

T.C.
BEYKENT ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ
BİLGİSAYAR MÜHENDİSLİĞİ ANABİLİM DALI
BİLGİSAYAR MÜHENDİSLİĞİ BİLİM DALI

**Kişilerin Konum Paylaşımlarını Kullanarak
Gelecekteki Konum Bilgilerinin Tahmini**
YÜKSEK LİSANS TEZİ

TEZİ HAZIRLAYAN:
ÖZGÜR KAPLAN

İstanbul, 2016

T.C.
BEYKENT ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ
BİLGİSAYAR MÜHENDİSLİĞİ ANABİLİM DALI
BİLGİSAYAR MÜHENDİSLİĞİ BİLİM DALI

**Kişilerin Konum Paylaşımlarını Kullanarak
Gelecekteki Konum Bilgilerinin Tahmini**

YÜKSEK LİSANS TEZİ

TEZİ HAZIRLAYAN:

ÖZGÜR KAPLAN

ÖRENCİ NO:

130820019

DANIŞMAN:

YARDIMCI DOÇENT DOKTOR EDİZ ŞAYKOL

İstanbul, 2016

YEMİN METNİ

Yüksek lisans tezi olarak sunduğum “Kişilerin Foursquare üzerindeki konum paylaşımlarını kullanarak, veri madenciliği yöntemleri ile kişilerin gelecekteki konum bilgilerinin tahmin edilmesi.” başlıklı bu çalışmanın, bilimsel ahlak ve geleneklere uygun şekilde tarafımdan yazıldığını, yararlandığım eserlerin tamamının kaynaklarda gösterildiğini ve çalışmamın içinde kullandıkları her yerde bunlara atıf yapıldığını belirtir ve bunu onurumla doğrularım. 23/12/2017


Özgür Kaplan

Adı ve Soyadı : Özgür Kaplan
Danışmanı : Yardımcı Doçent Doktor Ediz Şaykol
Türü ve Tarihi : Yüksek Lisans, 2016
Alanı : Bilgisayar Mühendisliği

T.C.
BEYKENT ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

YÜKSEK LİSANS TEZ SAVUNMA SINAVI SONUÇ TUTANAĞI

Beykent Üniversitesi
Fen Bilimleri Enstitüsü Müdürlüğü'ne,

Aşağıda tez adı belirtilen yüksek lisans öğrencisi.....no'lu.....'in 19/1/17 tarihinde yapılan tez savunma sınavı¹ sonucunda 45 dakika süreyle sunduğu ve savunduğu tezi hakkında² oybirliğiyle, KABUL kararı verilmiştir.

Bilgilerinize saygılarımızla arz ederiz.

Anabilim Dalı :
Programı :
Tez Başlığı³ :

Tez Sınav Jürisi

Öğretim Üyesi

Danışman

: Yrd. Doç. Dr. Ediz SAĞLIK

Üye

: Doç. Dr. Gökhan SİLİHTANIOĞLU

Üye

: Yrd. Doç. Dr. Turhan Kargılcı

İmza



¹ Jüri üyeleri söz konusu tezin kendilerine teslim edildiği tarihten itibaren en geç bir ay içinde toplanarak öğrenciyi tez savunma sınavına alır. Belirlenen günde yapılamayan jüri toplantısı, katılanların hazırladığı bir tutanakla enstitü yönetimine bildirilir. Bu durumda jüri en geç onbeş gün içinde toplanarak adayı tez savunma sınavına alır. Tez savunma sınav süresi en az 45 dakikadır. Yüksek lisans tez savunma sınavı, tez çalışmasının sunulması ve bunu izleyen soru-yanıt bölümlerinden oluşur ve dinleyiciye açıktır. (Beykent Lisansüstü eğitim ve Öğretim Yönetmeliği-Madde30-3)

² Tez sınavının tamamlanmasından sonra jüri, tez hakkında "kabul", "düzeltme" veya "red" kararı verir. Jüri başkanı, jüri üyelerince imzalanmış sınav tutanağını, tez sınavını izleyen üç gün içinde ilgili enstitü yönetimine teslim eder. Tezi başarısız bulunan öğrencinin Enstitü ile ilişkisi kesilir. Tezi hakkında düzeltme kararı verilen öğrenci en geç üç ay içinde gerekli düzeltmeleri yaparak ve yönetmelikte belirtilen usullere uygun olarak tezini aynı jüri önünde yeniden savunur. Bu savunma sınavında da tezi kabul edilmeyen öğrencinin enstitü ile ilişkisi kesilir. (Beykent Lisansüstü eğitim ve Öğretim Yönetmeliği-Madde30-4)

³ İleridedoğabilecek aksaklıkların engellenmesi için tezin başlığını yazılması gerekmektedir.

ÖZET

Kişilerin Konum Paylaşımını Kullanarak Gelecekteki Konum Bilgilerinin Tahmini

Son yıllarda çok hızlı bir şekilde artan akıllı telefon kullanımı ve bu telefonların sağladığı konum servislerini kullanan, Foursquare gibi uygulamaların popülerleşmesi, kişilerin hareketlilik alışkanlıkları hakkında detaylı bilgiler elde edilebilmesini sağlamıştır. Bu çalışmada kişilerin Foursquare üzerindeki geçmiş konum paylaşımları kullanılarak, kişilerin ziyaret edebilecekleri mekânların tahmin edilmesi hedeflenmiştir. Çalışmanın adli araştırmalara, işletme popülerlik ve kişilik analizi çalışmalarına katkı sağlayacağı öngörülmüştür. Çalışma kapsamında kişilerin Twitter üzerinden paylaştıkları ve Foursquare üzerinden gerçekleştirdikleri mekân paylaşımları analiz edilmiştir. Paylaşımlarda ziyaret edildiği belirtilen mekânların kategorilere Foursquare üzerinden alınmıştır. Toplanan mekân verileri, karar ağacı ve gradyan iyileştirmeleri karar ağacı öğrenme teknikleri kullanılarak analiz edilmiş ve ziyaret edilen mekânların kategorileri tahmin edilerek elde edilen bulgular incelenmiştir.

Anahtar Kelimeler: Konum, Sosyal Medya, Foursquare, Veri madenciliği

ABSTRACT

Forecasting people's future location information using data mining methods

The increasing popularity of smartphones over the last few years and the popularity of applications such as FourSquare, which use location services provided by these phones, have enabled us to obtain detailed information about people's mobility habits. In this study, it was aimed to estimate locations where people could visit using past location shares on FourSquare. It is envisaged that the study will contribute to the study of criminal investigations, enterprise popularity and personality analysis. Within the scope of the study, people's tweets about their FourSquare check-ins were analyzed. The categories of places that were visited on the Shares were taken from Foursquare. The collected spatial data were analyzed using decision tree and gradient boosted trees learner techniques and the findings obtained by estimating the categories of visited places were examined.

Key Words : Location, Social Media, Foursquare, Data mining

İÇİNDEKİLER

ÖZET	i
ABSTRACT	ii
İÇİNDEKİLER.....	iii
TABLolar LİSTESİ	v
ŞEKİLLER LİSTESİ.....	vii
KISALTMALAR	viii
1. GİRİŞ	1
2. BENZER ÇALIŞMALAR	4
3. KAVRAMLAR.....	7
3.1. Veri Madenciliği.....	7
3.2. Twitter	7
3.2. Foursquare	8
3.3. Uygulama programlama ara yüzü (API)	11
3.4. Twitter API.....	11
3.5. Foursquare API.....	12
3.6. RESTfull.....	12
3.7. OAuth	13
3.8. JavaScript Nesne Notasyonu (JSON)	14
3.9. Linq2Twitter	15
3.10. NoSQL.....	16
3.11. Azure	17
3.11. DocumentDB	17
3.12. Knime	18
3.13. Karar Ağacı.....	19
3.14. CSharp (C#) Programlama Dili	19

3.15. Gradyan İyileřtirmesi	20
4. YÖNTEM.....	21
4.1 Verilerin Toplanması	21
4.2 Verilerin Hazırlanması	31
4.3 Verilerin Filtrelenmesi.....	34
4.4 Verilerin Knime Programı ile Analiz Edilmesi	38
5. BULGULAR.....	47
6. SONUÇ	49
KAYNAKLAR.....	50
ÖZGEÇMİŐ	52

TABLÖLAR LİSTESİ

Tablo.1. Check-in Datası.....	29
Tablo.2. Kullanıcı Datası	30
Tablo.3. Mekân Datası	31
Tablo.4. Takipçi Sayısı Grupları	33
Tablo.5. Takip Edilen Sayısı Grupları	33
Tablo.6. Takip Edilen Sayısı Grupları	34



ŞEKİLLER LİSTESİ

Şekil.1. Tivit.....	8
Şekil.2. Foursquare Uygulaması	10
Şekil.3. API Akış Şeması	11
Şekil.4. RestFull Servis Akış Şeması.....	13
Şekil.5. JSON Formatı	15
Şekil.7. Azure Kontrol Paneli	17
Şekil.8. Knime Programı	18
Şekil.9. Karar Ağacı	19
Şekil.10. C# İle Uygulama Örneği	20
Şekil.11. Foursquare Konum Tahmini Programı Akışı.....	21
Şekil.12. Tivit Çeken Kod Parçağı.....	23
Şekil.13. Foursquare Check-In Anahtarının Bulunması	24
Şekil.14. Checkin Nesnesinin Doldurulması.....	25
Şekil.15. Kullanıcı bilgilerinin Alınması	26
Şekil.16. Mekân Bilgilerinin Toplanması	27
Şekil.17. Azure Periyodik İş Tanımı	28
Şekil.18. Numeric Binner Nesnesi Ayarları.....	32
Şekil.19. Knime Group By Nesnesi Ayarları	35
Şekil.20. Knime RowFilter Nesnesi Ayarları.....	36
Şekil.21. Knime Column Filter Nesnesi Ayarları	37
Şekil.22. Analiz Programı Akışı	38
Şekil.23. CSV Formatına Çevrilmiş Veriler	39
Şekil.24. Knime File Reader Nesnesi Ayarları	40
Şekil.25. Knime Partitioning Nesnesi Ayarları	41
Şekil.26. Decision Tree Learner Nesnesi Ayarları.....	42
Şekil.27. Knime Decision Tree Predictor Nesnesi Ayarları.....	43
Şekil.28. Gradient Boosted Decision Tree Learner.....	44
Şekil.29. Gradient Boosted Decision Tree Predictor	45
Şekil.30. Knime Scorer Nesnesi Ayarları	45

KISALTMALAR

API: Uygulama Programlama Ara yüzü (Application programming interface)

Check-in: Foursquare uygulamasını kullanarak bulunulan mekânın takipçiler ile paylaşılması.

Tivit: Twitter, kullanıcıların paylaştıkları 140 karakterlik mesaj.

JSON: JavaScript obje notasyonu



1. GİRİŞ

Son yıllarda çok hızlı bir şekilde artan akıllı telefon kullanımı ve bu telefonların sağladığı konum servislerini kullanan, Foursquare gibi uygulamaların popülerleşmesi, kişilerin hareketlilik alışkanlıkları hakkında detaylı bilgiler elde edilebilmesini sağlamıştır.

Günümüzde her saniye binlerce uygulama kullanıcısı gittikleri mekânlara check-in yaparak bu bilgiyi anlık olarak aile, arkadaş ve takipçilerinin görebileceği şekilde paylaşmaktadırlar. Benzer şekilde pek çok işletme, mekânlarına check-in yapan kullanıcılara ücretsiz ikramlar ve indirimler sağlamaktadırlar.

Başlangıçta oldukça faydalı görünen mekân check-in'leri, dikkatli kullanılmadığı durumlarda tehlikeli sonuçlar doğurabilmektedirler.

Tam konumun aile ve arkadaşlarla paylaşılması önemsenmeyebilir olsa da bu bilgilerin tanınmayan insanların eline geçmesi riskli olabilir. Konumun Facebook ve Twitter gibi sosyal ağlarda yayınlandığı durumlarda bilgilerin kontrolü kaybolur. Herkes bu bilgilere erişebilir. Örneğin tatildeki bir kişi bir otele check-in yaparak paylaşabilir. Bir hırsız, bu bilgiyi görebilir. Bu kişinin ev adresi için çevrimiçi bir arama yaparak adrese ulaşır ve bu evi soyabilir.

İşletmeler, uygulamalar ve reklam sağlayıcıları ise kişilerin check-in verilerini kullanarak çok değerli bilgilere ulaşabilmektedirler. Örneğin bir işletme, kendisini ziyaret eden kullanıcıların bilgilerini analiz ederek ziyaretçi kitlesine en uygun hizmetleri sunabilirler. Benzer şekilde uygulamalar kişilerin ziyaret ettikleri mekânlara en uygun olan aktiviteleri kişilere sunabilirler.

Çalışmanın kayıp veya bilinmeyen kişilerin tespit edilmesine katkıda bulunacağı öngörülmüştür. Benzer şekilde kitlelerin hareketlilik analizleri ve kişiye özel, reklam ve tavsiye çalışmalarına da katkıda bulunabileceği öngörülmüştür.

Çalışmanın ilk aşamasında kişilerin, Foursquare API'si üzerinde yaptığı check-in bilgileri toplanmıştır. Toplanan büyük miktarda verinin saklanıp, çok hızlı bir şekilde işlenebilmesi için bir NoSQL veri tabanı olan DocumentDB kullanılmıştır.

Foursquare arkadaş olunmayan kişilerin check-in verisine üçüncü şahıslar tarafından erişilmesine izin vermemektedir. Ancak kişi check-in bilgisini Twitter üzerinden yayınlarsa bu kısıt ortadan kalkmaktadır. Twitter üzerinde yayınlanan Foursquare mekân paylaşımlarında bu paylaşım ile ilgili olan benzersiz tanımlayıcı numara bulunmaktadır. Bu benzersiz tanımlayıcı kullanılarak Foursquare API'si üzerinden ilgili paylaşım ile ilgili konum, mekân adı, mekân kategorisi gibi detay bilgiye ulaşmak mümkündür.

Twitter API'si ve Foursquare API'sini kullanan ve kullanıcı check-in bilgilerini periyodik olarak toplayan program geliştirilmiştir. Bu program Microsoft firmasının bulut bilişim çözümü olan Azure üzerinde periyodik zamanla çalışması sağlanmıştır. Bu program kullanılarak 3151449 adet check-in kaydedilmiştir. Kaydedilen veri içerisinde, kullanıcının isim, soy isim, cinsiyet, yaşadığı şehir, yaşadığı ülke, Twitter'da takip ettiği kişi sayısı, Twitter'da bu kişiyi takip eden kişi sayısı, kişinin tivit sayısı, gittiği mekânın adı, mekân kategorisi, kullanıldığı telefon tipi, paylaşım tarihi, paylaşılan mekânın konumu, mekânın şehir ve ülke bilgisi toplanmıştır.

Toplanan veri içerisinde 726 adet kategori bulunmaktadır. Veri madenciliği tekniklerinin daha verimli uygulayabilmek için kategorilerden 10000'in üzerinde check-in yapılan 70 adet kategorinin kullanıldığı check-in bilgileri filtrelenmiş ve geriye kalan 2266834 adet check-in bilgisi kullanılmıştır.

Elde edilen veri kümesi veri analizi programlarından Knime kullanılarak 80'e 20 oranında öğrenme ve tahmin gruplarına ayrılmıştır. Öğrenme grubu karar ağacı ve gradyan iyileştirmeli karar ağacı yöntemleri kullanılarak analiz edilmiştir. Analiz sonuçları kullanılarak kalan %20'lik kısım tahmin edilmiştir. Tahminler sonucunda karar ağacı yöntemiyle %38,6'lık, gradyan iyileştirmeli karar ağacı ile ise %40,4'lük bir başarıya ulaşılmıştır.

