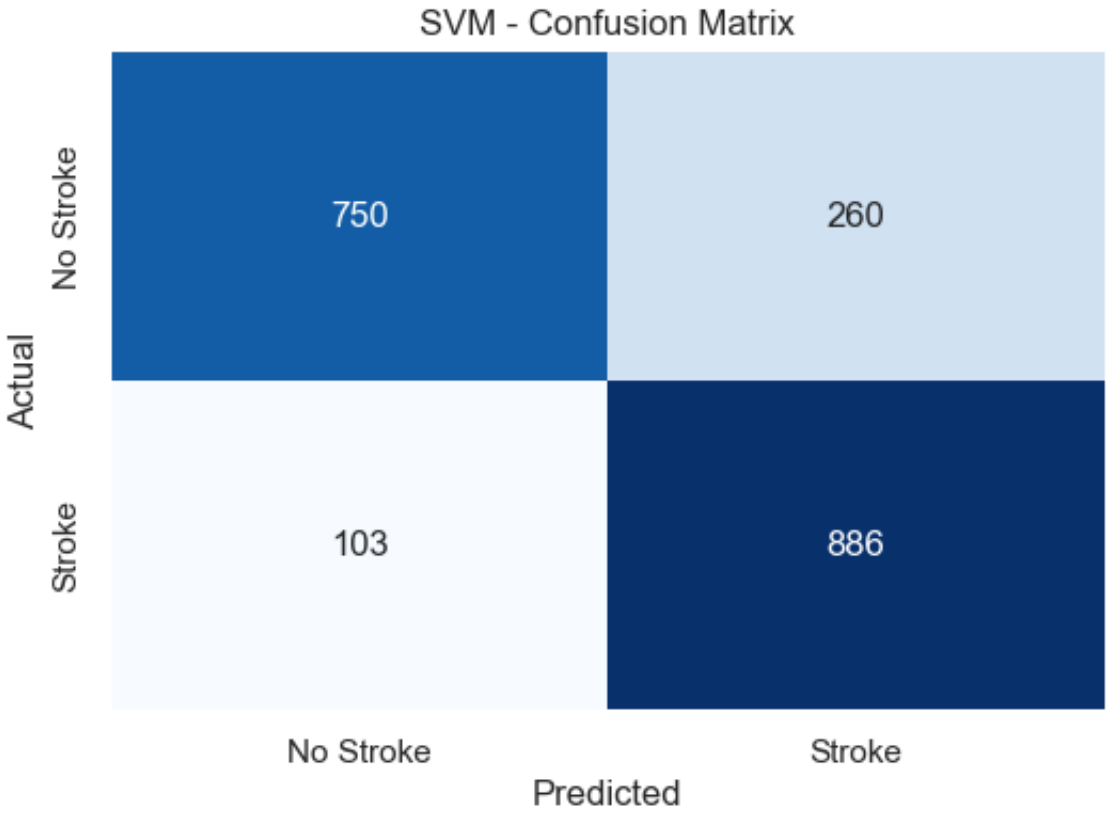

SVM - Confusion Matrix
793
233
93
892
No Stroke
Stroke
Actual
No Stroke
Stroke
Predicted

