

T.C.
İSTANBUL AYDIN ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ



No-SQL VERİ TABANLARI ÜZERİNDE
BİR METİN MADENCİLİĞİ UYGULAMASI

YÜKSEK LİSANS TEZİ
Gürkan AYDIN

Bilgisayar Mühendisliği Ana Bilim Dalı
Bilgisayar Mühendisliği Programı

OCAK 2017

TC.
İSTANBUL AYDIN ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ



No-SQL VERİ TABANLARI ÜZERİNDE
BİR METİN MADENCİLİĞİ UYGULAMASI

YÜKSEK LİSANS TEZİ

Gürkan AYDIN

Y1113.010024

Bilgisayar Mühendisliği Ana Bilim Dalı
Bilgisayar Mühendisliği Programı

Tez Danışmanı: Yrd. Doç. Dr. Metin ZONTUL

OCAK 2017



T.C.
İSTANBUL AYDIN ÜNİVERSİTESİ
FEN BİLİMLER ENSTİTÜSÜ MÜDÜRLÜĞÜ

Yüksek Lisans Tez Onay Belgesi

Enstitümüz Bilgisayar Mühendisliği Ana Bilim Dalı Bilgisayar Mühendisliği Tezli Yüksek Lisans Programı Y1113.010024 numaralı öğrencisi **Gürkan AYDIN**'ın “NO-SQL VERİTABANLARI ÜZERİNDE BİR METİN MADENCİLİĞİ UYGULAMASI” adlı tez çalışması Enstitümüz Yönetim Kurulunun 19.01.2017 tarih ve 2017/02 sayılı kararıyla oluşturulan jüri tarafından ile Tezli Yüksek Lisans tezi olarak edilmiştir.

Öğretim Üyesi Adı Soyadı

İmzası

Tez Savunma Tarihi : 26.01.2017

1) Tez Danışmanı: Yrd. Doç. Dr. Metin ZONTUL

.....
Metin Zontul

2) Jüri Üyesi : Prof. Dr. Ali GÜNEŞ

.....
Ali Güneş

3) Jüri Üyesi : Yrd. Doç. Dr. Ferdi SÖNMEZ

.....
Ferdi Sönmez

Not: Öğrencinin Tez savunmasında **Başarılı** olması halinde bu form **imzalanacaktır**. Aksi halde geçersizdir.

YEMİN METNİ

Yüksek Lisans tezi olarak sunduğum “**No-Sql Veri Tabanları Üzerinde Bir Metin Madenciliği Uygulaması**” adlı çalışmanın, tezin proje safhasından sonuçlanmasına kadarki bütün süreçlerde bilimsel ahlak ve geleneklere aykırı düşecek bir yardıma başvurulmaksızın yazıldığını ve yararlandığım eserlerin Bibliyografya’da gösterilenlerden oluştuğunu, bunlara atıf yapılarak yararlanılmış olduğunu belirtir ve onurumla beyan ederim. (28/12/2016)

Gürkan AYDIN
İmza

Bütünün parçaları, Onur 'uma, Efe 'me, Anneme, Babama, Kardeşime ve eşim Eylem 'e

ÖNSÖZ

Bizimle beraber hareket eden iletişim aracımız olan telefonlar Alexander Graham Bell 'in tasarımının çok ötesine gitmiştir. Akıllı telefonlar ve GSM şebekelerinin gelişmesi ile kişisel ofis merkezine ya da kişisel eğlence merkezine dönüşmüş durumdadır. İnternet iletişim hızının artması ile telefonlarımız sosyal bir görevde üstlenmeye başlamıştır. Telefonlarımız hayatımızın vazgeçilmez bir parçası olarak ve evrilerek hayatlarına devam etmekte.

Otobüste, şehirlerarası yolculukta ya da her neredesiniz elinizde olan telefon ile maillerinize, gazetenize, şarkılarınıza, filmlerinize veya oyunlarınıza ulaşmak istediğinizde GSM şirketlerinin kapsama ve hizmet kalitesi devreye girmektedir.

Bu projenin amacı GSM şirketleri için oluşan şikâyetler üzerinden metin madenciliği ile bir kaniya varmaktır.

Tez 'imi oluşturma süreçlerindeki katkılarından dolayı Hocam Yrd. Doç. Dr. Metin ZONTUL 'a, bir bütünün parçaları, Onur 'uma, Efe 'me, anneme, babama ve kardeşime bu süreçte tüm katkı ve desteğini i esirgemeyen eşim Eylem KILIÇ AYDIN 'a sonsuz teşekkürlerimi sunarım.

OCAK 2017

Gürkan AYDIN



İÇİNDEKİLER

Sayfa

ÖNSÖZ.....	vii
İÇİNDEKİLER	ix
KISALTMALAR	xi
ÇİZELGE LİSTESİ	xiii
ŞEKİL LİSTESİ	xv
ÖZET.....	xvii
ABSTRACT.....	xix
1. GİRİŞ.....	1
2. NoSQL VERİ TABANI SİSTEMLERİ.....	5
3. MongoDB VERİ TABANI SİSTEMİ.....	9
4. VERİ MADENCİLİĞİ.....	15
4.1 Veri Madenciliği Modelleri.....	17
4.2 Veri Madenciliğinde Yöntemler.....	18
4.2.1 Sınıflandırma	18
4.2.2 Kümeleme	18
4.2.3 Birliktelik kuralları	19
5. METİN MADENCİLİĞİ NEDİR?.....	21
5.1 Oluşturma Süreçleri.....	22
5.1.1 Toplama	22
5.1.2 Ön İşleme.....	22
5.1.3 Madencilik	23
5.1.4 Raporlama ve değerlendirme	23
5.2 Hazırlık.....	23
5.2.2 Gövdeleme.....	24
5.2.3 Sözlük oluşturma	25
5.2.4 Sözdizimsel analiz	27
6. PROJE YAZILIM KATMANLARI	29
6.1 Projenin amacı.....	29
6.2 Projenin Mimarisi.....	29
6.2.1 DAL layer :	30
6.2.2 Business layer :	30
6.2.3 Entities layer :	31
6.2.4 Common layer :.....	34
6.2.5 Email manager layer :	35
6.2.6 GSM layer :.....	35
7. METİN MADENCİLİĞİ UYGULAMASI.....	39
7.1 Uygulama Amacı.....	39

7.2 Projenin Mimarisi	39
7.2.1 Windows forms application teknolojisi	40
7.3 Veritabanı Oluşturma Aşaması	41
7.3.1 NoSQL mongoDB veri vabanı oluşturma	41
7.4 Veri İşlemleri:	49
7.4.1 Veri Toplama	49
7.4.3 Gövdeleme yöntemi	53
7.4.4 Sözlük oluşturma	54
7.5 Algoritma uygulama	56
8. SONUÇ	63
KAYNAKLAR	65
İNTERNET KAYNAKLARI	67
ÖZGEÇMİŞ	69

KISALTMALAR

RDBMS	: Relational Database Management System
ACID	: Atomicity – Consistency – Isolation – Durability
CAP	: Consistency, Availability, Partition, Tolerance
RAM	: Read Aktif Memory
SQL	: Structured Query Language
DB	: Database
JSON	: JavaScript Object Notation

ÇİZELGE LİSTESİ

SAYFA

Çizelge 3.1 RDBMS MongoDB Karşılaştırılması.....	11
Çizelge 3.2 RDBMS ve MongoDB Denk Terimler.....	12
Çizelge 3.3 RDBMS Veri Örneği.....	12
Çizelge 5.1 Oluşturulan Sözlük Tablosu	26

ŞEKİL LİSTESİ

SAYFA

Şekil 3.1 JSON Nesne Örnekleri	13
Şekil 5.1 Sözlük Veri Tabanı Bilgisi Kelime Frekansı.....	25
Şekil 6.1 Veri Katmanı	30
Şekil 6.2 Business Katmanı	31
Şekil 6.3 Entities Layer.....	32
Şekil 6.4 GSMTextMiningCategory Sınıfı	33
Şekil 6.5 Words Sınıfı.....	34
Şekil 6.6 Common Katmanı.....	34
Şekil 6.7 EmailManager Katmanı.....	35
Şekil 6.8 GSM Katmanı.....	36
Şekil 6.9 Proje Çalışma Ekranı	37
Şekil 7.1 Proje Solution Görseli.....	40
Şekil 7.2 Windows Forms Application Oluşturma Örneği.....	41
Şekil 7.3 Veri Tabanı MongoDB’de Barınan Collections Görseli	42
Şekil 7.4 Veri Tabanı MongoDB Collection Örneği	43
Şekil 7.5 Veri Tabanı MongoDB Veri HTML Data Örneği.....	44
Şekil 7.6 References NuGet.....	45
Şekil 7.7 MongoDB C# Driver Ekleme.....	46
Şekil 7.8 MongoDB.Driver lisans sözleşme onayı	47
Şekil 7.9 References.....	48
Şekil 7.10 HtmlAgilityPack kütüphanesinin eklenmesi	49
Şekil 7.11 HtmlAgilityPack Ekleme Değişiklik Onayı	50
Şekil 7.12 GSM Projesinin Referans Listesi	51
Şekil 7.13 Html Read Metodu	52
Şekil 7.14 Veri Toplama Metodu.....	52
Şekil 7.15 Veri Insert Metodu.....	53
Şekil 7.16 Proje Sözlük Yapısı	55
Şekil 7.17 Algoritma Mimarisi	56
Şekil 7.18 Proje Algoritması.....	57
Şekil 7.19 Program Ara Yüzü.....	59
Şekil 7.20 Program Turkcell Sonuçları.....	60
Şekil 7.21 Program Türk Telekom Sonuçları	61
Şekil 7.22 Program Vodafone Sonuçları	61

NO-SQL VERİ TABANLARI ÜZERİNDE BİR METİN MADENCİLİĞİ UYGULAMASI

ÖZET

Metin madenciliği metni veri kaynağı olarak dikkate alan veri madenciliği çalışmasıdır. Metin madenciliği düzensiz metinlerden oluştuğundan veri madenciliği algoritmaları ile yaklaşmak mümkündür. Asıl fark düzensiz metinleri anlamlı bir düzene oturtmaktır. Bu sebeple belirli bir düzene yaklaştırılan metinler üzerinde veri madenciliği yapmak kolaylaşmaktadır. Zor olan ise kaynaklar toplayıp derlemek ve anlamlı sayısal ifadelere çevirebilmektir.