Çalışmanın ikinci bölümünde benzer çalışmalar incelenip, mevcut çalışma ile karşılaştırmaları yapılmıştır. Üçüncü bölümde kullanılan araç ve kavramlar detaylı bir şekilde anlatılmıştır. Çalışmanın dördüncü bölümünde çalışma yöntem ve işleyişi anlatılmıştır. Çalışma sonucunda elde edilen bulgular dördüncü bölümde karşılaştırmalı olarak incelenip yorumlanmıştır. Sonuç ve gelecekte yapılması gerekenler ise beşinci bölümde incelenmiştir.



2. BENZER ÇALIŞMALAR

Bu bölümde Foursquare ve konum tahmini konusunda yapılmış benzer çalışmalar incelenmiştir.

2.1. You are where you eat: Foursquare checkins as indicators of human mobility and behaviour

G.B. Colombo, M.J. Chorley, M.J. Williams, S.M. Allen, R.M. Whitaker tarafından, Cardiff Üniversitesi Bilgisayar Bilimleri ve Enformatik bölümünde yapılan çalışmada, Foursquare kullanıcıların yaptıkları check-in bilgileri analiz edilmiştir. Kullanıcı gruplarının olası ortak davranışları ve bireylerin sistemi kullanma şekillerindeki farklılıkların tespit edilmesi amaçlanmıştır[1].

Çalışma Cardiff ve Cambridge şehirlerindeki check-in bilgileri incelemiştir. Kullanıcının check-in'leri arasındaki mesafe ve zaman farkları, check-in'ler arasındaki olası ilişkiler ve belirli bir süre içerisinde birden fazla yapılan check-in'ler tespit edilmeye çalışılmıştır.

Çalışma sonucunda kullanıcıların önemli bir yüzdesinin sınırlı sayıda mekân düzenli ve sık ziyaret ettikleri gibi, sosyal arkadaşlarının uğradığı yerleri sıklıkla ziyaret etme eğiliminde olduklarını gözlemlenmiştir.

Bu çalışmadan farklı olarak Cardiff Üniversitesi'nin çalışması veri madenciliği tekniklerinden ziyade istatistik ve veri analizine dayanmaktadır. Cardiff üniversitesinin çalışması kullanıcının check-in davranışlarını analiz etmeyi hedeflerken bu çalışma kullanıcıların gideceği mekânların kategorilerinin tahmin edilmesi amaçlanmıştır.

2.2. Understanding Venue Popularity in Foursquare

Xuelian Long, Lei Jin and James Joshi tarafından Pittsburgh Üniversitesi'nde yapılan çalışmada, mekân popürlüğünü etkileyebilecek bazı özellikler incelenmiştir. Pittsburgh bölgesinde hangi mekânların popüler olduğu ve bunları popüler kılan öğeler araştırılmıştır[2].

Çalışmada mekân popürlüğü ve kategori popürlüğü incelenmiş, genel mekân popürlüğünü artırmaya yardımcı olabilecek özellikler araştırılmıştır. Bu işlem için, bir süre boyunca yapılan check-in'ler, bir süre boyunca check-in yapan yeni kullanıcı sayısı, bir dönemdeki ortalama günlük check-in sayısı ve bir dönemdeki ortalama günlük check-in yapan yeni kullanıcı sayısı metrikleri kullanılmıştır.

Araştırma sonucunda üç farklı özelliğın mekân popürlüğüne etki ettiğı tespit edilmiştir. Bu üç madde promosyonlar, mekânın internette içeriklerinin olması ve menü zenginliğı olarak tespit edilmiştir.

Bu çalışmadan farklı olarak, Cambridge üniversitesinin çalışmasında mekân popürlüğü ve bu popürlüğe etkileyen etmenleri incelenmiştir.

2.3. Visiting Patterns and Personality of Foursquare Users

Martin J. Chorley, Gualtiero. B. Colombo, Stuart. M. Allen, Roger. M. Whitaker tarafından Cardiff Üniversitesinde yapılan çalışmada, Foursquare kullanıcılarının ziyaret ettiğı yerlerin türleri ile bu kişilerin kişilikleri arasındaki ilişki incelenmiştir[3].

Çalışma, kullanıcı konum verilerini, kullanıcının beş faktörlü kişilik profili ile birleştiren bir uygulamanın sonuçlarını sunmuştur. Bu beş kişilik faktörü; açıklık, vicdan, dışa dönüklük, katılğanlık ve nevroतिकlıktır. Aynı zamanda çalışma check-in t kalıpları ve kişilik arasındaki önceden bilinmeyen ilişkileri keşfetmeyi, insan

davranışını anlamaya yardımcı olan ve gelecekteki uygulamalarda kişiselleştirmeyi geliştirmeye yarayan göstergeler sağlamayı hedeflemiştir.

Çalışma sonucunda açıklık ile ziyaret edilen mekânlar arasında pozitif bir ilişki bulunmuştur. Nevrotiklik ile ziyaret edilen mekân sayısı arasında ise negatif bir ilişki bulunmuştur.

Bu çalışmada kullanıcının gelecekte bulunabileceği mekân kategorisi tahmin edilmeye çalışılırken, Cardiff Üniversitesinin çalışmasında kişilik profili ile mekân paylaşımları arasındaki ilişki incelenmiştir.

2.4. Beyond Sights: Large Scale Study of Tourists' Behavior Using Foursquare Data

Ana Paula G. Ferreira, Thiago H. Silva ve Antonio A. F. Loureiro tarafından Federal de Minas Gerais Üniversitesi'nde yapılan çalışmada dört farklı şehir için, bu şehirlerin sakinleri ve bu şehirleri ziyaret eden turistlerin zaman, etkinlik ve konum bazındaki davranışlarını incelemiştir. Bu iki kullanıcı grubunun mekân paylaşım kalıpları tespit edilmeye çalışılmıştır. Çalışma mekân öneri sistemlerine yardımcı olmayı hedeflemiştir[4].

Çalışmada incelenen şehirlerin turistlerinin ve şehir sakinlerinin kentsel hareketliliğinin mekânsal ve zamansal yönleri incelenmiş ve zamansal özelliklere sahip bir grafik modeli oluşturulmuştur.

Çalışma sonucunda turistler arasında popüler geçişleri bulmanın mümkün olduğunu ve turistlerin belli başlı yerleri ziyaret ettikleri tespit edilmiştir.

Bu çalışmada kullanıcının gelecekte bulunabileceği mekân kategorisi tahmin edilmeye çalışılırken, Federal de Minas Gerais Üniversitesi'nin çalışmasında turist ve şehir sakinlerinin mekân ve zamana bağlı ziyaret kalıpları oluşturulması hedeflenmiştir.

3. KAVRAMLAR

Bu bölümde çalışmada kullanılan teknoloji ve kavramları örnekler ile açıklanmıştır.

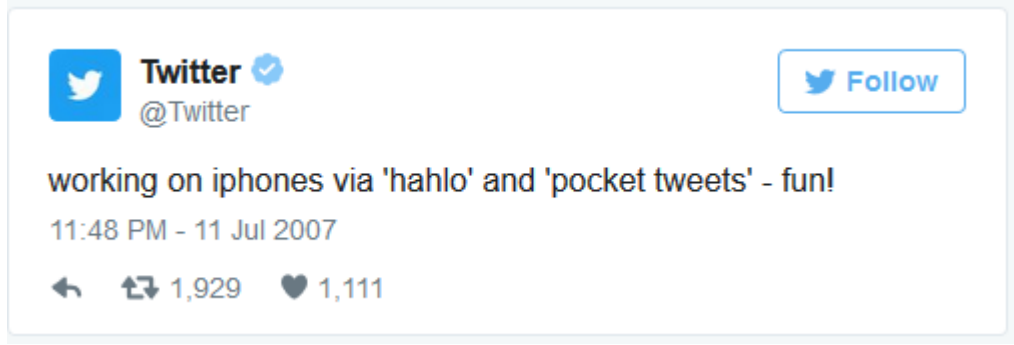
3.1. Veri Madenciliği

Veri madenciliği, yapay zekânın, makine öğrenmesinin, istatistiklerin ve veri tabanı sistemlerinin kesişme noktasındaki yöntemleri içeren geniş veri kümelerindeki kalıpları keşfetmenin hesaplama sürecidir. Veri madenciliği sürecinin genel amacı, bir veri kümesinden bilgi çıkarmak ve daha ileri bir kullanım için anlaşılabilir bir yapıya dönüştürmektir. Ham analiz aşamasının yanı sıra, veri tabanı ve veri yönetimi yönleri, veri ön işleme, model ve çıkarım düşünceleri, ilginçlik ölçümleri, karmaşıklık konuları, görselleştirme ve çevrimiçi güncelleme içerir.

Gerçek veri madenciliği görevi, veri kayıtlarının grupları (küme analizi), olağandışı kayıtlar (anormallik tespiti) ve bağımlılıkları (ilişki kuralı madenciliği, vb.) gibi önceden bilinmeyen ilginç kalıpları çıkarmak için büyük miktarda verinin otomatik veya yarı otomatik analizidir.

3.2. Twitter

Twitter, kullanıcıların tivit adı verilen 140 karakterlik mesaj yazıp, yazılan mesajları okuyabildiği bir sosyal paylaşım sitesidir. Twitter'a kayıtlı kullanıcılar, tivit yayınlayıp okuyabilir iken, kayıtsız olan kullanıcılar sadece bu tivitleri okuyabilirler. Kullanıcılar, web sitesi ara yüzü, SMS veya mobil cihaz gibi alternatif kanallar aracılığı ile Twitter'a erişebilirler.



Şekil.1. Tivit

Kaynak: (1 Haziran 2016) tarihinde <https://twitter.com/> adresinden alınmıştır.

Günümüzde artık dünya üzerindeki çoğunluk için Twitter sadece bir sosyal paylaşım sitesi olmaktan çıkmıştır. İnsanların hem duygu (kızgınlık, beğeni, sevinç, gibi) paylaşımı hem bilgi paylaşımı yaptıkları bir platform olmuştur. Aynı zamanda da haberlerin en önce duyulduğu yer olarak hayatın vazgeçilmez bir parçası olmuştur.

Twitter 2016 yılının bazı istatistikleri şu şekildedir[5].

- 310 milyon aylık aktif kullanıcısı vardır.
- 1.3 milyar hesap oluşturulmuştur
- 1.3 milyar hesabın %44'ü hiç tivit atmamıştır.
- 500 milyon kişi giriş yapmadan Twitter'ı ziyaret etmektedir.
- Kullanıcıların %80'i mobil cihazlar aracılığıyla siteye erişmektedir.
- Kullanıcıların ortalama takipçi sayısı 208'dir.

3.2. Foursquare

Foursquare, kullanıcılar için arama sonuçları sağlayan yerel bir mekân arama ve keşif servisi sağlayan mobil bir uygulamadır. Foursquare ile bir kullanıcı gittiği yerleri, beğendiğini mekânları ve tavsiyelerini güvendiği diğer kullanıcılar ile paylaşabilir.

Foursquare, kullanıcının bulunduğu konumdaki restoranlar, gece hayatı mekânları, mağazaları ve çevredeki diğer ilgi çekici yerleri aramasına olanak tanır. Kullanıcılar uzak bir alanın adını girerek başka alanlarda arama da yapmaları olasıdır. Uygulama, günün saatine göre, sabah kahvaltı yerleri, akşam saatlerinde akşam yemeği yerleri vb. kişiselleştirilmiş öneriler sunar. Öneriler, bir kullanıcıların mekân paylaşım geçmişine ve mekânlarının puanlamalarına göre oluşturulur.

2016 Aralık ayı verilerine göre Foursquare 'in bazı istatistikleri şu şekildedir[6].

- 55 milyon aktif kullanıcısı vardır.
- Toplamda check-in sayısı 10 milyarı geçmiştir.
- Günlük ortalama 8 milyon check-in yapılmaktadır.
- En çok check-in yapılan ülke Türkiye'dir.
- 87 milyon yorum yapılmıştır.
- 600 milyonun üzerinde fotoğraf paylaşılmıştır.
- 100 milyonun üzerinde kayıtlı mekân vardır.

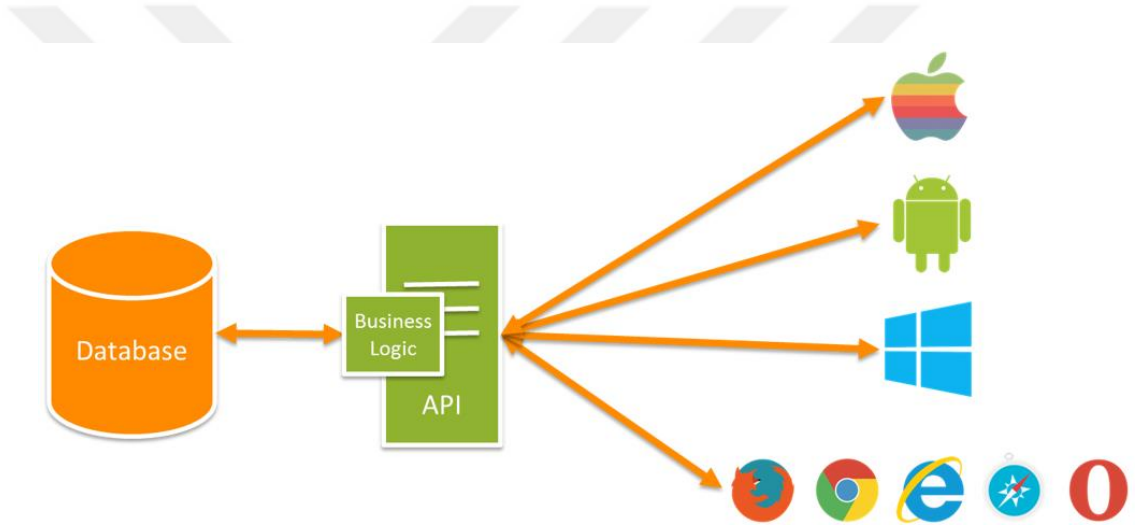


Şekil.2. Foursquare Uygulaması

3.3. Uygulama programlama ara yüzü (API)

Uygulama Programlama Ara yüzü bir bilgisayar yazılımının bir diğerine sağladığı bir sözleşme olarak düşünülebilir. Bir başka deyişle API, yazılımların birbiriyle etkileşim kurmak için kullandığı alternatif kullanıcı arabirimidir.

API'ler internet tabanlı olabildikleri gibi bir platforma özel olabilirler. Örneğin Google'ın arama, takvim, çeviriler vb. için API'leri vardır. Facebook ve Twitter, yazılımın durum güncellemelerini otomatik olarak yükleyebilmesini sağlayan API'lere sahiptir. Apple, iPhone uygulamaları oluşturmak için birçok API sunar.



Şekil.3. API Akış Şeması

Kaynak: (28 Kasım 2016) tarihinde

<https://blogs.msdn.microsoft.com/martinkearn/2015/01/05/introduction-to-rest-and-net-web-api/> adresinden alındı.

3.4. Twitter API

Twitter API'si Twitter verisine programatik olarak okuma ve yazma yapılmasını sağlar. Bu API kullanılarak tivit atılması, kullanıcı bilgileri ve kullanıcının takipçi bilgileri alınması gibi birçok işlem gerçekleştirilebilir. Twitter

API'sine API kullanıcıları OAuth ile erişim sağlarlar. API'nin dönüş tipi JSON formatındadır.

Twitter API'si 15 dakikalık dönem içerisinde maksimum 180 adet çağrı yapılmasına izin verir[7].

