VERİ MADENCİLİĞİ SINIFLANDIRMA ALGORİTMALARI İLE E-POSTA ÖNEMLİLİĞİNİN BELİRLENMESİ

DETERMINING EMAIL IMPORTANCE WITH DATA MINING CLASSIFICATION ALGORITHMS

Sena Özkara[1]

[1]MAKÜ Mimarlık Mühendislik Fakültesi Bilgisayar Mühendisliği Bölümü,Burdur.

Sorumlu yazar: senaozkaraa@hotmail.com

Öz

Günümüzde internet ortamında yaptığımız her hareket çeşitli araştırmacılar tarafından takip edilmekte ve kayıt altına alınmaktadır. Veri madenciliği, gizli veri tahminlerini büyük veri tabanlarından çıkaran, veri ambarındaki önemli bilgileri analiz etme potansiyeline sahip güçlü ve yeni bir teknolojidir. Bu çalışmada önemli e-postaları ayırt etmek için veri madenciliği algoritmaları kullanılmıştır. Çalışmada 5172 rastgele seçilmiş e-posta dosyasını içeren veri kümesi üzerinde çalışılmıştır. Naive Bayes, Support Vector Machines,Random Forest ve Logistic Regression sınıflandırma algoritmaları seçilmiş ve doğruluk, ağırlıklı ortalama anma, ağırlıklı ortalama duyarlılık ve kök hata kareler ortalaması istatistikleri kullanılarak karşılaştırılmış ve en iyi sınıflandırma algoritması belirlenmiştir. Analizler için Python programlama dilini kullanarak Jupiter Notebook programı kullanılmıştır.

Anahtar Kelimeler: Büyük Veri, Sınıflandırma Algoritmaları, Önemli E-postalar, Python

Abstract

Today, every move we make on the internet is tracked and recorded by various researchers. Data mining is a powerful new technology with the potential to analyze important information in the data warehouse, extracting confidential data predictions from large databases. In this study, data mining algorithms are used to distinguish important e-mails. In the study, a dataset containing 5172 randomly selected e-mail files was studied. Naive Bayes, Support Vector Machines, Random Forest and Logistic Regression classification algorithms were selected and compared using accuracy, weighted mean recall, weighted mean sensitivity and mean square root statistics, and the best classification algorithm was determined. For the analysis, Jupiter Notebook program was used by using the Python programming language.

Key Words: Big Data, Classification Algorithms, Important E-mails, Python

1. GİRİŞ

Günümüzde sadece bilgiye ulaşmak değil, gerekli koşullarda bilgi üretmek de önemli bir konu halini almıştır. Çığ gibi büyüyen sayısal veri ortamları arasından yararlı ve de gerekli olan bilgiye ulaşmayı sağlamak gerçek bir çaba haline gelmiştir. Veri madenciliği bu safhada göze çarpan bir olgudur. Veri madenciliği, gizli veri tahminlerinin büyük veri tabanlarından

çıkarılması, veri ambarındaki önemli bilgileri analiz etme potansiyeline sahip güçlü ve yeni bir teknolojidir

Bir veri ambarı, "İş ne olur?" diye sorar. Bu soruların cevabı, işletmenizin günümüzün enformasyonu için bir gereklilik olan reaktif değil, proaktif olmasını sağlayacaktır. Bugün endüstri eğilimi daha güçlü donanım ve yazılım konfigürasyonuna doğru ilerliyor, şimdi analitik olarak çok fazla bilgiyi işleme yeteneğine sahibiz ki bu altı yıl önce duyulmamıştı bile. Bugün bir işletme, bu yeni teknolojiyi kullanabilmeli veya riski kabul etmelidir. Risk ise bilgilendirilmiş kararlar almak için gerekli olan kritik bilgileri kaçırmaktır. Günümüz dünyasında, teknolojiyi büyük miktarlardaki bilginin üzerinden geçmek için etkin bir şekilde kullanmayan bir işletme, bilgi çağında hayatta kalmayacaktır. Bilgiye erişim ve bilgiyi anlama bir güçtür. Bu güç, rekabet avantajı ve hayatta kalmaya eşittir. Bir birim, bilgi çağında hayatta kalmayı ve gelişmeyi umuyorsa, işletmenin ihtiyaç duyduğu bilgileri depolamak için kendi veri ambarını inşa etmek zorundadır (Rai Technology University, 2018).

Barcelona'da düzenlenen Mobil Dünya Kongresi'nde (Mobile World Congress-MWC) Mobil Ekonomi 2017 başlıklı rapora göre, 2016 sonu itibariyle dünyada mobil abone sayısı 4,8 milyar oldu. 2020 sonundaki beklentisi ise 5,7 milyar aboneye ulaşılacağı yönünde. Bu veriye göreyse 2020'de dünyanın üçte ikisinin mobil kullanıcı olması bekleniyor (Aydın,2017). Küresel olarak, yılda yaklaşık 4.4 ZB elektronik veri üretilmektedir. IDC (International Data Corporation) kurumsal verilerin 2020'ye kadar 40 ZB'ye ulaşacağını iddia ediyor.

Dünya genelinde yaratılan e-posta hesaplarının sayısı 2012 yılında 3,3 milyardan 2016 yılının sonuna kadar 4,3 milyara yükselmiştir. 2012 yılında, gönderilen ve alınan e-posta sayısı günde 89 milyardı. Bu miktar 2016 yılı sonunda 143 milyarın üzerine çıkmıştır. 2016 yılında günde dünya genelinde gönderilen e-posta sayısı 170 milyar civarındadır. Her saniye gönderilen e-posta sayısı 2 milyondur. Son 3 yılda mobil cihazlarda e-posta açma oranı %180 yükselmiş durumdadır (Yaqoob ve ark., 2016).

Verilerin içsel değeri vardır. Ancak bu değer keşfedilene kadar hiçbir faydası yoktur. Bugün, büyük veri sermaye haline geldi. Dünyanın en büyük teknoloji şirketlerinden bazılarını düşünürsek, sundukları değerin büyük bir kısmının, daha fazla verimlilik sağlamak ve yeni ürünler geliştirmek için sürekli temkinli olarak ele almaktır. Bir başka örnek, finansal ve planlama konuları ile ilgili kararları iyileştirmek için veri analizlerini kullanmak; trendleri ve müşterilerin istediği yeni ürün ve hizmetleri incelemek ve dinamik fiyatlandırma yapmaktır (ORACLE, 2018)

Bu makalede kullanılan veri kümesi 5172 rastgele seçilmiş e-posta dosyasının ilgili bilgilerini ve bunların spam veya spam olmayan sınıflandırması için ilgili etiketlerini içeren bir csv dosyasıdır. *csv dosyası, her bir e-posta için her satırda 5172 satır içerir. 3002 sütun vardır. İlk sütun E-posta adını gösterir. Ad, gizliliği korumak için alıcıların adıyla değil, sayılarla belirlenmiştir. Son sütunda tahmin için etiketler bulunur: spam için 1, spam değil için 0. Kalan 3000 sütun, alfabetik olmayan karakterler/kelimeler hariç tutulduktan sonra , tüm e-postalarda en yaygın 3000 kelimedir .Veri seti kaggle'dan alındığı için veri setinin içeriğide İngilizcedir. Dolayısıyla en yaygın kullanılan kelimelerden İngilizce kelimelerdir.

