



**T.R.
GEBZE TEKNİK ÜNİVERSİTESİ
MÜHENDİSLİK FAKÜLTESİ
BİLGİSAYAR MÜHENDİSLİĞİ BÖLÜMÜ**

TREND YOUTUBE VİDEOLARI ANALİZİ

BİL 454 VERİ MADENCİLİĞİ DÖNEM PROJESİ RAPORU

ÖĞRENCİ
Şeyda Nur DEMİR
12 10 44 042

ÖĞRETİM ÜYESİ
Burcu YILMAZ

DERS ASİSTANI

-

KOCAELİ, 2021

PROJE AÇIKLAMASI

1.1.Gereklilikler :

In English :

- You should select a project topic related to data mining.
- You should not use deep learning models.
- You are expected to use as many data mining related models and data mining related techniques (preprocessing, postprocessing etc.) as possible.
- You are expected to analyze the effect of parameters on the results.
- You have to write a project report.
- You must present your project in the presentation lecture.
- You must attend demo section of your project.
- You will be graded according to how much effort you spend for your project.
- You will be graded according to the data mining related efforts.
- Do not spend so much effort on creating the dataset. It may not effect your grade that much.
- You are expected to implement the models by yourself. You can get help from tools but it shouldn't be more than %20 of your project.

1.2.Teslim Tarihi :

- Proje kaynak kodları ve raporu, 24 Ocak (2021) Pazar günü saat 23:55'e kadar, Moodle ortamı üzerinden teslim edilmelidir.

1.3.Demo :

- Projenin ilgili kodlarının demosu, 29 Ocak tarihinden önce, belirlenen bir gün ve saatte yapılacaktır.

1.4.Sunum :

- Sunumlar tüm sınıf ile birlikte, 29 Ocak (2021) Cuma günü, öğleden önce ve öğleden sonra olmak üzere iki kısımda gerçekleştirilecektir.

ÖZET

Bu projede, Youtube Trend Videoları, veri madenciliği konseptinde analiz edilir.

Youtube, kullanıcıların video izlemek, yüklemek ve paylaşmak üzere tercih ettiği yeni nesil internet uygulamalarındandır. Sağladığı imkanlar sayesinde internet kullanım becerisine sahip tüm insanlar, hem içerik üreticileri hem de içerik tüketicileri olarak söz konusu ortamı tecrübe edebilmektedir. Kişisel tercihler doğrultusunda çok çeşitli konularda farklı kullanım pratikleriyle karşımıza çıkan YouTube ortamı, üyelik şartı aramaksızın içeriklere ulaşım olanağı tanıyan yapısı ve paylaşılan içerikleri takip edilebilme özelliği ile her kuşaktan insanın farklı motivasyonlarla yararlanım durumunu beraberinde getirmektedir.

Bu çalışma, Youtube’da “en çok izlenen” videoların verileri incelenmiştir, videoların izlenme sayıları ile orantılı olan özellikleri ele alınmış, aralarındaki ilişki araştırılmıştır.

Veri Bilimi Konsepti :

Veri Madenciliği

Anahtar Kelimeler :

Veri Madenciliği, İlişki Matrisi, Lineer Regresyon Algoritması, Rastgele (Rassal) Orman Algoritması, Youtube Trend Videoları, Youtube Video İzlenme Sayısı, Youtube Video Beğen Sayısı, Youtube Video Beğenme Sayısı, Youtube Video Yorum Sayısı, Kullanıcı Davranış Analizi

A.PROJE İLE İLGİLİ BİLİNMEŞİ GEREKEN BAZI TERİMLER VE TANIMLAMALAR

Veri

İşlenmemiş, yorum yapmaya imkan verecek düzeyde birbiriyle ilişkilendirilmiş ham kayıtlar.

Enformasyon

Karar vermek için değeri olan ve organize edilmiş verilerin özetlenmesiyle elde edilen gerçekler.

Bilgi

Enformasyon verilerin analiz ve sentezlenmesi sonucu değer kazanmasıdır.

İstatistik

İstatistik veya sayıtlım, belirli bir amaç için veri toplama, tablo ve grafiklerle özetleme, sonuçları yorumlama, sonuçların güven derecelerini açıklama, örneklerden elde edilen sonuçları kitle için genelleme, özellikler arasındaki ilişkiyi araştırma, çeşitli konularda geleceğe ilişkin tahmin yapma, deney düzenleme ve gözlem ilkelerini kapsayan bir bilimdir. Belirli bir amaç için verilerin toplanması, sınıflandırılması, çözümlenmesi ve sonuçlarının yorumlanması esasına dayanır.

Olasılık

Olasılık ya da ihtimaliyet, bir şeyin olmasının veya olmamasının matematiksel değeri veya olabilirlik yüzdesi, değeridir. Olasılık kuramı istatistik, matematik, bilim ve felsefe alanlarında mümkün olayların olabilirliği ve karmaşık sistemlerin altında yatan mekanik işlevler hakkında sonuçlar ortaya atmak için çok geniş bir şekilde kullanılmaktadır.

Yapay Zeka

Yapay zekâ, bir bilgisayarın veya bilgisayar kontrolündeki bir robotun çeşitli faaliyetleri zeki canlılara benzer şekilde yerine getirme kabiliyeti. Yapay zekâ çalışmaları genellikle insanın düşünme yöntemlerini analiz ederek bunların benzeri yapay yönergeleri geliştirmeye yöneliktir.

Makine Öğrenmesi

Makine öğrenmesi yapısal işlev olarak öğrenebilen ve veriler üzerinden tahmin yapabilen algoritmaların çalışma ve inşalarını araştıran bir sistemdir. Bu tür algoritmalar statik program talimatlarını harfiyen takip etmek yerine örnek girişlerden veri tabanlı tahminleri ve kararları gerçekleştirebilmek amacıyla bir model inşa ederek çalışırlar.

Veri Madenciliği

Veri Bilimi (diğer bir deyişle Veri Madenciliği) geçmişi açıklamak ve veri analizi yoluyla geleceği tahmin etmekle ilgilidir. Veri bilimi, istatistik, makine öğrenimi, yapay zeka ve veritabanı teknolojilerini birleştiren çok disiplinli bir alandır. Veri bilimi uygulamalarının değerinin genellikle çok yüksek olduğu tahmin edilmektedir. Birçok işletme, yıllarca süren operasyonlar boyunca büyük miktarda veri depolamıştır ve veri bilimi bu verilerden çok değerli bilgiler çıkarabilir. İşletmeler daha sonra elde edilen bilgileri daha fazla müşteri, daha fazla satış ve daha fazla kar elde etmek için kullanabilirler. Bu aynı zamanda mühendislik ve tıp alanlarında da geçerlidir.

Veri Madenciliği

Basit bir tanım olarak veri madenciliği büyük ölçekli veriler arasında bilgiye ulaşma veya bilgiyi madenleme işidir. Büyük veri yığınları içerisinde gelecekle ilgili tahminde bulunabilmemizi sağlayabilecek bağıntıların bilgisayar programı kullanarak aranmasıdır. Bunun dışında bilgi keşfi, bilgi madenciliği, bilgi çıkarımı, veri ve model analizi, veri arkeolojisi... kullanılan bazı alternatif terimlerdir. Kısacası büyük veri yığınları arasında, görülmeyen bilgiyi ortaya çıkarma işlemidir.

Veri Madenciliğinin Kısa Tarihçesi

- 1950'lerde İlk bilgisayarlar matematiksel sayımlarda kullanıldı.
- 1960'larda Veri koleksiyonları, veri tabanı kullanımı başladı.
- 1970'lerde İlişkisel veri modeli ve ilişkisel RDMS uygulamaları geliştirildi.
- 1980'lerde İlişkisel RDMS kullanımı yaygınlaşmaya başladı.
- 1990'larda Günlük işlerde derlenen verinin nasıl değerlendirilebileceği sorgulanmaya başlandı.
- 1991'de Knowledge Discovery in Real Databases tanımı ve kavramları ortaya konuldu.
- 1992'de Veri madenciliği konusunda ilk yazılımın geliştirilmesi.
- 2000'lerde Veri ambarları ve veri madenciliği yaygınlaştı.

Veri Madenciliğinin Uygulama Alanları

- CRM Yönetimi (pazarlama kampanyasında getirinin maksimizasyonu, müşteri sadakatının artırılması)
- Pazarlama (müşteri satın alma alışkanlıkları belirlenmesi, pazar sepeti analizi, satışların tahmini)
- Bankacılık e Finans Sektörü (kredi kartı harcamalarına göre müşteri gruplarının belirlenmesi, kredi taleplerinin değerlendirilmesi)
- Elektronik Ticaret (saldırıların tespiti, web sayfalarına yapılan ziyaretlerin çözümlenmesi, kullanıcı davranışlarına göre web sitesinin yenilenmesi)
- Sigortacılık
- Sağlık ve İlaç Sektörü
- Spor Bilimleri
- Telekomünikasyon

ve benzeri alanlar, veri madenciliğinin kullanım alanlarıdır.

Veri Hazırlama

Veri hazırlama, keşif ve modelleme için kullanılmak üzere bir veya daha fazla veri kaynağından bir veri kümesi oluşturmakla ilgilidir. Verilere aşina olmak, verilerle ilgili ilk içgörülerini keşfetmek ve olası veri kalitesi sorunlarını iyi anlamak için ilk veri kümesiyle başlamak sağlam bir uygulamadır. Veri hazırlama genellikle zaman alan bir süreçtir ve hatalara büyük ölçüde eğilimlidir. Özellikle verilerin birçok geçersiz, aralık dışı ve eksik değerle toplandığı veri bilimi projeleri gibi, dikkatlice taranmamış verileri analiz etmek son derece yanıltıcı sonuçlar doğurabilir. Veri bilimi projelerinin başarısı büyük ölçüde hazırlanan verilerin kalitesine bağlıdır.

Veriseti

Veri kümesi, genellikle tablo biçiminde sunulan bir veri koleksiyonudur. Her sütun belirli bir değişkeni temsil eder ve her satır, verinin belirli bir üyesine karşılık gelir.

- **Sütunlar** : Alanlar, Nitelikler, Değişkenler
- **Satırlar** : Kayıtlar, Nesneler, Vakalar, Durumlar, Örnekler, Vektörler
- **Değerler** : Veriler

Veri Çıkarma, Dönüştürme, Yükleme

Veri kaynaklarından verileri çıkarır ve bir dizi dönüştürme işlevi kullanarak veri hedeflerine yükler.

- **Veri Çıkarma**
Veri çıkarma, düz dosyalar, ilişkisel veritabanları, akış verileri, XML dosyaları ve ODBC / JDBC veri kaynakları gibi çeşitli veri kaynaklarından veri ayıklama yeteneği sağlar.
- **Veri Dönüştürme**
Veri dönüştürme, verileri temizleme, dönüştürme, toplama, birleştirme ve bölme becerisi sağlar.
- **Veri Yükleme**
Veri yükleme, verileri güncelleme, açıklama ekleme veya silme yoluyla veya toplu olarak hedef veritabanlarına yükleme yeteneği sağlar.

Karar Destek Sistemleri

Karar Destek Sistemleri, deęişik kaynaklardan topladıęı bilgileri düzenleyerek, kararı modelleyerek, bilgileri analiz ederek ve deęerlendirme sonuçlarını sunarak karar vericiye seçim sırasında destek veren bilgisayar tabanlı sistemlerdir.

Veri Tabanları

Elde edilen verilerin tutulduęu alanlardır. Bir veri tabanı sistemi, birbiri ile iliřkili verilerin birikimini içeren, veriye eriřimi saęlayarak veriyi yönetmeye yardımcı olan yazılım programları kümesidir. Veri tabanları kullanım amaçlarına göre farklı isimler alırlar. Örnek olarak iliřkisel veri tabanları, işlemsel veritabanı, zaman serisi veritabanı verilebilir.

Veri Ambarları

Veri ambarları, tüm operasyonel işlemlerin en alt düzeydeki verilerine kadar inebilen, etkili analiz yapılabilmesi için özel olarak modellenen ve tarihsel derinlięi olan veri depolama sistematıęı olarak tanımlanabilir.

OLTP (Çevrimiçi İşlem Süreçleri)

Organizasyonda satın alma, kaydetme, muhasebe, bankacılık gibi günlük işlemlerin yapıldıęı işlemsel veritabanı sistemleridir. Detaylı bilgi içerir ve ayrıntılı görüntüye sahiptirler. Veriye eriřim saęlanabilir, üzerinde oynama yapılmasına izin verir. Saklanan kayıt sayısı sınırlıdır

OLAP (Çevrimiçi Analitik Süreçler)

Veri analizi ve karar verme için alt yapıyı oluşturan veri ambarı sistemleridir. İşlemsel veritabanı sistemlerinin aksine, bilgisel süreçler ile ilgilidir. Özet bilgi içerir ve çok boyutludurlar. Büyük boyutta kayıtlar saklanır.