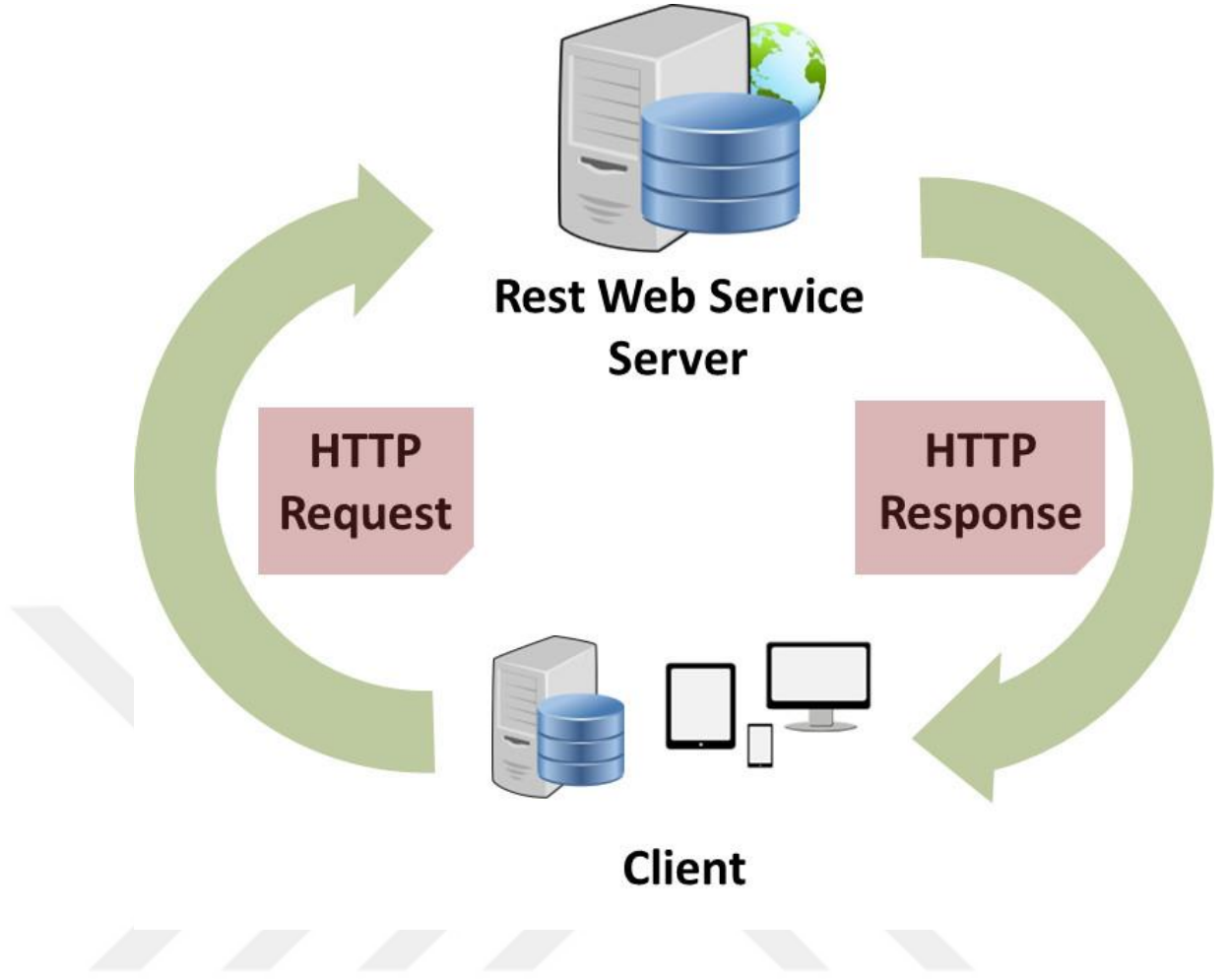
Klasik Twitter API'sinden farklı olarak Twitter Streaming API'si limitlere bağlı kalmadan belirli bir arama sonucu gelen tivitler devamlı bir şekilde ulaşılmasını sağlar[7].

3.5. Foursquare API

Foursquare API'si Foursquare'de bulunan yerler veri tabanına ve Foursquare kullanıcıları ve kayıtlı işletmeler ile programatik olarak etkileşim kurma olanağı sunar[8]. Genel olarak Foursquare API'si, isteklerin gönderebileceği bir RESTful bir adres kümesidir.

3.6. RESTfull

REST veya RESTful web hizmetleri Internet üzerindeki bilgisayar sistemleri arasında birlikte çalışabilirlik sağlamanın bir yoludur. REST uyumlu web hizmetleri, isteyen statik işlemlerden oluşan ve tek biçimli ve önceden tanımlanmış bir dizi kullanarak web kaynaklarının metinsel sunumlarına erişme ve bunları manipüle etme talep eden sistemlere izin verir[9].



Şekil.4. RestFull Servis Akış Şeması

Kaynak: (27 Kasım 2016) tarihinde

https://www.chemaxon.com/app/themes/chemaxon/images/product_pages/jws/rest.jpg adresinden alındı.

3.7. OAuth

OAuth açık standartlı bir kimlik doğrulama protokolüdür. Genellikle internet kullanıcıları tarafından kendi Google, Microsoft, Facebook, Twitter vb. hesaplarının şifrelerini açığa çıkarmadan farklı internet sitelerine bu hesaplardaki bilgilerle bağlanabilmelerini sağlar.

Örneğin Facebook ile bütünleşmiş bir S sitesine erişmek isteyen K kullanıcısı, S sitesine giriş yaparken bu sitedeki Facebook ile bağlan komutunu

kullanabilir. Bu komut K kişisini Facebook sayfasına yönlendirir. K kişisi Facebook'a giriş yapar ve S sitesine, Facebook'da kayıtlı olan Ad, Soyad ve E-Posta gibi bilgilerine erişmesine izin verir. İzin verildikten sonra K kişisi, S sitesine yönlendirilerek bu sitede oturum açması sağlanır. Bu işlem sırasında S sitesi hiçbir şekilde K kişinin Facebook şifresini öğrenmez.

3.8. JavaScript Nesne Notasyonu (JSON)

JavaScript Nesne Notasyonu (JSON) bir veri taşıma formatıdır. JSON insanlar tarafından kolaylıkla okunulabilir ve insanlar tarafından kolaylıkla yazılabilir bir formattır. Aynı şekilde JSON bütün programlama dilleri ile yazılmış olan yazılımlar tarafından kolaylıkla anlaşılabilir ve bu yazılımlar tarafından kolaylıkla üretilebilir bir formattır.

JSON iki ayrı formatta ifade edilir. Bu formatlardan ilki isim değer çiftleridir. Bu format çeşitli programlama dillerinde 'object', 'record', 'struct', 'dictionary', 'hash table', 'keyed list', ve 'associative array' şeklinde tanımlanmıştır.

Bu formatlardan ikincisi sıralı değer listesidir. Bu format çeşitli programlama dillerinde 'array', 'vector', 'list', ve 'sequence' şeklinde tanımlanmıştır.

```
1 {"results":[
2   {"text":"@twitterapi http://tinyurl.com/ctrefg",
3     "to_user_id":396524,
4     "to_user":"TwitterAPI",
5     "from_user":"jkoum",
6     "metadata":
7     {
8       "result_type":"popular",
9       "recent_retweets": 109
10    },
11    "id":1478555574,
12    "from_user_id":1833773,
13    "iso_language_code":"nl",
14    "source":"twitter< /a>",
15    "created_at":"Wed, 08 Apr 2009 19:22:10 +0000"}
16  ],
17  "since_id":0,
18  "max_id":1480307926,
19  "refresh_url":"?since_id=1480307926&q=%40twitterapi",
20  "results_per_page":15,
21  "next_page":"?page=2&max_id=1480307926&q=%40twitterapi",
22  "completed_in":0.031704,
23  "page":1,
24  "query":"%40twitterapi"}
25 }
```

Şekil.5. JSON Formatı

3.9. Linq2Twitter

Linq2Twitter Twitter API'sinin C# programlama dilinde kolayca kullanılabilmesi için geliştirilmiş açık kaynak kodlu bir yazılımdır.

```
var twitterCtx = new TwitterContext(...);

var searchResponse =
    await
    (from search in twitterCtx.Search
     where search.Type == SearchType.Search &&
           search.Query == "\"LINQ to Twitter\""
     select search)
    .SingleOrDefaultAsync();

if (searchResponse != null && searchResponse.Statuses != null)
    searchResponse.Statuses.ForEach(tweet =>
        Console.WriteLine(
            "User: {0}, Tweet: {1}",
            tweet.User.ScreenNameResponse,
            tweet.Text));
```

Şekil.6. LinqToTwitter Program Örneği

Kaynak: (8 Aralık 2016) tarihinde <https://linqtotwitter.codeplex.com/> adresinden alınmıştır.

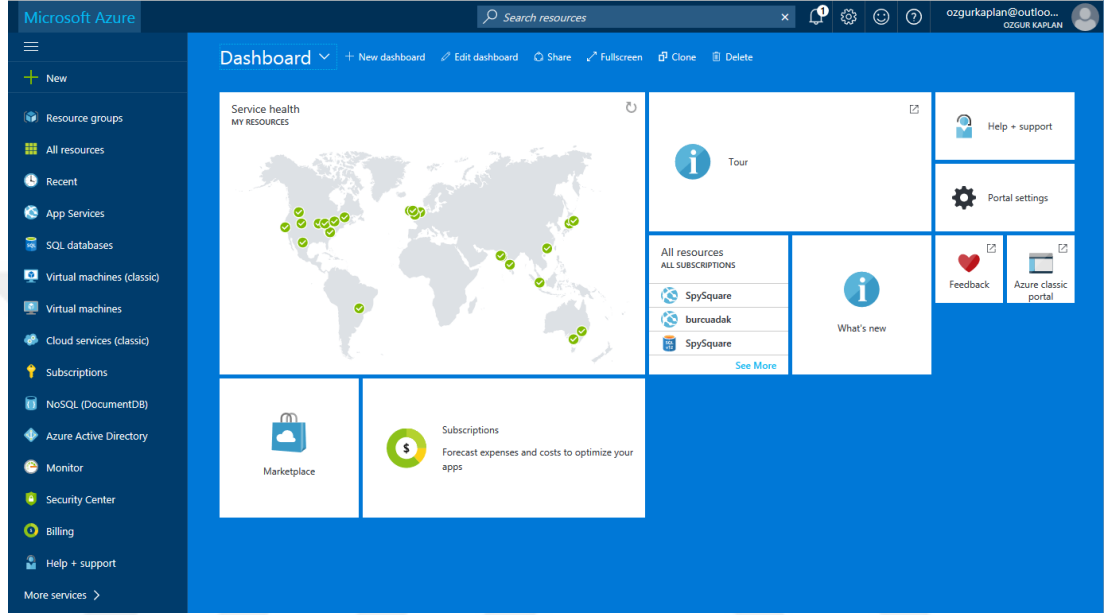
3.10. NoSQL

NoSQL kavramı, ilişkisel veri tabanlarındaki yetersizlikler sebebiyle bu sistemlerinin bir alternatifi olarak ortaya çıkmıştır. Tipik ilişkisel veri tabanları çok sayıda belgeyi indeksleme, yoğun trafiği olan Web sayfalarında ve medya sağlama gibi kimi yoğun veri uygulamalarında yetersizlikler göstermişlerdir. [10] Bu nedenle NoSQL ilişkisel veri tabanları gibi ilişkisel şemalar barındırmazlar. Bunun yerine tipleri önceden belirli olmayan ve yatay olarak genişletilebilen JSON formatında veriler saklarlar. NoSQL yapıları kolaylıkla farklı parçalara ayrılabilirler. Böylelikle her bir parça üzerindeki yük azaltılarak çok büyük bir performans kazancı sağlanmış olur.

Günümüzde Cassandra, Mongo ve Redis gibi popüler NoSQL çözümleri pek on binlerce uygulama tarafından kullanılmaktadırlar.

3.11. Azure

Azure, Microsoft tarafından yönetilen veri merkezlerinin küresel bir ağında uygulamalar ve hizmetleri oluşturmak, dağıtmak ve yönetmek için Microsoft tarafından oluşturulmuş bir bulut bilgi işlem hizmetidir.



Şekil.7. Azure Kontrol Paneli

3.11. DocumentDB

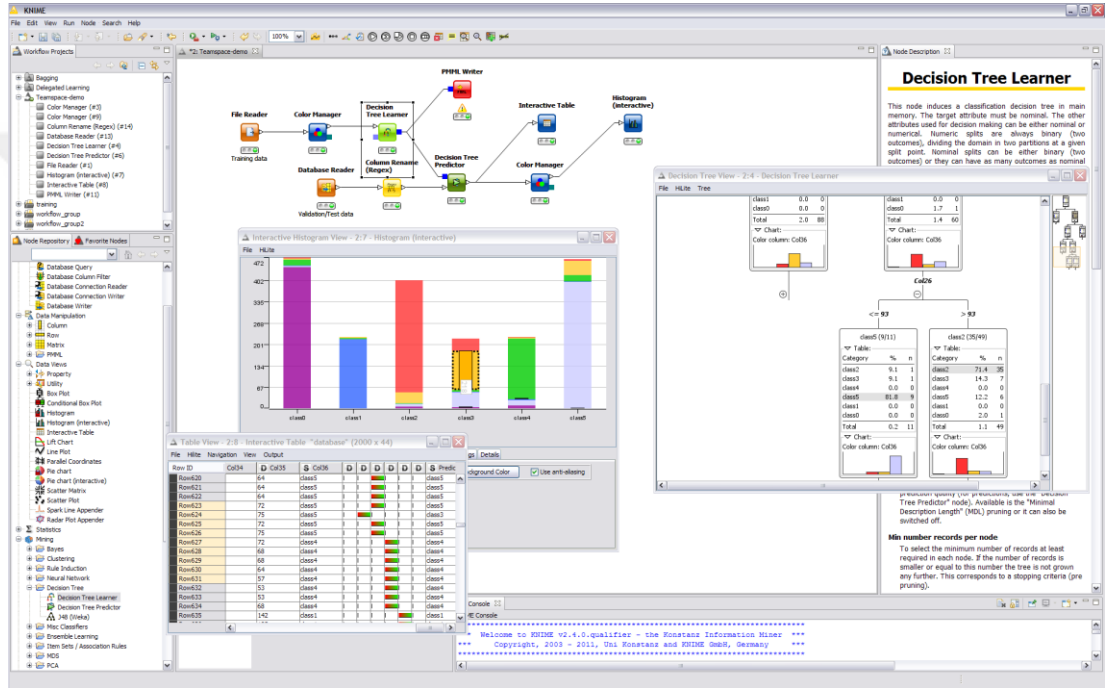
DocumentDB Microsoft firmasının bulut ürünü olan Azure üzerinden kullanıcılara sunduğu bir NoSQL çözümdür.

Azure DocumentDB, JSON verilerini iyi tanımlanmış veri tabanı kaynakları aracılığıyla yönetir. Bu kaynaklar yüksek kullanılabilirlik için çoğaltılır ve mantıksal URI'leri ile benzersiz olarak adreslenebilir. DocumentDB tüm kaynaklar için basit bir HTTP tabanlı RESTful programlama modeli sunar[11].

3.12. Knime

Knime açık kaynak kodlu veri analizi, raporlama ve entegrasyon platformudur. KNIME programı kullanılarak veriler üzerinde makine öğrenimi ve veri madenciliği teknikleri kolaylıkla uygulanabilir.

Knime programı, içerdiği eklentiler sayesinde oldukça güçlü olduğu gibi Knime'a üçüncü parti eklentiler de eklemek oldukça kolaydır[12].