Her satır için, o e-postadaki (satırdaki) her kelimenin (sütun) sayısı ilgili hücrelerde saklanır. Böylece, 5172 e-postanın tümüne ilişkin bilgiler, ayrı metin dosyaları yerine kompakt bir veri çerçevesinde depolanır.Python yazılımındaki sınıflandırma algoritmaları veriye uygunluğu göz önünde bulundurularak denemeler yapılacaktır, elde edilen verilerle sınıflandırma algoritmalarının nasıl sonuç verdiği gözlemlenecektir. Bu sınıflandırma algoritmaları; Naive Bayes, Random Forest, Logistic Regression ve Support Vector Machines algoritmalardır. Sınıflandırma algoritmaları karşılaştırılmış ve önemli e-posta olma olasılıkları belirlenmiştir

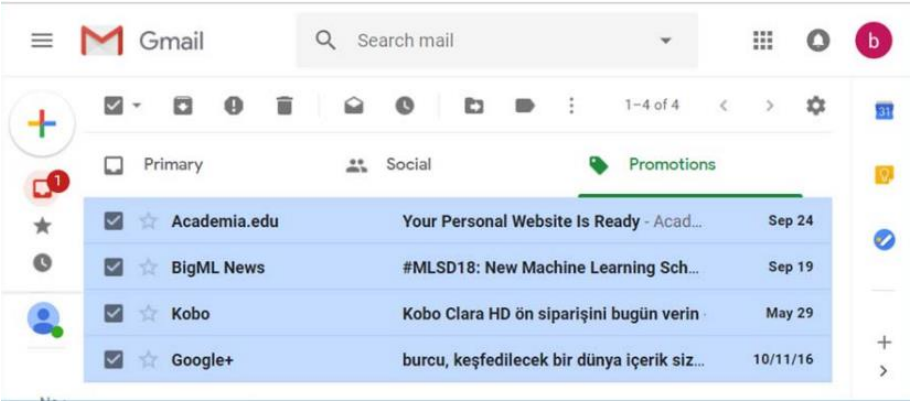
Algoritmaların performanslarının karşılaştırılması için doğru sınıflandırma yüzdesi (Accuracy), anma (Recall), duyarlılık (Precision) ve kök hata kareler ortalaması (RMSE) ölçütleri kullanılmıştır.

Sunulan makalenin amacı veri madenciliği algoritmalarını kullanarak önemli olan e-postaları ayırt edici belirgin farkları çıkarmak ve çıkan sonuçları karşılaştırıp bu problemde kullanılabilecek en iyi algoritmayı bulmaktır. Bu araştırmada, en uygun sınıflandırma algoritması veriye uygulanarak, hangi e-postaların önemlilik arz ettiğinin belirlenmiştir. Bu çalışma sayesinde büyük şirketlerde çalışanların iş gücü yükünü hafifletebiliriz. Gmail kullanıcıların önemli e-postaları algılaması için bir kural kümesi oluşturmasına izin verir, ama önemli postaların niteliği zamanla değiştikçe, bu kural kümeleri, kullanıcı tarafından sürekli olarak ayarlanmalı ve yeniden ayarlanmalıdır. Bu zaman alıcı ve genellikle hataya eğilimli, sıkıcı bir süreçtir. Yani kural tabanlı bir yaklaşımın önemli posta filtrelemede sınırlı bir faydası vardır. Bu çalışma sayesinde hangi e-postaların önemli olduğu daha kolay yoldan belirleyebiliriz. Ayrıca reklam içerikli ve okumaya gerek duymadığımız e-postalarda bu yolla daha kolay ortaya çıkar.

Materyal ve Yöntem

Önemli E-Posta Olma Olasılığı

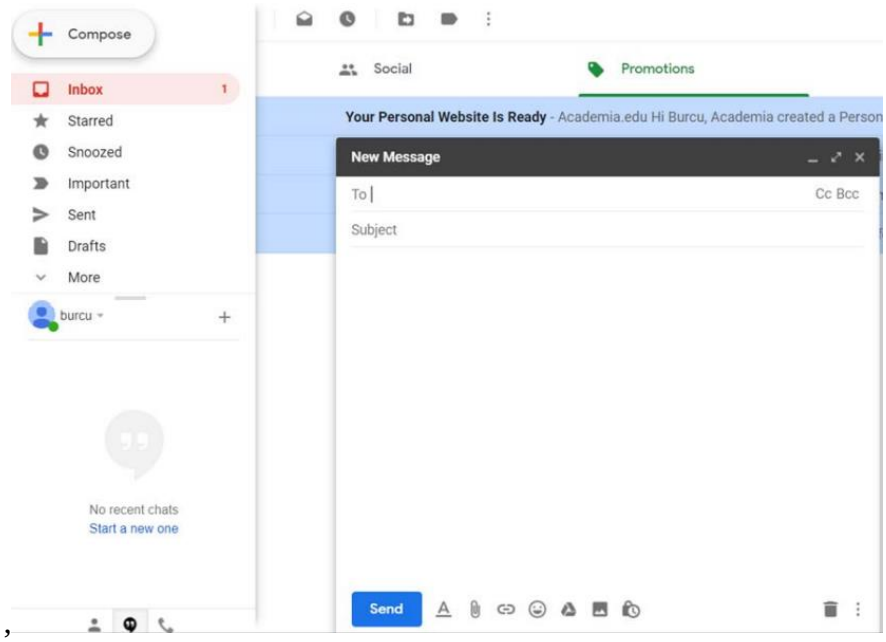
E-posta, 1971 yılında bilgisayar mühendisi olan Raymond Samuel Tomlinson tarafından ilk kez denenmiştir. Hemen yanı başındaki bilgisayara @ işaretini kullanarak posta atmasıyla başlamıştır. Daha sonra 26 Mart 1976 yılında İngiltere' de ARPANET adında internete dönüşebilecek bir bilgisayar ağı çıktı. 1978'de ise bugün e-posta trafiğinin çok büyük bir bölümünü oluşturan ilk reklam gönderimi yapıldı. 1982 yılında "e-mail" yani e-posta kelimesi ilk kez kullanılmaya başlandı. 21. yy 'da internet teknolojilerinin dolayısıyla da e-postanın gelişmesiyle birlikte bu defa güvenlik konusu ön plana çıktı. 2005'te, e-posta kimliklerinin doğrulanmasını sağlayan ilk teknoloji SPF (Sender Policy Framework) geliştirildi. Ardından yani Şubat 2007'de gmailin beta sürümü tamamen çıkarıldı ve tüm internet kullanıcılarının kullanabileceği şekilde kurgulandı (Gümüş, 2018).



Şekil 1 Gmail Kullanıcı Ara Yüzü

Öncelikle Şekil 1'deki gmail ara yüzüne bakıldığında ortada, Gelen Kutunuzdaki tüm e-postalar görülür. Gelen e-postaları çeşitli başlıklar altında kategorize edebilmektedir (Primary, Social, Promotions vs.). Bu ana bölümde soldan sağa e-postanın gönderenin adı, e-postanın konusu ve Gelen Kutunuza ulaştığı zaman bulunmaktadır. Solda gezinme öğeleri bulunur: Gelen Kutusu, Yıldızlı, Önemli, Gönderilmiş Postalar ve Taslaklar. (Telstra Tech Savvy Seniors, 2018).

Gmailin özelliklerinden biri de mobil uygulamasının bulunmasıdır ve uygulama 40'tan fazla dilde kullanılabilmektedir. Tüm gelen ve giden e-posta mesajlarındaki eklerde, virüs olup olmadığını otomatik olarak taramaktadır. Gönderilen elektronik postaları, göndereceğiniz kişiyi daha sonra otomatik olarak kaydetmektedir. Yenilikçi Google teknolojisi ile güçlü spam engelleme özelliğine sahiptir (Gümüş, 2018).



Şekil 2 Compose Kullanıcı Ara Yüzü

Şekil 2'de Compose sekmesi tıklandığında açılan pencerede görünenlerin anlamları şu şekildedir:

To: Mesajın gönderileceği e-posta adresinin yer aldığı bölümdür.

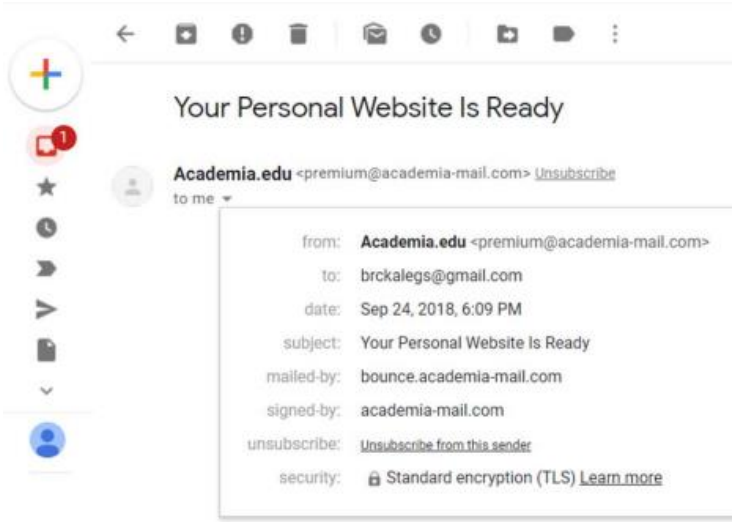
Subject: Mesajın konusu veya başlığının ifade edildiği bölümdür.