Veri Madencilięi Bilgi Keřif Süreci

- **Veri Temizlięi**
Bu adım eksik, gürültülü, tutarsız verilerin temizlenme sürecidir.
- **Veri Bütünleştirme**
Birçok veri kaynaęından alınan verilerin birleştirilme sürecidir.
- **Veri Seçme**
Veritabanından alınan analiz ile ilgili verilerden, probleme iliřkin olan verileri seçme sürecidir.
- **Veri Dönüřtürme**
Bu aşama verinin uygun formlara dönüřtürölüp, veri madencilięinde kullanılabilcek hale getirilme sürecidir.
- **Veri Madencilięi Uygulaması**
Hazırlanan veriler üzerinden, amacına göre veri madencilięi algoritmalarının uygulanma sürecidir.
- **Desenler**
Bazı ölçümlere göre elde edilmiř bilgiyi temsil eden örüntüler tanımlama sürecidir.
- **Bilgi Sunumu**
Veri madencilięi elde edilmiř bilginin kullanıcıya sunulmasıdır.

Veri Madenciliği Metodolojileri

Metodolojiler, önemli veri madenciliği sorunlarını daha iyi anlamaya yarayan ve süreçlerinin nasıl yapılması gerektiğini ifade eden yöntemlerdir. Bu metodolojiler arasında CRISP-DM ve SEMMA en çok kullanılan metodolojilerdir.

- **CRIPS-DM Metodolojisi (Cross-Industry Standard Process for Data Mining)**

CRIPS-DM analitik, veri madenciliği ve veri biliminde en popüler metodolojidir. Veri madenciliği projelerini planlama ve yürütmeye kullanılan bir süreç modelidir.

Bu model 6 aşamadan oluşmaktadır.

- **İş Tanımlama**

Başlangıç olarak proje hedeflerini ve ihtiyaçlarını anlama ve bunu veri madenciliği tanımına dönüştürme aşamasıdır.

- **Veriyi Anlama**

Bu aşamada veri toplama işlemiyle başlar, veri kalitesi problemlerini belirleme, veriden ilk görüşleri çıkartma.. diye verinin probleme ne kadar çözüm getirdiğiyle devam eder.

- **Veriyi Hazırlama**

Topladığımız veriden veri seçme, veri temizleme, veri dönüştürme... gibi model uygun son veri setini elde etmek için yapılan işlemlerdir.

- **Modelleme**

Bu aşamada çeşitli modelleme tekniklerinin ve algoritmalarının seçilmesi, parametrelerin seçilmesi ve uygulama işlemleri gerçekleştirilir.

- **Değerlendirme**

Bu aşamada oluşturulan modelin deneme ve gözden geçirilmesi yapılır, gerekiyorsa iyileştirmeler yapılır.

- **Uygulama**

Son aşamada ise modelin analistlere ve son kullanıcılara sunulup iş süreçlerinde kullanılacak hale getirilir.

- **SEMMA Metodolojisi (Sample, Explore, Modify, Model and Assess)**

İstatistik ve İş Zekası yazılımı geliştiren SAS Enstitüsü tarafından geliştirilen ardışık adımlar listesidir. CRISP-DM olduğu gibi bütün projenin metodolojisi iken, SEMMA ise veri madenciliği yapılan kısmın metodolojisidir.

- **Sample, Örnekleme**

Bu aşama, veri örnekleme ile başlar, yani modelleme için veri seti seçilir.

- **Explore, Araştırma**

Bu aşamada, beklenen ve beklenmeyen değişkenler arasında, ilişkileri ve anormallikleri keşfedilerek, veriler anlaşılır hale getirilir.

- **Modify, Düzenleme**

Bu aşamada modelleme süreci için, verilerin temizlenmesi ve dönüştürülmesi yapılır.

- **Model, Modelleme**

Bu aşama, eğilim ve tahminleri keşfetmek için, modelin verilere uygulanması aşamasıdır.

- **Assess, Değerlendirme**

Bu aşamada uyguladığımız modelin, sonucumuza uygunluğunun değerlendirilmesi yapılır.

Öğrenme

- **Gözetimli Öğrenme**
Eğitim verileri üzerinden bir fonksiyon üreten bir makine öğrenmesi tekniğidir. Başka bir deyişle, bu öğrenme tekniğinde, girdilerle istenen çıktılar arasında eşleme yapan bir fonksiyon üretilir. Eğitim verisi hem girdilerden hem çıktılarından oluşur. Fonksiyon, sınıflandırma veya eğri uydurma algoritmaları ile belirlenebilir.
- **Gözetimsiz Öğrenme**
Bu yöntem, işaretlenmemiş veri üzerinden, bilinmeyen bir yapıyı tahmin etmek için, bir fonksiyon kullanan, makine öğrenmesi tekniğidir. Burada girdi verisinin hangi sınıfa ait olduğu belirsizdir.

Veri Madenciliğinde Kullanılan Modeller

Veri madenciliğinde kullanılan modeller ikiye ayrılmaktadır :

- **Tahmin Edici Modeller**
Sonuçları bilinen verilerden hareket ederek bir model oluşturup, sonuçları bilinmeyen veri kümeleri için sonuç değerlerinin tahmin edilmesidir.
 - **Sınıflama**
Sınıflama modelinde, kategorik bağımlı değişken tahmin edilmeye çalışılır.
 - **Regresyon**
Regresyon modelinde, süreklilik gösteren bağımlı değişken tahmin edilmeye çalışılır.
 - **Zaman Serisi Analizi**
Zaman serisi analizi yapılırken, verilerin zamana bağlı değişimleri incelenir.
- **Tanımlayıcı Modeller**
Karar vermeye rehberlik etmede kullanılabilecek verilerdeki örüntülerin tanımlanmasını sağlamaktadır.
 - **Kümeleme Yöntemi**
Kümeleme yöntemi, veri tabanlarındaki verilerin, gruplar veya kümeler altında toplanarak, benzer özelliklere sahip nesnelerin bir araya gelmesini sağlar.
 - **Birliktelik Kuralı**
Birliktelik kuralı, olayların birlikte gerçekleşme durumlarını çözümler.

Veri Madenciliği Yöntemleri

- **Birliktelik Kuralları**
Büyük veritabanlarında birbiriyle ilişkili değişkenleri ve aralarındaki bağlantının büyüklüğünü tespit etmek için kullanılan bir yöntemdir. Apriori, Carma, Eclat, Sequence, GRI.. birliktelik yönteminde kullanılan algoritmalarıdır.
- **Sınıflandırma ve Tahmin**
Gelecekteki veri eğilimlerini açıklamak için bir nesnenin niteliklerini inceleme ve bu nesneyi önceden tanımlanmış bir sınıfa atamaktır. Decision Tree, Random Forest, Navie Bayes, KNN.. sınıflandırma yönteminde kullanılan algoritmalarıdır. Tahminleme, veri seti içinde bilinmeyen veya eksik olan sayısal verilerin tahmin edilmesidir.
- **Kümeleme Analizi**
Kümelemede amaç dağınık halde duran verileri özelliklerine göre birleştirip işlenebilir hale getirmektir. Sınıflandırmaya benzer ama aradaki fark, kümelerin önceden belirlenmemiş olmasıdır. K-Means, K-Metoids.. algoritmaları kümeleme yönteminde kullanılan algoritmalarıdır.
- **Aykırılık Analizi**
Verilerin algoritmalar ile kontrol edilerek verilerde aşırı sapma veya aykırı değerlerin bulunma sürecidir. Sıradışı veriler okuma, kayıt etme, ölçüm gibi hatalardan oluşmaktadır. Veri madenciliği algoritmaları ise bu sıradışı verileri en aza indirme veya ortadan kaldırmayı amaçlamaktadır.

Veri Analizi

Veri keşfi, verilerin istatistiksel ve görselleştirme teknikleriyle açıklanmasıyla ilgilidir. Daha fazla analiz için, bu verilerin önemli yönlerini odak noktasına getirmek adına veriler araştırılır.

- **Tek Değişkenli Veri Analizi**

Tek değişkenli analiz, değişkenleri (öznitelikleri) tek tek araştırır. Değişkenler kategorik veya sayısal olabilir. Her değişken türü için farklı istatistiksel ve görselleştirme araştırma teknikleri vardır. Sayısal değişkenler, sepetlere ayırma veya ayırıklaştırma adı verilen bir işlemle kategorik emsallere dönüştürülebilir. Kodlama adı verilen bir işlemle kategorik bir değişkeni sayısal karşılığına dönüştürmek de mümkündür. Son olarak, eksik değerlerin düzgün işlenmesi, madencilik verilerinde önemli bir konudur.

- **Kategorik (Ayrık) Veriler**

Kategorik veya ayrık bir değişken, iki veya daha fazla kategoriye (değer) sahip olandır.

- **Nominal**

Nominal bir değişkenin, kendi kategorilerine özgü bir sıralaması yoktur.

- **Sıralı**

Sıralı bir değişkenin net bir sıralaması vardır.

- **Sıklık Tablosu**

Söz konusu değişkenin her bir kategorisinin ne sıklıkta meydana geldiğini saymanın bir yoludur. Her bir kategoriye düşen yüzdelerin eklenmesiyle geliştirilebilir.

- **Sayısal (Sürekli) Veriler**

Sayısal veya sürekli bir değişken (öznitelik), sonlu veya sonsuz bir aralık içinde herhangi bir değeri alabilen bir değişkendir.

- **Aralık**

Bir aralık değişkeni, farklılıkları yorumlanabilir değerlere sahiptir, ancak gerçek sıfıra sahip değildir. Aralık ölçeğindeki veriler eklenebilir ve çıkarılabilir ancak anlamlı bir şekilde çarpılamaz veya bölünemez.

- **Oran**

Bir oran değişkeni gerçek sıfır değerlerine sahiptir ve eklenebilir, çıkarılabilir, çarpılabilir veya bölünebilir.

- **Çift Değişkenli Veri Analizi**

İki değişkenli analiz, iki değişkenin (özniteliklerin) eşzamanlı analizidir. İki değişken arasındaki ilişki kavramını, bir ilişki olup olmadığını ve bu ilişkinin gücünü veya iki değişken arasında farklılıklar olup olmadığını ve bu farklılıkların önemini araştırır.

- **Sayısal ile Sayısal**

Dağılım Grafiği ve Doğrusal Korelasyon kullanılabilir.

- **Kategorik ile Kategorik**

Yığın Sütun Grafiği, Kombinasyon Tablosu ve Ki-kare Testi kullanılabilir.

- **Sayısal ile Kategorik**

Hata Çubuklu Çizgi Grafik, Kombinasyon Tablosu, Z-Testi , T-Testi ve Varyans Analizi kullanılabilir.

Modelleme

- Tahmine dayalı modelleme, bir sonucu tahmin etmek için bir modelin oluşturulduğu süreçtir. Sonuç kategorik ise buna sınıflandırma, sonuç sayısal ise regresyon denir.
- Açıklayıcı modelleme veya kümeleme, gözlemlerin kümelere atanmasıdır, böylece aynı kümedeki gözlemler benzer olur.
- Son olarak, ilişkilendirme kuralları gözlemler arasında ilginç ilişkiler bulabilir.

Model Değerlendirme

Modeli değerlendirme, model geliştirme sürecinin ayrılmaz bir parçasıdır. Verilerimizi temsil eden en iyi modeli ve seçilen modelin gelecekte ne kadar iyi çalışacağını bulmamıza yardımcı olur.

- **Sınıflandırma Değerlendirmesi**

Karışıklık Matrisi, Birikimli Kazanım Grafiği, Kazanım Grafiği, K-S Grafiği, ROC Grafiği, AUC Grafiği kullanılabilir.

- **Regresyon Değerlendirmesi**

Bir dizi farklı regresyon modeli oluşturduktan sonra, değerlendirilebilecekleri ve karşılaştırılabilecekleri çok sayıda kriter vardır.

RMSE, RSE, MAE, RAE, R^2 , SST, SSR, SSE, SRE kullanılabilir.