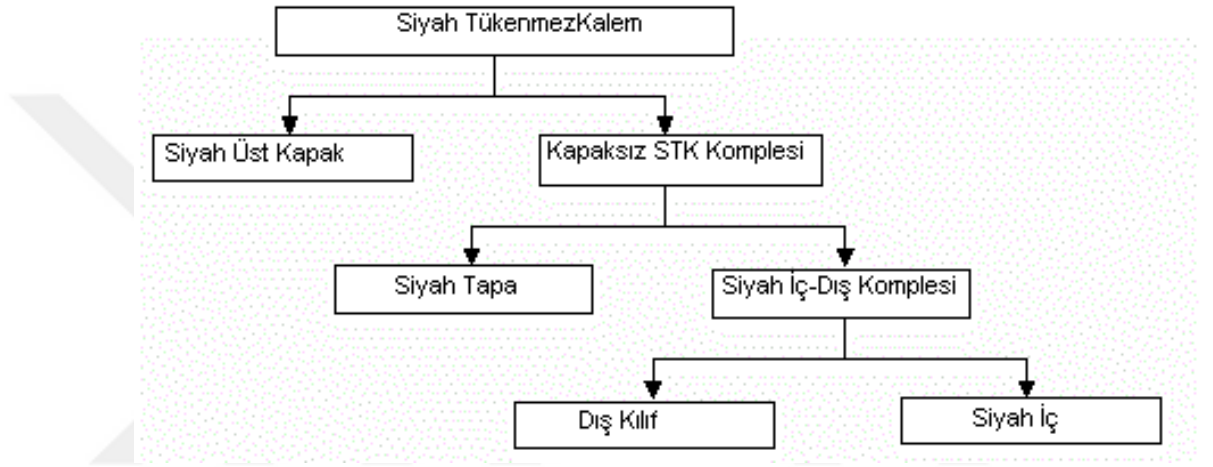
Şekil.8. Knime Programı

Kaynak: (15 Haziran 2016) tarihinde

<https://www.mapr.com/sites/default/files/knimesc2.png> adresinden alındı.

3.13. Karar Ağacı

Öğrenme elemanlarından karar ağaçları yöntemi en güçlü ve en yaygın sınıflandırma ve öngörü araçlarından birisidir. Ağaç yapılı yöntemlerin sık kullanılmasının nedeni ise yapay sinir ağlarının tersine ağaç yapılarının kuralları ifade edebilmesinden kaynaklanmaktadır. Oluşan kurallar çok sade ve nettir. Eğer Ağaç çok fazla büyük değilse her insanın kolaylıkla okuyup anlayacağı türden bir sınıflandırma gerçekleştirir.



Şekil.9. Karar Ağacı

3.14. CSharp (C#) Programlama Dili

C# Microsoft firması tarafından geliştirilmiş olan, nesne yönelimli ve bileşen odaklı programlama disiplinlerini kapsayan çok paradigmatlı bir programlama dilidir. Geliştirme ekibine Anders Hejlsberg liderlik etmiştir.

Microsoft'un .NET girişimiyle geliştirilmiş ve daha sonra Ecma ve ISO tarafından standart olarak onaylanmıştır. C#, ortak dil altyapısı için tasarlanmış programlama dillerinden biridir.

C # dilinin temel yazım şekli, C, C ++ ve Java gibi diğer C tabanlı dillere benzer. Örneğin;

- Noktalı virgöl, bir ifadenin sonunu belirtmek için kullanılır.
- Kodları gruplamak için küme parantezi kullanılır. Kodlar genellikle metot ve sınıflar içerisinde gruplandırılır.
- Değişkenlere değere ataması eşit karakteriyle atanır iken iki değer karşılaştırılması çift eşit karakterleriyle yapılır.
- Köşeli parantezler dizileri tanımlamak için kullanılır.

```
1 using System;
2
3 namespace OzgurKaplanConsoleApplication
4 {
5     class Program
6     {
7         static void Main(string[] args)
8         {
9             Console.WriteLine("C# Programla dili nedir?");
10        }
11    }
12 }
13
```

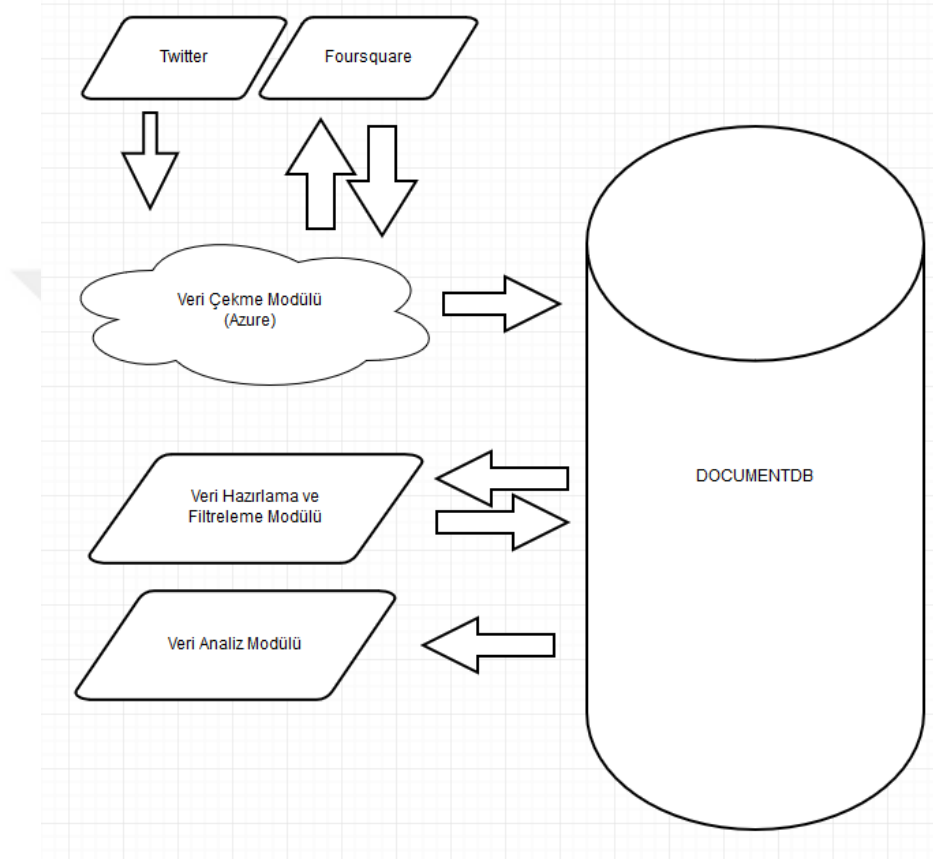
Şekil.10. C# İle Uygulama Örneği

3.15. Gradyan İyileştirmesi

Gradyan iyileştirmesi, regresyon ve sınıflandırma problemleri için bir makine öğrenme tekniğidir. Zayıf tahminsel modellerin, genelde karar ağaçlarının bir topluluğu biçiminde bir tahmin modeli üretir. Modeli, diğer artırma yöntemleri gibi aşamalı bir biçimde oluşturur ve keyfi ayrılabilir bir kayıp işlevinin optimizasyonuna izin vererek onları genelleştirir[13].

4. YÖNTEM

Bu bölümde Foursquare konum tahmini programının geliştirilme aşamaları ve işleyiş şekli açıklanmıştır.



Şekil.11. Foursquare Konum Tahmini Programı Akışı

4.1 Verilerin Toplanması

Bu modülün amacı, Foursquare’deki check-in bilgilerinin toplanmasıdır. Foursquare yapısı gereği arkadaş olunmayan kişilerin check-in bilgilerinin üçüncü şahıslar tarafından erişilmesine izin vermemektedir. Bu nedenle Foursquare API’sini kullanarak direkt olarak Foursquare check-in tanımadığımız kişilerin check-in verilerine ulaşmamız mümkün değildir.

Foursquare kullanıcıları yaptıkları check-in'leri Twitter üzerinde de kolaylıkla paylaşabilmektedirler. Eğer bir kullanıcı mekân paylaşımını Twitter'da yayınlarsa, Twitter'ın yapısı gereği artık bu paylaşım herkes tarafından erişilebilecek hale gelmiş olur. Ancak Twitter API'si de gönderilen bütün tivitlerin sadece %1'ine ulaşılmasına izin vermektedir[14]. Bu nedenle bir kişi bütün check-inlerini Twitter'da paylaşırsa dahi Twitter API'si kullanılarak bu check-inlerin sadece bir bölümüne ulaşabiliriz.

Kullanıcının Twitter'da paylaştığı Foursquare check-inleri `swarmapp.com/c/{benzersiz tanımlayıcı}` şeklinde bir link içermektedir. Bu linkteki {benzersiz tanımlayıcı} kısmı Foursquare check-in işleminin, Foursquare sisteminde kaydının benzersiz tanımlayıcısıdır. Bu benzersiz tanımlayıcıyı kullanarak Foursquare API'si üzerinden kişinin check-in'i ile ilgili mekân bilgisi, mekân koordinatları, ülke, şehir, mekân kategorisi, check-in tarihi gibi detay bilgilere ulaşmak mümkündür.

Verilerin toplanması için öncelikle Linq2Twitter yazılımı kullanılarak içerisinde "`swarmapp.com/c/`" geçen tivitler Twitter API'si kullanılarak çekilmiştir.

```

1 reference
private List<Status> GetTweets(UnitOfWork work)
{
    var auth = new SingleUserAuthorizer
    {
        CredentialStore = new SingleUserInMemoryCredentialStore
        {
            ConsumerKey = _twitterApiKey,
            ConsumerSecret = _twitterApiSecret,
            AccessToken = _twitterAccessToken,
            AccessTokenSecret = _twitterAccessTokenSecret
        }
    };
    var lastCheckIn = work.CheckInRepository
        .Get(null, p => p.OrderByDescending(o => o.CheckInDate))
        .FirstOrDefault();
    long lastTweetId = 0;
    if (lastCheckIn != null)
    {
        lastTweetId = lastCheckIn.TweetId;
    }
    var twitterCtx = new TwitterContext(auth);
    return (from search in twitterCtx.Search
        where search.Type == SearchType.Search &&
            search.Query == "swarmapp.com/c/" &&
            search.Count == 100 &&
            search.SinceID == (ulong)lastTweetId + 1
        select search.Statuses)
        .FirstOrDefault();
}

```

Şekil.12. Tivit Çeken Kod Parçasığı

Şekil 9'daki kod bloğunda LinqToTwitter ve Twitter Search API'si kullanılarak içerisinde "swarmapp.com/c/" geçen tivitlerin çekilmesi işlemi gerçekleştirilmiştir. Aynı tivitlerin tekrar tekrar gelmemesi için son çekilen tivitın Id alanı veri tabanında saklanmıştır. Kod bloğu her çalışmasında son çekilen tivitten itibaren atılan tivitleri aramaktadır.

Twitter API'si üzerinden gelen kullanıcının Twitter kullanıcı adı, Twitter'daki takipçi sayısı, Twitter'da takip ettiği kullanıcı sayısı, toplam yazdığı tivit sayısı ve Twitter'da belirttiği memleket bilgisi DocumentDB veri tabanına kaydedilmiştir. Daha çekilen tivitlerdeki “swarmapp.com/c/”den sonra gelen benzersiz tanımlayıcılar ayrıştırılmıştır.

```
1 reference
private string GetShortId(Status tweet)
{
    var url = tweet.Entities.UrlEntities.FirstOrDefault();
    if (url != null)
    {
        return url.ExpandedUrl.Split('/').Last();
    }
    return null;
}
```

Şekil.13. Foursquare Check-In Anahtarının Bulunması

Şekil 10'daki kod bloğunda çekilen tivit'deki Foursquare linki üzerinden linkin son kısmında bulunan Id alanının bulunması işlemi yapılmaktadır.

Ayrıştırılan benzersiz tanımlayıcılar kullanılarak Foursquare API'si üzerinden check-in detay bilgileri çekilmiştir.

```

var tweets = GetTweets(work);
foreach (var tweet in tweets)
{
    var shortId = GetShortId(tweet);
    if (string.IsNullOrEmpty(shortId))
    {
        continue;
    }
    var square = new FourSquare.SharpSquare.Core.SharpSquare(_clientId[counter % 14],
        _clientSecret[counter % 14],
        _accessToken[counter % 14]);
    var checkin = square.GetCheckinShort(shortId);
    if (checkin == null)
    {
        continue;
    }

    CheckIn check = new CheckIn()
    {
        Shout = checkin.shout,
        CheckInDate = Convert.ToInt64(checkin.createdAt).UnixTimeToDate(),
        CheckInDateEpoch = Convert.ToInt64(checkin.createdAt),
        Source = checkin.source.name,
        TweetId = (long)tweet.StatusID,
        CheckInId = checkin.id,
        Tweet = tweet.Text
    };
}

```

Şekil.14. Checkin Nesnesinin Doldurulması

Şekil 11'deki kod bloğunda Foursquare Api'sine check-in'in Id değeri parametre olarak gönderilerek check-in'in detay bilgileri elde edilmiştir. Elde edilen veri check-in nesnesi üzerinde saklanmıştır.

Check-in yapılan mekânın, mekân adı, mekân kategorisi, mekân koordinatları, mekân şehri, mekân ülkesi, kullanıcının kullanılan telefon tipi (andorid veya ios) ve kişinin cinsiyet bilgisi verileri DocumentDB veri tabanına kaydedilmiştir.

```

CheckIn check = new CheckIn()
{
    Shout = checkin.shout,
    CheckInDate = Convert.ToInt64(checkin.createdAt).UnixTimeToDate(),
    CheckInDateEpoch = Convert.ToInt64(checkin.createdAt),
    Source = checkin.source.name,
    TweetId = (long)tweet.StatusID,
    CheckInId = checkin.id,
    Tweet = tweet.Text
};

long userId = Convert.ToInt64(checkin.user.id);
var user = work.UserRepository
    .SingleOrDefault(u => u.FoursquareId == userId);
if (user == null)
{
    var u = new SpySquare.Core.Entities.User
    {
        TwitterName = tweet.User.ScreenNameResponse,
        FirstName = checkin.user.firstName,
        LastName = checkin.user.lastName,
        FoursquareId = Convert.ToInt64(checkin.user.id),
        Gender = checkin.user.gender,
        TwitterId = Convert.ToInt64(tweet.User.UserIDResponse),
    };
    check.User = u;
    check.FoursquareId = Convert.ToInt64(u.FoursquareId);
}
else
{
    check.User = user;
    check.FoursquareId = user.FoursquareId;
}