Projedeki amaç müşteri şikâyetlerini anlamlandırıp ölçümlemektir. Projede GSM firmaları dikkate alınmıştır. Sonraki süreçte ise hedef öğrenci yazılılarını okumaktır.

Öğrencilere sınavlarına uygulanacak metin madenciliği çalışması ise farklı bir uzmanlık gerektirmektedir. Sözlük yaklaşımında olduğu gibi her soru için kategori ve kategori altındaki ağırlıklı kelimeler bu uzmanlar tarafından belirlenmelidir. Doğru sözlük tablosu oluşturulduğunda yazılım hızla değerlendirmesini yapacaktır. Milyonlarca öğrencinin cevap kâğıdının okunduğunu düşünürsek yazılımın milyonlarca işlem karşısında yorgunluk ve dikkatsizlik gibi davranışlarda bulunması beklenemez.

Sonuç olarak milyonlarca iş yükünün altında bile metin madenciliği yaklaşım projesinin eşit, adil ve hızlı bir değerlendirme yapacağı aşikârdır.

Anahtar Kelimeler: Metin Madenciliği, Joker Yöntemi, Sözlük Yöntemi, Gövdeleme, MongoDB, NoSQL

ON NOSQL DATABASES TEXT MINING APPLICATION

ABSTRACT

Text mining is a data mining operation that considers text as a data source. Since text mining is composed of irregular texts, it is possible to approach it with data mining algorithms. The main difference is to place irregular text on a meaningful level. For this reason, it is easy to make data mining on a certain level of approximated text. The hard one is to collect resources and translate meaningful digital expressions. The purpose of the project is to understand and measure customer complaints. GSM companies are considered in the project. The next step is to read the target student's writings.

The study of text mining that will be applied to the students' exams requires different expertise. As with the dictionary approach, the weighted words under the category and category for each question must be determined by these experts. When the correct dictionary table is created, the software will make a quick evaluation. If we think that millions of learners read the answer paper, it can not be expected that the software will be found in behaviors like fatigue and carelessness in the face of millions of transactions. As a result, even under millions of workloads, the text mining approach project is likely to make an equitable, fair and rapid assessment.

Keywords : Text Mining, Joker Method, Dictionary Method, Mocking, MongoDB, NoSQL.

1. GİRİŞ

Mobil; Fransızcadan dilimize katılan bir terimdir. Fransızcada “hareketli, hareket eden” anlamına gelmektedir. Sıfat olarak hareketli, taşınabilir anlamına gelir (TDK, 2016). Zamanımızda ise Mobil terimi “Hareketli iletişimi” temsil etmektedir.

Bilgi çağında teknolojinin de ilerlemesiyle giyilebilir giysi ya da cihazlarda üretilmeye başlamıştır. Bununla beraber televizyonlar, telefonlar, saatler, dijital bileklikler, ayakkabılar dahi sanal dünyaya anlık olarak veri gönderebilmektedirler. Bu kadar teknolojik hareketlerde farklı cihazlarla farklı uygulamaların doğmasına sebep olmaktadır. Bu cihazların ortak değeri ise veridir. Tüm bu cihazlar sosyal paylaşım sitelerine otomatik olarak veri yüklemek suretiyle büyük verinin oluşmasına katkıda bulunmaktadır. Dolayısıyla, sosyal medyada yer alan herhangi bir konu üzerinde metin madenciliğinin uygulanması mümkün olabilmektedir.

Dünyadaki insanların yaklaşık yarısı internet kullanmaktadır(Digital in 2016, 2016). Bu insanlar ya mobil cihazlar üzerinden ya da online web siteleri üzerinde anlık veri girmekte ve internet üzerinde bir saniyelik zaman diliminde terabyte’lar düzeyinde büyük veriyi oluşturmaktadırlar. Oluşan bu veri üzerinde gizli bilginin mantıksal bir zaman diliminde açığa çıkartılabilmesi için akıllı analiz algoritmalarının kullanılması gerekmektedir. Bu algoritmalar veri madenciliği ya da metin madenciliği başlığı altında incelenebilirler.

Veri madenciliği, bilginin gizli olduğu veri kaynaklarından faydalı olan bilginin elde edilmesidir. Bu da; kümeleme, veri özetleme, analiz gibi yöntemleri kapsamaktadır.

Büyük veri dendiği zaman Veri Madenciliği akla gelmektedir. William Frawley ve Gregory Piatetsky-Shapiro tarafından “önceden bilinmeyen, fakat yarar sağlama potansiyeli yüksek olan verinin keşfedilmesi” ‘dir. M. Berry ve G. Linoff ise “Anlamlı kuralların ve örüntülerin bulunması için geniş veri yığınları üzerine yapılan keşif ve analiz işlemleridir” şeklinde tanımlamışlardır.

Veri madenciliği bilişim literatüründe; veri tabanlarında bilgi madenciliği (knowledge mining from databases), bilgi çıkarımı (knowledge extraction), veri ve örüntü analizi (data/pattern analysis), veri arkeolojisi gibi kavramlarla da açıklanmaya çalışılmıştır (ÖZCAN,2014).

Karar alma mekanizmalarının doğru tercihler yapması adına bu verilerin basitleştirilmesi, anlamlandırılması ya da görselleştirilmesi veri madenciliğinin kapsamına girmektedir. Veri madeni raporları doğrultusunda stratejik karar mekanizmaları bilimsel çıkarım yapabilir ya da bilimsel tercihlerde bulunabilirler.

Büyük veri, yapılandırılmış veya yapılandırılmamış veri şeklinde olabilir. Yapılandırılmış verilerde her satırdaki sütun sayısı sabitken yapılandırılmamış verilerde her bir satırda farklı sayıda sütun olabilir. Veri madenciliği genellikle yapılandırılmış verilerle çalışırken metin madenciliği ise yapılandırılmamış verilerle uğraşır.

Büyük veri üzerinde metin madenciliği uygulaması geliştirebilmek; Seçim, Aktarım, Barınma, İşlem ve Çıktı aşamalarından oluşmaktadır. Bu adımların gerçekleştirilmesi adına birçok uzmanlık alanına ihtiyaç duyulmaktadır:

Seçim : Akademik, sosyolojik, ticari ya da siyasal analizlerini dikkate aldığımızda içerik ve kapsam olarak birbirlerinden çok farklı alanları teşkil etmektedirler. Seçim; başvurulacak verilerin ihtiyaca bağlı olarak kategori ve alan belirlenmesi işlemi kapsamaktadır.

Aktarım : Seçim işleminden sonraki, verinin nasıl ve ne şekilde toplanma sürecinin gerçekleştiği aşamadır. Veriye ulaşma şekli ve barınma alanına depolama sürecini kapsar.

Barınma : Toplanan tüm verinin korunduğu alandır. Veriyi işleme sürecinin ihtiyaç duyduğu veriye erişim noktasıdır. Veriyi barındırma teknolojileri olarak SQL, Hadoop ya da NoSQL gibi yapıları örnek verebiliriz. Barındırma için bir sunucuya da ihtiyaç duyulacaktır.

İşlem : İşlem aşaması işlem tipini belirlemeden önce veriye ihtiyaç duyar. Veriye ulaşma şeklimiz ve verinin tipi uygulayacağımızın işlem sürecini doğrudan etkiler. Bu süreçte verilerin çeşitli algoritmalarla işlenmesi ve çıktı oluşturacak verilerin bu algoritmalar doğrultusunda elde edilmesini kapsayan bir süreçtir.

Çıktı : İşlem süreci sonrasında oluşan raporun görsel ve ya doküman olarak elde edilmesi. Karar mekanizmalarının karar alma süreçlerinde fikir sahibi olmalarını sağlayacak ve çıktı öncesini kapsayan tüm süreçlerin sonucunu barındıran süreçtir. Bu süreçle karar verici elde ettiği veriden yola çıkarak iş sürecine bağlı olarak stratejilerine karar verir.

Şirketlerin vizyon analizi yapılan metin madenciliği çalışmasında kaynak olarak 2010 Yılı Capital dergisinde bulunan Türkiye'nin En Beğenilen Şirketleri araştırması kaynak olarak kullanılmıştır. Ayrıca kaynak olarak yöneticilerle uygulanan online anket de kullanılmıştır. Çalışmada geliştirme teknolojileri olarak Visual Basic ve Statistica programı kullanılmıştır. MELEK çalışmasında "Tekil değer ayrışımı özellik çıkarımı" ve anlam endeksini oluşturabilmek için Statistica programının analizini kullanmıştır (MELEK, 2012).