Model Değerlendirme İçin Veri Setinin Bölünmesi

Eğitim için kullanılan verilerle model performansının değerlendirilmesi, veri biliminde kabul edilemez çünkü kolayca aşırı iyimser ve aşırı uyumlu modeller oluşturabilir. Aşırı uyumu önlemek için, model performansını değerlendirmek için bir test seti (model tarafından görülmeyen) kullanır.

- **Bekletme**

Bu yöntemde, çoğunlukla büyük veri kümesi rastgele üç alt gruba bölünür :

- **Eğitim Seti**

Tahmine dayalı modeller oluşturmak için kullanılan veri kümesinin bir alt kümesidir.

- **Doğrulama Seti**

Eğitim aşamasında oluşturulan modelin performansını değerlendirmek için kullanılan veri setinin bir alt kümesidir. Modelin parametrelerinin ince ayarını yapmak ve en iyi performans gösteren modeli seçmek için bir test platformu sağlar. Tüm modelleme algoritmalarının bir doğrulama setine ihtiyacı yoktur.

- **Test Seti**

Görünmeyen örnekler, bir modelin gelecekteki olası performansını değerlendirmek için kullanılan veri setinin bir alt kümesidir. Eğitim setine bir model, test setine uyduğundan çok daha iyi uyuyorsa, bunun nedeni muhtemelen aşırı uyumdur.

- **Çapraz Doğrulama**

Yalnızca sınırlı miktarda veri mevcut olduğunda, model performansının tarafsız bir tahminini elde etmek için k-kat çapraz doğrulama kullanırız. K-kat çapraz doğrulamada, verileri eşit büyüklükte k alt kümeye böleriz. Her seferinde alt kümelerden birini eğitimden çıkararak k kez modeller oluştururuz ve bunu test kümesi olarak kullanırız. Eğer k örneklem büyüklüğüne eşitse, buna "birini dışarıda bırak" denir.

Doğruluk (Accuracy)

Doğru olan toplam tahmin sayısının oranı.

Pozitif Öngörücü Değer (Positive Predictive Value) veya Kesinlik (Precision)

Doğru şekilde tanımlanan olumlu vakaların oranı.

Negatif Öngörücü Değer (Negative Predictive Value)

Doğru şekilde tanımlanan olumsuz vakaların oranı.

Hassasiyet (Sensitivity) veya Geri Çağırma (Recall)

Doğru şekilde tanımlanan gerçek olumlu vakaların oranı.

Özgüllük (Specificity)

Doğru şekilde tanımlanan gerçek olumsuz durumların oranı.

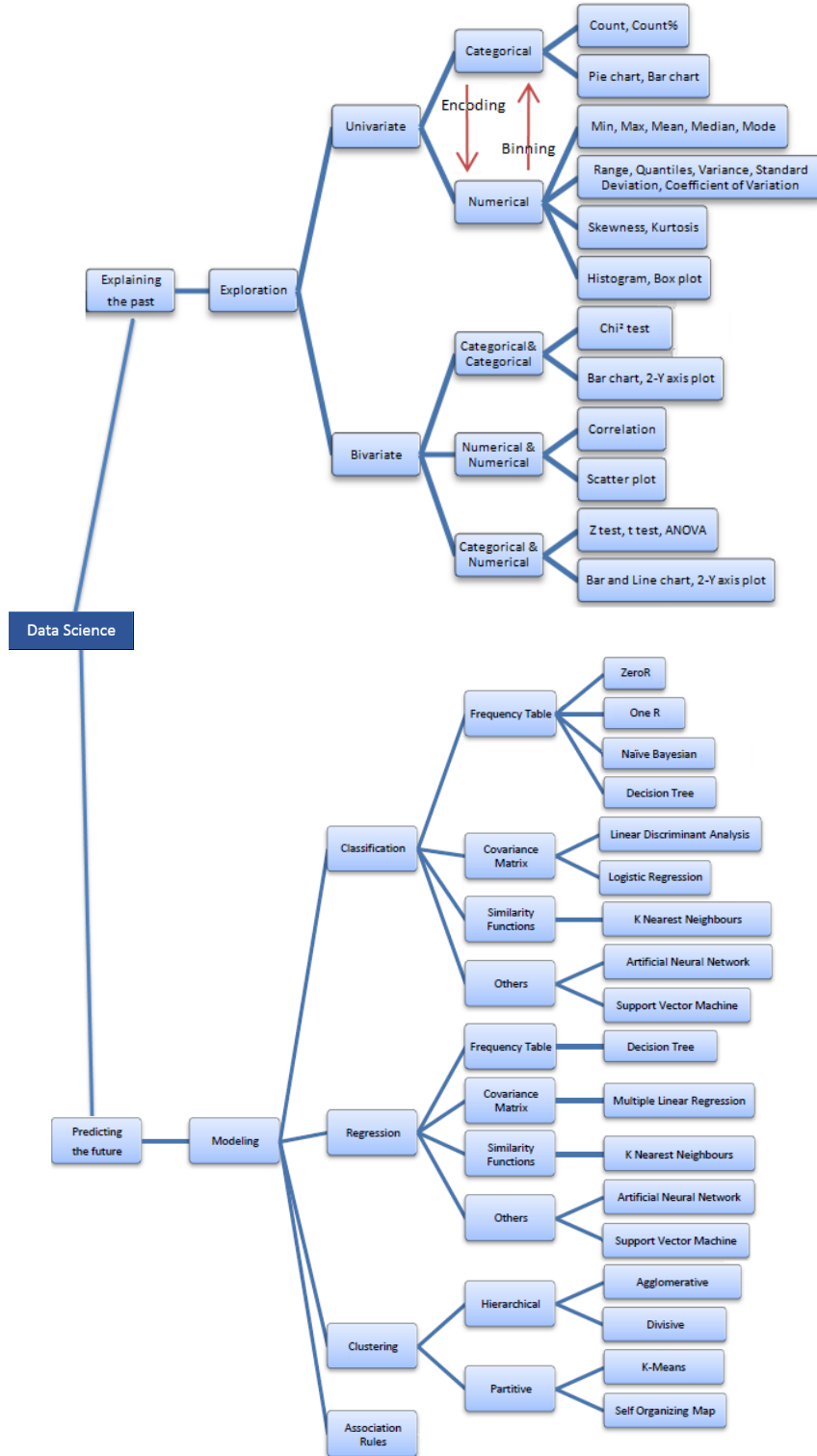
Veri Madenciliğinde Kullanılan Yazılımlar, Uygulama Dilleri ve Kullanım Alanları

- **Rapid Miner (Java)**
Veri madenciliği, veri ön işleme ve görselleştirme, öngörülü analitik ve istatistiksel modelleme, değerlendirme ve uygulama gibi işlevsellik de sağlar; WEKA ve R komut dizilerinden öğrenme şemaları, modeller ve algoritmalar sunar.
- **Weka (Java)**
Veri ön işleme, kümeleme, sınıflandırma, regresyon, görselleştirme ve özellik seçimi gibi birkaç standart veri madenciliği görevini destekler.
- **R Programming (R)**
Veri madenciliği, doğrusal ve doğrusal olmayan modelleme, klasik istatistiksel testler, zaman serileri analizi, sınıflandırma, kümeleme ve bazı istatistiksel ve grafik teknikleri sağlar.
- **Orange (Python)**
Makine öğrenimine, biyoinformatik ve metin araştırmasında kullanılabilecek eklentiler için bileşenlere ve veri analitiği özelliklerine sahiptir.
- **KNIME (Java)**
Veri işleme, makine öğrenimi ve veri madenciliği için çeşitli bileşenlerin entegrasyonu, iş zekası ve finansal veri analizi sağlar.
- **NLTK (Python)**
Dil işleme, veri madenciliği, makine öğrenmesi, veri kazıma, duygu analizi ve diğer çeşitli dil işleme görevleri de dahil olmak üzere dil işleme araçlarının bir havuzunu sağlar.
- **Rattle GUI (R)**
Verilerin istatistiksel ve görsel özetlerini sunar, verileri kolayca modellenebilecek şekilde dönüştürür, veriden denetlenmemiş ve denetlenen makine öğrenme modelleri oluşturur, modellerin performansını grafiksel olarak sunar ve üretime geçiş için yeni veri setlerini puanlandırır.
- **Oracle DB (SQL)**
Veri tabanında çeşitli veri madenciliği algoritmaları uygulanır.
- **Tanagra (HTML)**
Görselleştirme, tanımlayıcı istatistikler, örnek seçimi, özellik seçimi, özellik oluşturma, regresyon, faktör analizi, kümeleme, sınıflandırma ve ilişki kuralı öğrenimi gibi birkaç standart veri madenciliği görevini desteklemektedir.
- **SAS (-)**
Çeşitli kaynaklardan veri toplamayı, değiştirmeyi, yönetmeyi ve almayı ve bunun üzerine istatistiksel analiz yapabilmeyi sağlar.
- **SPSS (-)**
Veri madenciliği, metin analizi, tahmini modeller oluşturma, ve diğer analitik görevleri yapmamızı sağlar.

B. PROJEDE TAKİP EDİLEN GENEL SIRA

Veri Bilimine Giriş Haritası (Saed Sayad)

Projem boyunca çokça faydalandığım, veri madenciliği konularına çalışırken sürekli başvurduğum bu haritayı, raporumda paylaşmak istedim.



1.GİRİŞ

Projeye Samimi Bir Bakış

Öncelikle, projeye başlamadan önce temelde ne python bilgimin, ne de veri madenciliği hakkında bir deneyiminin olmadığını belirtmeliyim. Dolayısıyla bu rapor benim için biraz da, dönemin özeti niteliğindedir. Yaptığım bazı işlemler çok basit veya yanlış olabilir, ancak hepsi projemde ilerlerken edindiğim bilgilerdir, kazandığım deneyimdir, dolayısıyla benim için çok önemlidir. Projenin verileceğini duyduğum ilk dersten beri, veri madenciliği konseptine çalıştım, yapılan işlemleri öğrendim, örnek çalışmaları inceledim, Udemy ve Youtube üzerinden kısa eğitim videoları serileri izledim, genel bir bilgi ve birikim edindim. Bu ilerleme sırasında, öğrendiklerimi yukarıda (BÖLÜM A) kısa kısa temel bilgiler şeklinde yazdım. Faydalandığım kaynak oldukça rapor sonuna kaynakça şeklinde ekledim, doğru bir kaynak gösterme biçimi olmasa da, faydalandığım tüm içerikleri aşağıda belirttim.

Projemi Seçme Amacım

Projemizde bir konu belirlememiz, bu konuda ulaşmak üzere bir amaç belirlememiz, bu amacı gerçekleştirmek için bir veriseti bulup üzerinde çalışmamız gerekmektedir. Ben veriseti araştırırken, bu konuyu gördüm, ve bana yakın geldiği için seçtim. Kredi kartı dolandırıcılığı, market alışveriş alışkanlıkları analizi, bitki türleri vb. gibi genel konular da seçebildim, ancak bu konular hakkında kendim bilgi sahibi olmadığım için, üzerinde nasıl çalışabileceğim konusunda herhangi bir şey canlandıramadım kafamda. Dolayısıyla, nedir ne değildir bildiğim, konu ve amaç belirleyebileceğim bir şey seçmeye karar verdim.

Projemin Konusu

Projemin konusu, Youtube’da izlenme sayısı ilk 200’e girmiş olan, yani “trend” olarak adlandırdığımız videoların analizinin yapılmasıdır.

Projemin Önemi

Her geçen gün, Youtube’da içerik üreten içerik üreticileri artmakta, ve Youtube içeriklerinden faydalanan kullanıcı sayısı da aynı şekilde artmaktadır. Son istatistikler, YouTube’a gelen veya YouTube’dan gelen trafiğin toplam web trafiğinin %20’sinden fazlasını ve tüm internet trafiğinin %10’unu oluşturduğunu göstermektedir. Amazon’un sahip olduğu web trafiği izleme hizmeti Alexa’ya göre YouTube, her dakika 300 saatin üzerinde video yükleyen ve her gün 5 milyar video izlenen dünyanın en popüler ikinci web sitesidir. Burada içerik üreticiler için, videoların izlenme sayısı büyük önem arz etmektedir. İçerik üreticileri birbirlerinin önüne geçiren konu, videolarına aldıkları etkileşimdir. Etkileşim sayısı yüksek olan kanalların sahipleri, bundan, hem maddi hem manevi kazanç sağlamaktadır. Dolayısıyla bir içerik üretici için, daha çok nasıl etkileşim kasabileceği, büyük bir sorun haline gelmiştir. Eğer içeriğin bir özelliği ile aldığı etkileşimleri ilişkilendirebilirsek, bu içerik üreticiler için büyük fayda sağlayabilir.