```

Şekil.15. Kullanıcı bilgilerinin Alınması

Şekil 12'deki kod bloğunda check-in bilgisiyle check-in yapan kullanıcı ilişkisi kurulmuştur.

```

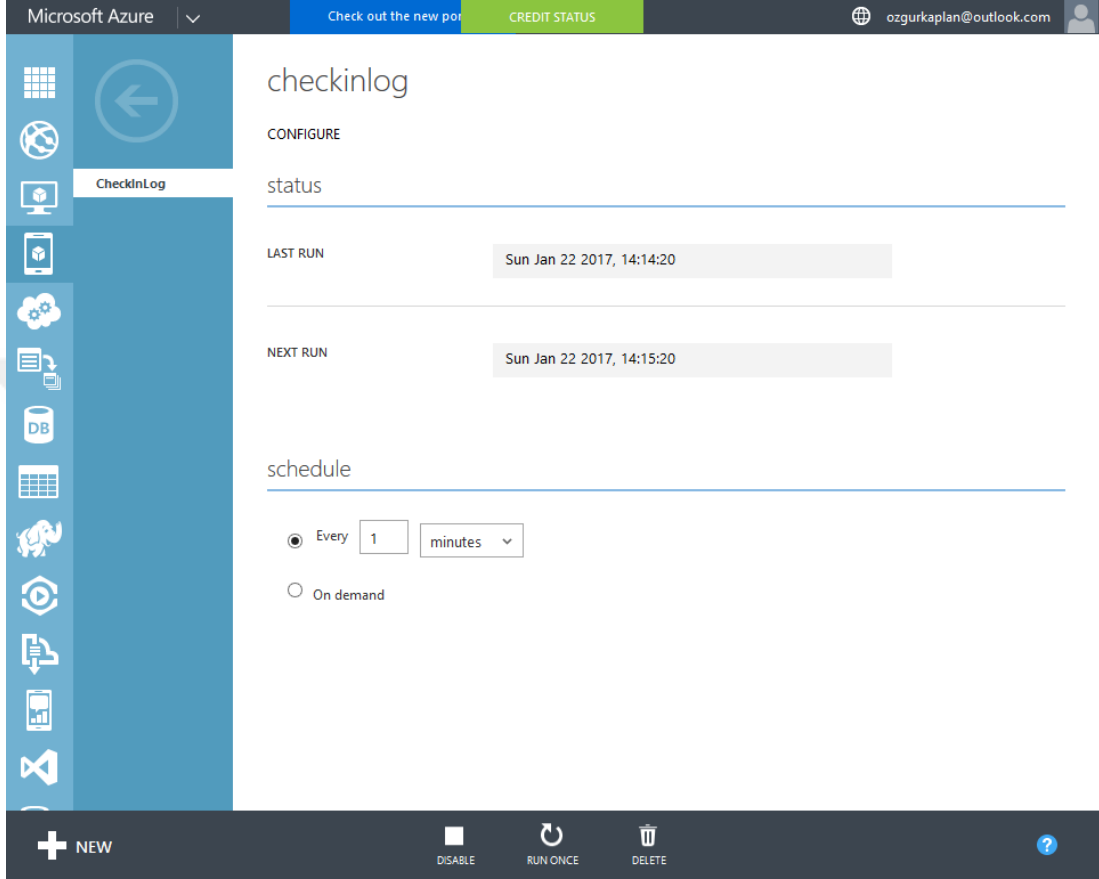
var venue = work.VenueRepository
    .SingleOrDefault(v => v.VenueId == checkin.venue.id);
if (venue == null)
{
    var v = new SpySquare.Core.Entities.Venue()
    {
        VenueId = checkin.venue.id,
        Name = checkin.venue.name,
        Country = checkin.venue.location.country,
        City = checkin.venue.location.city,
        Adress = checkin.venue.location.address,
        Latitude = checkin.venue.location.lat,
        Longitude = checkin.venue.location.lng,
    };
    v.Categories = new List<SpySquare.Core.Entities.Category>();
    foreach (var c in checkin.venue.categories)
    {
        var category = work.CategoryRepository
            .SingleOrDefault(ca => ca.CategoryId == c.id);
        if (category == null)
        {
            var cat = new SpySquare.Core.Entities.Category()
            {
                CategoryId = c.id,
                Name = c.name,
                ShortName = c.shortName
            };
            v.Categories.Add(cat);
        }
        else
        {
            v.Categories.Add(category);
        }
    }
    check.Venue = v;
    check.VenueId = v.VenueId;
}
else
{
    check.Venue = venue;
}
try
{
    work.CheckInRepository.Insert(check);
    work.Save();
}

```

Şekil.16. Mekân Bilgilerinin Toplanması

Şekil 13'deki kod bloğunda check-in ile mekân bilgileri eşleştirilmiştir. Ayrıca mekân ile mekân kategorisi arasındaki ilişki de kurulmuştur.

Veri toplama işleminin sürekli bir şekilde çalışabilmesi için Microsoft'un bulut bilişim çözümü olan Azure üzerinde bir planlı iş tanımlaması yapılmıştır. Geliştirilen program bu iş tanımı ile ilişkilendirilmiştir. Böylelikle programın periyodik olarak sürekli çalışması sağlanmıştır.



Şekil.17. Azure Periyodik İş Tanımı

Uygulama 1 Ocak 2015 ile 1 Temmuz 2015 tarihleri arasında çalışarak 3151449 adet check-in bilgisi toplamıştır.

Tablo.1. Check-in Datası

CheckInId	54c56d9d498e61bb4182973f
TweetId	559477468063875000
Tweet	Ngopi kopsus again (@ Rumah Kopi Sarina in Minahasa, Sulawesi Utara) https://t.co/8OCIFrtpbb
FoursquareId	4140983
VenueId	4c441b71429a0f47d441481e
Source	Swarm for iOS
CheckInDate	2015-01-25 22:26:37.000
CheckInId	54c56dcc498ebc40c570e246
TweetId	559477665619783680
Tweet	A bebeeeerrrsssss (@ DUBLIN Public House in Celaya, Guanajuato) https://t.co/TYkFkTObMi
FoursquareId	103546641
VenueId	51b2a27a498e8da92f8b09c3
Source	Swarm for Android
CheckInDate	2015-01-25 22:27:24.000

Tablo.2. Kullanıcı Datası

FoursquareId	37
TwitterId	7482
FirstName	Alex
LastName	Rainert
Gender	male
TwitterName	arainert
TwitterLocation	New York
FollowersCount	9027
FollowingCount	535
TweetCount	37080

Tablo.3. Mekân Datası

VenueId	4b14cf5af964a520b8a623e3
Name	Fuji Xerox Towers
Adress	80 Anson Rd
Latitude	1,273323438
Longitude	103,8439274
City	Singapore
Country	Singapore
CategoryId	4bf58dd8d48988d124941735
CategoryName	Office

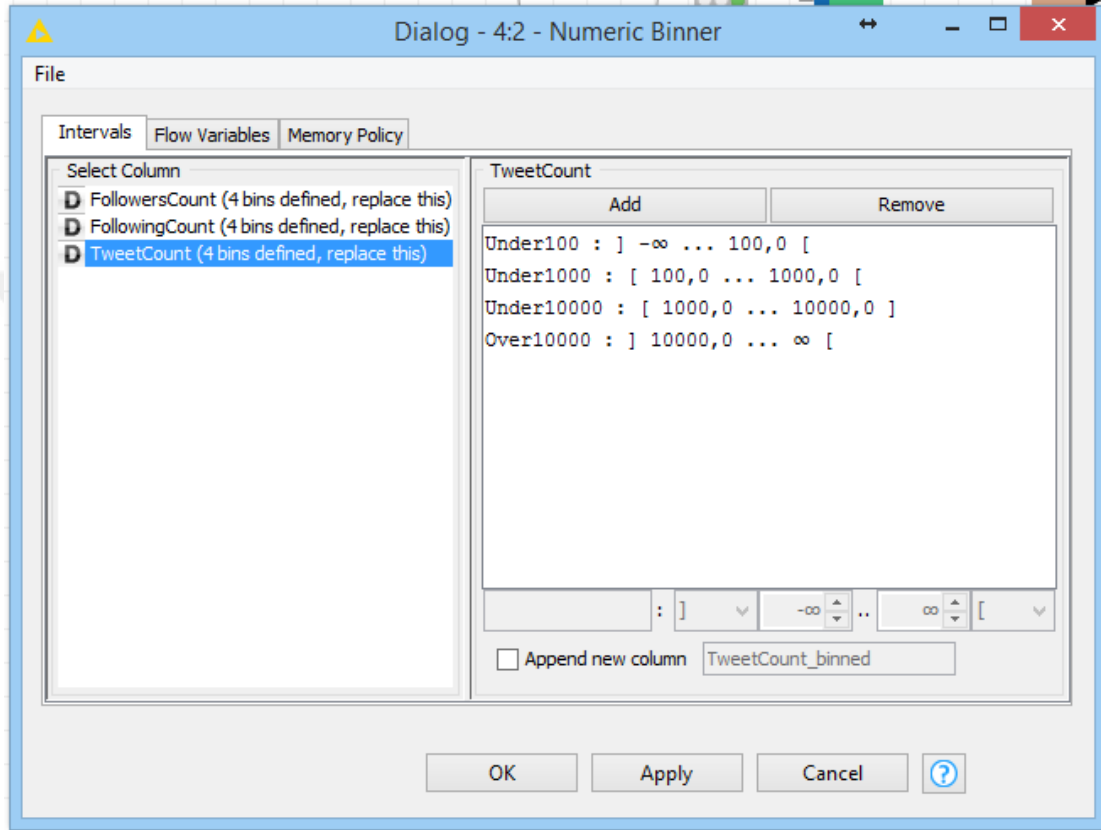
4.2 Verilerin Hazırlanması

Veri madenciliği algoritmaları ham veri üzerinde başarılı sonuçlar verememektedir. Ham verinin işlenerek veri madenciliği algoritmalarıyla uyumlu hale getirilmesi istenilen sonuçlara ulaşmak için oldukça önemlidir.

Bu bölümde toplanılan ham veri üzerinde yapılan veri hazırlama çalışmaları anlatılmıştır.

Çalışmada Twitter takipçi sayısı, Twitter'da takip edilen sayısı ve toplam paylaşılan tivit alanları sayı olarak kaydedilmiştir. Veri madenciliği algoritmalarının düzgün çalışabilmesi için bu verinin gruplanmasının yapılması gerekmektedir. Bunun nedeni algoritmaların her bir sayıyı tamamen farklı olarak algılayarak işlem yapmasından kaynaklanmaktadır. Örneğin 1567 takipçisi olan bir kullanıcı ile 1566 adet takipçisi olan kullanıcı bir işlem yapılmadığı takdirde algoritmalar bu iki veriyi

birbirinden tamamen bağımsız olarak değerlendirirler. Birbirine yakın sayıların tamamen farklı olarak algılanması, algoritmaların istediğimizin ilişkisel sonuçları bulmasını engel olacaktır. Bu durumu ortadan kaldırmak için Knime programında bulunan NumericBinner nesnesi kullanılarak gruplama işlemi yapılmıştır.



Şekil.18. Numeric Binner Nesnesi Ayarları

Takipçi sayıları NumericBinner kullanılarak Tablo 4’de belirtilen kriterlere göre 4 farklı kategoriye ayrılmıştır.

Tablo.4. Takipçi Sayısı Grupları

Kategori	Asgari Takipçi Sayısı	Azami Takipçi Sayısı
Under100	0	99
Under1000	100	999
Under10000	1000	9999
Over10000	100000	-

Takip edilen sayıları NumericBinner kullanılarak Tablo 4’de belirtilen kriterlere göre 4 farklı kategoriye ayrılmıştır.

Tablo.5. Takip Edilen Sayısı Grupları

Kategori	Asgari Takipçi Sayısı	Azami Takipçi Sayısı
Under100	0	99
Under1000	100	999
Under10000	1000	9999
Over10000	100000	-

Atılan tivit sayıları NumericBinner kullanılarak Tablo 4’de belirtilen kriterlere göre 4 farklı kategoriye ayrılmıştır.

Tablo.6. Takip Edilen Sayısı Grupları

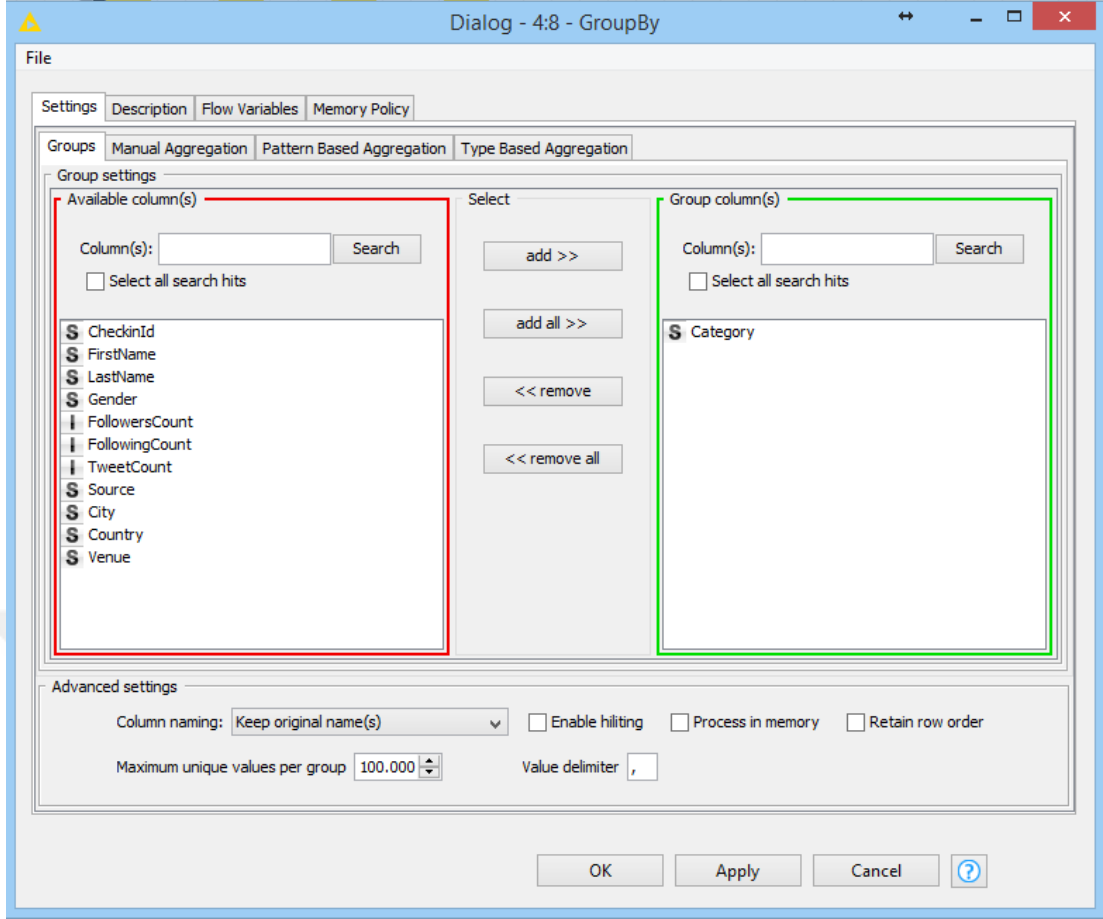
Kategori	Asgari Takipçi Sayısı	Azemi Takipçi Sayısı
Under100	0	99
Under1000	100	999
Under10000	1000	9999
Over10000	100000	-

4.3 Verilerin Filtelenmesi

Veri madenciliği algoritmalarının düzgün çalışabilmesi için anlam ifade etmeyen alanların bu algoritmalar çalışırken kullanılmaması gerekmektedir. Id alanları, ad, soyad alanları gibi alanlarının filtrelenerek öğrenme verisinden çıkartılmaları gerekmektedir.