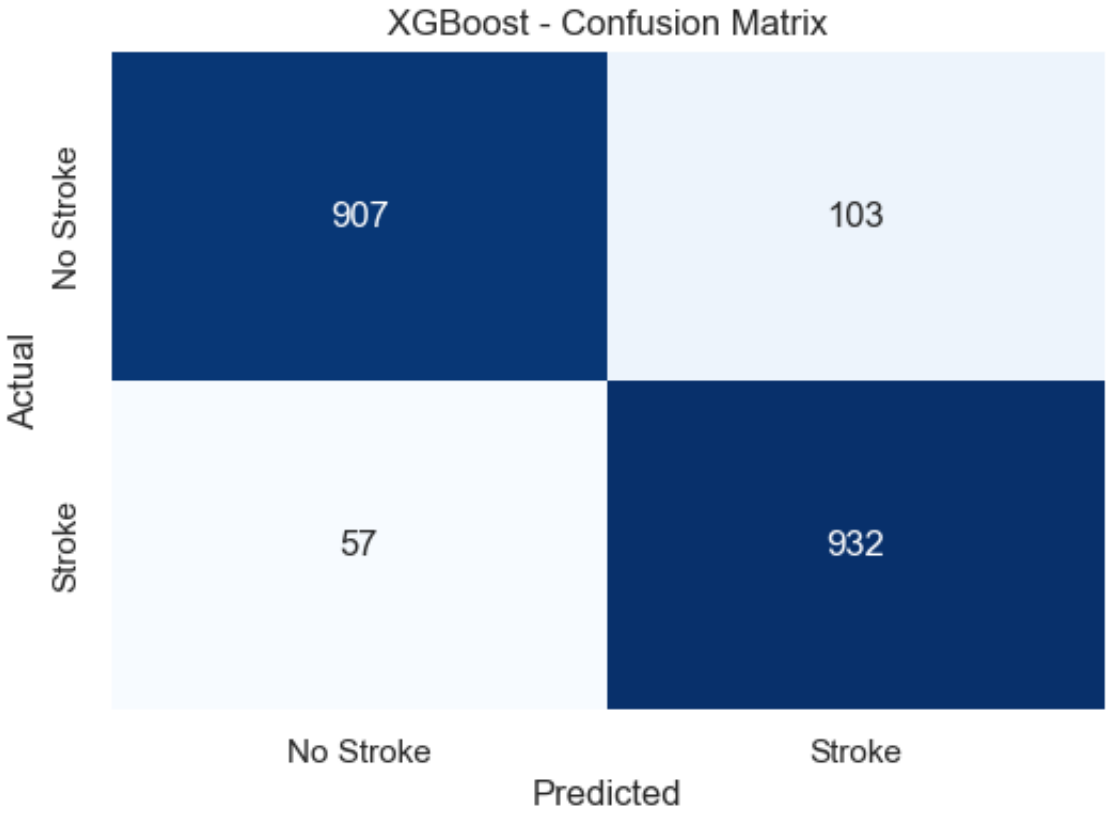

XGBoost - Confusion Matrix
949
77
69
916
No Stroke
Stroke
Actual
No Stroke
Stroke
Predicted