Mesaj alanı: Mesajın tam içeriğinin yer aldığı bölümdür.

CC: Mesajın gönderileceği bir başka e-posta adresinin yazılacağı bölümdür.

BCC: Mesajın bir kopyasının gönderileceği ancak diğer alıcılardan gizlenen e-posta adreslerinin yazıldığı bölümdür.

Ek (attachment): Yazıya ek olarak gönderilmek istenen dosyaların yüklendiği bölümdür.



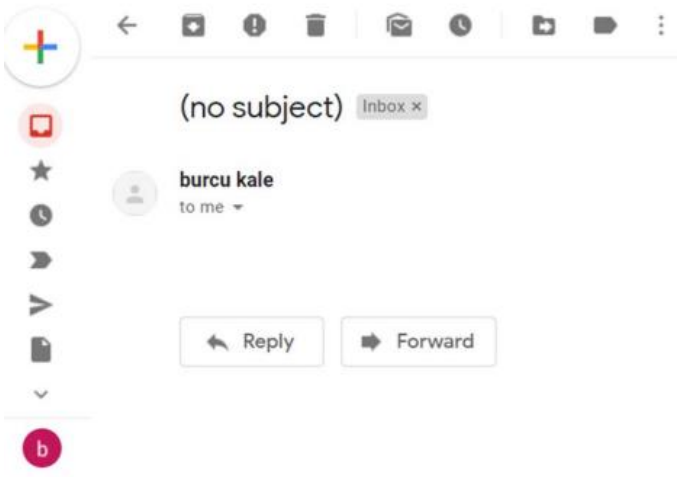
Şekil 3 Gelen E-postanın Bilgilerinin Yer Aldığı Kullanıcı Ara Yüzü

Şekil 3'te gelen e-postanın bilgilerinin bulunduğu alanda görünenlerin anlamları şu şekildedir:

From: E-postanın kimden geldiğini gösteren başlık alanıdır.

Date: E-postanın geliş zamanını (yıl, ay, gün, saat, dakika, saniye olarak) belirtir.

Subject: Gelen e-postanın konusunun olduğu alandır (Ceyhan ve Sağıroğlu, 2015).



Şekil 4 Reply veya Forward Simgelerinin Yer Aldığı Kullanıcı Ara Yüzü

Şekil 4’te reply ve forward simgelerinin anlamları şu şekildedir:

Reply: Bir e-postayı yanıtlamaktır (Reply). Orijinal e-postanın içeriği ana penceredir. Yanıtlar, konuşmanın tam geçmişini tutar. Kural olarak, cevabınızı orijinal mektubun üzerindeki alana yazmanız gerekir (böylece alıcının e-postayı açtığında gördüğü ilk şey olur). Yazdıkça, orijinal e-posta aşağı itilir. Sonra Gönder'e sol tıklanır. Orijinal e-postayı gönderen kişi, cevabınızı hemen almalıdır. Konu satırı, öncekiyle aynı olacaktır, ancak başlangıçta bulunan bir re: bunun bir yanıt olduğunu belirtir.

Forward: Açılan e-postanın altındaki İlet (Forward) sekmesinin tıklanması dışında, yukarıdakilerle aynı adımlar uygulanır. Fakat kime alanındaki alıcı gönderici kişi için doldurulmamıştır. E-postayı yönlendirmek istenilen kişinin adının yazılması gerekir. E-postaya eklemek istediğiniz herhangi bir şey varsa, orijinal içeriğin üzerine yazmanızın beklenmesi gerekir. E-postayı iletilmeye hazır olduğunda Gönder tıklanır. Bu kişi e-postayı aldığında, konu satırı Fwd'ye sahip olacaktır (Telstra Tech Savvy Seniors, 2018).

İnternet'e bağlanan kullanıcı sayısı hızla ilerlerken, elektronik posta (E-posta) hızlı bir şekilde mevcut en hızlı ve en ekonomik iletişim biçimlerinden biri haline gelmektedir.

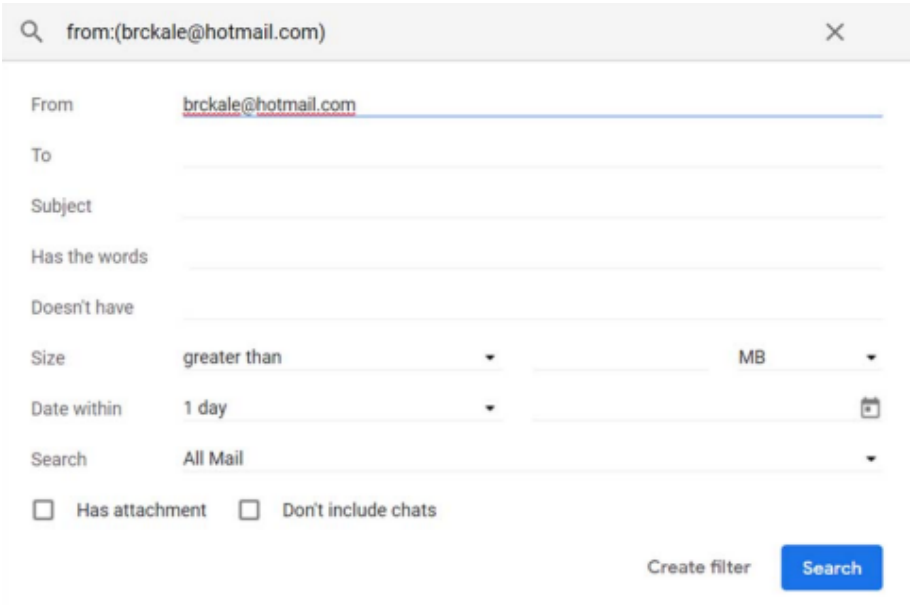
E-posta son derece ucuz ve kolay gönderilebildiğinden, sadece arkadaşlara ve meslektaşlara mesaj alışverişinde bulunmanın bir aracı olarak değil, aynı zamanda elektronik ticaret yapmak için bir ortam olarak da muazzam halk kitleleri kazanmıştır.

İnternet'teki doğrudan pazarlamacıların çoğalması ve E-posta adresi posta listelerinin kullanılabilirliğinin artmasıyla, önemsiz postaların hacmi (genellikle toplu olarak "spam" olarak anılacaktır) son birkaç yılda muazzam bir şekilde artmıştır (Sahami ve ark., 1998).

Gmailin tasarlanma amaçlarından biri e-postanın önemli ya da önemli değil olarak belirlenebilmesidir. Ayrıca çok önem verilenlere yıldızlama yapılabilir. İşaretleme yapmak istedikleriniz için sarı yıldız veya daha önemlilere başka yıldızlar konulabilir.

Her postanın önünde iki farklı simge bulunur. Bu simgelerden soldakini işaretleyip “More” sekmesinden “önemli olarak işaretle” sekmesini işaretleyince e-postamız önemli duruma gelir. Bunu Google, maillerinizi bir algoritma çevresinde değerlendirerek otomatik yapar. Ama bu işaretleme de değiştirilebilir. Örneğin bir yerden gelen e-postaları, gmail önemli olarak işaretledi ise bu işaret kaldırıldığında önemli olmadığı belirtilmiş olur. Tabi bu işaretleme sadece gönderici için kolay bir kullanım olmaktadır . Ayrıca sol tarafta Önemli sekmesi tıklanarak sadece önemli olarak işaretlenen iletilerde görülebilir (Durmaz, 2013).