Projemin Amacı

Youtube’da trend olan videoların, belirli özellikleri vardır, bu özellikler daha çok izlenmelerini sağlamıştır. Projemde genel amaç, bir videonun hangi özellikleri izlenme sayısının artmasını sağlar, bunun hakkında bir fikir sahibi olmak, bir görüş kazanmaktır.

Projeme Benzer Çalışmalar

Bu şekilde, yine trend olmuş Youtube videoları üzerinde analiz yapan birçok çalışma mevcuttur. Sadece trend videoları değil, belirli bir konuda üretilen video içerikleri ve bunların belirli hedef kitleler üzerindeki etkileri de araştıran çalışmalar mevcuttur.

Projeme Başlamadan Önce Yaptığım Literatür Çalışması

Projeme başlamadan önce, akademik makaleleri, okudum ve inceledim. Çoğunluğu tabii ki İngilizce dilinde yapılan bu yayınların, bana en büyük katkısı, Youtube videolarının sandığımdan daha büyük bir önem arz ettiğini görmek oldu. Ayrıca online video içeriği üzerine, nasıl bir çalışma yapılabilir, hangi algoritmalar ne amaçla faydalanılabilir bunu görmüş oldum.

Projemle İlgili Yaptığım Pazar Araştırması

Açıkçası, bu konuda bir üründen çok, video içerik üreticilerine danışmanlık sağlayan ajans şirketleri mevcut. Bu ajanslar, video içerikleri hakkında içerik üreticileri yönlendiren, onlara içerik üretebilmeleri için gerekli uygun koşulları sağlayan şirketler şeklinde düşünülebilir.

Ayrıca, beni veri madenciliği konsepti ile tanıştıran, bana bu deneyimi tecrübe etme fırsatı veren, Burcu YILMAZ hocama, teşekkürlerimi sunarım.

2.KULLANILAN METOTLAR

Kullandığım Veriseti

Öncelikle belirtmeliyim ki, daha önce bu alanda böyle bir çalışmam olmadığı için, karar vermekte çok zorlandım. Araştırdıkça, birçok çalışma ve veriseti buldum. Hangi verisetlerinde ne çalışmalar yapılmış, hangi konularda nasıl verisetleri mevcut, bunları inceledim. Verisetlerine özel websitelerinin olduğunu da bu proje sayesinde öğrendim. “Kaggle” websitesini keşfettim, içerisindeki verisetlerini ve verisetlerinde yapılan çalışmaları inceledim. Verisetleri aranabiliyor, türlerine boyutlarına göre filtrelenebiliyor ve sıralanabiliyordu. Ben daha çok bildiğim bir konu olsun istedim. Çok bilindik “Iris” gibi verisetlerini kullanmamamız isteniyordu. Benim “Iris” verisetinden haberim bile yoktu, araştırdığımda gördümki “Iris” veriseti kütüphane import eder gibi koda dahil edilebiliyor bile. Bunu görünce böyle bir veriseti olmaması için de çalıştım. Listede ilk çıkan verisetlerini seçmedim. Ayrıca verisetimizin mükemmel olmaması, üzerinde çalışma yapabilmemize uygun olması isteniyordu. Dolayısıyla mükemmel olmayan verisetlerine baktım. Bu esnada da gördüm ki, makine öğrenmesi için, model oluştururken kullanılmak üzere, modelin daha doğru sonuçlar vermesi isteniyor ve onlara ait verisetleri mükemmele yakın. Dolayısıyla buna benzer verisetleri gördüğümde onları da seçmedim. Veriseti araştırırken, Youtube video istatistiklerini içeren verisetleri gördüm. Kendim daha önce de çalışırken, sosyal medya reklamcılığı üzerine hem eğitim almıştım, hem de daha verimli reklamlar vermek için, önceden verdiğimiz reklam sonuçlarını grafiklerden değerlendirip, reklam hedef kitlemizi reklam konusuna göre belirliyorduk. Bu verisetini görünce, hem tanıdık hissettim, hem de bu konuda yapılabilecek çalışmalar gelmeye başladı aklıma. Dolayısıyla verisetimi bu konuda tekrar araştırdım. Hem proje kurallarına uygun, hem de içeriği bakımından çalışmaya müsait olduğu için, “Kaggle” websitesi üzerinden, “Mitchell J” kullanıcısına ait, “Trending YouTube Video Statistics - Daily statistics for trending YouTube videos” verisetine karar verdim.

Kullandığım Veri Madenciliği Aracı

Derslerimizin başladığı ilk haftalarda, projemizin olduğunu, ve araç kullanabileceğimizi, ancak kullanmaz kendimiz gerçekleştirsek daha iyi sonuçlar alabileceğimizi öğrendiğimde, hemen veri madenciliğinde kullanılan araçlar nelerdir, nasıl kullanılır diye araştırdım. Youtube üzerinde “Şadi Evren Şeker”in birkaç veri madenciliği aracı (Orange, Knime, Weka, R Programming) ile ilgili yapmış olduğu eğitimler buldum, izledim. Araçlarla gerçekten işin çok kolaylaşacağını görmüş oldum. Hem de bu esnada, bir veriseti üzerinde yapılabilecek işlemler öğrenmiş oldum. Ancak puanımı etkileyeceği düşüncesi ile, kendim çalışmaya karar verdim. Dolayısıyla herhangi bir araç **kullanmadım**.

Kullandığım Programlama Dili

Projemde herhangi bir araç kullanmadım, ancak zaten python programlama dilinin de veri madenciliğine yardımcı birçok kütüphanesi olduğunu öğrenmiştim. Dolayısıyla programlama dili olarak “Python”ı tercih ettim.

Kullandığım Geliştirme Ortamı

Python geliştirirken, ilk aracın Anaconda olacağı, Spyder IDE’sinin çok verimli olduğu, vb. birçok bilgiyle karşılaştım. Hepsini, defalarca indirip, sürüm sorunları yaşayıp, mecburen kaldırıp, farklı sürümlerini indirip, tekrar deneyip deneyip, en son hiçbir araç kullanmamaya karar verdim. İlerlerken kullanmak istediğim bir fonksiyon bulduysam, bunun hangi

kütüphanede, hangi sürümde doğru çalıştığına bakıp, gerekli kurulumları yapıp, fonksiyonları ve kütüphaneleri kullandım. Sadece kodlarımı yazarken ve derlerken, bazen “**Jupyter-Notebook**” kullandım, bazen de “**MS Visual Studio Code**” uygulamasını kullandım.

Kullandığım Yazılım Kütüphaneleri

Öncelikle Python’ın belli başlı kütüphanelerine sıklıkla başvurdum, bunlar “**numpy**”, “**pandas**”, ve görselleştirmede “**matplotlib**”, “**seaborn**”. Bunlara ek, deneme amaçlı “**wordcloud**” kütüphanesini, ve algoritmalarda kullanmak üzere “**scikitlearn**” kütüphanesini kullandım.

Kullandığım Veri Madenciliği Algoritmaları

Projemde, faydalı olacağını düşündüğüm için, “**Lineer Regresyon**” ve “**Rastgele (Rassal) Orman**” algoritmalarını kullandım.

Kaynak Kod ve Çıktılar Üzerinden Adım Adım Projemin Detayları

Bu kısımda, “**MS Visual Studio Code**” ortamında, “**Python Notebook**” üzerinde, adım adım çalıştırdığım kaynak kodum ve aldığım çıktılarımdan ekran görüntüleri üzerinden, sırayla projemde ne yaptığımdan bahsedeceğim.

Raporumda kendimi ve yaptıklarımı daha rahat ifade edebileceğimi düşündüğüm için “**Türkçe**” kullanmama rağmen, dönem sonunda açık kaynak olarak paylaşacağım ve daha çok insana hitap etmesini istediğim için, kaynak kodumu “**İngilizce**” yorumlandırımdı.

CSE 454 - Data Mining Course

Trending YouTube Video Statistics Project

By Seyda Nur DEMİR

Burada markdown ile projeme giriş bilgileri yazdım.

```
Importing libraries

import pandas as pd
import seaborn as sns
import numpy as np
import matplotlib
import matplotlib.pyplot as plt
from wordcloud import WordCloud, STOPWORDS
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
from sklearn.metrics import mean_squared_error
from sklearn.metrics import r2_score
from sklearn.ensemble import RandomForestRegressor
from sklearn.model_selection import GridSearchCV
import warnings; warnings.simplefilter('ignore')
```

Burada gerekli kütüphaneleri import ettim.

Loading the data											
<pre> > MI videos = pd.read_csv("USvideos.csv") videos.head() </pre>											
video_id	trending_date	title	channel_title	category_id	publish_time	tags	views	likes	dislikes	comment_count	
0	2ky56svSYSE	17.14.11	WE WANT TO TALK ABOUT OUR MARRIAGE	CaseyNeistat	22	2017-11-13T17:13:01.000Z	SHANTell martin	748374	57527	2966	1595
1	1ZAPwfrtAFY	17.14.11	The Trump Presidency: Last Week Tonight with J...	LastWeekTonight	24	2017-11-13T07:30:00.000Z	last week tonight trump presidency "last week ...	2418783	97185	6146	1270
2	5qpjK5DgCt4	17.14.11	Racist Superman Rudy Mancuso, King Bach & Le...	Rudy Mancuso	23	2017-11-12T19:05:24.000Z	superman "rudy" "mancuso" "king" "bach"...	3191434	146033	5339	818
3	puqawrEC7tY	17.14.11	Nickelback Lyrics: Real or Fake?	Good Mythical Morning	24	2017-11-13T11:00:04.000Z	rhett and link "gmm" "good mythical morning" "...	343168	10172	666	214
4	d380meDdW0M	17.14.11	I Dare You: GOING BALDI?	nigahiga	24	2017-11-12T18:01:41.000Z	ryan "higa" "higatv" "nigahiga" "i dare you" "...	2095731	132235	1989	1751

Burada, pandas kütüphanesinin bir fonksiyonu olarak, verisetimi tek satırda dataframe içerisine okudum. Ve içeriği hakkında fikir sahibi olabilmek adına, ilk 5 satırı ekrana yazdırdım.

Loading the category data						
<pre> > MI category_id = pd.read_csv("US_category_id.csv") category_id.head() </pre>						
kind	etag	id	snippet/channelId	snippet/title	snippet/assignable	
0 youtube#videoCategory	"m2yskbQFythfE4irbTieOgYyFBU/Xy1mB4_yLrHy_BmKm...	1	UCBR8-60-B28hp2BmDPdntcQ	Film & Animation	True	
1 youtube#videoCategory	"m2yskbQFythfE4irbTieOgYyFBU/UZ1oLI1z2dx1h045Z...	2	UCBR8-60-B28hp2BmDPdntcQ	Autos & Vehicles	True	
2 youtube#videoCategory	"m2yskbQFythfE4irbTieOgYyFBU/nqRIq97-xe5XRZTxb...	10	UCBR8-60-B28hp2BmDPdntcQ	Music	True	
3 youtube#videoCategory	"m2yskbQFythfE4irbTieOgYyFBU/hwXkamMIQ20q9BN-o...	15	UCBR8-60-B28hp2BmDPdntcQ	Pets & Animals	True	
4 youtube#videoCategory	"m2yskbQFythfE4irbTieOgYyFBU/9GQMSRjrZdHeb10EM...	17	UCBR8-60-B28hp2BmDPdntcQ	Sports	True	

Veriseti, kategori bilgilerini ayrıca içeriyordu, iki verisetini birleştirmek istedim, ancak önce bu verisetini de okudum ve içeriği hakkında bilgi edinmek adına ilk 5 satırı ekrana yazdırdım.

We get only "id" and "snippet/title" columns.

►  ML

```
category_id = category_id.rename(columns={'id': 'category_id'})
category_id = category_id[['category_id', 'snippet/title']]
category_id.head(20)
```

	category_id	snippet/title
0	1	Film & Animation
1	2	Autos & Vehicles
2	10	Music
3	15	Pets & Animals
4	17	Sports
5	18	Short Movies
6	19	Travel & Events
7	20	Gaming
8	21	Videoblogging
9	22	People & Blogs
10	23	Comedy
11	24	Entertainment
12	25	News & Politics
13	26	Howto & Style
14	27	Education
15	28	Science & Technology
16	29	Nonprofits & Activism
17	30	Movies
18	31	Anime/Animation
19	32	Action/Adventure

İki verisetini bağlamak istediğim asıl konu, kategori isimleri ve id'leriydi. Dolayısıyla, ikinci verisetinden sadece bu iki sütunu çektim.