SMOTETomek

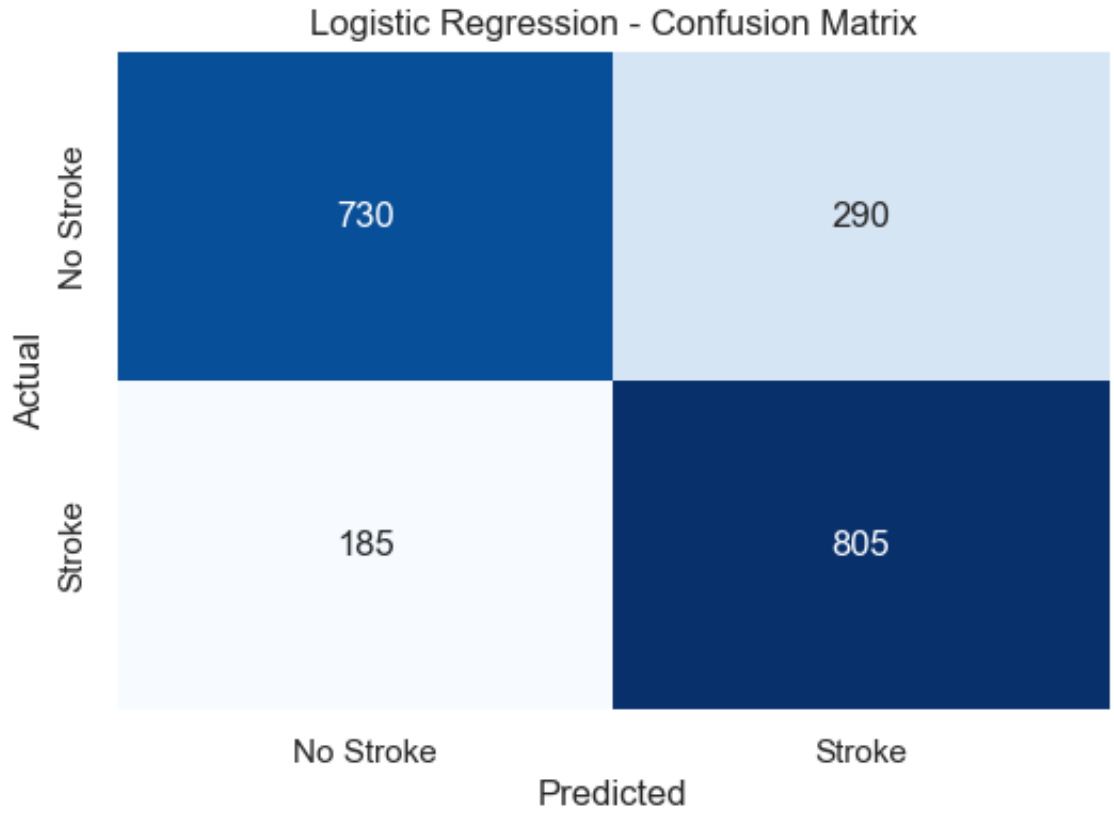

Logistic Regression - Confusion Matrix
746
264
199
790
No Stroke
Stroke
Actual
No Stroke
Stroke
Predicted