Kullanıcıların önemli e-postaları algılaması için bir kural kümesi oluşturmasını gerektiren sistemler, kullanıcılarının sağlam kurallar oluşturabilecek kadar bilgili olduğunu varsayar. Üstelik, önemli postaların niteliği zamanla değiştikçe, bu kural kümeleri, kullanıcı tarafından yeniden ayarlanmalıdır. Bu zaman alıcı ve genellikle hataya eğilimli, sıkıcı bir süreçtir. Yani kural tabanlı bir yaklaşımın önemli posta filtrelemede sınırlı bir faydası vardır. Aşağıda gmail ortamında e-postalarınızı filtrelemek için oluşturulan kuralın aşamaları anlatılmıştır.



Şekil 5 Filtreleme yapılan Kullanıcı Ara Yüzü

Şekil 5'te filtreleme yapılan alanda görünenlerin anlamları şu şekildedir:

1. Şekil 1'de olduğu gibi en üstteki arama kutusunda, aşağı ok simgesi tıklanır.

2. Şekil 5'te görüldüğü gibi arama ölçütleri girilir. Aramanın düzgün şekilde çalışıp çalışmadığını kontrol etmek için Arama sekmesi tıklanır ve açılan pencerede görüntülenen e-postalara buradan göz atılabilir.

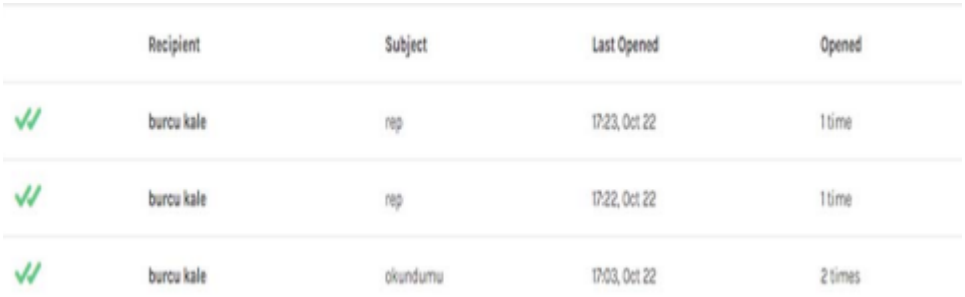
3. Arama penceresinin alt kısmındaki Filtre Oluştur tıklanır.

4. Filtrenin yapması istenilen işlem belirtilir.

5. Filtre Oluştur tıklanır (gmail yardım, 2018).

Ayrıca Mailtrack uzantısı, gmailde mesajlaşmada faydalı olan bir yoldur. Bu uzantı sayesinde e-postanın gönderildiğinden ve alıcının onu açtığından emin olunur. En önemlisi kaç kere okunduğu da görülür.

Şekil 6'da Mailtrack uzantısının oluşturduğu raporun gösterildiği pencere şu şekildedir:



Şekil 6 Mailtrack Uzantısının Oluşturduğu Raporun Gösterildiği Pencere

Bu çalışmanın amacı e-posta filtrelenmesinin veri madenciliği sınıflandırma algoritmaları kullanılarak birbiri ile karşılaştırılıp sınıflandırmada en başarılı olanın belirlenmesidir.

Python

Python, Hollandalı bir yazılım geliştirici olan Guido van Rossum tarafından 1990 yılında tasarlanmış bir programlama dilidir. Programlama dilinin adı, yaygın bilinenin aksine piton yılanından değil, Guido van Rossum'un çok sevdiği Monty Python adlı, altı kişiden oluşan İngiliz komedi grubunun oynadığı, Monty Python's Flying Circus isimli gösteriden gelmektedir.

Açık kaynak kod lisansına sahip olan ve ücretsiz yazılım geliştirilmesine imkân veren Python programlama dilinin; Windows, Unix/Linux ve MacOS işletim sistemleri üzerinde farklı yöntemlerle çalıştırılması mümkündür.

Veri Bilimi ve Makine Öğreniminde Python'un Yeri : Sofistike veri analizleri günümüzde IT için en önemli konular haline gelmiştir. Python ise bu durumlar için en elverişli programlama dili olmuştur. Python arayüzündeki kütüphanelerin birçoğu makine öğrenimi ve veri bilimi üzerine elverişlidir. Bu alanlardaki kütüphanelerdeki yüksek kaliteli komutları, makine öğrenimi kütüphanelerinin ve diğer nümerik algoritma kütüphanelerinin sürekli gelişmesine çok yardımcı olmuştur.

Jupyter Notebook

Kod tabanlı metinsel, grafiksel sonuçların ve belgelerin hücre tabanlı bir ortamda birleştirilebildiği tarayıcı tabanlı interaktif bir programlama aracıdır.

Sınıflandırma Algoritmaları ve E-mail Önemlilik Analizi

Python yazılımındaki sınıflandırma algoritmaları veriye uygunluğu göz önünde bulundurularak denemeler yapılacaktır, elde edilen verilerle sınıflandırma algoritmalarının nasıl sonuç verdiği gözlemlenecektir. Bu sınıflandırma algoritmaları; Naive Bayes, Random Forest, Logistic Regression ve Support Vector Machines algoritmalardır. Sınıflandırma algoritmaları karşılaştırılmış ve önemli e-posta olma olasılıkları belirlenmiştir

Bayes Sınıflandırması

Bayes sınıflandırması istatistiksel sınıflandırıcılardır. Bayes teoremi, 18. yüzyılda olasılık ve karar teorisinde çalışma yapan, bir İngiliz papaz olan Thomas Bayes'den alınmıştır. Bayes sınıflandırması, Bayes'in teoremine dayanarak tanımlanmıştır. Sınıflandırma algoritmalarını karşılaştıran çalışmalarda Bayes sınıflandırması, büyük veri tabanlarına uygulandığında yüksek doğruluk ve hız sergilemiştir.

Bayes teoremine göre: X bir veri grubu olsun. X, bir dizi n özniteliğin üzerinden yapılan ölçümlerle tanımlanır. H, veri grubu X belirtilen bir C grubuna aittir hipotezidir.

Sınıflandırma problemleri için, P (H \ X)'i, diğer bir deyişle, X veri grubunun C sınıfına ait olma olasılığını ararız. Burada P(H\X) koşullu olasılıktır.

$$\mathrm{P}\ (\mathrm{H} \setminus \mathrm{X}) = \frac{P\left(\frac{X}{H}\right) P(H)}{P(X)}$$

P(H\X): X olayı gerçekleştiği durumda H olayının meydana gelme olasılığıdır.

P(X\H): H olayı gerçekleştiği durumda X olayının meydana gelme olasılığıdır.

P(H) ve P(X): H ve X olaylarının önsel olasılıklarıdır.

Burada önsel olasılık Bayes teoremine öznellik katar. Diğer bir ifadeyle örneğin P(H) henüz elde veri toplanmadan H olayı hakkında sahip olunan bilgidir. Diğer taraftan P(X\H) ardıl olasılıktır çünkü veri toplandıktan sonra, H olayının gerçekleşmiş olduğu durumlarda X olayının gerçekleşme ihtimali hakkında bilgi verir.

Naive Bayes

Veri madenciliğinde, Naive Bayes sınıflandırıcıları, verideki tüm değişkenlerin birbirlerinden bağımsız ve eşit derecede önemli olduğu varsayılarak analiz gerçekleştiren, varsayımlarıyla Bayes teoreminin uygulanmasına dayanan basit olasılıksal sınıflandırıcılar ailesidir. Örneğin, bir meyve kırmızı, yuvarlak ve çapı yaklaşık 10 cm ise bir elma olarak kabul edilebilir. Bir Naive Bayes sınıflandırıcı, bu özelliklerin her birinin; renk, yuvarlaklık ve çap özellikleri arasındaki olası korelasyonlardan bağımsız olarak, bu meyvenin bir elma olması olasılığına bağımsız olarak katkıda bulunduğunu göstermektedir.

Naive Bayes sınıflandırıcılar gözetimli öğrenme ortamında çok verimli bir şekilde eğitilebilir. Naive Bayes'ın bir avantajı, sadece sınıflandırma için gerekli parametreleri tahmin etmek için az sayıda eğitim verisi gerektirmesidir (Han ve ark., 2012).