Then we merge two data frames. Now we have category for each id.

▶  MI

```
USvideos = pd.merge(videos, category_id, on='category_id')
USvideos
```

video_id	trending_date	title	channel_title	category_id	publish_time	tags	views	likes	dislikes	comment_count
0	2kyS6svSYSE	17.14.11	WE WANT TO TALK ABOUT OUR MARRIAGE	CaseyNeistat	22	2017-11-13T17:13:01.000Z	SHANTell martin	748374	57527	2966
1	0mlNzVSJrT0	17.14.11	Me-O Cats Commercial	Nobrand	22	2017-04-21T06:47:32.000Z	cute "cats" "thai" "eggs"	98966	2486	184
2	STI2fI7sKMo	17.14.11	AFFAIRS, EX BOYFRIENDS, \$18MILLION NET WORTH	Shawn Johnson East	22	2017-11-11T15:00:03.000Z	shawn johnson "andrew east" "shawn east" "shaw...	321053	4451	1772
3	KODzih-pVlU	17.14.11	BLIND(folded) CAKE DECORATING CONTEST (with Mo...	Grace Helbig	22	2017-11-11T18:08:04.000Z	itsgrace "funny" "comedy" "vlog" "grace" "helb...	197062	7250	217
4	8mhTWqWlQzU	17.14.11	Wearing Online Dollar Store Makeup For A Week	Safiya Nygaard	22	2017-11-11T01:19:33.000Z	wearing online dollar store makeup for a week ...	2744430	115426	1110
...
40944	V6ElE2xs48c	18.02.06	Game of Zones - S5:E5: The Isle of Van Gundy	Bleacher Report	43	2018-05-10T21:01:22.000Z	bleacher report "br" "nba" "Stan Van Gundy" "J...	1324482	22413	608
40945	V6ElE2xs48c	18.03.06	Game of Zones - S5:E5: The Isle of Van Gundy	Bleacher Report	43	2018-05-10T21:01:22.000Z	bleacher report "br" "nba" "Stan Van Gundy" "J...	1332252	22461	610
40946	V6ElE2xs48c	18.04.06	Game of Zones - S5:E5: The Isle of Van Gundy	Bleacher Report	43	2018-05-10T21:01:22.000Z	bleacher report "br" "nba" "Stan Van Gundy" "J...	1340039	22504	615
40947	V6ElE2xs48c	18.05.06	Game of Zones - S5:E5: The Isle of Van Gundy	Bleacher Report	43	2018-05-10T21:01:22.000Z	bleacher report "br" "nba" "Stan Van Gundy" "J...	1345086	22542	615
40948	V6ElE2xs48c	18.06.06	Game of Zones - S5:E5: The Isle of Van Gundy	Bleacher Report	43	2018-05-10T21:01:22.000Z	bleacher report "br" "nba" "Stan Van Gundy" "J...	1351321	22587	616

40949 rows x 17 columns

İki verisetini, kategori id'leri üzerinden eşleştirerek ilk dataframe üzerinde birleştirdim.

There are 40949 observations.

▶  MI

```
USvideos.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 40949 entries, 0 to 40948
Data columns (total 17 columns):
video_id          40949 non-null object
trending_date     40949 non-null object
title             40949 non-null object
channel_title     40949 non-null object
category_id       40949 non-null int64
publish_time      40949 non-null object
tags              40949 non-null object
views            40949 non-null int64
likes            40949 non-null int64
dislikes         40949 non-null int64
comment_count     40949 non-null int64
thumbnail_link    40949 non-null object
comments_disabled 40949 non-null bool
ratings_disabled  40949 non-null bool
video_error_or_removed 40949 non-null bool
description       40379 non-null object
snippet/title     40949 non-null object
dtypes: bool(3), int64(5), object(9)
memory usage: 4.8+ MB
```

Şimdi verisetim hangi başlıklara sahip, nasıl veriler içeriyor, kaç adet veri var bir gözlemlemek için veriseti bilgilerini bastırdım.

```
USVideos.describe()
```

	category_id	views	likes	dislikes	comment_count
count	40949.000000	4.094900e+04	4.094900e+04	4.094900e+04	4.094900e+04
mean	19.972429	2.360785e+06	7.426670e+04	3.711401e+03	8.446804e+03
std	7.568327	7.394114e+06	2.288853e+05	2.902971e+04	3.743049e+04
min	1.000000	5.490000e+02	0.000000e+00	0.000000e+00	0.000000e+00
25%	17.000000	2.423290e+05	5.424000e+03	2.020000e+02	6.140000e+02
50%	24.000000	6.818610e+05	1.809100e+04	6.310000e+02	1.856000e+03
75%	25.000000	1.823157e+06	5.541700e+04	1.938000e+03	5.755000e+03
max	43.000000	2.252119e+08	5.613827e+06	1.674420e+06	1.361580e+06

Daha önce de bahsettiğim gibi, python programlama dili, zengin kütüphane içeriğiyle, tam bir veri madenciliği aracı gibi görev görüyor. Burada, bu özelliklerinden faydalanarak, sadece tek satırda, verisetimin temel özelliklerini, yani ortalama, minimum, maksimum, çeyrekler, standart sapma gibi özellikleri, kullanabilip kullanamayacağımı, kullanabilirsem hangi sütunlar sayısal değerler içeriyor, hangileri için kullanabilirim öğrenmek için bastırdım.

```
We delete the columns which are unnecessary for the analysis.
```

```
USVideos.drop(['video_id', 'thumbnail_link', 'comments_disabled', 'ratings_disabled', 'video_error_or_removed'], axis=1, inplace=True)
```

```
USVideos.head()
```

	trending_date	title	channel_title	category_id	publish_time	tags	views	likes	dislikes	comment_count
0	17.14.11	WE WANT TO TALK ABOUT OUR MARRIAGE	CaseyNeistat	22	2017-11-13T17:13:01.000Z	SHANTell martin	748374	57527	2966	15954
1	17.14.11	Me-O Cats Commercial	Nobrand	22	2017-04-21T06:47:32.000Z	cute "cats" "thai" "eggs"	98966	2486	184	532
2	17.14.11	AFFAIRS, EX BOYFRIENDS, \$18MILLION NET WORTH	Shawn Johnson East	22	2017-11-11T15:00:03.000Z	shawn johnson "andrew east" "shawn east" "shaw...	321053	4451	1772	895
3	17.14.11	BLIND(folded) CAKE DECORATING CONTEST (with Mo...	Grace Helbig	22	2017-11-11T18:08:04.000Z	itsgrace "funny" "comedy" "vlog" "grace" "helb...	197062	7250	217	456
4	17.14.11	Wearing Online Dollar Store Makeup For A Week	Safiya Nygaard	22	2017-11-11T01:19:33.000Z	wearing online dollar store makeup for a week ...	2744430	115426	1110	6541

Verisetimde, zaten konumu ve amacımı kafamda belirlediğim için gerekli olmayacağını düşündüğüm veriler vardı, birkaç önçalışma sonrasında kullanmayacağımı kesinleştirdiklerim oldu, dolayısıyla bu sütunları, yani video başlığı, video kapak resminin bağlantısı, videonun yorumlara kapalı olup olmadığı bilgisi gibi verileri verisetimden (dataframe'den) çıkardım. Ve tekrar elimde neler kaldı şöyle bir görmek için ilk 5 satırı ekrana yazdırdım.

```
Any missing values?

In the description column.

> MI
USVideos.isnull().sum()

trending_date    0
title            0
channel_title    0
category_id      0
publish_time     0
tags            0
views           0
likes           0
dislikes        0
comment_count    0
description      570
snippet/title    0
dtype: int64
```

Eksik veya kayıp veri, veri madenciliğinde çok önemlidir, bunlar bulunmalı, ya kaldırılmalı, ya da diğer veriler baz alınarak doldurulmalı. Öncelikle eksik veya kayıp veri var mı diye, yine python'ın özelliği sayesinde, tek satırda baktım ve toplam sayıyı bastırdım.

```
We dropped the "description" column which contains 570 Nan values.

> MI
USVideos = USVideos.dropna(how='any',axis=0)
```

Burada, eksik verileri doldurmamın, benim verisetime herhangi bir katkısı olmayacağı için, ben bu boş veriye sahip satırları direk kaldırdım.

```
We can see number of unique values in each column.

> MI
USVideos.apply(lambda x: len(x.unique()))

trending_date    205
title            6357
channel_title    2142
category_id      16
publish_time     6172
tags            6008
views           39927
likes           29664
dislikes        8460
comment_count    13684
description      6901
snippet/title    16
dtype: int64
```

Verisetimde, her sütunda, kaç farklı unique değer var, görmek için bastırdım. Belki ileride bana bir fikir oluşturmamda katkı sağlayabilir diye düşündüm.

```

> MI
publish_time = pd.to_datetime(USvideos['publish_time'], format='%Y-%m-%dT%H:%M:%S.%fZ')
USvideos['publish_time'] = publish_time.dt.time
USvideos['publish_date'] = publish_time.dt.date
USvideos['publish_weekday']=publish_time.dt.weekday_name
> MI
USvideos

```

_time	tags	views	likes	dislikes	comment_count	description	snippet/title	publish_date	publish_weekday
13:01	SHANTEll martin	748374	57527	2966	15954	SHANTELL'S CHANNEL - https://www.youtube.com/s...	People & Blogs	2017-11-13	Monday
47:32	cute "cats" "thai" "eggs"	98966	2486	184	532	Kittens come out of the eggs in a Thai commerc...	People & Blogs	2017-04-21	Friday
00:03	shawn johnson "andrew east" "shawn east" "shaw...	321053	4451	1772	895	Subscribe for weekly videos ► http://bit.ly/sj...	People & Blogs	2017-11-11	Saturday
08:04	itsgrace "funny" "comedy" "vlog" "grace" "helb...	197062	7250	217	456	Molly is an god damn amazing human and she cha...	People & Blogs	2017-11-11	Saturday
19:33	wearing online dollar store makeup for a week ...	2744430	115426	1110	6541	I found this online dollar store called ShopMi...	People & Blogs	2017-11-11	Saturday
...

Youtube’da içerik üreticilerin düzenli video yüklemeleri, ve bunu belirli gün saatlerde aksatmadan yapmaları gerekir. Dolayısıyla videonun yüklenme gün ve saati, benim çalışmam için önem arz etmekte. Ancak verisetimde bu bilgi, bitişik yazmaktaydı. Ben de bunları, ayrı sütunlara bölerek, saat, tarih ve gün bilgisi olarak veri setime ekledim.

```

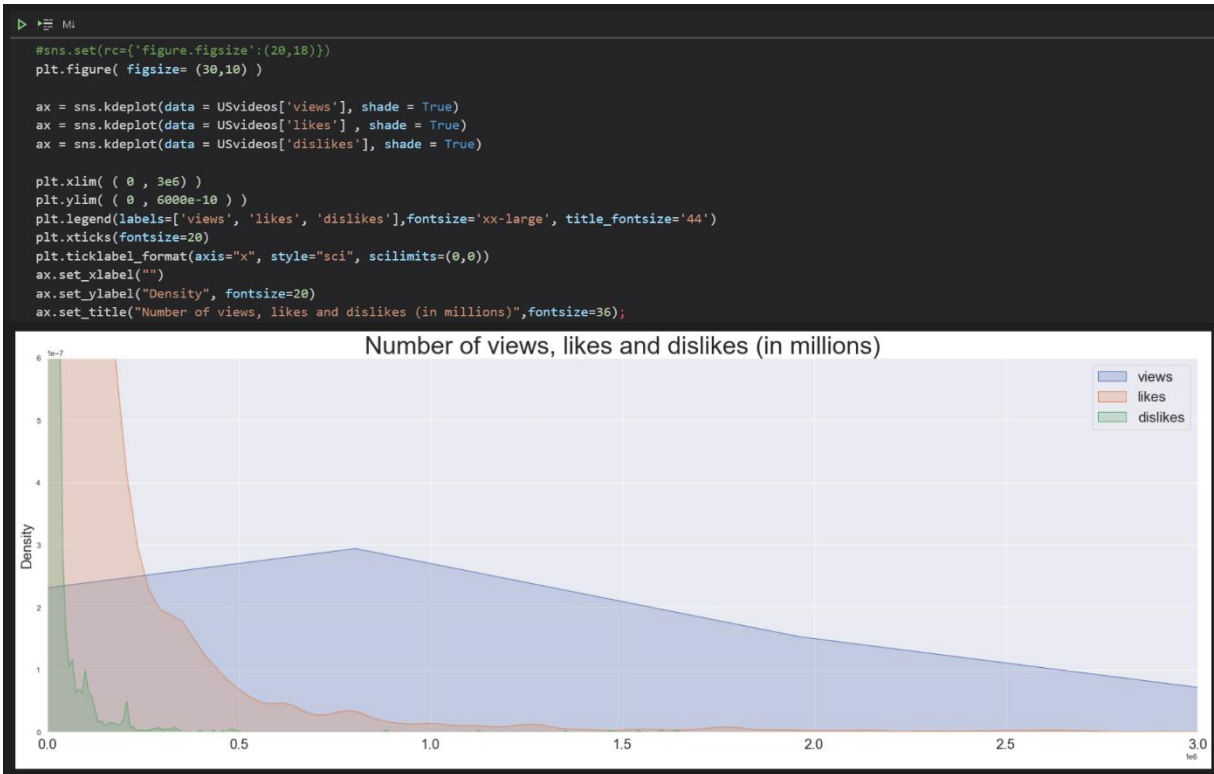
> MI
USvideos['publish_weekday'] = USvideos['publish_weekday'].replace({'Monday':1,
                                                                    'Tuesday':2,
                                                                    'Wednesday':3,
                                                                    'Thursday':4,
                                                                    'Friday':5,
                                                                    'Saturday':6,
                                                                    'Sunday':7})