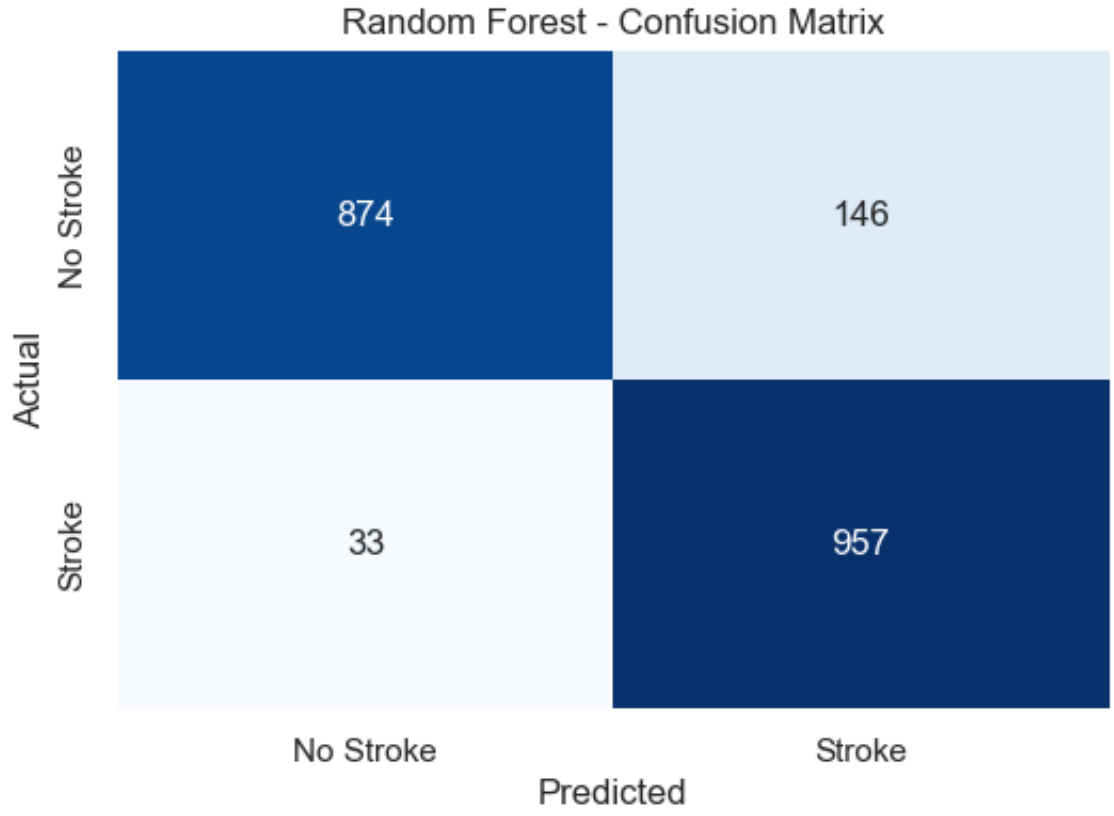

Random Forest - Confusion Matrix
881
129
51
938
No Stroke
Stroke
Actual
No Stroke
Stroke
Predicted

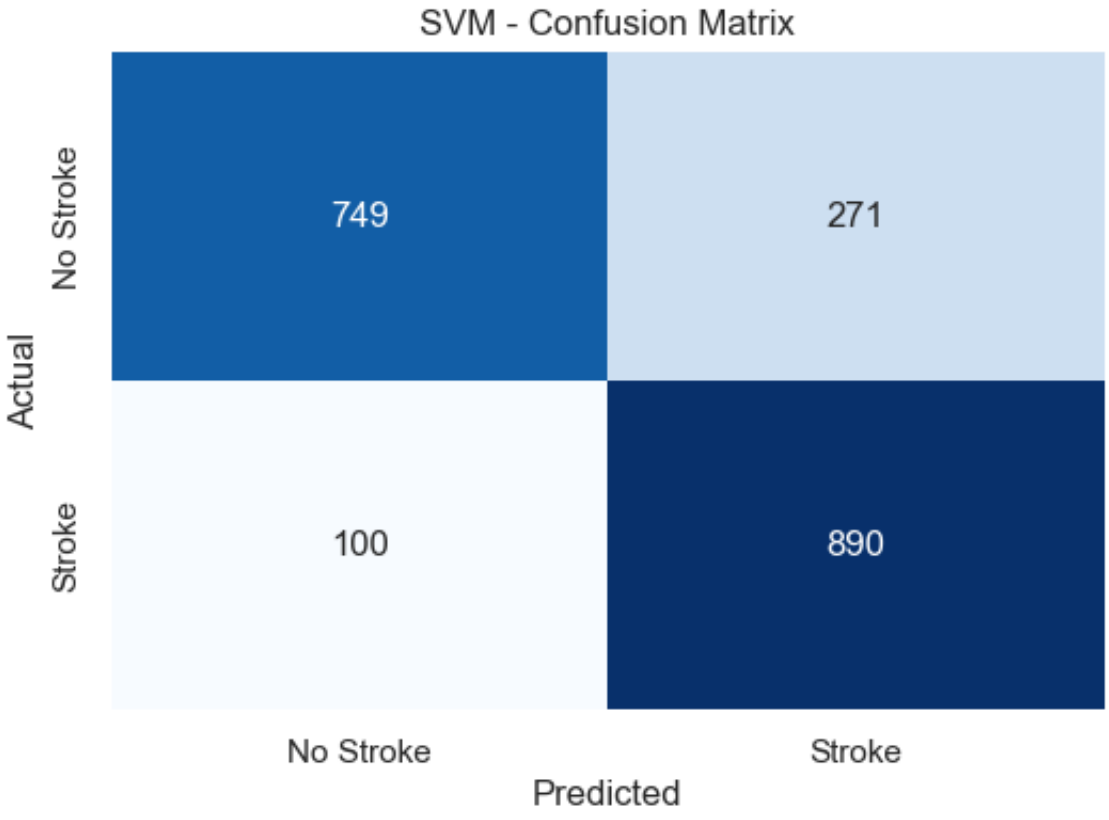

SVM - Confusion Matrix
750
260
103
886
No Stroke
Stroke
Actual
No Stroke
Stroke
Predicted