Sosyal medya üzerinde duygu analizi ve pazar değerlendirme çalışmasında yazılımsal teknoloji için Botego firmasının destek aldıklarını beyan ederler. Sosyal medya da kaynak olarak Twitter seçilmiştir. Kendi ifadesi ile "Twitter verileri, yaratmış oldukları yazılımın alt yapısını oluşturan makine öğrenmesi temeline dayalı duygu analizi ile pozitif, negatif ve nötr olarak sınıflandırılmıştır." Şeklinde açıklamıştır. Metin madenciliği yaklaşımı olarak Kümeleme Yöntemi kullanılmıştır. İnternet, spor alış veriş, romantik, reklam ve diğerleri başlıkları altında kümeleri toplanmıştır (BEYHAN, 2014).

Sosyal medya üzerinde yapılan bir metin madenciliği çalışmasında, fast-food hizmeti veren ünlü zincir markaları müşteri şikâyetlerine göre hijyen, hız, lezzet ve memnuniyet kalitesi gibi kategorilere ayrılmıştır. İlgili çalışmada sözlük algoritması ve joker yaklaşımı kullanılarak firmalar hakkındaki şikâyetler kategorilere ayrılarak derecelendirilmiştir. Sözlükte yer alan kelimeler negatif (-1) veya pozitif (+1) olarak değerlendirilmiştir. Çalışmada geliştirilen yazılım, Entity Framework, WPF ve MSSQL teknolojileri kullanılarak C# programlama dili ile yazılmıştır (ARAVİ, 2014).

Müşteri memnuniyeti çalışmasında ise oluşturulan veri tabanına verilerin yazılan bir program aracılığıyla girilmesi sağlanmıştır. Metin madenciliği kaynağı olan veriler düzensiz yapıda olduğundan elle giriş sırasında kullanıcı tarafından mutlu ya da mutsuz diye ayrıştırılmaktadır. Oluşturulan girdi verisinin toplam kelimesi ile yargıya varılmıştır.

Ayrıca bir ikinci yöntem olarak Karşılık Matrisini kullanarak sonuca ulaşılmıştır (KUZUCU, 2015).

Sosyal medyada bulunan metin ve fotoğraf eşleşmesinin incelenmesi üzerine metin madenciliği yapılmıştır. Sosyal medyalarda bulunan fotoğraf ve fotoğrafla ilgili metnin bağının incelenmesi üzerine öncelikle fotoğrafların Pixel bazında her hücresi incelenerek renk ağırlığı tespit edilmiştir. Tespit sonrası veri analiz programı olarak Knime kullanılmıştır. Metin madenciliğinde ise Kümeleme Yöntemi kullanılmıştır. Program üzerinden elde edilen analizlerle renk uzayı ve metin madenciliğinin sonuçları birlikte yorumlanmıştır (DEMİREL, 2015).

Talep tanıma ve yönlendirme çalışmasında ise dilekçe tanıma ve anlamlandırma konusu üzerinde çalışılmıştır. Dilekçeleri önce optik karakter tanıma yöntemi ile karakter taraması sırasında eksik ya da yanlış kelime var ise burada da Levenshtein algoritmasından faydalanılarak oluşan veriler veri tabanına gönderilmektedir. Naive Bayes Multinomial kullanılarak dilekçe sınıflandırılması yapılmıştır. Sınıflandırılan dilekçeler veri tabanı üzerinde kayıt kişilere belge talep, ders kayıt, harç, sınav, ve staj gibi kategorilere ayrıştırılarak ilgililere aktarılması sağlanmıştır. (SANCAR, 2016).

Bu tez çalışmasında, sikayetvar.com üzerinden GSM firmaları hakkında alınan şikâyetlerin kategorilere ayrılarak derecelendirilmesi metin madenciliği ile yapılmıştır. Bu çalışmada da sözlük algoritması ve joker yaklaşımı kullanılmakla beraber sözlükte kullanılan kelimelere (0,1) aralığında ağırlıklar verilmiştir. Ayrıca, geliştirilen yazılımın performansını artırmak için class yapısı C# dilinde elle oluşturulmuş ve ilişkisel veri tabanı yerine NoSQL veri tabanı tercih edilmiştir. Oluşturulan yazılım mimarisi bulut mimarisi ile uyumlu olup ileride web servis şeklinde hizmet vermesi düşünülmektedir.

2. NoSQL VERİ TABANI SİSTEMLERİ

NoSQL ilişkisel (RDBMS) (Relational Database Management System) veri tabanlarına alternatif olarak çıkan, nesneyi dikkate alan SQL 'deki gibi belirli kolonlara ihtiyaç duymayan veri depolama mekaniğidir. NoSQL dendiğinde bilinen Hadoop, MongoDB, ElasticSearch vb. gibi birçok marka bulunmaktadır. Bu çalışmada, NoSQL veri tabanı olarak MongoDB tercih edilmiştir.

NoSQL yapılar ilişkisel veri tabanlarının performans sorunlarının yanında lisans ve işletme maliyetlerinin de karşısına çıkmıştır. NoSQL ACID garantisi vermeyen dağıtık veri depolarını dikkate alarak yola çıkmıştır (NoSQL, 2016).

Bir transaction oluşabilmesi için ACID ilkelerine uyması gerekir. ACID, Atomicity – Consistency – Isolation – Durability (parçacık – tutarlılık - izolasyon - dayanıklılık) anlamlarına gelen kelimelerin baş harfleridir (ACID, 2016):

Atomicity (Parçacık) : Transaction'ı oluşturan tüm işlem parçaları hataya düşmemelidir ve her bir işlemin başarılı olarak sonuçlanması gerekmektedir. Aksi halde başlatılan işlem geri alınır. "Ya hep Ya hiç" mantığına dayanır.

Atomicity fiziksel olaylar dahil olmak üzere elektrik kesintisi, çökme, hata benzeri durumlarda işlem gerçekleşmesini garanti eder. Dışardan bakıldığında (atom) bölünmez olmak ilkesi göze çarpar.

Consistency : Tutarlılık anlamına gelmektedir. Yazılımcının yazılım tarafında ki tüm işlemlerin tutarlılığını garanti edemez. Tutarlılığın garanti edilebilmesi için veri tabanında trigger, cascade, constraints ve ya bunların herhangi bir kombinasyonu için tanımlı SQL ("Structured Query Language", Yapılandırılmış Sorgu Dili) komutları için geçerlidir.

Isolation : Veritabanında oluşan Transaction -lar için eş zamanlı çalışma sürecinde birbirlerini etkilemeden çalışma esasıdır. Ek olarak kuyruk yönetimi ile bu çalışma süreci kontrol altına alınmaktadır.

Durability(Rollback) : Atomcitiy içerisinde bahsettiğimiz transaction işlemlerini konu almaktadır. Elektrik kesintisi, çökme ve ya hata gibi durumlarda oluşabilecek durumlar veri tabanındaki tüm verileri kaybetmemize sebep olabilir. SQL Transaction işlemlerinin ızlı olabilmesi için Caching modeli kullanılarak RAM 'den faydalanılmaktadır. Elektrik kesintisinde ise RAM 'de bulunan işlemlere erişme imkânı kalmayacağı gibi hata durumlarında RAM 'e erişim yönetimi imkânsız olacaktır.

Durability 'nin esas aldığı kelimeninde karşılığı olduğu gibi dayanıklılıktır. Hata durumlarına karşın RAM yönetimi yerine temp file ya da log yönetimi esas alınır.

Yukarıda belirtilen durumlar ilişkisel veri tabanı sistemlerinin sahip olduğu güçlü yönlerdir. Bu veri tabanı sistemleri bazı dezavantajlara da sahiptir.

Veri tabanında yer alan veriler büyüdükçe ve tabloların sayısı artarak ilişkilerinde bu oranda büyüdüğünü dikkate alırsak donanımsal ve yazılımsal maliyetler artmaktadır. Büyük verileri hızla yönete bilmek için daha güçlü, hızlı bileşenler ve bunlar üzerinde çalışacak veri tabanı yönetim yazılımlarının da lisanslama maliyetleri de aynı oranda büyümetedir.

NoSQL bu noktada alternatif olarak, nesne tabanlı ve file stream veri tabanı olarak ortaya çıkmıştır. Büyük veri tabanlarında ise Storage ve RAM artışları yeterli olmaktadır. Bununla beraber Open Source (açık kaynak) yazılımlarla desteklendiğinden lisan bedeli ortadan kalktığı gibi toplamda maliyetleri SQL 'e göre çok daha ucuzdur.

NoSQL 'in asıl hedefi büyük veri olduğundan performanslı olarak çalışabilmesi için dağıtık kurgu mimarisinde daha yüksek performans ve güvenirlilik sağlarlar. NoSQL sistemlerde “transaction” dikkate alınmamaktadır. Fakat özellikle MongoDB 3.2 versiyon ile Atomicity kurallarıyla doküman yazma işleminde destek vermektedir. Zaten dağıtık sistemlerde CAP (Consistency, Availability, Partition, Tolerance) teorisine göre o an için Tutarlılık, Müsaitlik ve Parçalanma payı kavramlarının tümünü birden sağlayamaz. Şöyle ki dağıtık bir sistemin her bir bünyesinin aynı dataya erişebilme, kaybolan paketlere rağmen verinin tamamını kaybetmeme özelliğine aynı anda sahip olamaz.

