Futbol Maçları Sonuçları Tahmini

Makina Öğrenme Projesi

Çalışanlar:

Mohammad Hajjar 180290608

Ders Sorumluları:

Dr.Öğr.Üyesi

Feyza ALTUNBEY ÖZBAY

Arş.Gör.

Kübra ARSLANOĞLU

İÇİNDEKİLER

1. Giriş:

Bir insan, bir görevi nasıl yapacağını öğrenmek için tekrar tekrar uygulayarak ve tekrarlayarak bir görevi yapmayı öğrenir, ve öylece kazandığı bilgileri sonucuyla tahminedebilir. Bizim projemiz hakkında konuşacaksak; geçen maçların sonuçlarından öğrenerek yeni maçların sonuçlarını tahmini classification modelleri yardımıyla.

1.1.Kaynak Özeti:

- 9 dan 18 mevsimine kadar Excell olarak .csv uzantılı dosyalardır.
- 5 ülkenin takımların mevsimleri içerir: english, french, italian, ispan, german.
- Bilgileri (özellikleri) :
  index Div, Date, HomeTeam, AwayTeam, FTHG, FTAG, FTR, HTHG, HTAG, ... BbMxAHH, BbAvAHH, BbMxAHA, BbAvAHA, PSH, PSD, PSA, PSCH, PSCD, PSCA.
- Veriseti sayesi:
  Her Exell dosyasında 307 veri(satır) vardır, her ülke için 10 excell dosyası, ve bizde 5 ilke olur böylece: 307*10*5=15 350 veri vardır.
- Verisetiyi «datahub.io» adresinden aldık [1].

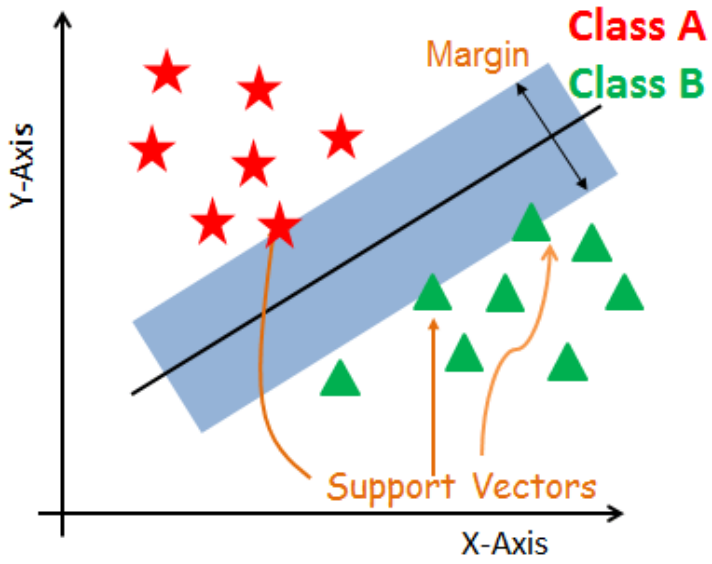
2. YÖNTEM:

Kullandığımız classification metodları:

- SVM (Support Vector Machines)
- Logistic Regression
- Gaussian Nave Bayes
- Decision Tree
- Random Forest

2.1.SVM (Support Vector Machines)

Genel olarak, support vector machines bir sınıflandırma yaklaşımı olarak kabul edilir, ancak hem sınıflandırma hem de regresyon problemlerinde kullanılabilir. Birden çok sürekli ve kategorik değişkeni kolayca işleyebilir. SVM, farklı sınıfları ayırmak için çok boyutlu uzayda bir hiperdüzlem oluşturur. SVM, bir hatayı en aza indirmek için kullanılan yinelemeli bir şekilde en uygun hiper düzlemi oluşturur. DVM'nin temel fikri, veri kümesini sınıflara en iyi şekilde bölen bir maksimum marjinal hiperdüzlem (MMH) bulmaktır.

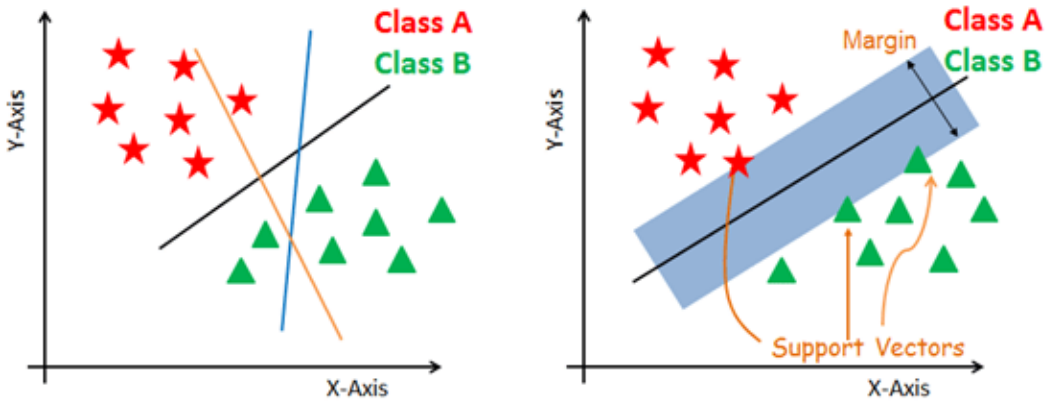


Support Vectors Machin: Support Vectors, en yakın olan veri noktalarıdır. Bu noktalar, marjları hesaplayarak ayırma çizgisini daha iyi tanımlayacaktır. Bu noktalar daha çok sınıflandırıcının yapısıyla ilgilidir.