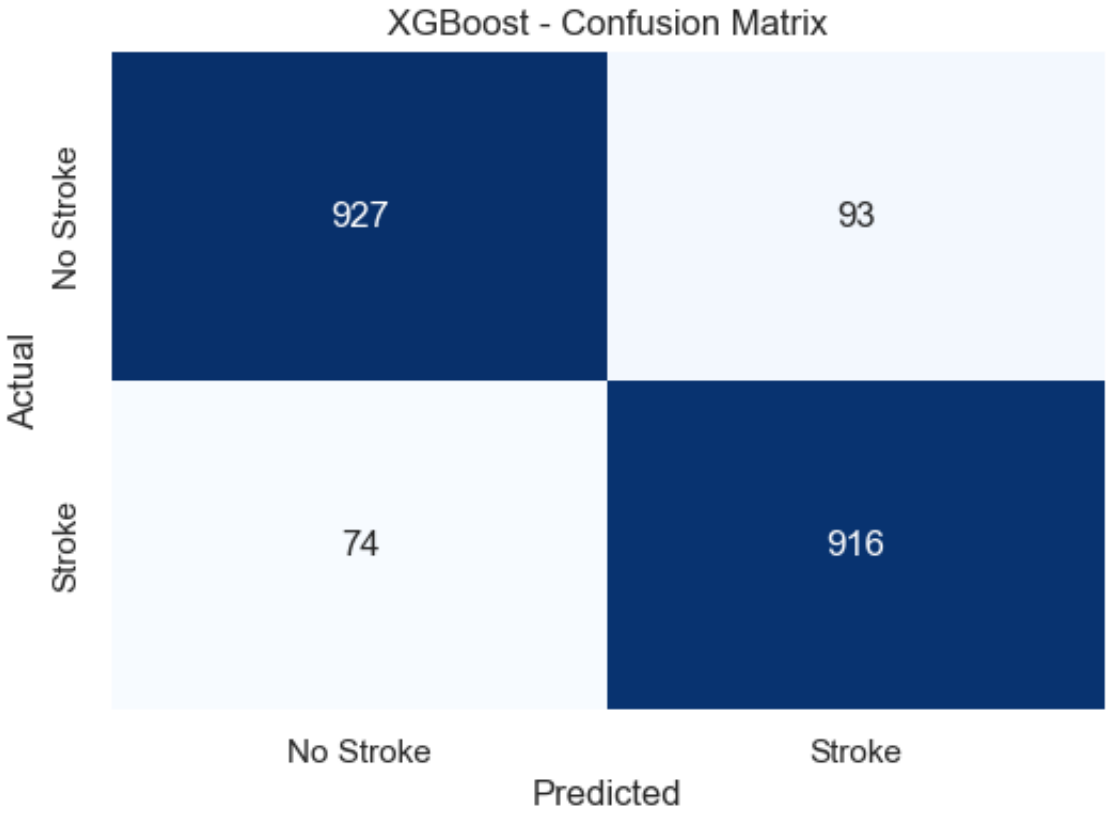

XGBoost - Confusion Matrix
907
103
57
932
No Stroke
Stroke
Actual
No Stroke
Stroke
Predicted

ADASYN


Logistic Regression - Confusion Matrix
730
290
185
805
No Stroke
Stroke
Actual
No Stroke
Stroke
Predicted


Random Forest - Confusion Matrix
874
146
33
957
No Stroke
Stroke
Actual
No Stroke
Stroke
Predicted


SVM - Confusion Matrix
749
271
100
890
No Stroke
Stroke
Actual
No Stroke
Stroke
Predicted


XGBoost - Confusion Matrix
927
93
74
916
No Stroke
Stroke
Actual
No Stroke
Stroke
Predicted