NoSQL sistemlerde tüm bu kavramlar yatayda büyüme ile performans artırılması hedeflenmiştir. Örnek olarak veriyi parçalayarak, farklı kopyalarını da dağıtık sistemin bünyesindeki farklı birimlere göndererek tutarlılık sağlanır. Paket kaybı yaşansa da verinin bütünlüğü bozulmaz. Bununla birlikte veri dengeli parçalara bölüdüğü için her bir parçaya düşen yükünde balansı sağlanmış olur.

Bu projede NoSQL örnekleri içerisinde MongoDB kullanılacaktır.

3. MongoDB VERİ TABANI SİSTEMİ

HuMONGOUS DB isminden türemektedir. MongoDB yüksek performans, yüksek kullanılabilirlik ve otomatik ölçeklendirme sağlayan bir açık kaynak belge veri tabanıdır (HuMONGOUS, 2016).

MongoDB yapı olarak dokümanları dikkate alır. MongoDB dokümanları JSON nesnelere benzemektedir, alanların değerleri diğer dokümanları ya da dizileri içerebilir (MongoDB, 2016).

Veri tabanı sistemlerinde nesne kullanmanın bazı avantajları vardır. Nesnelere birçok programlama dillerinde yerel veri türlerine karşılık gelmektedir. Birbirine bağımlılığı olmayan nesnelere MongoDB’de aynı Collection üzerinde tutulabilirler.

MongoDB ‘nin NoSQL veri tabanlarında öne çıkmasını sağlayan bazı özellikler aşağıda verilmiştir.

NoSQL veri yapılarının geneli KEY ile sorgulamaya izin vermektedir. Fakat MongoDB QUERY desteği ile veriye ulaşılmasını sağlamaktadır.

Sorgu oluşturulan alanlara performans artırımını oluşturması adına secondary index ‘e destek vermektedir.

Master-Slave Replication destekleri ile Master sunucu yazma işlemi yaparken Slave sunucu okuma işlemi yapmaktadır. Master sunucu herhangi bir sebeplerle cevap veremez durumda olursa Slave ‘lerden biri belirlenen koşullarda Master görevini üstlenmektedir (Replication, 2016).

MongoDB ‘nin en önemli özelliklerinden biri ise **Sharding** ‘dir. Replication yapısı ile beraber kullanılan bir sistemdir. Büyük veri ve yoğun şekilde veri yazma işlemi kullanılacaksa ve tek MongoDB işleme yetiştirmiyorsa, sistem ölçek olarak yetmiyorsa “ram ve ya disk gibi” devreye Shard Mimarisi girer. Shard ün ana bölümden oluşur. Shards, Mongos ve Config Server (Sharding, 2016).

Shards : Verinin Mongod işlemlerine bölündüğü alandır. Ayrı ayrı Mongod 'ların yönetilmesini Mongos sağlar. Her bir veri belirli Chunk 'lara bölünür. Aynı verinin bölünmüş Chunk 'ları farklı Shard 'larda barınır. Her bir Shard Replica Set olarak çalışmaktadır.

Mongos "Query Router" : İstekler direk olarak Shard 'larla bağ kuramazlar. Çünkü hangi veri hangi Shard 'da bilemezler. Bu işi üzerine Mongos alır. Bu sebepler Query Router denmektedir. İstek oluştuğunda Mongos devreye girerek ilgili Shard 'lara görev yönlendirmesi yapar ve dönen sonuçları birleştirerek oluşan isteğin cevabını iletir. Mongos veri tutmaz sadece işlem yönetimini üstlenir. İşleyişle ilgili süreci takip edebilmesi için işlem süreç verilerine ihtiyaç tutar.

Config Server: Mongos 'un görev sürecindeki işleme durumlarının veri olarak tutulduğu Mongod yapılandırma örneğidir. Kısaca Cluster 'ın tüm bilgisinin barındıran veri alanıdır. Her 200MB 'lık veri için 1KB 'lık Config Server alanı ayrılması önerilir. Mongos 'a gelen isteklerle ilgili hangi Shard 'da hangi Chunk barınır bilgisini bununla beraber serverların durum bilgisini barındırırlar.

MongoDB aynı zamanda **MapReduce** desteği sağlamaktadır. Oluşturacağımız analizleri görebilmenizi kolaylaştıracak bir sistemdir MapReduce ilişkisel veritabanı sistemleri ile (RDBMS) karşılaştırırsak, Map select ifadesine ve where kıstaslarını belirlemeye, Reduce ise having, sum, count, average gibi hesaplamalar yapmamıza benzemektedir (MapReduce, 2016).

MongoDB Java, PHP, C, Python, C++, Erlang, C# gibi dillere desteğini sunmaktadır.

Çizelge 3.1 RDBMS MongoDB Karşılaştırılması

RDBMS vs MongoDB		
	RDBMS	MongoDB
Veri Yapısı	Sabit	Düzensiz
Join	Dahil	İç içe yapı
Transaction	Dahil	Atomic operasyonlar
Ölçeklenebilme	Dikey	Yatay
SQL Dili	Dahil	API 'ler ile
Primary Key	Var	Var
Foreign Key Constraint	Var	Referans

Çizelge 3.1 'de RDMS ve MongoDB 'nin yetenekleri gösterilmektedir. MongoDB verilerini **Collection** içerisinde Document olarak tutar. Veri yapılarını dikkate alırsak SQL veri tabanları ile karşılaştırırsak Collection tabloya, Document ise satırlara karşılık gelir.

Çizelge 3.2 RDBMS ve MongoDB Denk Terimler

SQL Terms/Concepts	MongoDB Terms/Concepts
database	database
table	collection
row	document
column	field
İndex	index

Çizelge 3.2 'de RDMS ve MongoDB 'deki denk gelen terimler gösterilmektedir.

Çizelge 3.3 RDBMS Veri Örneği

Soyadi	Adi	Yas
AYDIN	Onur Ekin	16
AYDIN	Efe Çınar	9
AYDIN	Gürkan	42

Çizelge 3.3 'de RDBMS 'lerde yer alan veri tabanlarında ki tablolarda tüm kayıtlar aynı veri tipinde olmak zorundadır.

```
{
  "_id": ObjectId("124ega5c2b7d284dad101e4bc9"),
  "Soyadi": "AYDIN",
  "Adi": "Gürkan",
  "Yas": 42
},
{
  "_id": ObjectId("124efa8d2b7d284dad101e4bc8"),
  "Last Name": " AYDIN ",
  "First Name": "Onur Ekin",
  "Age": 16,
  "Adres": "Bulgurlu Mh. Karlidere Cd.",
  "Sehir": "İstanbul"
},
{
  "_id": ObjectId("124efa8d2b7d284dad101e4bc9"),
  "Last Name": " AYDIN",
  "First Name": "Efe Çınar",
  "Age": 9,
  "Adres": "Bulgurlu Mh. Karlidere Cd.",
  "Okul": "Faik Reşit Unat",
  "Sehir": "İstanbul"
}
```

Şekil 3.1 JSON Nesne Örnekleri

Şekil 3.1 'de MongoDB nesnesi olan bir Collection görünmektedir. Colleciton içerisinde ki her bir Document 'lar farklı alanlar içerebilir.

MongoDB'de her dokümanın benzersiz kimliklikleri olmak zorundadır. Benzersiz kimlik alanı varsayılan olarak bu alan adı **_id** tanımlanmıştır. Eklenen document 'a bir tekil anahtar atanmamışsa MongoDB ObjectId tipinde otomatik olarak benzersiz bir kimlik atar.

4. VERİ MADENCİLİĞİ

Veri “enformasyon”, sadece verinin kendisi bir anlam ifade etmez. Ancak bir içerisindeki bilgi keşfedildiğinde bir anlamı olacaktır. Enformasyon sözlükte “haber alma, haber verme, haberleşme” olarak açıklanırken, bilişimde ise bilgi işlemde kullanılan kurallara uyularak veriye çıkarılan anlam olarak tanımlanabilir. Analiz süreci sonunda bir anlam çıkarmak adına, kurgulanacak ya da ulaştırılması gereken bilgiye haiz olabilmek içinde ayrı bir uzmanlık ve enformasyon gereklidir. Stratejik amaçlara yönelik çıkarımlarda bulunmak için verinin analiz edilmesi gerekir. Veriye anlam yüklemek veri analizinin konusudur. “Bilgi” ise amaç doğrultusunda ihtiyaç duyulan bilgiye cevap oluşturmak için veriden elde edilen anlamlı bilgilerdir. Bu bağlamda bilgi ve veri arasında kuvvetli bir bağ vardır (DURMUŞ, 2005).

Veri Madenciliği büyük hacimli verilerin karar sürecimize yönelik çıkarımlarda bulunmamıza olanak sağlayacak modellere ve bağlantılara göre analiz sonuçlarının çıkarılmasıdır. Kaynak olarak bilgiyi veri tabanlarında tutmak ve bu veri tabanlarına günümüzde ulaşmak hayli kolaylaşmıştır. Bu kolaylık neticesinde de kaynaklarından amaca yönelik bilgileri çıkarmak ve gelecek tahminlerinde bulunmak için veri madenciliğine günümüzde daha çok ihtiyaç duyulmaktadır.

Veri madenciliği istatistiksel ve bellek tabanlı yöntemler diye ikiye ayrabiliriz: (PEKTAŞ, 2013)

İstatistiksel Yöntem: Veri parametrik modele sahiptir. Bu parametrelerin bağlamında regresyon, sınıflandırma, öbekleme, boyut azaltma, hipotez testi, varyans analizi, bağıntı kurma gibi teknikler istatistik biliminin de kullandığı yöntemlerdir. Bu esasla veri madenciliği bir istatistik uygulamasıdır.