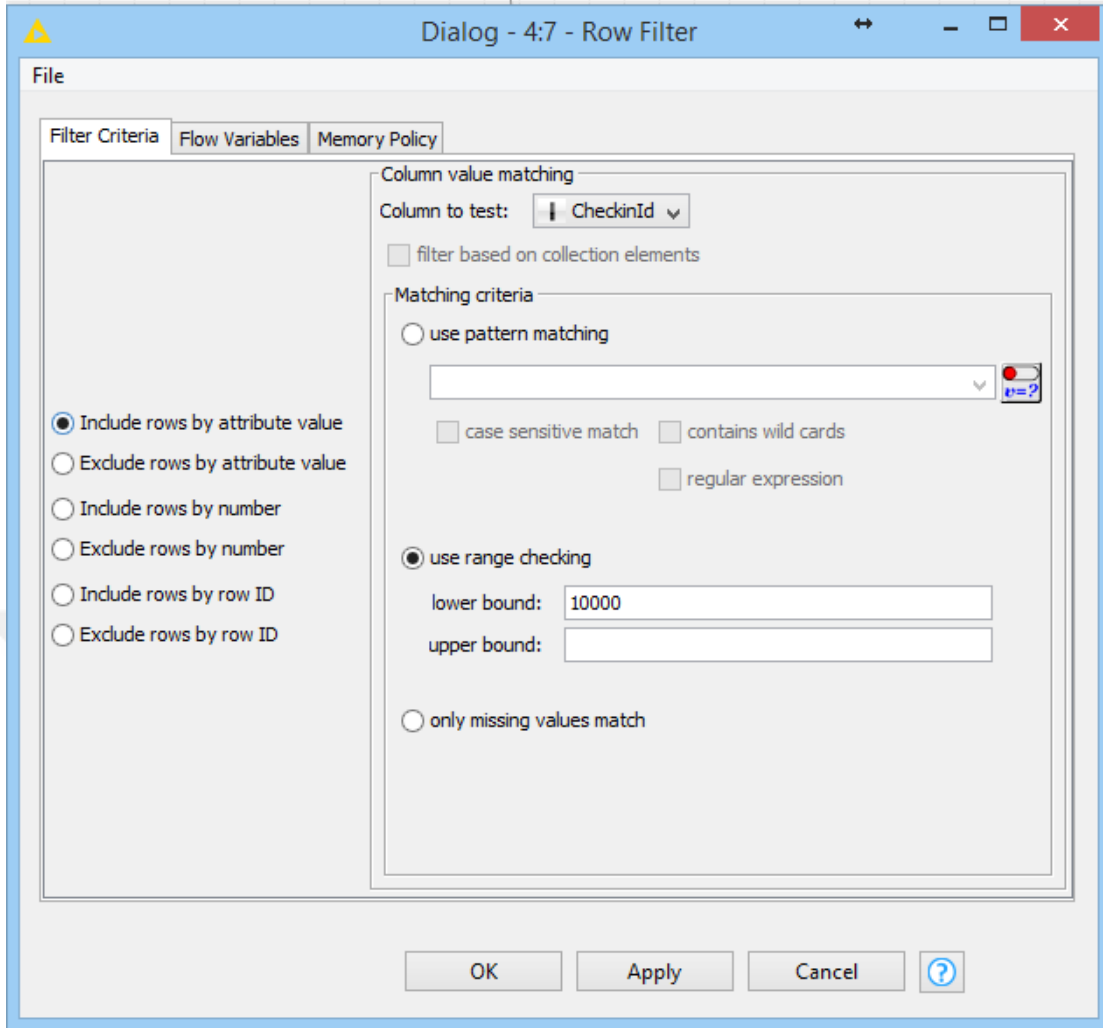
Veri madenciliği veriler içerisindeki ilişkilerden bir desen yakalamayı amaçladığı için veri kümesinin tekrar eden ilişkiler içermesi gerekmektedir. Çalışmada kişilerin gidebileceği mekânların kategorileri tahmin edilmek istendiği için bu kategorilerin veri kümesinde belirli âdetin üzerinde bulunuyor olması öğrenme algoritmalarının daha verimli çalışmasını sağlayacaktır. Bu verimliliği yakalamak için 10000'in altında ziyaret edilen kategoriler ve bu kategorilerle ilgili check-in'lerin filtrelenmesine karar verilmiştir.

Knime programındaki “Group By” nesnesi kullanılarak check-in'ler mekan kategorisi alanına göre gruplanmıştır.



Şekil.19. Knime Group By Nesnesi Ayarları

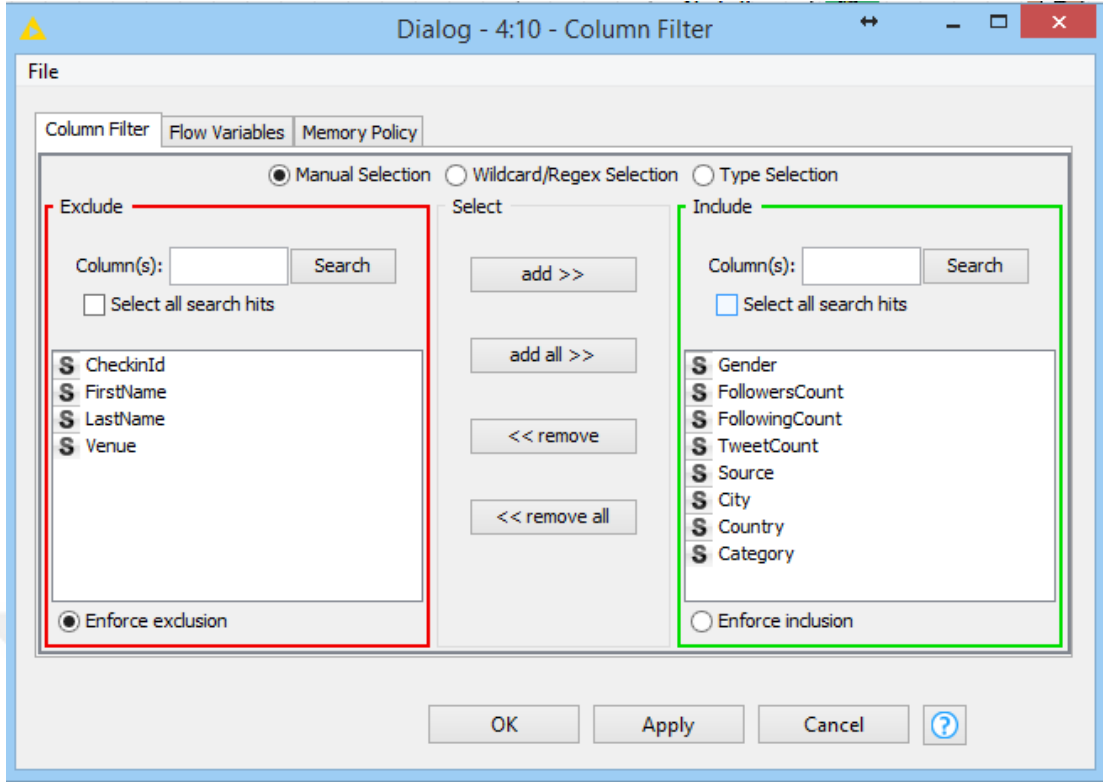
Veriler gruplandıktan sonra “Row Filter” nesnesi ile 10000’in altında geçen kategoriler filtrelenmiştir.



Şekil.20. Knime RowFilter Nesnesi Ayarları

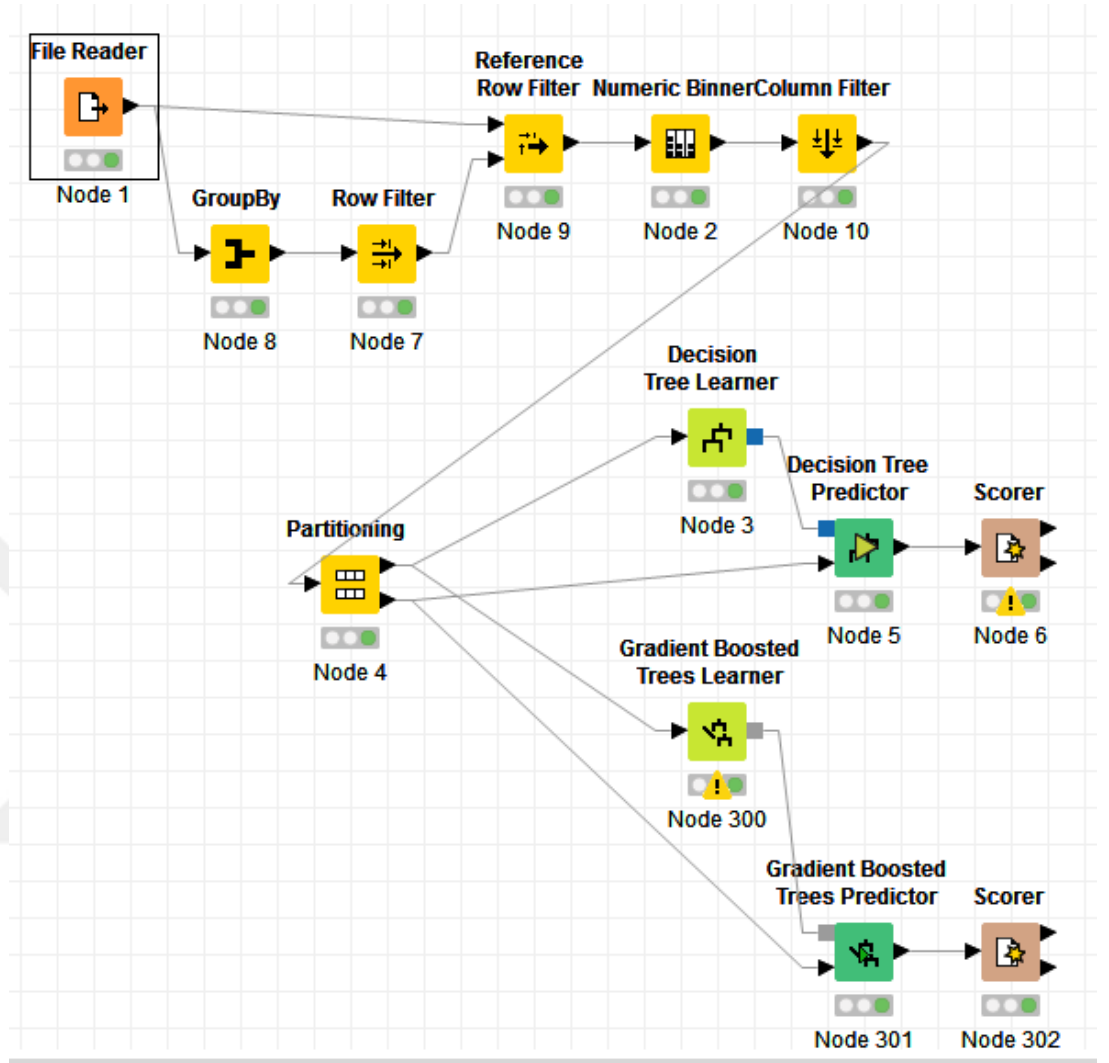
Filtrelenen veriler ile ana veriler Knime üzerindeki “Reference Row Filter” nesnesi ile birleştirilerek son durumda kullanılacak olan check-in verileri elde edilmiştir.

Veri filtreleme aşamasının son işlemi olarak anlam taşımayan tekil belirleyici alanları, mekân ve kişi adları Knime programı üzerinde “Column Filter” kullanılarak filtrelenmiştir.



Şekil.21. Knime Column Filter Nesnesi Ayarları

4.4 Verilerin Knime Programı ile Analiz Edilmesi



Şekil.22. Analiz Programı Akışı

Knime programı DocumentDB desteklemediği için gerekli veriler CSV formatına dönüştürülmüştür.

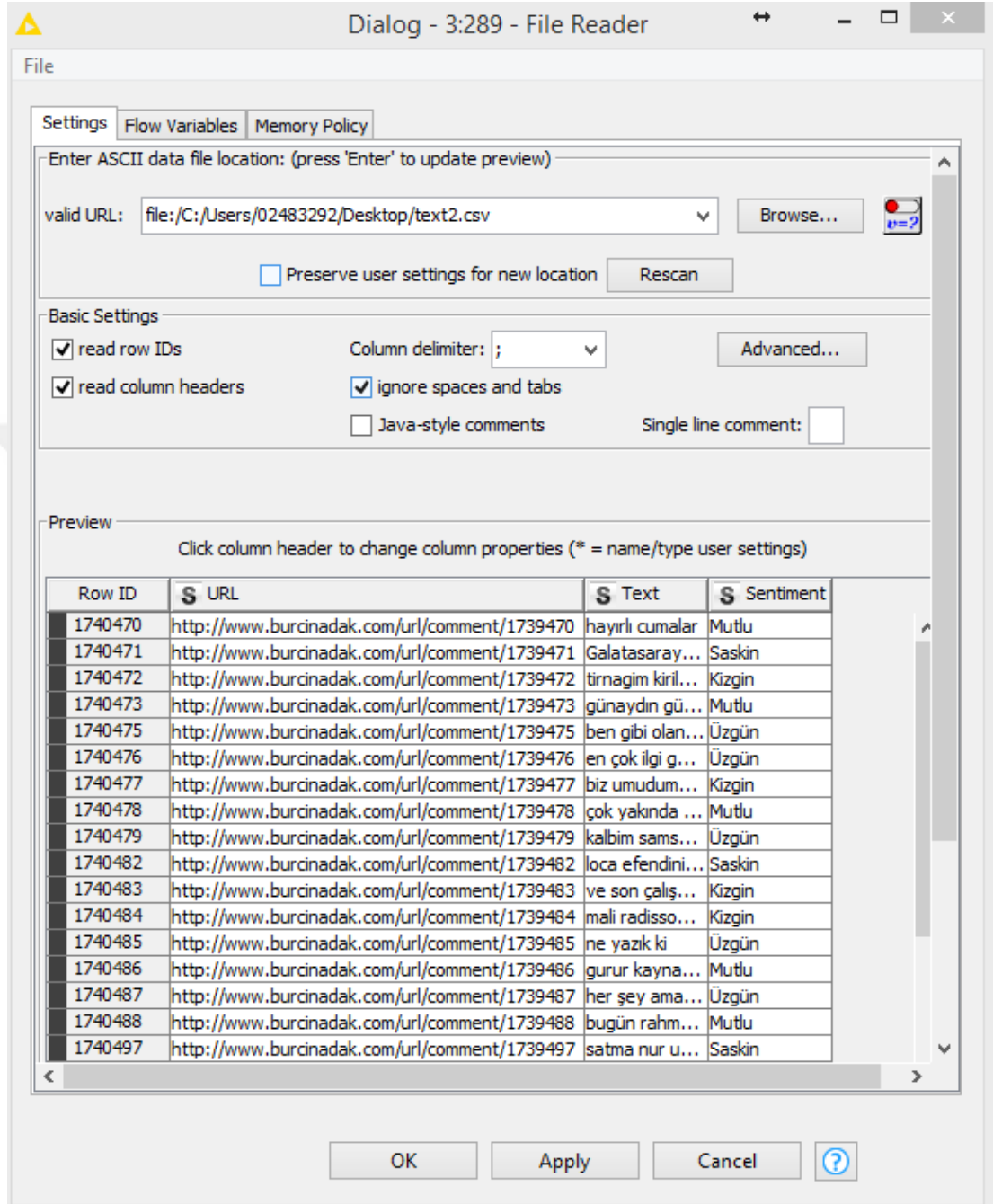
```

1 CheckinId;FirstName;LastName;Gender;FollowersCount;FollowingCount;TweetCount;S
source;City;Country;Venue;Category
2 54efd511498e76a36a91c09e;Phyo;Ba Kyu;male;118;257;21045;Swarm for
iOS;Philadelphia;United States;Pat's King of Steaks;Sandwiches
3 555b9dde498e7e012a466341;Jahmal;Mayers;male;731;540;46747;Swarm for
iOS;Philadelphia;United States;Pat's King of Steaks;Sandwiches
4 56765874498e1256cd86fc3a;Brandon;Slattery;male;343;504;6080;Swarm for
iOS;Philadelphia;United States;Pat's King of Steaks;Sandwiches
5 555b9dc8498e447b98afcb13;Yani;Iman;female;353;630;762;Swarm for iOS;New
York;United States;Cafe Habana;Cuban
6 555bcadd498e4e628d6fe062;Kristin;Brooks;female;1226;494;65087;Swarm for
iOS;New York;United States;Cafe Habana;Cuban
7 5686ecd7498e312224362b5e;Julio;Gallego;male;37;188;283;Swarm for iOS;New
York;United States;Cafe Habana;Cuban
8 5547ec66498e763d3f4346b0;Scott;Gold;male;150;434;2234;Swarm for iOS;New
York;United States;Cafe Habana;Cuban
9 5525d893498e18cbdd0b01e2;La Fer;@lenubienne;female;5230;662;146748;Swarm for
Windows Phone;New York;United States;S.O.B.'s;Bar
10 5552943d498ec25d4c6a133e;Kristin;Brooks;female;1226;494;65087;Swarm for
iOS;New York;United States;S.O.B.'s;Bar
11 555a9dad498ef90126f95ccd;Rebe;Espinosa;female;473;560;3953;Swarm for iOS;New
York;United States;S.O.B.'s;Bar
12 555bf188498e13f4ccd01f8d;Josh;Clarke;male;243;180;15336;Swarm for iOS;New
York;United States;S.O.B.'s;Bar
13 54c2dc1f498ecd28b6e960f1;Chris;Session;male;146;87;1765;Swarm for
Android;New York;United States;Raoul's Restaurant;French
14 554d5245498ef3d5d375af87;Chris;Session;male;146;87;1765;Swarm for
Android;New York;United States;Raoul's Restaurant;French
15 567c837a498e4f4bc4101597;Raul;Jimenez II;male;493;1772;4178;Swarm for
Android;New York;United States;Raoul's Restaurant;French
16 54c56dcc498e6fc833f7ce33;Chris;Session;male;146;87;1765;Swarm for
Android;New York;United States;Raoul's Restaurant;French
17 54cd58f7498e4b619b2640e9;Chris;Session;male;146;87;1765;Swarm for
Android;New York;United States;Raoul's Restaurant;French
18 54c6c613498eb6cf38065741;Chris;Session;male;146;87;1765;Swarm for
Android;New York;United States;Raoul's Restaurant;French
19 555a5c3e498eb5ee9c9d01bc;Chris;Session;male;146;87;1765;Swarm for
Android;New York;United States;Raoul's Restaurant;French
20 54eba84a498ec43d80f2657a;Chris;Session;male;146;87;1765;Swarm for
Android;New York;United States;Raoul's Restaurant;French
21 54cc0aea498e7474da4ff19a;Chris;Session;male;146;87;1765;Swarm for
Android;New York;United States;Raoul's Restaurant;French
22 5688b8ea498e6eeebc608f2;Veronika;Mata;female;837;994;26016;Swarm for
iOS;New York;United States;Niagara;Restaurant
23 5531b915498e6942955cce9d;Steven;Pesantez;male;0;0;0;Swarm for iOS;New
York;United States;Pianos;Rock Club
24 56723e9b498ed2a7ef793e69;Mathew;Thomas;male;101;176;375;Swarm for iOS;New
York;United States;Pianos;Rock Club