Bayes teoreminin m tane örneğe uygulandığı varsayılsın. S= {S1, S2, …, Sm} veri kümesi eğitim veri kümesidir. S veri kümesindeki her örnek n boyutlu {X1, X2, …. , Xn} vektörlerle ifade edilsin ve bu veri kümesinde k tane C1,C2,……..,Ck ile gösterilen sınıf değişkeni olsun. Her örnek bu sınıflardan birine ait olur. Bu verilere ek olarak sınıfı bilinmeyen X veri kümesi verildiğinde bu kümenin sınıfları Bayes teorimi geliştirilerek P(Ci/X) olasılığı ile tahmin edilir (Çığşar, 2017).

Rastgele Orman (Random Forest)

Rastgele orman birden fazla karar ağacı oluşturur ve daha doğru ve istikrarlı bir tahmin elde etmek için onları birleştirir. Her bir ağaç farklı eğitim kümeleriyle eğitilerek sonuçlar kullanır. Her ağaç sınıflandırıcı üretir. Üretilen bu sınıflandırıcılar kendi arasında oylama yapar ve en fazla oyu alan sınıflandırıcıyı algoritma belirler. Seçilen bu sınıflandırıcı yeni veriler verildiğinde, veriyi sınıflandırması için kullanılır (Çığşar, 2017).

Boosting (Shapire, 1996) ve Bagging (Breiman, 1996) ağaçların sınıflandırılmasında toplu öğrenme için çok iyi iki yöntem olarak bilinir. Bagging'de eğitim verisi kullanılarak her

bir ağaç inşa edilir. Ardışık gelen ağaçlar bir öncekinden bağımsızdır ve en büyük oyu alan ağaç tahmin için alınır. Boosting'de, ardışık gelen ağaçlar bir öncekine bağımlıdır. Bir önceki öncüller tarafından yanlış tahmin edilmiş noktalar için ekstra ağırlık verilir. 1996'da Berkeley, California Üniversitesi'nde Leo Breiman (CART®'ın babası) tarafından Bagging tekniği kullanılarak her bir ağacın birbirinden bağımsız olarak eğitim verileri oluşturulmuştur (Zavoral ve ark., 2010).

Rastgele orman, özellikle büyük ölçekli sistemlerde yazılım kalitesi tahmini için iyi bir adaydır, çünkü:

- Mevcut sınıflandırma algoritmaları ile karşılaştırıldığında tutarlı bir şekilde doğru olduğu bildirilmiştir.
- Büyük veri kümelerinde etkili bir şekilde çalışır.
- Eksik verilerin tahmin edilmesi için etkin bir yönteme sahiptir ve verinin büyük bir kısmı eksik olduğunda doğruluğu korur.
- Sınıfta hangi niteliklerin önemli olduğuna dair tahminlerde bulunur.
- Diğer yöntemlere göre gürültü açısından daha sağlamdır (Guo ve ark., 2004).

Rastgele ormanda iki farklı rastlantısallık kaynağı vardır: rastgele eğitim seti ve rastgele nitelik seçimi. Her bir düğümü ayırmak için rastgele bir özellik seçkisinin kullanılması, uygun hata oranları sağlar ve gürültüye göre daha sağlamdır. Çeşitlilik, bir ağacın her düğümündeki özelliklerin rastgele seçilmesi ve en yüksek düzeyde öğrenmeyi sağlayan özellikler kullanılarak elde edilir (Zavoral ve ark., 2010).

Ağaçlar ikili bölümleme kullanılarak yetiştirilir (her ebeveyn düğüm ikiden fazla çocuğa bölünmez) (Guo ve ark., 2004).

Her ağaç, bilgi kaybı nedeniyle daha fazla düğüm oluşturulmadan, budama olmadan mümkün olan en geniş alana kadar büyütülür (Zavoral ve ark., 2010).

Rastgele orman algoritması (hem sınıflandırma hem de regresyon için) aşağıdaki gibidir:

1. Orijinal verilerden yer değiştirmeli olarak N sayıda rastgele eğitim verisi elde edilir.
2. Her bir düğüm için M toplam girdi değişkenlerinden rastgele $m \leq M$ olacak şekilde değişkenleri seçilir. Bu m değeri orman geliştirme süresince sabittir.
3. Her bir ağaç muhtemel en geniş oranda geliştirilir. Sınıflandırma sırasında, kuralı durdurma ya da budama işlemleri yapılmaz.
4. Ağaçların tahminleri toplanarak yeni veriler tahmin edilir (örn. sınıflandırma için çoğunluk oyu, regresyon için ortalama).
5. m azalınca korelasyon ve güç azalır, m artınca korelasyon ve güç artar (Guo ve ark., 2004).

Lojistik Regresyon (Logistic Regression)

Lojistik regresyon, sınıflandırma sürecinin gerçekleştirilmesine yardımcı olan bir regresyon yöntemidir. Lojistik regresyondaki nokta, bağımlı ve bağımsız değişkenler

arasındaki ilişkiyi belirleyen ve en az değişken kullanarak en uygun olanın bilimsel olarak kabul edildiği bir model sunmaktır. Bilinen doğrusal regresyon analizinde bağımlı değişken ve bağımsız değişken(ler) sayısal (ölçümle belirtilen sürekli ya da kesikli sayısal) olarak belirtilir. Bağımlı değişken nitelik olarak belirtilirse, bağımsız değişken ya da değişkenlerle arasındaki ilişki lojistik regresyon yöntemiyle aranır. Lojistik regresyon, bağımlı değişkenler ikili, üçlü ve dörtlü olduğunda da kullanılan bir yöntemdir (Korkmaz ve ark., 2012).

Support Vector Machines (Destek Vektör Makineleri)

Makine öğrenmesinde , destek vektör makineleri (SVM'ler vektörel ağları destekler), sınıflandırma ve regresyon analizi için kullanılan veriyi analiz eden ilişkili öğrenme algoritmalarıyla denetimli öğrenme modelleridir. Her biri, her iki kategoriden birine ya da diğerine ait olarak işaretlenmiş bir dizi eğitim örneği verildiğinde, bir SVM eğitim algoritması, bir olasılık dışı ikili doğrusal sınıflandırıcı haline getirerek bir kategoriye ya da diğerine yeni örnekler atayan bir model oluşturur (metodlar olsa da SVM'yi olasılıksal bir sınıflandırma ayarında kullanmak için Platt ölçeklendirme gibi).

Doğrusal sınıflandırma gerçekleştirmenin yanı sıra, SVM'ler, çekirdek numarası diye adlandırılanları kullanarak doğrusal olmayan sınıflandırmayı verimli bir şekilde gerçekleştirebilir ve girişlerini yüksek boyutlu özellik alanlarına örtülü olarak eşlerler.

Veriler etiketlendirilmediğinde, denetimli öğrenme mümkün değildir ve verilerin gruplara kümelenmesini ve daha sonra bu gruplara yeni verilerle eşleştirmeyi deneyen denetimsiz bir öğrenme yaklaşımı gereklidir. Destek vektör makinelerine bir iyileştirme sağlayan kümeleme algoritmasına, destek vektör kümeleme adı verilir ve endüstri uygulamaları için ya veri işaretlenmediğinde ya da sadece bazı veriler bir sınıflandırma için bir ön işleme olarak etiketlendiğinde kullanılır.

Uygulama ve Analizler

Veri Kümesi ve Yapısı

Kullanılan veri kümesi 5172 rastgele seçilmiş e-posta dosyasının ilgili bilgilerini ve bunların spam veya spam olmayan sınıflandırması için ilgili etiketlerini içeren bir csv dosyasıdır.

*csv dosyası, her bir e-posta için her satırda 5172 satır içerir. 3002 sütun vardır. İlk sütun E-posta adını gösterir. Ad, gizliliği korumak için alıcıların adıyla değil, sayılarla belirlenmiştir. Son sütunda tahmin için etiketler bulunur: spam için 1, spam değil için 0. Kalan 3000 sütun, alfabetik olmayan karakterler/kelimeler hariç tutulduktan sonra , tüm e-postalarda en yaygın 3000 kelimedir .Veri seti kaggle'dan alındığı için veri setinin içeriğide ingilizcedir. Dolayısıyla en yaygın kullanılan kelimelerden ingilizce kelimelerdir. Her satır için, o e-postadaki (satırdaki) her kelimenin (sütun) sayısı ilgili hücrelerde saklanır. Böylece, 5172 e-postanın tümüne ilişkin bilgiler, ayrı metin dosyaları yerine kompakt bir veri çerçevesinde depolanır.