Bellek Tabanlı Yöntem: 1950 'li yıllarda önerilen yöntem zamanın teknolojik yetersizliğinden dolayı kullanılamamıştır. Bellek tabanlı yönteme en yakın K komşu algoritmasıdır. Bellek kullanımına ihtiyaç duyan yapay sinir ağlarını da örnek verebiliriz.

Ham verilerin algoritmalar tarafından kaynak olarak kullanılması için ön çalışmaların yapılması gerekmektedir. Veri ön işleme; büyük verinin analize tabi tutulmadan önce istatistiksel olarak uygun hale getirilmesini amaçlamaktadır. Veri ön işleme, temizleme, birleştirme, dönüştürme ve indirgeme basamaklarından oluşmaktadır (ŞENTÜRK, 2006).

Bilişim ekipmanlarındaki gelişme ve maliyetlerin düşmesi sonucunda birçok veri tabanları teknikleri gelişmiştir. Veri madenciliği konusunda ilk olarak ihtiyaçlar belirlenmelidir. Sonrasında ise verilerin nasıl temin edileceği önem arz etmektedir. Veri kaynaklarına eklenecek verilerin ne şekilde olacaklarının da belirlenmesi gerekmektedir.

Süreç sonrasında ise verilerin temizlenmesi ve buna ek olarak verideki gürültünün de temizlenmesi gerekir. Veri sağlama sürecinde oluşacak veri ihtiyacımız kirli olmayan verilerin bütünüdür.

Bir veri madenciliği sistemi, aşağıdaki temel bileşenlere sahiptir:

1. Veri tabanı, veri ambarı ve diğer depolama teknikleri
2. Veri tabanı ya da Veri Ambarı Sunucusu
3. Veri Madenciliği Motoru
4. Örüntü Değerlendirme
5. Kullanıcı Ara yüzü.

Verilere kaynak olarak çıkış özeti, fatura, slip, köşe yazısı, web sayfası gibi benzer dokümanlar ya da sayısal verilerden oluşabilir. Bu durumda veri madenciliği yapılandırılmış veriler ile işlem yaparken yapılandırılmamış veriler üzerinde çalışan alana metin madenciliği diyebiliriz.

Bilgi toplumu; veri, veriye erişim ve bilgi iletişim teknolojilerini yoğun şekilde kullanmak olarak tanımlanmıştır. İnternet'in günümüzde yoğun şekilde kullanılması ve hatta cep telefonlarına kadar indirgenmesi tüketici davranışlarını değiştirirken tahmin edilebilir hale getirmiştir.

Online ticaret günümüzde internet üzerinde oluşan değeri gün geçtikçe artan bir olgudur. Böylelikle online sipariş ve pazarlama uygulamalarına teknolojinin gelişmesi ile beraber kullanıcı deneyimi uzmanlığı gibi alanlarda öne çıkmaktadır. Online pazarın büyümesi hacminin genişlemesi online verilerin analiz edilmesi ve içindeki bilginin keşfedilmesinde ki arz ihtiyacı teknolojinin gelişmesi ile doğru orantılı olarak büyümektedir. Günümüzde oluşan bu online ticaret büyük şirketlerin en temel gelir kaynaklarını oluşturmaktadır (CHUNG & YOUNG, 2003). Özellikle Google Adsense reklam yönetiminde kullanıcıların eğilimlerini tespit ederek hedefe dönük reklam sürecini yönetmektedirler. Tüm bu sürecin kapsamı da metin madenciliğinin konusudur

4.1 Veri Madenciliği Modelleri

Veri madenciliği modelleri ikiye ayrılmaktadır.

Tanımlayıcı Modeller: Karar verme mekanizmalarının amaç doğrultusunda hedeflenen desenlerin çıkarılması gerekmektedir. Geliri üç bin ve beş bin para birimi aralığında olan ve iki veya daha fazla arabası olan çocuklu aileler ile çocuğu olmayan ve geliri bin beş yüz ve üç bin para birimi aralığından düşük olan ailelerin satın alma tercihlerinin benzerliğinin belirlenmesi tanımlayıcı modellere bir örnektir (Silahtaroglu, 2008)

Tahmin Edici Modeller: Tahmin edici modeller tersine işleyen bir süreçtir. İstenen sonuca göre model geliştirilir. Böylelikle öğrenme modeli gibi elde edilen model tanınmayan verilerin üzerine uygulanması sağlanır (GÜRSOY, 2010).

4.2 Veri Madenciliğinde Yöntemler

Veri Madenciliği modelleri temel olarak şu şekildedir (MELEK, 2012):

Sınıflandırma

Kümeleme

Birliktelik Kuralları

4.2.1 Sınıflandırma

Sınıflandırma veriler içerisinde ki gizli desen ve ya modelleri belirlemede kullanılır. Verilerin sınıflandırılması için belirli bir süreç izlenir. Hızlılık açısından var olan verilerin bir kısmı ile sistemi eğitmek adına sınıflandırma kurallarının oluşturulması sağlanır. Bu çalışma ile sınıf sayısını ve sınıfların barındırdıkları örneklerin belirlenmesi sağlanmış olur.

4.2.2 Kümeleme

Kümelemenin sınıflandırmadan farkı denetimsiz olmasıdır. Hedef ise ilişkili verilerin alt kümelere gruplanmasıdır.

Bir okula ait öğrenci kayıt verileri kaynak olarak kullanıldığında öğrencilerin yaşları, cinsiyetleri, sınıfları, bağlı oldukları kulüpler ve benzeri karşılaştırılarak alt kümelerin oluşturulması hedeflenir. Oluşan çıktı ile öğrencilere bilginin aktarılması konusunda ya da eğitim tekniklerinin geliştirilmesi alanlarında kullanılabilir.

4.2.3 Birliktelik kuralları

Birliktelik kuralları, farklı veriler arasındaki ilişkinin belirlenmedir. Bu ilişkiler arasındaki benzerliklerin oranının tespit edilmedir. Böylelikle veriler arasındaki bağıntı modeli de belirlenmiş olur. Öğrenci cinsiyet ve ders arasındaki bağıntı analizleriyle parametreler arasındaki benzerlikler belirlenerek cinsiyet ve eğitim kurgusu oluşturulabilir. Benzeri bağıntılar öğrencilerin davranışlarının tespiti ile eğitim modelinin geliştirilmesinde kullanılabilir.

Bir eğitim kurumundaki öğrencileri ele alırsak erkek öğrencilerin Türkçe dersi ile ilgili iken Müzik dersi ile ilgilendiğinde başarısının arttığı gözlenirse eğitim modelinin bu modelle yenilenmesi sağlanabilir. Süreç her seferinde ele alınarak çıktılar analiz edilip modelin dinamik bir şekilde güncellenmesi sağlanabilir. Bu da eğitim kurumunun başarısında ve tercih edilmesinde artış sağlayacaktır.

5. METİN MADENCİLİĞİ NEDİR?

Metin madenciliği, metin kaynaklarını girdi olarak kullanır ve bu metinler üzerinden anlamlı çıktılar elde etmeyi amaçlar. Metin madenciliğine kaynak olarak internet çevrimiçi ulaşılması kolay bir veri kaynağıdır. Veri tabanına aktarımı zor olan lokal veri ya da kağıt üzerinde bulunabilirler. Örneğin fatura, hesap belgesi, mektup, mail, epikriz gibi. Bu dokümanları toplamak, derlemek ve veri tabanına doğru bir şekilde aktarmak çevrimiçi sistemlere göre daha fazla zaman alabilirler.

Metin madenciliğinde amaca hizmet eden algoritmalar seçilerek veri tabanı incelemesi yapılır. Günümüzde internet kullanımının yaygınlaşmasıyla beraber bireylerinde sosyal mecralar üzerinden kişisel paylaşımları oldukça artmıştır. Buda internet dünyasını metin madenciliğinin merkezine oturtmaktadır.

Metin madenciliği, biçimi olmayan ifadelerin içinden bilgiler çıkarılması ve yapılandırılmamış verilerin biçimlendirilmesi sürecidir. Metin madenciliği dört adımdan oluşmaktadır (OĞZULAR, 2011);

1. Metin madeni oluşturma
2. Metin işleme
3. Veri Analizi
4. Değerlendirme ve Raporlama

Metin madenciliğinden birçok alanda yararlanılmaktadır. Bunlardan en önemlilerinden bir tanesi de Tıp 'tır. Tıp 'ta ki verilerin genel olarak serbest metin biçiminde olması hasta ile ilgili önemli bilgilerin gözden kaçırılmasına, bilgiye erişimin zorlaşmasına sebep olmaktadır. Sağlık sistemin başarısının, klinik dokümantasyonun serbest metin biçiminde oluşmasından dolayı bu tür sistemlere olan ihtiyacı ortaya çıkarmıştır. Yapılan klinik çalışmalar, araştırma raporları, hastane kayıtları, doktor notları ve ya faturalar tıptaki en önemli veri kaynaklarıdır (OĞUZ, 2009).

Metin madenciliği, son zamanlarda daha çok sanal dünyanın sosyal etkileşiminin popülerliğinden dolayı sanal ticaret konularında daha çok rastlanmaktadır. Ürün pazarlama ve tanımlama sosyal medyalar üzerinde daha rahat geliştirilip daha rahat incelenebilmektedir. Sosyal medyalarda sayısal ya da metin verilerini toplamak, derlemek ve analiz etme süreçleri daha kısa ve düşük maliyetli olduğundan pazarlama beceri analizleri bu mecralar üzerinde daha çok yapılmaktadır. Reklam ve tüketim dünyasının ilişkisini düşünüldüğünde ihtiyacın karşılığı daha net görülecektir.