Hyperplane: Hyperplane, farklı sınıf üyeliklerine sahip bir dizi nesne arasında ayrım yapan bir karar düzlemidir.

Margin: Kenar boşluğu, en yakın sınıf noktalarındaki iki çizgi arasındaki boşluktur. Bu, vektörleri veya en yakın noktaları desteklemek için hattan dikey mesafe olarak hesaplanır. Sınıflar arasındaki marj daha büyükse, o zaman iyi bir marj olarak kabul edilir, daha küçük bir marj kötü bir marj olarak kabul edilir.

Temel amaç, verilen veri setini mümkün olan en iyi şekilde ayırmaktır. Amaç, verilen veri setindeki destek vektörleri arasında mümkün olan maksimum marjı olan bir hiperdüzlem seçmektir. SVM, maksimum marjinal hiper düzlemi arar.



2.2.Logistic Regression

Lojistik Regresyon, denetimli bir Makine Öğrenimi algoritmasıdır; bu, eğitim için sağlanan verilerin etiketlendiği, yani yanıtların eğitim kümesinde zaten sağlandığı anlamına gelir. Algoritma bu örneklerden ve bunlara karşılık gelen cevaplardan öğrenir ve ardından bunu yeni örnekleri sınıflandırmak için kullanır.

Matematiksel terimlerle, bağımlı değişkenin Y ve bağımsız değişkenler kümesinin X olduğunu varsayalım, o zaman lojistik regresyon bağımlı değişkeni P(Y=1) bağımsız değişkenler kümesi olan X'in bir fonksiyonu olarak tahmin edecektir.

2.3.Naive Bayes

Naive Bayes, Bayes Teoreminden ilham alan, makine öğreniminde basit ama güçlü bir olasılıksal sınıflandırma modelidir.

Bayes teoremi, başka bir B olayının zaten gerçekleşmiş olması koşuluyla, bir A olayının gerçekleşmesinin koşullu olasılığını veren bir formüldür. Formülü $P(A|B) = \frac{P(B|A).P(A)}{P(B)}$ şeklinde yazılabilir.

Not: A ve B iki olaydır

P(A|B), verilen B olayının zaten meydana gelmiş olması durumunda A olayının olasılığıdır.

P(B|A), A olayının zaten gerçekleşmiş olması durumunda B olayının olasılığıdır.

P(A), A'nın bağımsız olasılığıdır

P(B), B'nin bağımsız olasılığıdır

Naive Bayes Sınıflandırıcılarının Türleri

3 tip Naive Bayes Sınıflandırıcısı vardır:

1) Gaussian Naive Bayes

Bu sınıflandırıcı, tahmin edicilerin değerleri doğada sürekli olduğunda ve Gauss dağılımını takip ettikleri varsayıldığında kullanılır.

2) Bernoulli Naive Bayes

Bu sınıflandırıcı, tahmin ediciler doğada boole olduğunda ve Bernoulli dağılımını izledikleri varsayıldığında kullanılır.

3) Multinomial Naive Bayes

Bu sınıflandırıcı çok terimli dağıtım kullanır ve çoğunlukla belge veya metin sınıflandırma problemleri için kullanılır.

2.4.Decision Tree

Karar ağacı sınıflandırıcıları, denetimli makine öğrenimi modelleridir. Bu, tahmin yapmak için kullanılabilecek bir algoritmayı eğitmek için önceden etiketlenmiş verileri kullandıkları anlamına gelir. Karar ağaçları, regresyon problemleri için de kullanılabilir.

Karar ağacı sınıflandırıcıları, akış şemaları gibi çalışır. Bir karar ağacının her bir düğümü, iki yaprak düğüme ayrılan bir karar noktasını temsil eder. Bu düğümlerin her biri kararın sonucunu temsil eder ve kararların her biri aynı zamanda karar düğümlerine dönüşebilir. Sonunda, farklı kararlar nihai bir sınıflandırmaya yol açacaktır.

Karar Ağacı Sınıflandırıcıları Neden Öğrenmek İçin İyi Bir Algoritmadır?

Karar ağaçları, birçok nedenden dolayı öğrenmek için harika bir algoritmadır. Yeni başlayanlar için harika olmasının ana nedenlerinden biri, bunun bir "beyaz kutu" algoritması olmasıdır, yani algoritmanın karar verme sürecini gerçekten anlayabilirsiniz. Bu, özellikle makine öğreniminin "nasıl"ını anlamak için yeni başlayanlar için yararlıdır.

karar ağaçları harika algoritmalardır çünkü:

• Genellikle sinir ağları gibi diğer algoritmalardan daha hızlı eğitilirler.