Çalışmada Python yazılımı kullanılmıştır. Python yazılımındaki sınıflandırma algoritmaları veriye uygunluğu göz önünde bulundurularak denenmiş, elde edilen sonuçlarda dört sınıflandırma algoritmasının iyi sonuç verdiği gözlemlenmiştir. Bu sınıflandırma algoritmaları; Naive Bayes, Random Forest, Logistic Regression ve Support Vector Machines algoritmalardır. Seçilen dört sınıflandırma algoritması da kendi içinde doğruluk, anma, duyarlılık ve RMSE kriterlerine göre karşılaştırılmıştır. Karşılaştırma sonucuna göre belirlenen en iyi veri madenciliği sınıflandırma algoritması; benzer özelliklere sahip veri kümelerine uygulanabilir ve e-postanın önemli olma olasılığı bu algoritma kullanılarak hesaplanabilir.

Veri kümesi e-posta %70 ve %30 olacak şekilde 2 parçaya bölünmüştür. Bir parçası "train" eğitim için sınıflandırma algoritmasına gider sonrasında %30'luk unlabeled olarak gelen kısma uygular.

Performansların Karşılaştırılması

Veri madenciliğinde sınıflandırıcıların karşılaştırılması ve en iyi sınıflandırıcının belirlenmesi çok önemlidir. Algoritmaların performanslarının karşılaştırılması doğru sınıflandırma yüzdesi (accuracy), anma(recall), duyarlılık(precision) ve kök hata kareler ortalaması (RMSE) ölçütlerine göre yapılmıştır.

Doğruluk Oranı (accuracy): Gözlemlerin doğru sınıflandırılma yüzdelerini verir.

Kök hata kareler ortalaması (RMSE): Hatanın ortalama büyüklüğünü ölçen ikinci dereceden bir puanlama kuralıdır. Formül kelimelerle ifade edilirse, tahmin ile karşılık gelen gözlemlenen değerler arasındaki farkın her birinin karesi alınır ve daha sonra örnek üzerinde ortalaması alınır. Son olarak, ortalamanın karekökü alınır. RMSE'nin düşük olması beklenir (Chai ve Draxler, 2014)

Navie Bayes Algoritması

Şimdi kelimeleri neden maillerden ayırdığımızı anlayalım. Bunun nedeni, bunun bir metin sınıflandırma problemi olmasıdır. Bir spam sınıflandırıcı bir postaya baktığında, önceki spam e-postalarında gördüğü olası kelimeleri arar. Bu kelimelerin çoğunu bulursa, bunu 'Spam' olarak etiketler. Çoğunluğa göre ayrılmasının sebebi :

Durum 1: Diyelim ki bir 'Selamlar' kelimesini alalım. Diyelim ki hem 'Spam' hem de 'Spam Değil' postalarında mevcut.

Durum 2 : Bir 'piyango' kelimesini ele alalım. Diyelim ki sadece 'Spam' maillerde var.

Durum 3: Bir 'ucuz' kelimesini ele alalım. Diyelim ki, yalnızca spam'de bulunur.

Şimdi bir test e-postası alırsak ve yukarıda bahsedilen üç kelimeyi de içeriyorsa, bunun bir 'Spam' postası olma olasılığı yüksektir.

Metin sınıflandırma problemleri için en etkili algoritma, klasik Bayes teoremi üzerinde çalışan Naive Bayes algoritmasıdır. Bu teorem, tahminlerde bulunmak için test verilerindeki her bir kelime üzerinde çalışır.

Diyelim ki test e-postamız "Bir piyango kazandınız" Navie Bayes bu veriler üzerinde şu şekilde çalışır:

P(S) = P('Bir') P('Piyango') P('kazandınız') (1)

Bu nedenle, P(S|Spam) = P('Bir'|Spam) P('Piyango'|Spam) P('kazandınız'|Spam) (2)

P(S|Not_Spam) için aynı hesaplama (3)

Eğer 2 > 3 ise, 'Spam' Değilse, 'Spam Değil'.

Olasılık sıfırsa : P(kelime) = (kelime_sayısı + 1)/(toplam_hayır_kelimesi + hayırın_unique_kelimeleri) olduğu Laplace Smoothing kavramı geliyor.

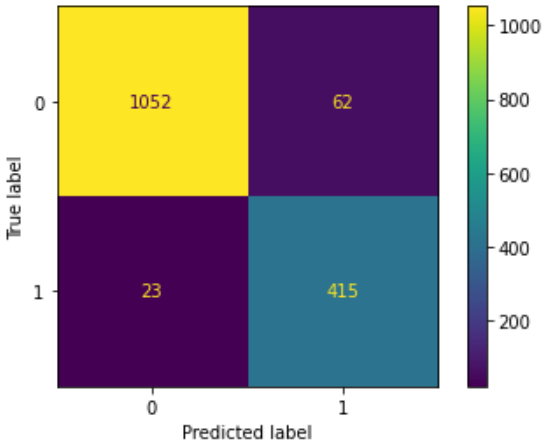
Burada, mevcut Multinomial Naive Bayes sınıflandırıcısı üzerinde çalıştık. (scikitlearn altında). Naive Bayes'in metin sınıflandırma için ne kadar iyi çalıştığını daha iyi anlamak için, iki modelin nasıl performans gösterdiğini görmek için başka bir standart sınıflandırıcı olan SVC'yi kullandık.

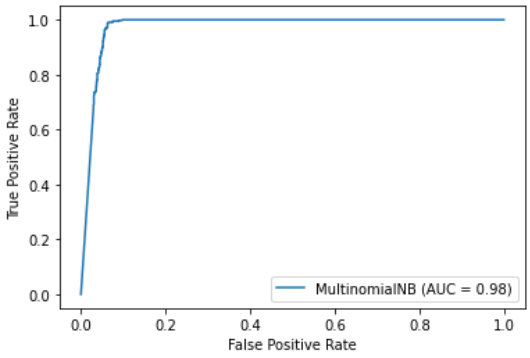
```
In [9]:
mnb = MultinomialNB(alpha=1.9)  # alpha varsayılan olarak 1'dir ve alpha her zaman > 0 olmalıdır.
# alfa, Laplace Düzgünleştirme formülündeki '1'dir (P(kelimeler))
mnb.fit(train_x,train_y)
y_pred1 = mnb.predict(test_x)
print("Accuracy Score for Naive Bayes : ", accuracy_score(y_pred1,test_y))

Accuracy Score for Naive Bayes :  0.9381283836040216
```

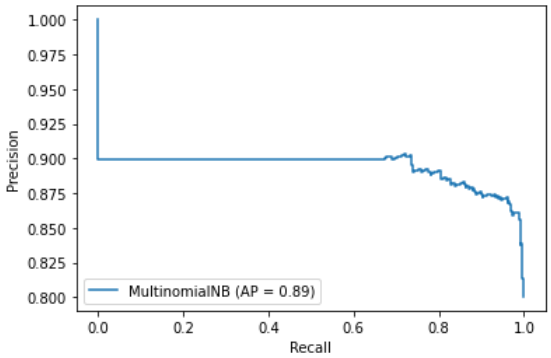
Şekil 7 Naive Bayes Algoritması

```
NAIVE BAYES MODEL
              precision    recall  f1-score   support

           0       0.94      0.98      0.96      1075
           1       0.95      0.87      0.91       477

    accuracy                           0.95      1552
   macro avg       0.95      0.92      0.93      1552
weighted avg       0.95      0.95      0.94      1552
```

Şekil 8 Navie Bayes Modeli







Algoritmamız başarılı bir şekilde sınıflandırma yapmıştır ama diğer algoritmalarımızdan daha düşük performans göstermiştir.

Random Forests Algoritması

Ensemble methods , herhangi bir zayıf modeli çok güçlü bir modele dönüştürür.