Scimedirect 'de ise “advertise text mining” 1997 ve 2016 yılları arasında 5.000 ‘den fazla adet sonuç bulunmaktadır. Bu tezler ise birçok bilim dalında yer almaktadır (Scimedirect , 2016).

5.1 Oluşturma Süreçleri

Metin madenciliği süreçleri toplama, ön işleme, madencilik ve raporlama başlıkları altında toplanabilir (OĞUZLAR, 2011):

5.1.1 Toplama

Metin madenciliğine konu olan verinin toplama aşamasıdır. Metin toplama işlemleri işlemin nevine göre internette çevrimiçi faaliyet gösteren sitelerden ya da sosyal ağlardan elde edilebilir ve yahut hazır anlamsız metin kümeleri de bu işleme konu olabilir.

5.1.2 Ön İşleme

İşaretleme, gövdeleme, sözlük oluşturma ve gereksiz kelimeleri ayıklama sürecidir. Cümle içerisindeki kelimelerin ya da cümlenin kendisinin Yazım kurallarına uygunluğunu tespit eden aşamadır.

5.1.3 Madencilik

Veri madenciliğinde (Data Mining), düzenli bir yapıya göre (structured) oluşturulmuş örneğin veri küpü olarak saklanmış verileri dikkate almaktadır. Genellikle veri ambarlarındaki veriler üzerinde çalışılır. Metin madenciliğinde ise düzensiz metin verileri (Unstructured) dikkat noktasını oluşturmaktadır.

Farklı bir deyişle veri madenciliği, çok fazla miktarda bilginin depolandığı veri tabanlarından, amacımız ile ilgili olarak tahminler yapmamızı sağlar. Düzenli veri depolarına ulaşma ve bilgiyi kullanma işlemidir.

5.1.4 Raporlama ve değerlendirme

Veri depolarının işlenmesi sonucunda çıktının son kullanıcının anlayabileceği bilgiler ve görsellerle ifade şeklidir.

5.2 Hazırlık

Metin madenciliği öncesinde verilerin işlemlere uygun hale getirilmesi ve hazırlanması gerekmektedir. Ham veri üzerinden çalışılması sonuçlar üzerinde farklı etkiler oluşturacağı gibi sürecinde uzamasına sebep olacaktır. Ön hazırlık için ham veriyi temizlemekle işe başlanmalıdır. Veri tabanımızda bulunan ham verilerin madencilik işlemlerine uygun hale gelmesi ve optimum sonucun elde edilmesi adına madencilikten önce hazırlık işlemleri uygulanmalıdır.

Metin ile ilgili çalışmada atılacak ilk adım işaretleme işlemidir. Ham metin verilerinde bulunan bütün tümcelerin sağlıklı bir şekilde edilmesi işlemine işaretleme denir. Elimizdeki ham verinin daha kaliteli hale getirilmesi, veri boyutunun da küçültülerek işlem kabiliyetimizin artırılması adına metnin sadeleştirilmesi gerekmektedir. Bu sebeple her kelimeyi ayrıştırabilmemiz için toplam metni sadeleştirmek ve işaretlememiz gerekmektedir. Bu esnada metin içerisinde bulunan noktalama işaretleri, tek boşluk karakterinden fazla olan boşluklar “white space” ve diğer metine konu olmayan

karakterlerin temizlenmesi şekliyle işaretleme gerçekleşir. Böylelikle metin olarak geriye kelimeler ve kelimeler arasındaki birer boşluklar halinde ki sade metin kalır.

5.2.2 Gövdeleme

Veri kümesinin işaretlenmesidir. İşaretlemeler belirlendikten sonra bu işaretlemelerin tüm veri için uygun olacak halde standardize edilmesi gerekir. Gövdeleme aşaması her uygulamada farklı olabilir. Bu farklılıklar fayda sağlayacağı gibi işleme zarar da verebilir. Gövdeleme iki ana başlıktan oluşmaktadır (ÇELİKİYAY, 2010).

Joker Yöntemi: Türk dili sondan eklemeli yapıya sahip bir dildir. “okul” kelimesi ile “okuldan”, “okulu”, “okula”, ve “okulun” kelimeleri aynı kelime köküne farklı anlamlar yüklemektedir. Konunun temeli olan “okul” kelimesi bütünlüğünü korumaktadır (ÖNER, 1999), Eğer her ek almış kelimenin hallerini farklı kelimeler olarak kabul edilebilir. Kelimenin ek almış halleri işlem sürecinin uzatacağı gibi sözlük de barınacak kelime sayısının da artması ile başarı oranı düşecektir. Joker kelime, aynı köke sahip farklı ekler alan kelime öbeğidir. Bu farklı öbekleri bir çatı altında toplayan kelimeler joker kelimelerdir. Belirtilen örnek kelime “ okul* “ (kelimenin joker olduğunu * işaretinden anlıyoruz), “okul” kelimesinden sonra gelecek tüm ekler “okul” terimi olarak kabul görmüş olacaktır (OĞUZLAR, 2011).

Gövdeleme Yöntemi: Gövdeleme yönteminin kuralları daha katı kurallar içermektedir. Bu katı kurallar ile joker yönteminden ayrılır. joker yöntemine göre daha katı kurallardan oluşan bir yöntemdir. Gövdeleme kelimenin kökünü hedef almaktadır. Buda çakışmalar ile karşılaşmamıza sebep olacaktır. (ŞENTÜRK & GÜNDÜZ, 2014). Gövdeleme yöntemi sondan eklemeli olan Türkçe ‘de çekim eklerinin filtrelenerek yapım eklerinin korunması ile sağlanmaktadır.

5.2.3 Sözlük oluşturma

Şekil 5.1 'de görüldüğü üzere veri tabanımızda bulunan tüm kelimelerin dağılım frekansını tespit eden bir program yazıldı. Bu programın analizi sonucunda en çok kullanılan kelimeler tespit edildi. Bu kelimelerden kategorilere uygun olanlar belirlenerek aşağıdaki sözlük tablosu oluşturulmuştur. Sözlük oluşturulurken kalabalık edecek gereksiz kelimelere yer verilmemesi önem arz eder. Dikkat edilmemesi durumunda çıktının sağlayacağı faydanın değeri de düşecektir.

{ kelime = "TL", frekans = 10816 }
{ kelime = "Fatura", frekans = 4664 }
{ kelime = "Müşteri", frekans = 4350 }
{ kelime = "İnternet", frekans = 4174 }
{ kelime = "müşteri", frekans = 3653 }
{ kelime = "internet", frekans = 3604 }
{ kelime = "Haksız", frekans = 3539 }
{ kelime = "fatura", frekans = 2715 }
{ kelime = "önce", frekans = 2495 }
{ kelime = "de", frekans = 2361 }
{ kelime = "Hat", frekans = 2294 }
{ kelime = "Tarife", frekans = 2277 }
{ kelime = "tarihinde", frekans = 2250 }
{ kelime = "GB", frekans = 2103 }
{ kelime = "rağmen", frekans = 2006 }

Şekil 5.1 Sözlük Veri Tabanı Bilgisi Kelime Frekansı

Çizelge 5.1 Oluşturulan Sözlük Tablosu

Kapsama	Ağırlı	İnternet	Ağırlık	Fatura	Ağırlı	Hizmet	Ağırlı
Sokak	0.1	Yavaş	0.1	Tarife	0.1	Hizmet	0.15
Sinyal	0.1	Hız	0.1	Ücret	0.1	Alay	0.1
İletişim	0.1	Kısıtlam	0.05	Fiyat	0.1	Yapılmadı	0.15
Göbeği	0.3	Kopma	0.05	Aboneli	0.1	Ukala	0.1
Ev	0.1	İnternet	0.1	Fatura	0.05	Ulaşamıyoru	0.1
Çek*	0.3	GB	0.2	Faiş	0.05	Saygısızlık	0.05
Kapsam*	0.1	Kota	0.1	Kapama	0.1	Çağrı	0.1
Cayma	0.1	3G	0.1	Kredi	0.1	Umursamaz	0.05
Bölge	0.1	4.5G	0.2	TL	0.1	Kapat	0.05
				Ödeme	0.1	Davranış	0.05
				Paha	1.5	Sorun	0.05

Çizelge 5.1 Kategori altındaki kelimeler ve ağırlık puanları görülmektedir. Örnek olarak Kapsama kategorisi altındaki “çek*” kelimesini incelersek “çekim, çekmiyor, çekmez, çekmeyen” gibi veri tabanımızda bulunan kelimelerin Joker ifadesini oluşturmaktadır. Kapsam* Joker ifadesi ise “kapsam, kapsamıyor, kapsamamakta” gibi kelimelerin joker ifadesidir.

Bir dilin veya dillerin kelime haznesini (sözvarlığını), söyleyiş ve yazılış şekilleriyle veren, kelimenin kökünü esas alarak, bunların başka unsurlarla kurdukları sözleri ve anlamlarını, değişik kullanışlarını gösteren eserlere sözlük denir. Bir başka kaynağa göre sözlük, bir dilin veya dillerin kelime haznesini (sözvarlığını), söyleyiş ve yazılış şekilleriyle veren, sözcüğün kökünü esas alarak, bunların başka unsurlarla kurdukları sözleri ve anlamlarını, değişik kullanışlarını gösteren yazılı eserdir (BOZ, 2015).