```

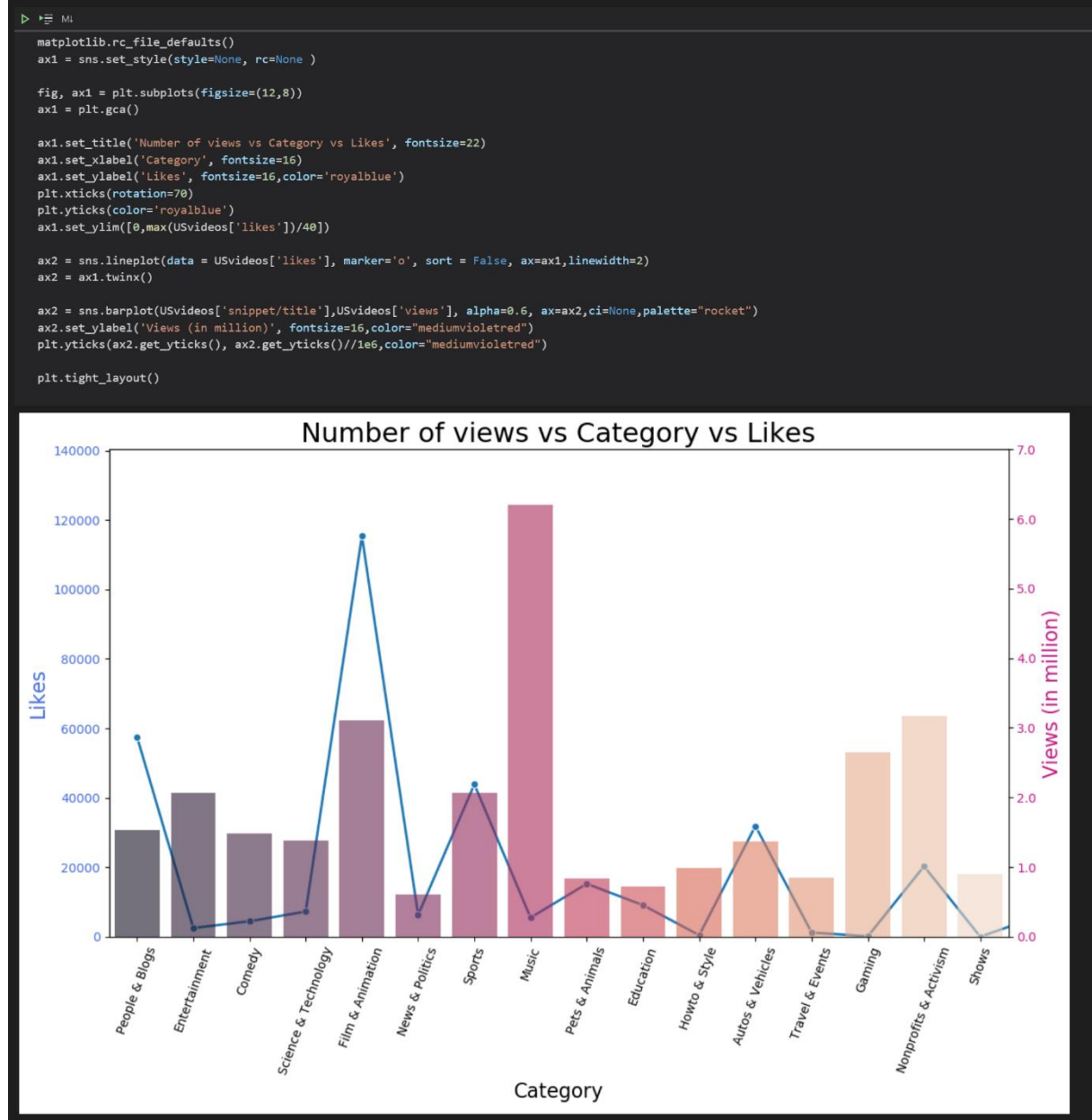
Böldüğüm fonksiyon, string değerler içeriyordu, ancak ben işlem yapabilmek adına bunları numaralandırdım ve verisetimi düzenledim.



Artık yavaş yavaş verisetimi işlemeye başladım, ilk olarak hem bir sonuç üretmek adına, hem merak ettiğim için, videoları yüklenme günlerine göre grafiğe döktüm. Sonuç açıkçası beni şaşırttı, ben Pazar günleri daha çok video yüklendiğini düşünürken, grafiğe göre hafta içi daha çok video yüklenmekte.



Tüm sosyal medya mecralarında olduğu gibi, Youtube’da da bir video için en ama en önemli şey, aldığı etkileşimdir. Youtube’da etkileşim “izlenme sayısı, beğenme, beğenmeme, ve yorum” şeklindedir. Burada ben, izlenme sayıları ile beğenme-beğenmeme sayılarını grafiğe döktüm.



Videoların kategorilerine göre, izlenme sayılarını ve beğenme sayılarını, grafiğe döktüm. Burada ben “People&Blogs” kategorisinin daha ön planda olduğunu düşünürken, “Film&Animation” kategorisi gözle görülür bir farkla daha ön planda. Ve yine bu grafik beni şaşırtan sonuçlar arasında.


```

matplotlib.rcParams.update({'font.size': 12})
ax1 = sns.set_style(style=None, rc=None)

fig, ax1 = plt.subplots(figsize=(12,8))
ax1 = plt.gca()

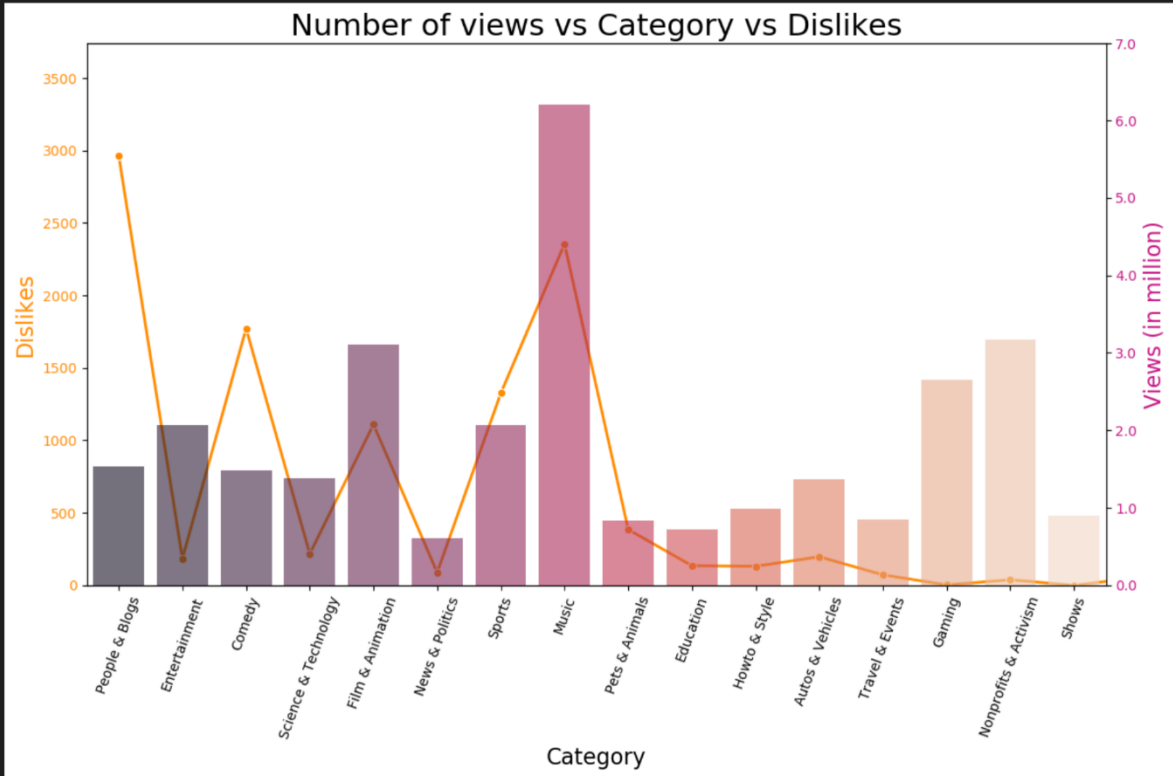
ax1.set_title('Number of Views vs Category vs Dislikes', fontsize=22)
ax1.set_xlabel('Category', fontsize=16)
ax1.set_ylabel('Dislikes', fontsize=16, color='darkorange')
plt.xticks(rotation=70)
plt.yticks(color='darkorange')
ax1.set_ylim([0, max(USvideos['likes'])/1500])

ax2 = sns.lineplot(data = USvideos['dislikes'], marker='o', sort = False, ax=ax1, linewidth=2, color='darkorange')
ax2 = ax1.twinx()

ax2 = sns.barplot(USvideos['snippet/title'], USvideos['views'], alpha=0.6, ax=ax2, ci=None, palette="rocket")
ax2.set_ylabel('Views (in million)', fontsize=16, color="mediumvioletred")
plt.yticks(ax2.get_yticks(), ax2.get_yticks()//1e6, color="mediumvioletred")

plt.tight_layout()

```



Aynı şekilde kategorilere göre, izlenme sayısı ile beraber bu kez beğenmeme sayılarını grafiğe döktüm. Tahminimin tam tersine, “People & Blogs” kategorisindeki videolar, izlenme sayısına oranla yoğun bir şekilde beğenmeme etkileşimi almış durumdadılar.

[illegible]

Correlation Matrix

	category_id	views	likes	dislikes	comment_count	publish_weekday
category_id	1	-0.17	-0.17	-0.047	-0.087	0.012
views	-0.17	1	0.85	0.56	0.66	0.041
likes	-0.17	0.85	1	0.51	0.85	0.049
dislikes	-0.047	0.56	0.51	1	0.62	0.022
comment_count	-0.087	0.66	0.85	0.62	1	0.038
publish_weekday	0.012	0.041	0.049	0.022	0.038	1

Bilindiği üzere, ilişki matrisi, veri madenciliğinde büyük yardımcımızdır. Eğer python'da bu özelliği verimli kullanabilirsek, verisetimiz de uygunsa, üzerinde araştırma yapabileceğimiz birbiriyle ilişkili özellikleri seçmemizde çok faydalıdır. Burada zaten belirlediğim 6 özellik üzerinde durdum, ve bunların birbirleriyle olan ilişkilerini inceledim. Korelasyon matrisinde, eğer bir şey birbiriyle tam ilişkiye sahipse, bunun değeri 1'dir, ve bu değeri sadece bir özellik kendisi ile karşı karşıya gelince alırız. Bunun dışında, değerin negatif olması, bize iki özellik arasında ters ilişki olduğunu, pozitif olması ise doğru orantı olduğunu gösterir. Ayrıca iki özellik arasında, mutlak 0.70 ve üzerinde bir ilişki görüyorsak, bu iki özellik birbiriyle ilişkilidir diyebiliriz. Burada biz izleme sayısı ile beğenme sayısı arasında 0.85 değerini, izleme sayısı ile beğenme sayısı arasında 0.56 değerini, izleme sayısı ile yorum sayısı arasında 0.66 değerini, yorum sayısı ile beğenme sayısı arasında 0.62 değerini okuyoruz. Bu şekilde birbirleriyle ilişkileri hakkında artık daha çok fikir sahibiyiz, ve doğru özellikler üzerinde çalıştığımızdan biraz daha emin olmuş durumdayız.

```
Removing non correlated features

> ML
USvideos.drop(['trending_date','publish_date','publish_time','tags','title','description','channel_title','snippet/title'],axis=1,inplace=True)
```

Dolayısıyla artık, geriye kalan gereksiz özellikleri de verisetimden çıkarıyorum, elimde sadece üzerinde çalışmaya devam edeceğim verileri bırakıyorum.