In [11]:
```
rfc = RandomForestClassifier(n_estimators=100,criterion='gini')

# n_estimators = Ormandaki ağaç sayısı
# criterion = ya gini impurity('gini') ya da bilgi kazancı('entropy') üzerinde karar ağacını bölmenin temeli

rfc.fit(train_x,train_y)
y_pred3 = rfc.predict(test_x)
print("Accuracy Score of Random Forest Classifier : ", accuracy_score(y_pred3,test_y))
```

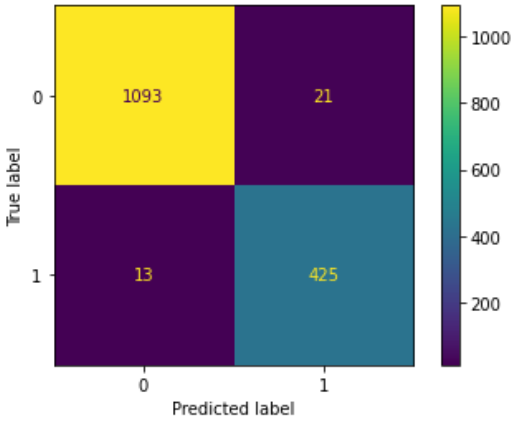
```
Accuracy Score of Random Forest Classifier :  0.9760247486465584
```

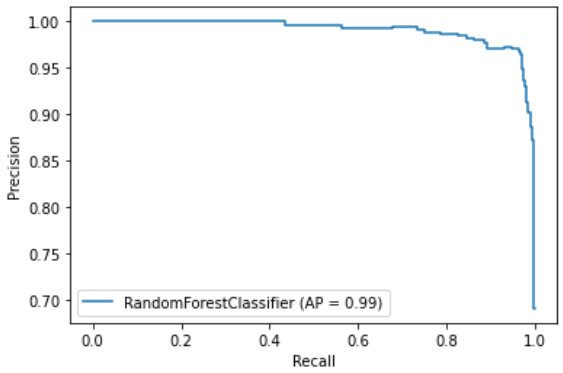
Şekil 9 Random Forests Algoritması

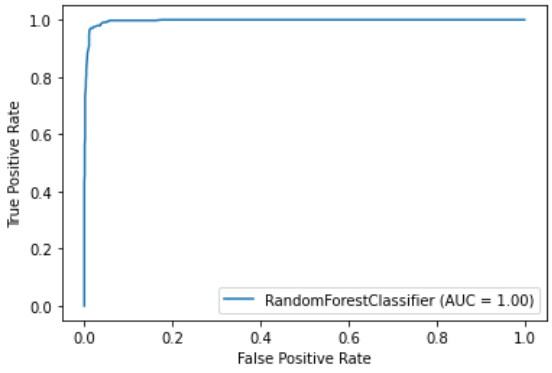
RANDOM FOREST MODEL

	precision	recall	f1-score	support
0	0.98	0.99	0.98	1106
1	0.97	0.95	0.96	446
accuracy			0.98	1552
macro avg	0.98	0.97	0.97	1552
weighted avg	0.98	0.98	0.98	1552

Şekil 10 Random Forests Algoritması Modeli







Algoritmamız başarılı bir şekilde sınıflandırma yapmıştır. Logistic Regression algoritmasından sonra en iyi performansı göstermiştir.

Support Vector Machines Algoritması

Destek Vektör Makinesi, klasik sınıflandırma problemleri için en çok aranan algoritmadır. SVM'ler, iki sınıfın destek vektörleri arasındaki maksimum marjı veya eşiği bulmak için (ikili sınıflandırmada) Maksimal Marj algoritması üzerinde çalışır. En etkili

Destek vektör makineleri, bir yanlış sınıflandırmaya izin veren yumuşak maksimal marj sınıflandırıcıdır, yani model daha sonra düşük varyansı sağlamak için düşük sapmayla (biraz düşük performans) başlar.

```
In [10]:
svc = SVC(C=1.0,kernel='rbf',gamma='auto')

# C burada düzenleme parametresidir. Burada L2 cezası kullanılır (varsayılan). Düzenlileştirme gücünün tersidir.
# C arttıkça model fazla sığar.( overfits)
# Buradaki çekirdek, radyal tabanlı fonksiyon çekirdeğidir.
# gama (yalnızca rbf çekirdeği için kullanılır) : Gamma arttıkça model fazla sığar.
svc.fit(train_x,train_y)
y_pred2 = svc.predict(test_x)
print("Accuracy Score for SVC : ", accuracy_score(y_pred2,test_y))
```

```
Accuracy Score for SVC :  0.9010054137664346
```

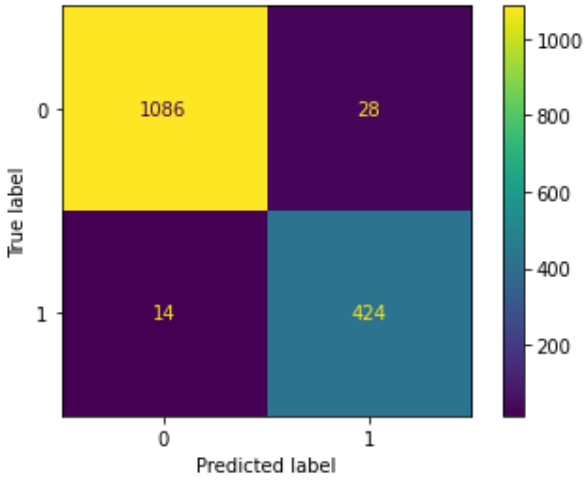
Şekil 11 Support Vector Machines Algoritması

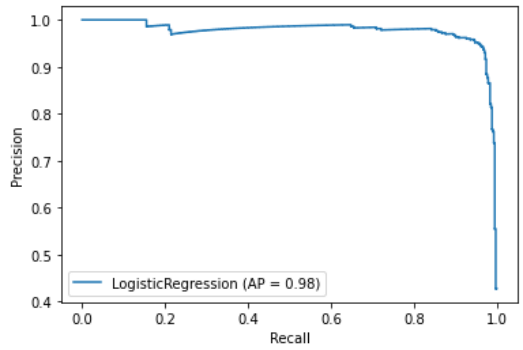
Beklendiği gibi, Support Vector Machines 'nin performansı Multinomial Naive Bayes'ten ve Random Forests'tan ve Logistic Regression'dan biraz daha zayıf performans göstermiştir.

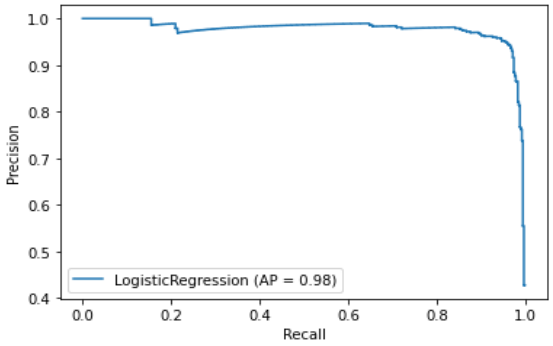
Logistic Regression Algoritması

```
LOGISTIC REGRESSION MODEL
              precision    recall  f1-score   support

           0       0.97      0.99      0.98      1100
           1       0.97      0.94      0.95       452

    accuracy                           0.97      1552
   macro avg       0.97      0.96      0.97      1552
weighted avg       0.97      0.97      0.97      1552
```

Şekil 12 Logistic Regression Algoritması







En başarılı sonuç elde ettiğimiz algortima Logistic Regression Algoritması oldu. Sınıflandırmayı çok başarılı şekilde en az hatayla gerçekleştirdi.

Sonuçlar

Dijital verinin toplanması ve saklanmasındaki gelişmeler, saklanan verilerin üstel bir şekilde büyümesine sebep olmuştur. Diğer yandan hayatın hızla elektronikleşiyor olması, internetin günlük yaşamın ayrılmaz bir parçası haline gelmesi toplanan veri artışını hızlandırmıştır. Veri madenciliği, veri tabanlarının araştırılması, gizli veri tahminlerinin büyük veri tabanlarından çıkarılması, veri ambarındaki önemli bilgileri analiz etme potansiyeline sahip güçlü ve yeni bir teknoloji olmakta ve beklentilerinin ötesine geçmektedir.