```

Şekil.23. CSV Formatına Çevrilmiş Veriler

CSV formatındaki veriler Knime üzerindeki “FileReader” nesnesiyle Knime platformuna yüklenmiştir.

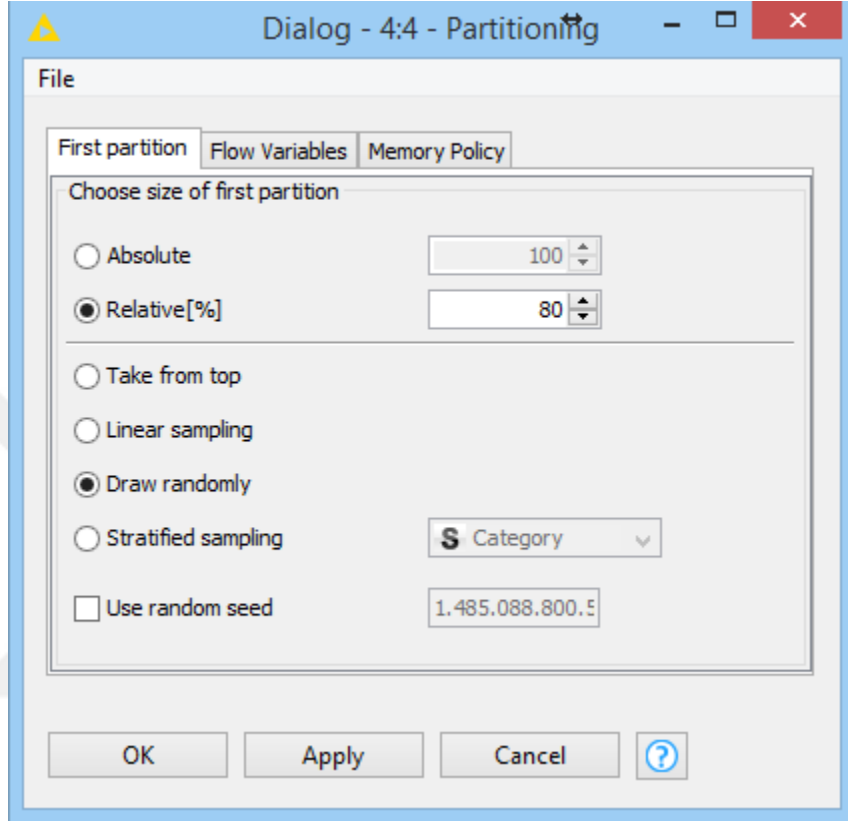


Şekil.24. Knime File Reader Nesnesi Ayarları

Dosyalar Knime programına aktarıldıktan sonra veri filtreleme ve veri hazırlama adımları anlatılan işlemler sırasıyla veriye uygulanmıştır. Böylelikle veri öğrenme algoritmalarının çalışmasına uygun hale getirilmiştir.

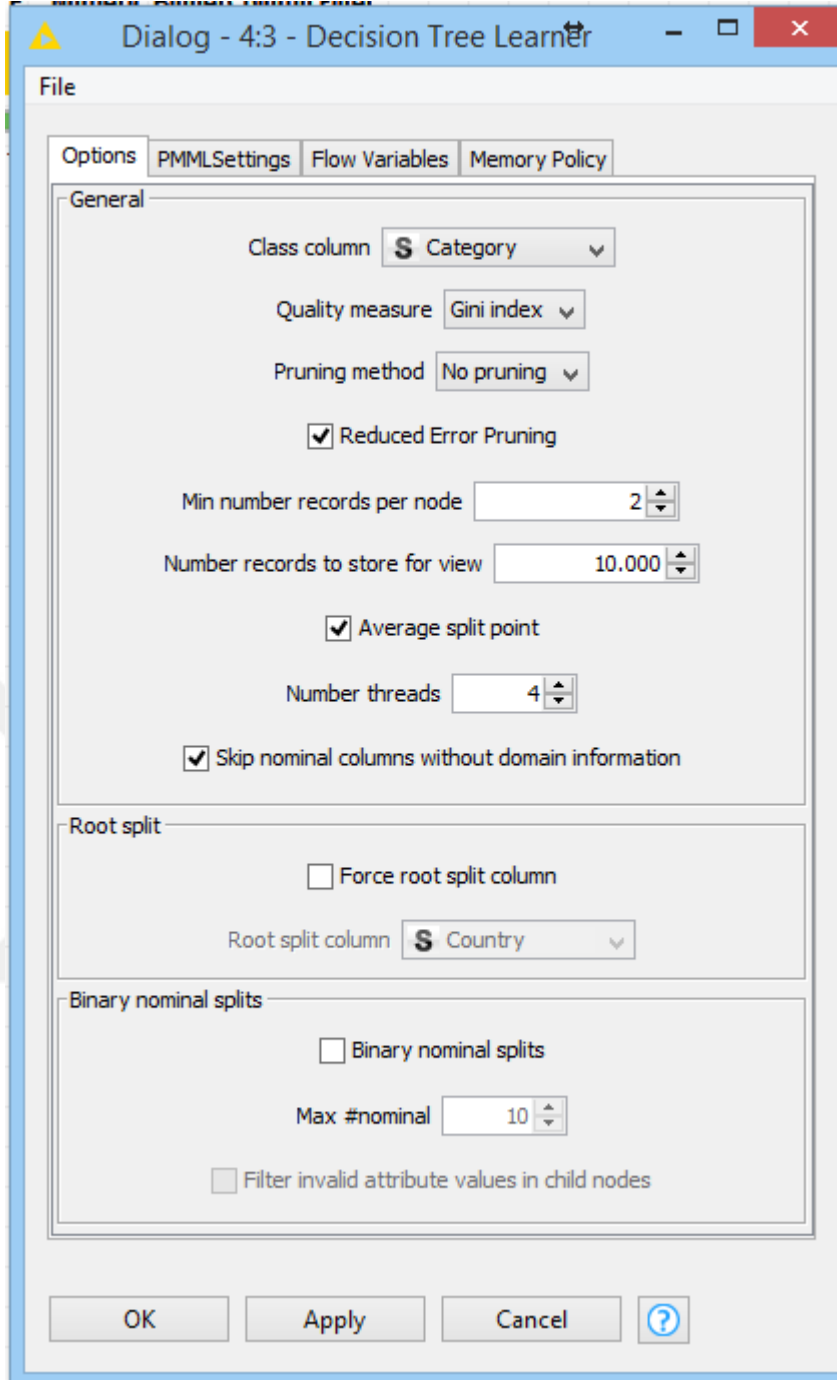
Uygun haldeki veri iki gruba ayrılarak öğrenme modülünün çalışması amaçlanmıştır.

Knime üzerindeki “Partitioning” nesnesi kullanılarak veri %80’i öğrenme %20’si tahmin grubu olmak üzere ikiye ayrı gruba ayrılmıştır.



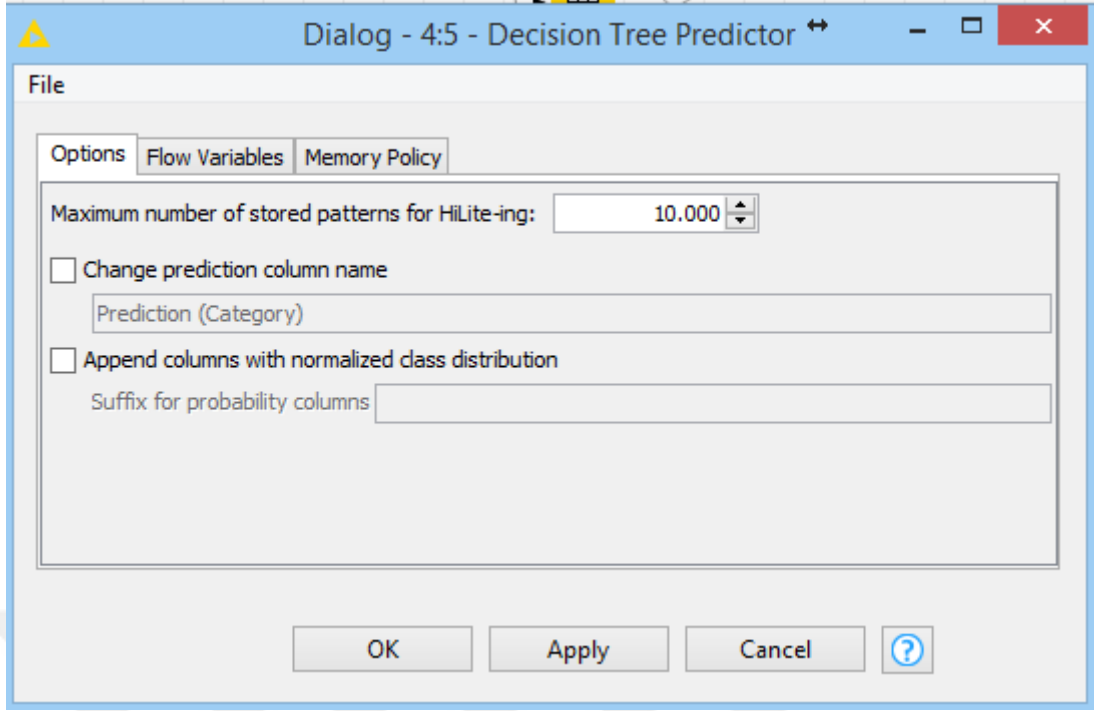
Şekil.25. Knime Partitioning Nesnesi Ayarları

%80’lik öğrenme kısmı “Decision Tree Learner” nesnesine bağlanmıştır.



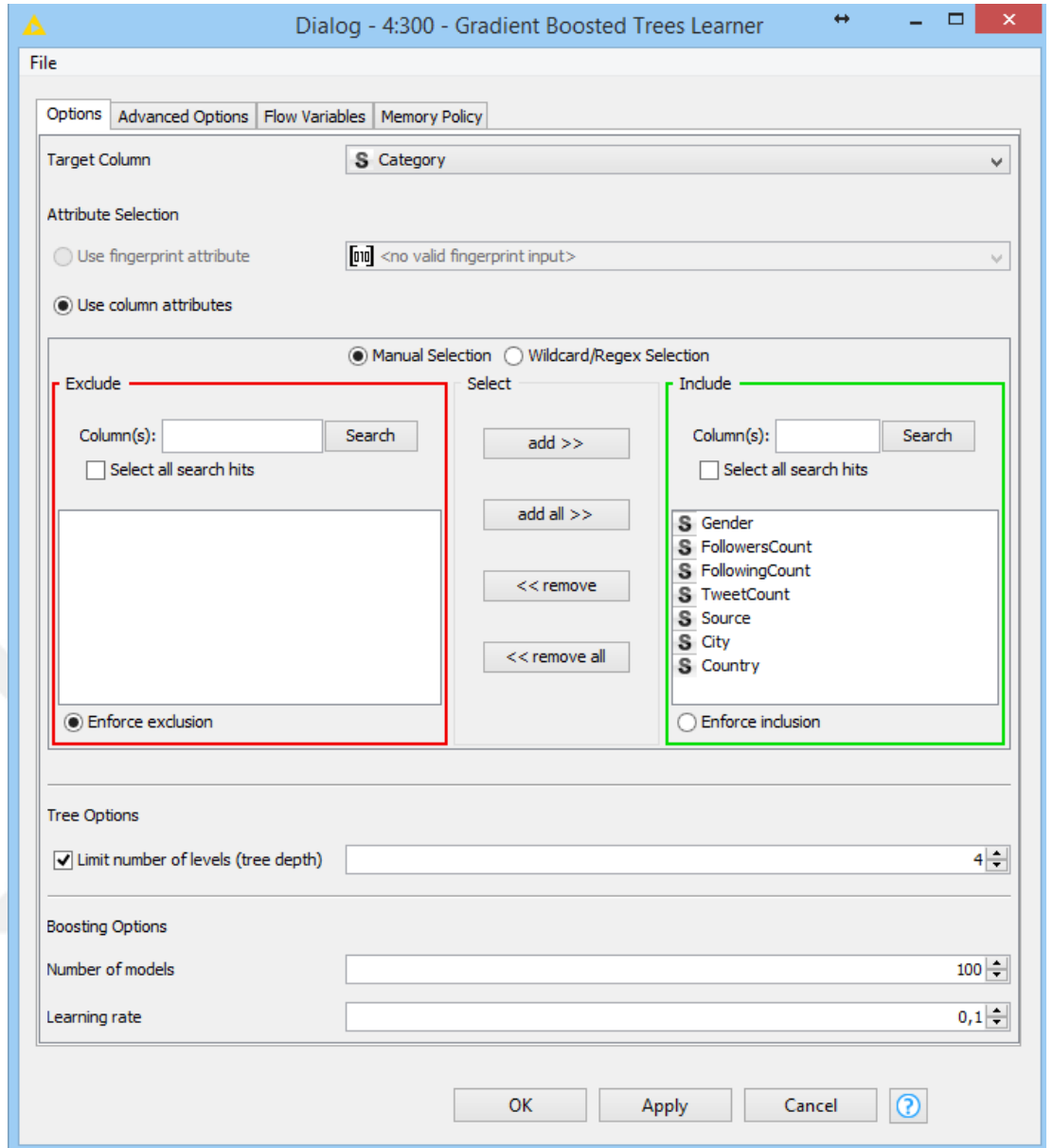
Şekil.26. Decision Tree Learner Nesnesi Ayarları