Machine Learning Algorithms

Artık, verim hakkında fikir sahibiyim, özelliklerimi seçtim, onlar hakkında ve birbirleriyle olan ilişkileri konusunda belirli bir görüşe vardım. Şimdi üzerinde algoritmaları kullanarak çalışacağım. Ben burada kullanmak üzere, “Lineer Regresyon” ve kıyaslamak için “Rastgele (Rassal) Orman” algoritmalarını seçtim, bunları seçmemdeki temel amaç, öncelikle araştırırken ve algoritmalara çalışırken, verisetime uygun olduklarını düşünmem oldu. Ayrıca, literatür taraması yaparken de, bu algoritmaların kullanıldığını gördüm, ve doğru ilerlediğimi düşünerek ikisi üzerine ilerledim.

Önemli Not : Algoritmalar kısmını, sürem yetecek olursa raporumda açıklayacağım ve yorumlayacağım, daha detaylı olduğu için sürem yetmeyecek olursa demoda zaten gösterileceği için, yalnızca kaynak kodlarımı ve aldığım çıktıları eklediğim hali ile bırakacağım.
Şeyda Nur DEMİR
(Son Düzenleme : 24 Ocak 2021, 21:30)

Prediction for views

Splitting the data into train (70%) and test(30%)

```
> % ML
views=USvideos['views']
USvideos_view=USvideos.drop(['views'],axis=1,inplace=False)

> % ML
X_train,X_test,y_train,y_test=train_test_split(USvideos_view,views, test_size=0.3,shuffle=False)

> % ML
print(X_train.shape,X_test.shape,y_train.shape,y_test.shape)
(28265, 5) (12114, 5) (28265,) (12114,)
```

We have 27697 data to train and 27697 data to test.

Linear Regression

Machine learning part

```
> % ML
reg = LinearRegression()
reg.fit(X_train,y_train)

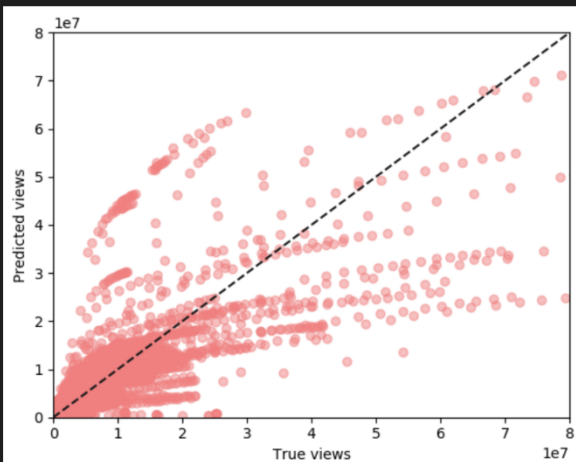
LinearRegression(copy_X=True, fit_intercept=True, n_jobs=None, normalize=False)
```

Prediction part

```
> % ML
y_pred = reg.predict(X_test)
print('Variance score: %.2f' % r2_score(y_test, y_pred)) # The higher the R-squared, the better the model fits your data
print('Root means score', np.sqrt(mean_squared_error(y_test, y_pred))) # It indicates the absolute fit of the model to the data-how close
#the observed data points are to the model's predicted values.
print("Result :",reg.score(X_test, y_test))
```

Variance score: 0.78
Root means score 5313690.482665997
Result : 0.780908966101816

```
> % ML
plt.scatter(y_test,y_pred,c='lightcoral',alpha=0.5)
plt.plot([0, 2e8], [0, 2e8], '--k')
plt.ylim([0,8e7])
plt.xlim([0,8e7])
plt.xlabel('True views')
plt.ylabel('Predicted views')
plt.show()
```



Random Forest

```
> % ML
n_estimators = [30,40,50,60]
depth = [3,4,5,6]
RF = RandomForestRegressor()
hyperParam = [{'n_estimators':n_estimators,'max_depth': depth}]
gsv = GridSearchCV(RF,hyperParam,cv=5,verbose=1,scoring='r2',n_jobs=-1)
gsv.fit(X_train, y_train)
print("Best HyperParameter: ",gsv.best_params_)
print(gsv.best_score_)
scores = gsv.cv_results_['mean_test_score'].reshape(len(n_estimators),len(depth))
```

```
Fitting 5 folds for each of 16 candidates, totalling 80 fits
[Parallel(n_jobs=-1)]: Using backend LokyBackend with 4 concurrent workers.
[Parallel(n_jobs=-1)]: Done 42 tasks | elapsed: 20.9s
[Parallel(n_jobs=-1)]: Done 80 out of 80 | elapsed: 42.8s finished
Best HyperParameter: {'max_depth': 4, 'n_estimators': 40}
0.6457130118961093
```

Then we apply the best hyperparameter

Machine learning part

```
> % ML
depth_best=gsv.best_params_['max_depth']
n_estimators_best=gsv.best_params_['n_estimators']
RF_best = RandomForestRegressor(n_estimators = n_estimators_best,max_depth=depth_best)
RF_best.fit(X_train, y_train)
```

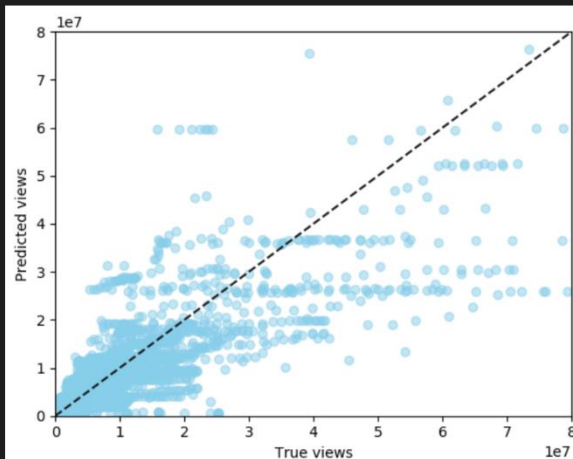
```
RandomForestRegressor(bootstrap=True, criterion='mse', max_depth=4,
                        max_features='auto', max_leaf_nodes=None,
                        min_impurity_decrease=0.0, min_impurity_split=None,
                        min_samples_leaf=1, min_samples_split=2,
                        min_weight_fraction_leaf=0.0, n_estimators=40,
                        n_jobs=None, oob_score=False, random_state=None,
                        verbose=0, warm_start=False)
```

Predicting part

```
> % ML
# predicting the test set results
y_pred = RF_best.predict(X_test)
print('Root means score', np.sqrt(mean_squared_error(y_test, y_pred)))
print('Variance score: %.2f' % r2_score(y_test, y_pred))
print("Result :",RF_best.score(X_test, y_test))
```


```
Root means score 5633891.776498567
Variance score: 0.75
Result : 0.7537086846905366
```


```
> % ML
plt.scatter(y_test,y_pred,c='skyblue',alpha=0.5)
plt.plot([0, 2e8], [0, 2e8], '--k')
plt.ylim([0,8e7])
plt.xlim([0,8e7])
plt.xlabel('True views')
plt.ylabel('Predicted views')
plt.show()
```



Prediction for likes


Splitting the data into train (70%) and test(30%)

```
>  ML
likes=USvideos['likes']
USvideos_likes=USvideos.drop(['likes'],axis=1,inplace=False)

>  ML
X_train2,X_test2,y_train2,y_test2=train_test_split(USvideos_likes,likes, test_size=0.3,shuffle=False)
```


Linear Regression

Machine learning part

```
>  ML
reg.fit(X_train2,y_train2)

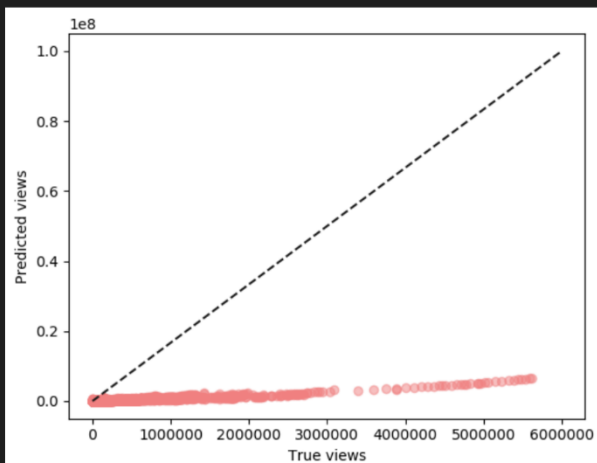
LinearRegression(copy_X=True, fit_intercept=True, n_jobs=None, normalize=False)
```

Prediction part

```
>  ML
y_pred2 = reg.predict(X_test2)
print('Variance score: %.2f' % r2_score(y_test2, y_pred2)) # The higher the R-squared, the better the model fits your data
print('Root means score', np.sqrt(mean_squared_error(y_test2, y_pred2))) # It indicates the absolute fit of the model to the data-how close
#the observed data points are to the model's predicted values.
print("Result :",reg.score(X_test2, y_test2))
```

Variance score: 0.91
Root means score 104190.61871930702
Result : 0.9119752818242721

```
>  ML
plt.scatter(y_test2,y_pred2,c='lightcoral',alpha=0.5)
plt.plot([0, 6e6], [0, 10e7], '--k')
plt.xlabel('True views')
plt.ylabel('Predicted views')
plt.show()
```



Random Forest

```
> % ML

n_estimators = [60,80,100,120]
depth = [3,4,5,6]
RF = RandomForestRegressor()
hyperParam = [{'n_estimators':n_estimators,'max_depth': depth}]
gsv = GridSearchCV(RF,hyperParam,cv=5,verbose=1,scoring='r2',n_jobs=-1)
gsv.fit(X_train2, y_train2)
print("Best HyperParameter: ",gsv.best_params_)
print(gsv.best_score_)
scores = gsv.cv_results_['mean_test_score'].reshape(len(n_estimators),len(depth))
```

Fitting 5 folds for each of 16 candidates, totalling 80 fits
[Parallel(n_jobs=-1)]: Using backend LokyBackend with 4 concurrent workers.
[Parallel(n_jobs=-1)]: Done 42 tasks | elapsed: 32.2s
[Parallel(n_jobs=-1)]: Done 80 out of 80 | elapsed: 1.2min finished
Best HyperParameter: {'max_depth': 5, 'n_estimators': 80}
0.7525291046281363

Then we apply the best hyperparameter

```
{}
```

Machine learning part

```
> % ML

depth_best=gsv.best_params_['max_depth']
n_estimators_best=gsv.best_params_['n_estimators']
RF_best = RandomForestRegressor(n_estimators = n_estimators_best,max_depth=depth_best)
RF_best.fit(X_train2, y_train2)

RandomForestRegressor(bootstrap=True, criterion='mse', max_depth=5,
                       max_features='auto', max_leaf_nodes=None,
                       min_impurity_decrease=0.0, min_impurity_split=None,
                       min_samples_leaf=1, min_samples_split=2,
                       min_weight_fraction_leaf=0.0, n_estimators=80,
                       n_jobs=None, oob_score=False, random_state=None,
                       verbose=0, warm_start=False)
```

Predicting part

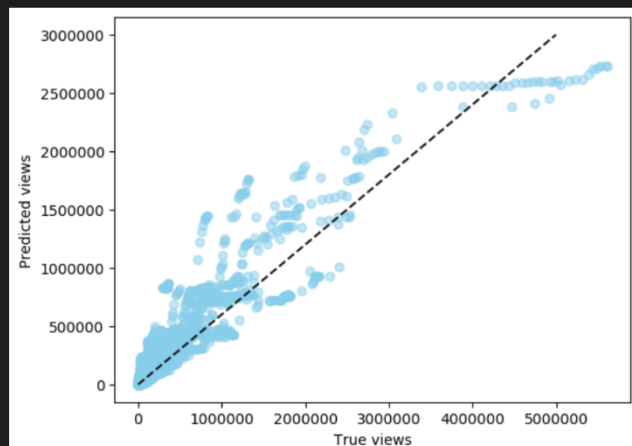
```
> % ML

# predicting the test set results
y_pred2 = RF_best.predict(X_test2)
print('Root means score', np.sqrt(mean_squared_error(y_test2, y_pred2)))
print('Variance score: %.2f' % r2_score(y_test2, y_pred2))
print("Result :",RF_best.score(X_test2, y_test2))
```