Sunulan makalenin amacı veri madenciliği algoritmalarını kullanarak önemli olan e-postaları ayırt edici belirgin farkları çıkarmak ve çıkan sonuçları karşılaştırıp bu problemde kullanılabilecek en iyi algoritmayı bulmaktır. Bu araştırmada, en uygun sınıflandırma algoritması veriye uygulanarak, hangi e-postaların önemlilik arz ettiğinin belirlenmiştir. Çalışmada Python yazılımı kullanılmıştır. Python yazılımındaki sınıflandırma algoritmaları veriye uygunluğu göz önünde bulundurularak denenmiş, elde edilen sonuçlarda sınıflandırma algoritmalarının nasıl sonuç verdiği gözlemlenmiştir. Bu makalede kullanılan veri kümesi 5172 rastgele seçilmiş e-posta dosyasının ilgili bilgilerini ve bunların spam veya spam olmayan sınıflandırması için ilgili etiketlerini içeren bir csv dosyasıdır. *csv dosyası, her bir e-posta için her satırda 5172 satır içerir. 3002 sütun vardır. İlk sütun E-posta adını gösterir. Ad, gizliliği korumak için alıcıların adıyla değil, sayılarla belirlenmiştir. Son sütunda tahmin için etiketler bulunur: spam için 1, spam değil için 0. Kalan 3000 sütun, alfabetik olmayan karakterler/kelimeler hariç tutulduktan sonra , tüm e-postalarda en yaygın 3000 kelimedir. Kullanılan sınıflandırma algoritmaları; Naive Bayes, Random Forest, Logistic Regression ve Support Vector Machines algoritmalardır. Sınıflandırma algoritmaları karşılaştırılmış ve önemli e-posta olma olasılıkları belirlenmiştir. Algoritmaların performanslarının karşılaştırılması doğru sınıflandırma yüzdesi (Accuracy), ağırlıklı ortalama anma (Recall), ağırlıklı ortalama duyarlılık (Precision) ve kök hata kareler ortalaması (RMSE) ölçütlerine göre yapılmıştır. Buna göre Random Forest ve Logistic Regression sınıflandırma algoritmalarının en iyi performansa sahip algoritma olduğu görülmüştür.

Bu makalede, en uygun sınıflandırma algoritması veriye uygulanarak, hangi e-postaların önemlilik arz ettiği belirlenmiştir. Bu uygulama sayesinde büyük şirketlerde çalışanların iş gücü yükünü hafifletilebilir. Gmail kullanıcıların önemli e-postaları algılaması için bir kural kümesi oluşturmasına izin verir, ama önemli postaların niteliği zamanla değiştikçe, bu kural kümeleri, kullanıcı tarafından sürekli olarak ayarlanmalı ve yeniden ayarlanmalıdır. Bu zaman alıcı ve genellikle hataya eğilimli, sıkıcı bir süreçtir. Yani kural tabanlı bir yaklaşımın önemli posta filtrelemede sınırlı bir faydası vardır. Bu çalışma sayesinde hangi e-postaların önemli olduğu daha kolay yoldan belirlenebilir. Ayrıca reklam içerikli ve okumaya gerek duyulmayan e-postalarda bu yolla daha kolay ortaya çıkar.Bu makalede veri madenciliği algoritmaları ile e-posta önemliliğini sınıflandırma yaparak belirlemiş olduk.

Kaynaklar

[1] Ajayi, O., V., 2017.Advance Statistical Methods in Education. Phd Thesis, Benue. State University, Nijerya

[2] Alpaydın, E., 2014. lntroduction to Machine Learning. The MIT Press, England,517

[3] Avcılar, M., Y., 2014. Association Rules in Data Mining: An Application on a Clothing and Accesory Speciality Store. Canadian Social Science, 10, 3:75- 83.

[4] Baydemir, M., B., 2014. Lojistik Regresyon Analizi Üzerine Bir İnceleme. İnönü Üniversitesi Sosyal Bilimler Enstitüsü Ekonometri Ana Bilim DalıMalatya.

[5] Bigml, 2018. Machine Learning made beautifully simple for everyone. https://bigml.com/. (Erim tarihi 18 Ekim 2018).

[6] Chai, T., Draxler, R., R., 2014. Root mean square error (RMSE) or mean absolute error (MAE)? Geoscientific Model Development, 7: 1247-1250.

[7] Coşkun, C., Baykal, A., 2011. Veri Madenciliğinde Sınıflandırma Algoritmalarının Bir Örnek Üzerinde Karşılaştırılması. 13. Akedemik Bilişim Konferansı Bildirileri, İnönü Üniversitesi, Malatya, 51-58

[8] Çığşar, B., 2017. Kredi Risklerinde Veri Madenciliği Sınıflandırma Algoritmaları. Çukurova Üniversitesi, Adana

[9] Dietrich, D., Heller, B., Yang, B., 2015. Data Science & Big Data Analytics. John Wiley & Sons, U.S.A., 409.

[10] Doğan, F., Türkoğlu, İ., 2018. Derin Öğrenme Algoritmalarının Yaprak Sınıflandırma Başarımlarının Karşılaştırılması. Sakarya University Journal of Computer and Information Sciences, 1: 1-2018.

[11] Durmaz, G., 2013.Gmail'de Bilmeniz Gereken 5 Özellik. Weblopedi.net/s=gmailde+bilmemiz+gereken. (Erim tarihi 18 Ekim 2018).

[12] Gmail Yardım, 2018. E-postalarınızı filtrelemek için kural oluşturma. http://support .google.com/mail/answer/6579?hl=tr#. (Erişim tarihi: 18 Ekim 2018).

[13] Grivas, S., G., 1999. Data Warehousing: concepts and Mechanisms, Wirtschaftsinformatik als mitter Zwischen Technic, Ökonomie und Gesselschaft, 61-69.

[14] Gua, L., Ma, Y., Cukic, B., Singh, H., 2004. Robust Prediction of Fault-Proneness by Random Forests. 15th International Symposium on Software Reliability Engineering. Fransa, 417-428.

[15] Gunjal, B., 2003. Database Managements: Concepts and Design. 24. IASLIC, Hindistan, 516-2003.

[16] Gümüş, G., 2018. Dünden Bugüne E-Posta: E-mailin Tarihçesi.http://www.brandingturkiye.com/dünden-bugüne-e-posta-e-mailin- tarihcesi. (Erişim tarihi: 18 Ekim 2018).

[17] Jovic, A., Brkic, K., Bogunovic, N., 2014. An Overview of free software tools for general data mining. 37th lnternational Convention on lnformation and communication Technology, Electronics and Microelectronics (MIPRO), Hırvatistan.

[18] Korkmaz, M., Güney, S., Yiğiter, Ş., Y., 2012. The Importance of Logistic Regression Implementations in The Turkish Livestock Sector and Logistic Regression Implementations/ Fields. Harran Üniversitesi Ziraat Fakültesi Dergisi, 16(2): 25-36.

[19] Krishnan, K., 2013. Data Warehousing in the Age of Big Data. Morgan Kaufmann, 335.

[20] Kuonen, D., 2004.Data Mining and Statistics: What is the connection? The Data Administration Newslatter, 30.

[21] Liew, A., 2013. DIKIW: Data, Information, Knowledge, Intelligence, Wisdom and Their lnterrelationships. Business Management Dynamics, 2, 10: 49-62. Mailtrack Support Center, 2018. What is Mailtrack?

[22] NIST Big Data Public Working Group. 2015. NIST Big Data lnteroperability Framework: Volume 3, Use Cases and General Requirements. 5-43.

[23] Ok, A., Güngör, O., Akar, Ö., 2011. Rastgele Orman Sınıflandırıcısı ile Arazi Kullanım Alanlarının Belirlenmesi. 3. Uzaktan Algılama ve coğrafi Bilgi sistemleri Sempozyumu, Kocaeli, 142-152.

[24] Oracle, 2018. What is Big Data? http//www.oracle.com/bidata/guide/what-is-big - data.html.

[25] Öğüt, S., 2005. Veri Madenciliği Kavramı ve Gelişim Süreci. Türkiye Bilişim Derneği, 1-12.

[26] Sahami, M., Dumais, S., Heckerman, D., Horvitz, E., 1998. A Bayesian Approachto Filtering Junk E-mail. AAAI workshop, U.S.A.