Kelime sayısının niceliğinden çok niteliği önemlidir. Bu sebeple çok sayıda kelime tespit etmektense faydaya yakın kelimeleri belirlemek daha da önemlidir. Bu sebeple kelime sayısı az olan sözlüklerin işlem süreçlerinin kısa olduğu gibi başarı yüzdesinin de yüksek olması gözlemlenebilir.

5.2.4 Sözdizimsel analiz

Sözdizimsel analiz, cümle içinde geçen her kelimenin diğer kelimelere olan bağlantısı ve cümle içinde ki rolünü bulabilmektedir. Her dilin kendine göre birçok farklı Sözdizimsel analizi türü bulunmaktadır (ALTAN, 2016).

5.3.3 Çalışma Tablosu Oluşturulması

Ön işleme metotlarından sonra oluşturulan koleksiyonlar çalışma tablosuna çevrilmek üzere belirli algoritmalarından geçtikten sonra, çalışma tablosuna her bir satır bir doküman ve her sütun bir özelliğe denk gelecek şekilde yerleştirilir.

Bu noktadan sonra yapılması gereken işlem belirlenen kelimelerin doküman içerisinde bulunup bulunmadığını incelemektir. Her kelimenin karşısında bulunup bulunmadığına bağlı olarak bir veya sıfır şeklinde belirtilmelidir. Bu yöntemden farklı olarak kelimelerin karşısına kelimenin görülme sayısı da yazılabilir. Üçüncü bir yöntem olarak ta kelimelerin görülme sayısı için bir eşik konulabilir. Bu kelimenin bir değer ifade etmesi için bu eşiği aşması beklenebilir. Bu eşiği aşamayan kelimeler ise sıfır kabul edilir.

6. PROJE YAZILIM KATMANLARI

Proje gerekleřirken aynı zamanda da yeni teknolojilere adapte olunması hedeflenmiřtir. Visual Studio teknolojisi ile geliřtirilen projede katman mimarisi uygulanmıřtır. Ana gvde olarak proje bir Solution iinde bulunurken her katman bir Project 'e denk gelmektedir. Her katmanın kendi zg iřlem yeteneđi ve iř becerisi bulunmaktadır.

6.1 Projenin amacı

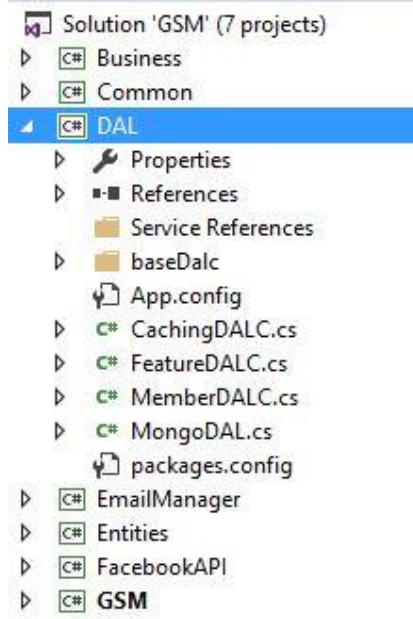
Metin madenciliđi projesinde dzensiz metinler olan mřteri řikyetleri kaynak olarak seilmiřtir. Kategorisi olmadıđından ve tanımlı bir yapıya bađlı olmayan řikyetler metin olarak dzensiz bir řekilde barınmaktadırlar ve uzman insan desteđiyle ayrıřtırılarak konular ele alınmaktadır. Projede ise her bir řikyetin anlam olarak ne ifade ettiđini zmleyeceđiz. Bu řikyetleri kategorilendirdiđimiz; kapsama, internet, cret ve hizmet bařlıkları altında gruplayarak daha ok hangi bařlık altında sorun yařandıđının kurum bazında tespiti yapılarak incelenecektir. Metin madenciliđinin szlk yaklařımından faydalanılarak szlkte bulunan kategoriler altında ki kelimelerin ađırlık puanları bazında her bir řikyet incelenecektir.

6.2 Projenin Mimarisi

Proje Visual Studio 2015 platformunda C# 6.0 yazılım dili Framework 4.6 ile geliřtirilmiřtir. Veri Tabanı olarak MongoDB kullanılmaktadır. Projede katmanlı mimari ve nesnel yaklařım esasları dikkate alınmıřtır. Nesnelliđin haricinde her bir katman grev ve yetenek olarak ayrıřtırılmıřtır.

6.2.1 DAL layer :

Şekil 6.1 'de görüldüğü üzere Data Access Layer data erişim katmanı projesidir. Veri tabanına sadece bu katmandan erişebilir. Her katmanın veri katmanına erişmeye yetkisi yoktur. Sadece iş katmanının sağlayıcıları erişebilmektedir. Projede veri tabanına erişim



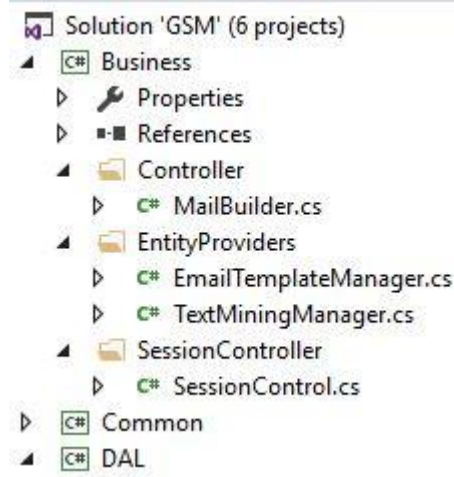
MongoDAL sınıfı üzerinden sağlanmaktadır.

Şekil 6.1 Veri Katmanı

6.2.2 Business layer :

Şekil 6.2 'de Business katman görseli yer almaktadır. Business katmanı Data katmanından aldığı verilerin kontrolünü ve yönetimlerini yapmaktadır. EntityProviders klasörü adından da anlaşılacağı gibi varlık sağlayıcıdır. Controller klasörü altında MailBuilder sınıfı görülmektedir. Örnek olarak burada mail tasarım olarak oluşturulur ve kimlere iletileceği ile bilgiler kontrol edilerek Mail katmanına iş olarak iletir.

Sağlayıcılar klasörü içerisindeki TextMiningManager sınıfı bir yönetim sınıfıdır. TextMiningManager yönetim sınıfı içerisindeki tüm işlem plana bağlanmıştır. Plan harici bir işlem gerçekleştirilemez. Önyüz katmanı bilgi ister ve plana bağlı kalan TextMiningManager sınıfı DAL ile haberleşerek veriyi elde edip önyüze iletir.



Şekil 6.2 Business Katmanı

6.2.3 Entities layer :

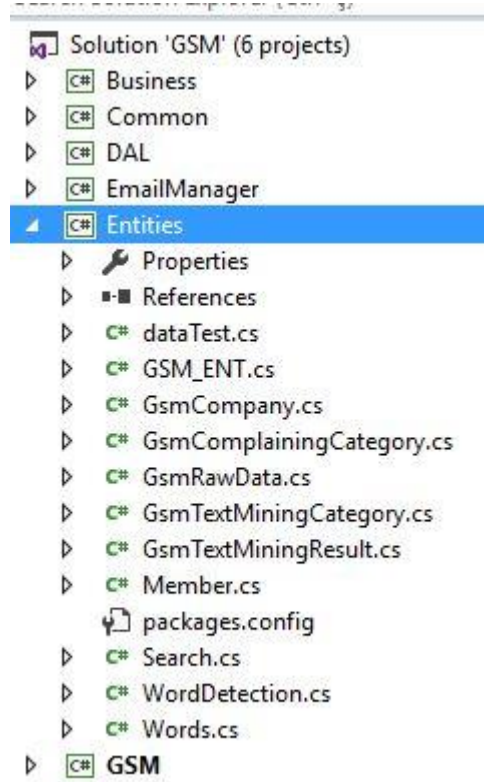
Şekil 6.3 'de görüldüğü gibi varlıklar katmanı projesi. Katmanların kullanacağı entity (varlık) sınıfları bu katmanda barınmaktadır. Projenin ihtiyacına göre belirlenip özellikleri ile beraber tüm sınıflar elle oluşturulmuştur.

Şekil 6.4 'de ise Entities katmanında var olan GSMTextMiningCategory sınıfı örneği görülmektedir. Metin madenciliğinde uygulanacak kategori bilgisi ve ağırlık puanları ile beraber kategoriler altındaki bulunan kelimeler de veri tabanında barınmaktadır. GSMTextMiningCategory sınıfı Id, CategoryName, WordList özelliklerinden oluşmaktadır.

Id :Projeye eklenen MongoDB kütüphanesinin Bson sınıfının ObjectId özelliğinden türetilmektedir. ObjectId 12 Byte 'lık hexadecimal (onaltılık sayı tabanı) değer barındırabilir. Bu değer elle verilebilir. Değer oluşturulmadığı takdirde MongoDB her nesneyi ObjectId tekil kimliği ile ayrıştıracağından benzersiz bir anahtar otomatik olarak da atar.

CategoryName :C# kütüphanelerinden System altında ki String sınıftan türetilmiştir. Projede metin madenciliğine konu olan kategorilerin isimlerini taşıyan özelliktir.

WordList : C# kütüphanelerinden System sınıfı altındaki Colletions.Generic.List sınıfı kabul olarak elle oluşturulan Words sınıfını tutmaktadır. Birbirinden farklı Words sınıflarını dizi gibi liste halinde tek yapıda barındırır. Words ise yine elle oluşturulan bir sınıftır. Şekil 6.5 'de görüldüğü gibi Words sınıfı ise string sınıfindan Name ve system altındaki double sınıfindan türeyen WeightScore özelliklerinden oluşmaktadır. WeightScore özelliği 0,1 ağırlığında sayılar ile tanımlandığından ondalıklı sayıları karşılayacak olan Double sınıfınsan türemiştir.