Root means score 155784.1158728512
Variance score: 0.80
Result : 0.8032141998627278

```
> % ML

plt.scatter(y_test2,y_pred2,c='skyblue',alpha=0.5)
plt.plot([0, 5e6], [0, 3e6], '--k')
plt.xlabel('True views')
plt.ylabel('Predicted views')
plt.show()
```



Prediction for number of comments

Splitting the data into train (70%) and test(30%)

```
> % ML
comments=USvideos['comment_count']
USvideos_comments=USvideos.drop(['comment_count'],axis=1,inplace=False)

> % ML
X_train3,X_test3,y_train3,y_test3=train_test_split(USvideos_comments,comments, test_size=0.3,shuffle=False)
```

Linear Regression

Machine learning part

```
> % ML
reg.fit(X_train3,y_train3)

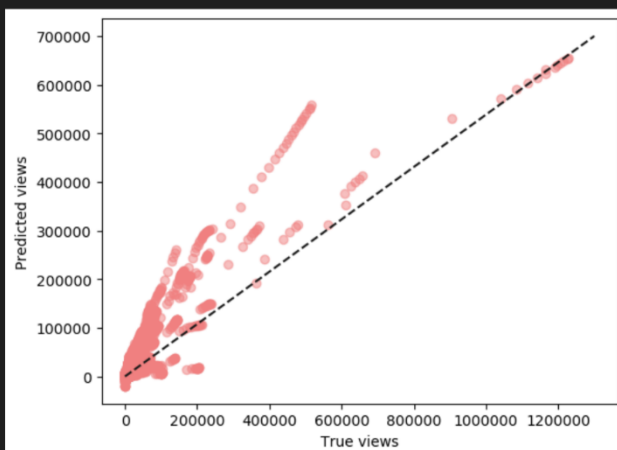
LinearRegression(copy_X=True, fit_intercept=True, n_jobs=None, normalize=False)
```

Prediction part


```
> % ML
y_pred3 = reg.predict(X_test3)
print('Variance score: %.2f' % r2_score(y_test3, y_pred3)) # The higher the R-squared, the better the model fits your data
print('Root means score', np.sqrt(mean_squared_error(y_test3, y_pred3))) # It indicates the absolute fit of the model to the data-how close
#the observed data points are to the model's predicted values.
print("Result :",reg.score(X_test3, y_test3))
```

Variance score: 0.80
Root means score 23067.33374075203
Result : 0.802691124715655

```
> % ML
plt.scatter(y_test3,y_pred3,c='lightcoral',alpha=0.5)
plt.plot([0, 13e5], [0, 7e5], '--k')
plt.xlabel('True views')
plt.ylabel('Predicted views')
plt.show()
```



Random Forest


```
>  ML

n_estimators = [60,80,100,120]
depth = [3,4,5,6]
RF = RandomForestRegressor()
hyperParam = [{'n_estimators':n_estimators,'max_depth': depth}]
gsv = GridSearchCV(RF,hyperParam,cv=5,verbose=1,scoring='r2',n_jobs=-1)
gsv.fit(X_train3, y_train3)
print("Best HyperParameter: ",gsv.best_params_)
print(gsv.best_score_)
scores = gsv.cv_results_['mean_test_score'].reshape(len(n_estimators),len(depth))
```

Fitting 5 folds for each of 16 candidates, totalling 80 fits
 [Parallel(n_jobs=-1)]: Using backend LokyBackend with 4 concurrent workers.
 [Parallel(n_jobs=-1)]: Done 42 tasks | elapsed: 35.8s
 [Parallel(n_jobs=-1)]: Done 80 out of 80 | elapsed: 1.4min finished
 Best HyperParameter: {'max_depth': 3, 'n_estimators': 80}
 0.5194875856762318

Then we apply the best hyperparameter


Machine learning part

```
>  ML

depth_best=gsv.best_params_['max_depth']
n_estimators_best=gsv.best_params_['n_estimators']
RF_best = RandomForestRegressor(n_estimators = n_estimators_best,max_depth=depth_best)
RF_best.fit(X_train3, y_train3)
```


```
RandomForestRegressor(bootstrap=True, criterion='mse', max_depth=3,
                        max_features='auto', max_leaf_nodes=None,
                        min_impurity_decrease=0.0, min_impurity_split=None,
                        min_samples_leaf=1, min_samples_split=2,
                        min_weight_fraction_leaf=0.0, n_estimators=80,
                        n_jobs=None, oob_score=False, random_state=None,
                        verbose=0, warm_start=False)
```

Predicting part

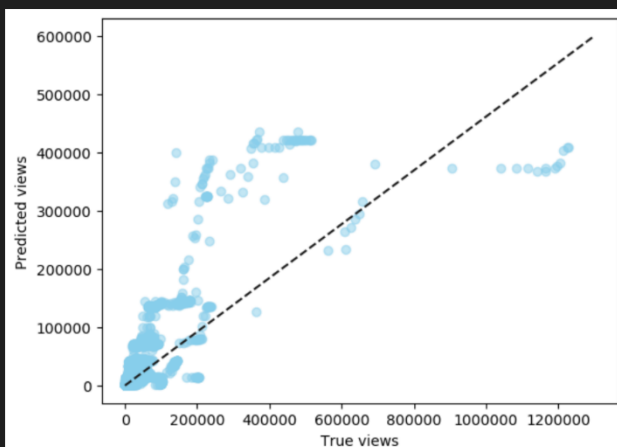
```
>  ML

# predicting the test set results
y_pred3 = RF_best.predict(X_test3)
print('Root means score', np.sqrt(mean_squared_error(y_test3, y_pred3)))
print('Variance score: %.2f' % r2_score(y_test3, y_pred3))
print("Result :",RF_best.score(X_test3, y_test3))
```

Root means score 30899.963882460474
 Variance score: 0.65
 Result : 0.645947417601674

```
>  ML

plt.scatter(y_test3,y_pred3,c='skyblue',alpha=0.5)
plt.plot([0, 13e5], [0, 6e5], '--k')
plt.xlabel('True views')
plt.ylabel('Predicted views')
plt.show()
```



End of the project.

▶ ML

```
print("Seyda Nur DEMIR, 12 10 44 042")
```

Seyda Nur DEMIR, 12 10 44 042

3.ELDE EDİLEN SONUÇLAR

Projenin Amacı ve Bu Doğrultuda Yaptıklarım

Projenin amacı, daha önce de belirttiğim gibi, izlenme sayıları ve bunu etkileyen değişkenler var mıdır, varsa bu faktörler nelerdir, bu konular hakkında bir fikir sahibi olmak, belirli bir görüş kazanmaktır. Bu doğrultuda, yaptığım tüm araştırmalar üzerine çalıştım, ve bir veriseti üzerinde kendi yöntemlerimle sonuca ulaşmaya çalıştım.

Projemde İlerlerken Karşılaştığım Sorunlar

Projemde ilerlerken karşılaştığım en büyük sorun, python yüklemeleri, kütüphaneler ve sürümlerin çakışması idi. Birçok defa bu sorunları çözmekle zaman kaybettim. Ayrıca, Türkçe kaynak bulamamak da karşılaştığım sorunlar arasında sayabileceğim önemli bir eksiklik.

Elde Ettiğim Sonuçlar

Ben bu proje sonunda, her şeyden önce, veri madenciliği konsepti hakkında geniş bir bilgi birikimine sahip oldum. Artık, bir yöntemi bilsem de bilmesem de, önüme çıkan probleme, kazandığım bakış açısı sayesinde, kısa sürede bir çözüm önerebilirim. Kısa bir araştırma, benim için bir konuda çalışmama artık yetecektir.

Ulaştığım Amaçlar

Kişisel bu kazanımlarım dışında, projemde amaçlayıp ulaştığım şeyleri, şu şekilde sıralayabilirim :

- Youtube videolarının kategorileri, izlenme sayılarını önemli ölçüde etkileyebilir.
- Youtube videolarının kategorileri, beğenme ve beğenmeme sayılarını önemli ölçüde etkileyebilir.
- Youtube videolarının izlenme sayısı arttıkça, aldığı beğenme ve beğenmeme sayıları da artmaktadır.
- Youtube’a, haftaiçi yüklenen video sayısı, haftasonuna göre çok daha fazladır.
- Youtube’da en çok izlenen içerik kategorisi, “Music” (Müzik) kategorisidir.
- Youtube’da en az izlenen içerik kategorisi, “News&Politics” (Haberler ve Politika) kategorisidir.
- Youtube’da en çok kullanılan kelime etiketleri, “music”, “makeup tutorial”, “star wars”, “funny video”, “super bowl”, “talk show”, “will smith”tir.
- Youtube’da “kim kardashian”, “galaxy s9”, “iphone x”, “full face”, “family friendly”, az kullanılan kelime etiketleri olarak sayılabilir.
- Bir Youtube içeriğinin beğenme sayısına, yorum sayısına, ve kategorisine bakarak, izlenme sayısı hakkında tahmin yürütebiliriz.

4.TARTIřMA

Proje Hakkında Genel Deęerlendirme

Proje sonunda, veri madencilięi, veri seti, ve konum sayesinde Youtube video ierikleri ve izlenme sayıları hakkında, geniř bir grř aısı kazandım.

Gelecek alıřma

Bu alıřma zerine, daha ok veri ieren veri setleri ile aynı alıřma yapılabilir, farklı algoritmalar kullanılarak aynı zellikler hakkında sonulara ulařılabilir, veya farklı zellikler seilerek farklı grřler elde edilebilir. Benim setięim veriseti, algoritma ve zellikler kullanılarak, daha doęru verilerle daha doęru sonulara da ulařılabilir. Ayrıca, tm bunların dıřında, belirli konularda belirli video ierik analizleri yapılabilir.

Projemin Bana Kattıkları

Projenin bana kattıęı en gzel řey, bir veri bilimi hakkında daha bilgi sahibi olmak, ve yetkinlik kazanmaktır. Bana bu imkanı tanıdıęı iin, tekrar Burcu YILMAZ hocamıza teřekkr bor bilirim.

KAYNAKLAR

1. <https://www.saedsayad.com/>
2. <http://nek.istanbul.edu.tr:4444/ekos/TEZ/45671.pdf>
3. <https://medium.com/@furkanalaybeg/veri-madencili%C4%9Fi-ve-y%C3%B6ntemleri-d0e2fd238e44>
4. <https://medium.com/veri-madencili%C4%9Fi/s%C4%B1n%C4%B1fland%C4%B1rma-regresyon-k%C3%BCmeleme-ve-birlikteilik-kurallar%C4%B1-e8ee1e47aeed>
5. <https://vizyonergenc.com/icerik/5-temel-soruda-veri-madenciligi-data-mining-nedir>
6. https://www.researchgate.net/publication/286048668_Veri_Madenciligi_Kumeleme_ve_Siniflama_Algoritmaları
7. <https://towardsdatascience.com/applying-the-universal-machine-learning-workflow-to-the-uci-mushroom-dataset-1939442d44e7>
8. <https://ieeexplore.ieee.org/abstract/document/8258541>
9. http://sbp-brims.org/2018/proceedings/papers/latebreaking_papers/LB_14.pdf
10. <https://slideplayer.biz.tr/slide/9467037/>
11. http://ybsansiklopedi.com/wp-content/uploads/2015/09/zaman_serileri.pdf
12. https://erdincuzun.com/makine_ogrenmesi/hangisini-secmeliyim-supervised-ve-unsupervised-learning/
13. <http://www.rcl.ece.iastate.edu/sites/default/files/papers/KriZam13A.pdf>
14. <https://ojs.aaai.org/index.php/ICWSM/article/view/14607>
15. <https://dl.acm.org/doi/abs/10.1145/2433396.2433443>
16. http://dataminingcasestudies.com/DMCS_WorkshopProceedings25.pdf
17. <https://www.youtube.com/watch?v=iButBaRPpBM&list=PLh9ECzBB8tJNScCBWJFoMdpMkCdpmwUEI>
18. <https://www.kaggle.com/datasnaek/youtube-new?select=USvideos.csv>

END OF THE REPORT

UPDATED : 24.01.2021 21:30

STUDENT

**Şeyda Nur DEMİR
12 10 44 042**

LECTURER

Burcu YILMAZ

TEACHING ASSISTANT

-

KOCAELİ, 2020