%20'lik tahmin grubu Decision Tree Predictor nesnesine bağlanmıştır



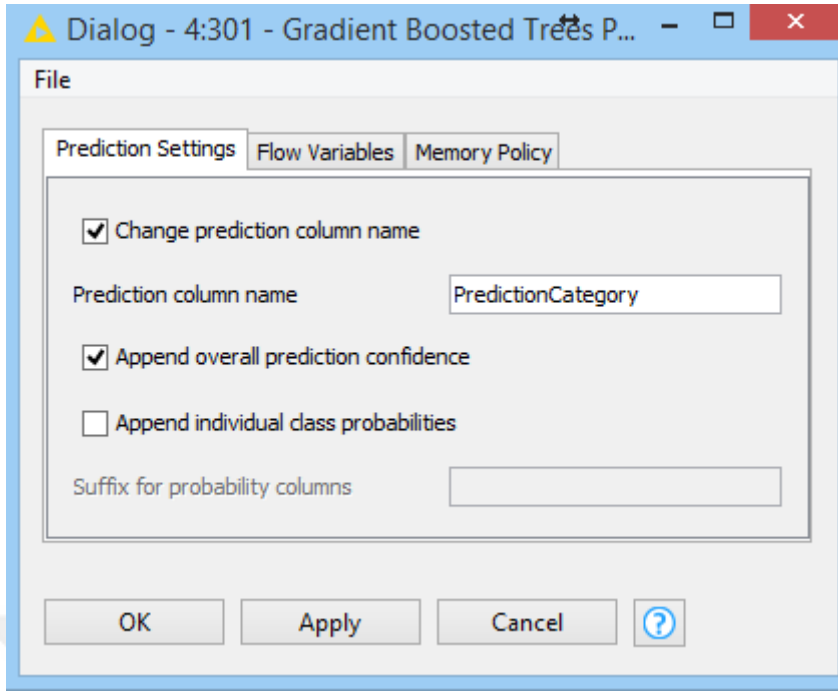
Şekil.27. Knime Decision Tree Predictor Nesnesi Ayarları

Benzer şekilde %80'lik öğrenme kısmı “Gradient Boosted Decision Tree Learner” nesnesine bağlanmıştır.

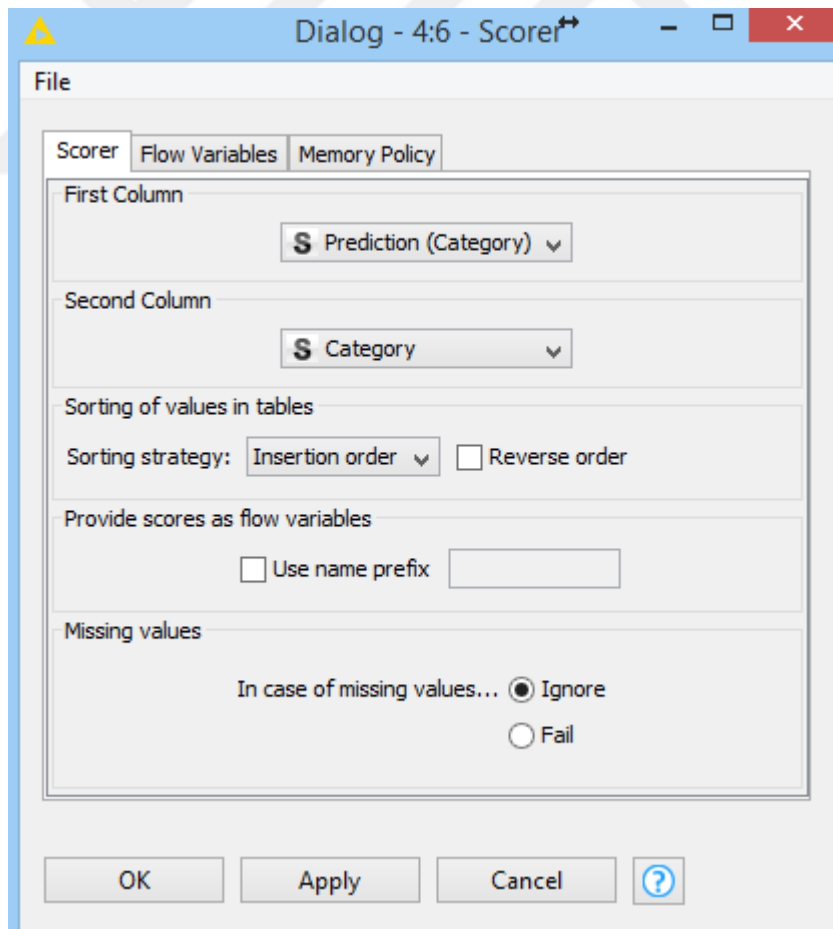


Şekil.28. Gradient Boosted Decision Tree Learner

%20'lik tahmin grubu Decision Tree Predictor nesnesine bağlanmıştır



Şekil.29. Gradient Boosted Decision Tree Predictor



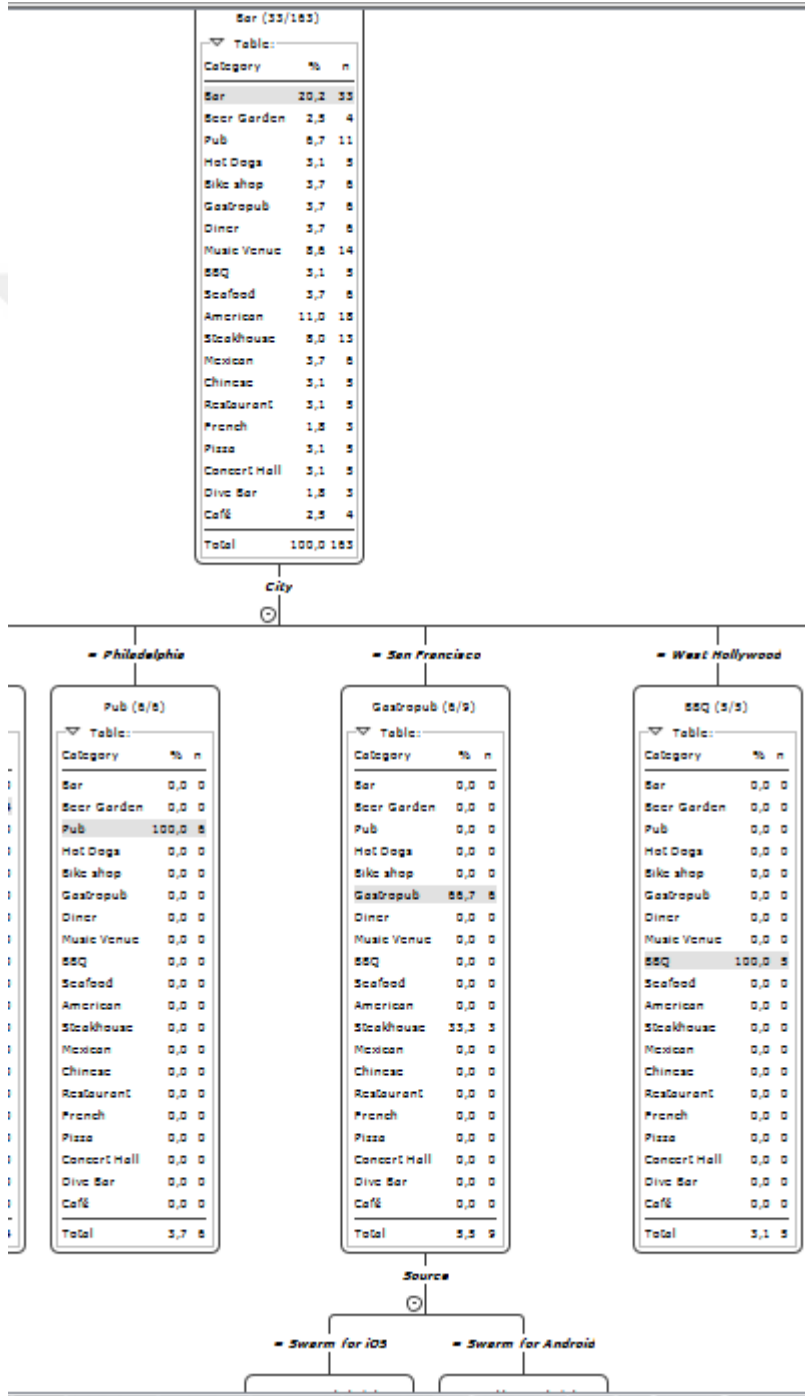
Şekil.30. Knime Scorer Nesnesi Ayarları

Decision Tree Predictor” ve “Gradient Boosted Trees Predictor” nesnelerinin sonuna “Scorer” nesneleri bağlanarak kategori ile tahmin edilen kategori değerleri karşılaştırılarak tahmin sonuçları alınmıştır.



5. BULGULAR

Karar ağacı ile yapılan tahminimde %38,6'lık bir başarı elde edilmiştir. Gradyan iyileştirmeli karar ağacı ile yapılan tahminde ise %40,4'lük bir başarı elde edilmiştir.



Şekil.31. Analiz Sonucu Karar Ağacı

Karar ağacı verisi incelendiğinde kişilerin mekân kategorileri tercihlerini etkileyen en önemli etmenin buldukları şehir olduğu tespit edilmiştir. Örneğin Chicago şehrinde attıkları tivit sayısı 10000'in üzerinde olan kullanıcıların en çok check-in yaptıkları mekân türü "Home" yani ev olduğu gözlemlenmiştir.

New York şehrinde bulunan kişilerin en çok tercih ettikleri mekân kategorisinin Bar olduğu gözlemlenirken, Boston şehrindekilerin en çok check-in yaptıkları mekân türü American'dır.



6. SONUÇ

Çalışmada hiç tanımadığımız kişilerin mekân paylaşımlarına Twitter ve Foursquare üzerinden kolaylıkla ulaşabileceğimiz görüldü. Ancak Foursquare ve Twitter API'lerinin kısıtları bu açık verinin sadece çok ufak bir kısmını kaydedebilmemize olanak sağlamaktadır. Kaydedilen verinin birbirinden kopuk olması kişilerin check-in'lerini bütün olarak takip etmemizi engellemiştir.

Çalışmada 726 adet kategori toplanmış filtreleme sonrası kalan 70 adet kategori tahmin programıyla değerlendirilmiştir.

Karar ağacı ile %38,6 'lık başarıya ulaşılırken Gradyan iyileştirmeli karar ağacıyla başarı oranı %40,4'e yükselmiştir. Böylelikle Gradyan iyileştirmeli karar ağacının, klasik karar ağacına göre ufak bir miktar daha sağlıklı sonuç verdiği görülmüştür.

%40,4'lük başarı oranı ile istenilen düzeye erişilememiş olsa da eldeki kategori miktarının 70 gibi yüksek olması ve mekân dağılımının belirli bir bölge ile kısıtlanmamasının tahmin işlemini zorlaştırması göz önüne alındığında sonuçlar umut vericidir.

İleriki çalışmalarda Twitter üzerinden gelen kısıtlı veriyi kullanmak yerine gönüllü kullanıcıların periyodik check-in'lerini incelemek elde edilen verideki bütünlüğü arttıracaktır. Alternatif olarak veri kümesine hava durumu, yerel saat, yerel özel günler ve promosyonlar gibi kullanıcı eğilimlerini değiştirecek kritik bilgilerin toplanması sonuçlardaki tutarlılıkları arttıracığı ön görülmektedir.

KAYNAKLAR

[1] G. B. Colombo, M. J. Chorley, M. J. Williams, S. M. Allen, R. M. Whitaker, "You are where you eat: Foursquare checkins as indicators of human mobility and behaviour," in 2012 IEEE International Conference on Pervasive Computing and Communications Workshops (PERCOM Workshops), (Lugano, Switzerland), 19-23 Mart 2012.

[2] Xuelian Long, Lei Jin and James Joshi "Understanding Venue Popularity in Foursquare" in 2013 9th International Conference Conference on Collaborative Computing: Networking, Applications and Worksharing (Collaboratecom), 9-13 Haziran 2013

[3] Martin J. Chorley, Gualtiero. B. Colombo, Stuart. M. Allen, Roger. M. Whitaker, "Visiting Patterns and Personality of Foursquare Users", in 2013 Third International Conference on Cloud and Green Computing (CGC), 30 Eylül - 2 Ekim 2013

[4] Ana Paula G. Ferreira, Thiago H. Silva, Antonio A. F. Loureiro, "Beyond Sights: Large Scale Study of Tourists' Behavior Using Foursquare Data" in 2015 IEEE International Conference on Data Mining Workshop (ICDMW), 14-17 Kasım 2015

[5] 4 Aralık 2016 <https://www.brandwatch.com/blog/44-twitter-stats-2016/>

[6] 4 Aralık 2016 <http://expandedramblings.com/index.php/by-the-numbers-interesting-foursquare-user-stats/>

[7] Twitter Developer Documentation "API Rate Limits" <https://dev.twitter.com/rest/public/rate-limiting> 1 Eylül 2016

[8] Foursquare API <https://developer.foursquare.com/> 12 Ekim 2016

[9] Representational state transfer https://en.wikipedia.org/wiki/Representational_state_transfer 12 Ekim 2016

[10] "NoSQL Relational Database Management System". <http://www.strozzi.it/> 1 Eylül 2016

[11] DocumentDB'ye giriş: Bir NoSQL JSON Veritabanı "DocumentDB verileri nasıl yönetir?" <https://azure.microsoft.com/tr-tr/documentation/articles/documentdb-introduction/> 1 Eylül 2016

[12] Comprehensive Study of Data Analytics Tools (RapidMiner, Weka, R tool, Knime) Information Technology MITM Indore, India 2016

[13] Gradient boosting https://en.wikipedia.org/wiki/Gradient_boosting 1 Eylül 2016

[14] Potential adjustments to Streaming API sample volumes <https://twittercommunity.com/t/potential-adjustments-to-streaming-api-sample-volumes/31628> 15 Şubat 2016



ÖZGEÇMİŞ

ÖZGÜR KAPLAN

E-Posta : ozgurkaplan@live.com

Adres Bilgileri : Türkiye - İstanbul(Asya) - Ataşehir - Küçükbakkalköy

Cep Telefonu : 90 (537) 792 55 88

Doğum Tarihi : 15.01.1988

EĞİTİM

(2006-2012) Yıldız Teknik Üniversitesi – Matematik Mühendisliği
(Bitirme Tezi: Nümerik Analizi Algoritmalarının Programlarının Optimizasyonu)

(2002-2006) Köy Hizmetleri Anadolu Lisesi

TECRÜBE

(2016 Temmuz-Halen) KoçSistem – Yazılım Geliştirme Lideri

(2015 Ocak – 2016 Haziran) KoçSistem – Yazılım Geliştirme Kıdemli Danışmanı

(2012 Şubat- 2015 Ocak) Kariyer.Net – Yazılım Geliştirme Uzmanı

(2011 Ocak- 2011 Temmuz) Veripark – Yazılım Geliştirme Uzmanı

YABANCI DİLLER

İngilizce (İleri)

TEKNİK YETKİNLİKLER

C#, Asp.Net (MVC and Web Forms) ,Entity Framework, WPF, WCF

Ms-Sql, Oracle, RavenDB, IndexedDB

JavaScript, AngularJs, JQuery

Apache Cordova

OpenXml

Swift

HOBİLER

Basketbol, seyahat