• Karmaşıklıkları, verilerin niteliklerinin ve boyutlarının bir yan ürünüdür.

• Olasılık dağılımı varsayımlarına bağlı olmadıkları anlamına gelen parametrik olmayan bir yöntemdir.

• Yüksek boyutlu verileri yüksek doğruluk derecesinde işleyebilirler.

2.5.Random Forest

Rastgele ormanlar, denetimli bir öğrenme algoritmasıdır. Hem sınıflandırma hem de regresyon için kullanılabilir. Aynı zamanda en esnek ve kullanımı kolay algoritmadır. Orman, ağaçlardan oluşur. Bir ormanın ne kadar çok ağacı varsa o kadar sağlam olduğu söylenir. Rastgele ormanlar, rastgele seçilen veri örnekleri üzerinde karar ağaçları oluşturur, her ağaçtan tahmin alır ve oylama yoluyla en iyi çözümü seçer. Ayrıca, özelliğin öneminin oldukça iyi bir göstergesini sağlar.

Rastgele ormanlar, öneri motorları, görüntü sınıflandırma ve özellik seçimi gibi çeşitli uygulamalara sahiptir. Sadık kredi başvuru sahiplerini sınıflandırmak, hileli faaliyetleri belirlemek ve hastalıkları tahmin etmek için kullanılabilir. Bir veri kümesindeki önemli özellikleri seçen Boruta algoritmasının temelinde yer alır.

Avantajlar:

- Rastgele ormanlar, sürece katılan karar ağaçlarının sayısı nedeniyle oldukça doğru ve sağlam bir yöntem olarak kabul edilir.
- Aşırı uyum sorunu yaşamaz. Ana sebep, önyargıları ortadan kaldıran tüm tahminlerin ortalamasını almasıdır.
- Algoritma hem sınıflandırma hem de regresyon problemlerinde kullanılabilir.
- Rastgele ormanlar da eksik değerleri işleyebilir. Bunları ele almanın iki yolu vardır: sürekli değişkenleri değiştirmek için medyan değerleri kullanmak ve eksik değerlerin yakınlık ağırlıklı ortalamasını hesaplamak.
- Sınıflandırıcı için en fazla katkıda bulunan özelliklerin seçilmesine yardımcı olan göreli özellik önemini elde edebilirsiniz.

Dezavantajları:

- Rastgele ormanlar, birden fazla karar ağacına sahip olduğu için tahmin üretmede yavaştır. Ne zaman bir tahmin yapsa, ormandaki tüm ağaçların aynı girdi için bir tahmin yapması ve ardından bunun üzerinde oylama yapması gerekir. Tüm bu süreç zaman alıcıdır.
- Modelin yorumlanması, ağaçtaki yolu takip ederek kolayca karar verebileceğiniz bir karar ağacına kıyasla zordur.

3. DENEY SONUÇLARI:

	Support Vector Machine(SVM)	**Logistic Regression**	**Gaussain Naive Bayes**
Accuracy(doğruluk oranı)	0.59 = 59%	0.66 = 66%	0.63 = 63%
Confusion Matrix	[[2382 0 1690] [864 0 2754] [511 0 6100]]	[[3093 281 698] [1157 548 1913] [503 305 5803]]	[[2709 717 646] [1030 1001 1587] [596 666 5349]]

	Decision Tree	**Random Forest**
Accuracy(doğruluk oranı)	0.99993 = 100%	0.99993 = 100%
Confusion Matrix	[[4072 0 0] [1 3617 0] [0 0 6611]]	[[4072 0 0] [1 3617 0] [0 0 6611]]

4. TARTIŞMA:

Seçtiğimiz deneme test 0.2 demekki öğrenme 0.8 dir.

Biz test 0.5 demekki öğrenme 0.5 dir ama daha kötü doğruluk oranı verdi randomForest Classifier ile gösterirsek:

Test 0.2 verdiğimizde:

Test 0.5 verdiğimize:

SVM de:

Random satate vardır:

random stat arttık zaman(defalar) öğrenme ve test zamanı azaldı ama doğruluk oranı ve score artmışlar.

LogisticRegression da:

max-iterator vardır:

max-iterator 300... verdiğimizde “STOP: TOTAL NO. of ITERATIONS REACHED LIMIT.” bir uyarı çıktı. Max-iterator biz azalttık zaman doğruluk oranı artar, biz 600 denedik doğruluk oranı düştü.

5. KAYNAKÇA:

[1] https://datahub.io/collections/football#football-data-guides-articles

[2] https://scikit-learn.org/stable/index.html

[3] https://asperbrothers.com/blog/logistic-regression-in-python/

[4] https://www.datacamp.com/tutorial/random-forests-classifier-python

[5] https://scikit-learn.org/stable/supervised_learning.html#supervised-learning

[6] https://www.analyticsvidhya.com/blog/2021/03/machine-learning-with-python-gaussian-naive-bayes/

[7] https://stackabuse.com/random-forest-algorithm-with-python-and-scikit-learn/

[8] https://www.geeksforgeeks.org/decision-tree-implementation-python/