Şekil 6.3 Entities Layer


```

using MongoDB.Bson;
using MongoDB.Bson.Serialization.Attributes;
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace Entities
{
    7 references
    public class GsmTextMiningCategory
    {
        0 references
        [BsonId]
        public ObjectId Id { get; set; }
        1 reference
        public string CategoryName { get; set; }
        1 reference
        public List<Words> WordList { get; set; }
    }
}

```

Şekil 6.4 GSMTextMiningCategory Sınıfı

```

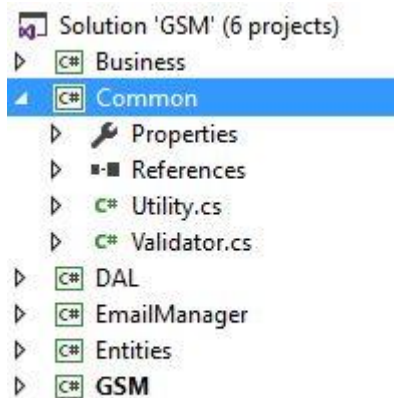
namespace Entities
{
    1 reference
    public class Words
    {
        1 reference
        public string Name { get; set; }
        1 reference
        public double WeightScore { get; set; }
    }
}

```

Şekil 6.5 Words Sınıfı

6.2.4 Common layer :

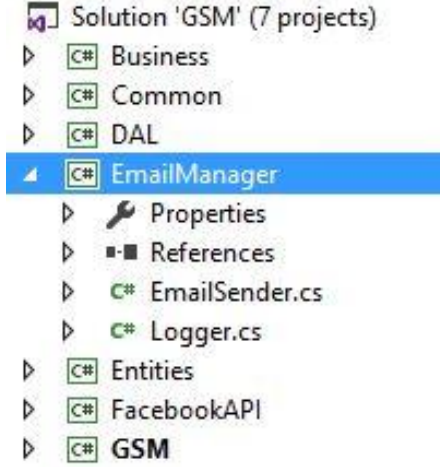
Şekil 6.6 'da görülen Common katmanı projesidir. Solution içerisinde bulunan proje katmanlarının kullanacağı ya da yapılması gereken basit işlemlerin metotları bu katmanda barınmaktadır. Metin madenciliğinde kullanılan Stop Word ve White Space temizleme işlemi bu katmanda gerçekleştirilir.



Şekil 6.6 Common Katmanı

6.2.5 Email manager layer :

Email Manager katmanı Şekil 6.7 'de görüldüğü gibi email yönetim katmanı projesidir. Email ile haberleşme bu katman üzerinden gerçekleşir. Kuyruk ve mail kategori yönetim sınıfları bu katmanda barınmaktadır. Proje sonucunun ilgili kişilere mail yolu ile iletilmesi sağlanır.

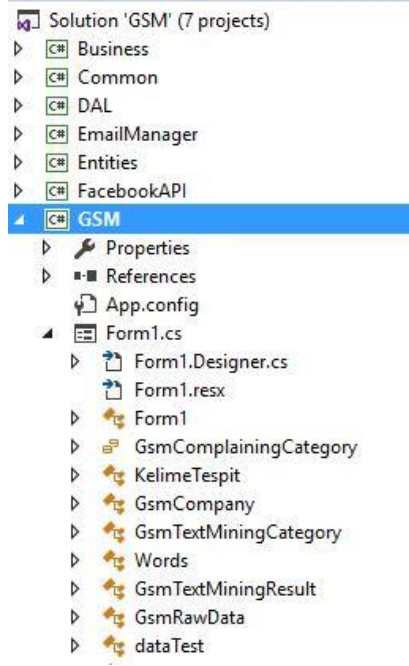


Şekil 6.7 EmailManager Katmanı

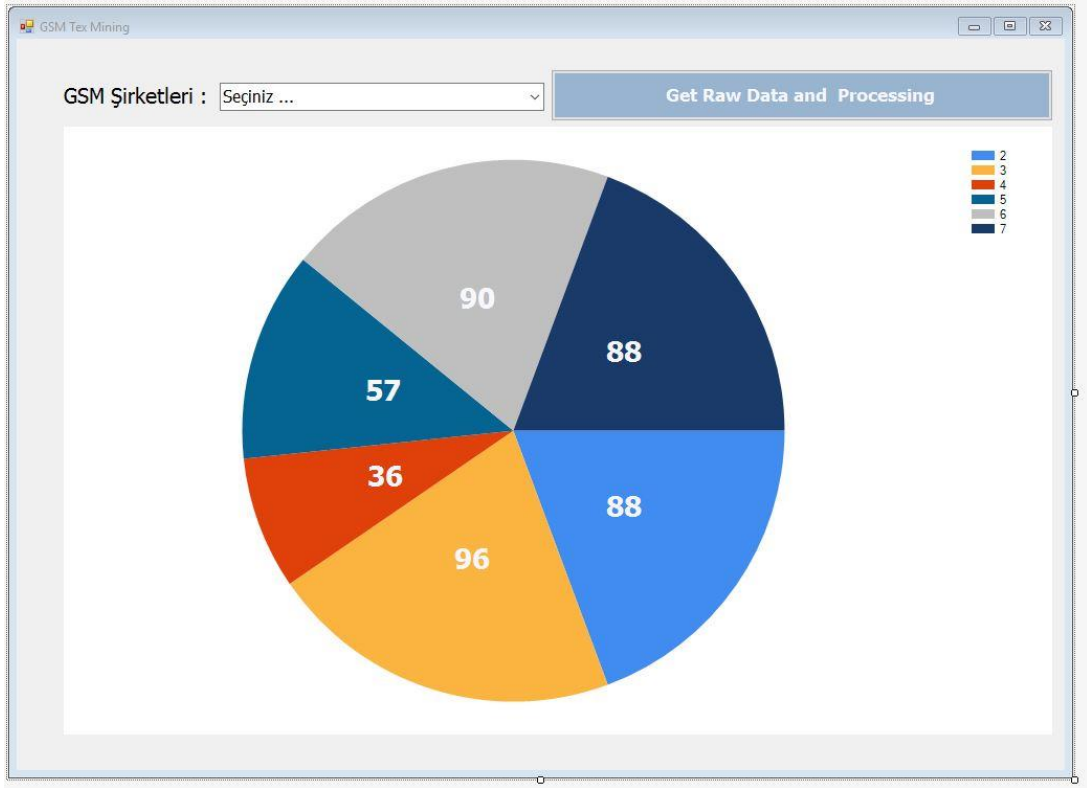
6.2.6 GSM layer :

Şekil 6.8 DE görülen önyüz katmanı projesidir. Girdi ve çıktıların görselleştiği katmandır. C# Windows Forms Application teknolojisi ile geliştirilmiştir.

Şekil 6.9 'da görülen projenin çalışma ekranıdır. Get Raw Data and Processing yazan butona basıldığında GSM Şirketleri seçeneğinden seçilen firma ismine göre veriler veri tabanından ham olarak çekilir sonrasında metin madenciliği uygulanarak işlem sonucunun analiz ekranının da görselleşmesi sağlanır.



Şekil 6.8 GSM Katmanı



Şekil 6.9 Proje Çalışma Ekranı

7. METİN MADENCİLİĞİ UYGULAMASI

7.1 Uygulama Amacı

Uygulamanın amacı GSM iletişim hizmeti veren firmaların aldığı şikâyetler üzerinden kapsama, internet, ücret ve hizmet gibi kategorilerin hangisinde daha çok şikâyet aldığını metin madenciliği yöntemlerini kullanarak tespit etmek ve raporlamak.

7.2 Projenin Mimarisi

Metin madenciliği uygulaması Visual Studio 2015 kullanılarak C# 6.0 yazılım dili altyapısı ile geliştirilmiştir. Projemiz Business, Common, DAL, EmailManager, Entities, FacebookAPI, GSM adında katman mimarisini kullanan yedi adet projeden oluşmaktadır.

Layer : Veri erişim katmanıdır. Haricinde hiçbir projeden veriye erişim sağlanmaz.

Business Layer : İş uygulama katmanıdır. Veriye erişim sağlayacak katmanların yetki kontrolleri, verilerin dönüştürülme işlemleri ve benzeri tüm iş süreci uygulamalar bu katmanda barınır.

Common Layer : Tüm katmanların eriştiği bir ortak alandır. Common katmanında oturum kontrolü ya da validasyon gibi işlemlere ihtiyaç duyan tüm katmanlar erişerek nesnel yaklaşım gereği bu katmanı kullanırlar.

EmailManager Layer : Email yönetim katmanıdır. Business Layer bu katmana bağlanarak yetki ve posta düzen işlemlerini yaptıktan sonra raporların ilgili kişilere iletimini sağlamak adına EmailManager katmanını kullanır.

Entities Layer : Varlık katmanı. Kullanacağımız objelerin oluşturulduğu ve barındığı katmandır. Business, Common ve GSM katmanları yoğun olarak buradaki class'lar ile çalışırlar.

GSM Layer : Frontend katmanıdır. Windows Forms Application teknolojisi kullanılarak geliştirilmiştir. Analiz görselinin haricinde metin madenciliği uygulamasının algoritması da bu katmanda barınmaktadır.



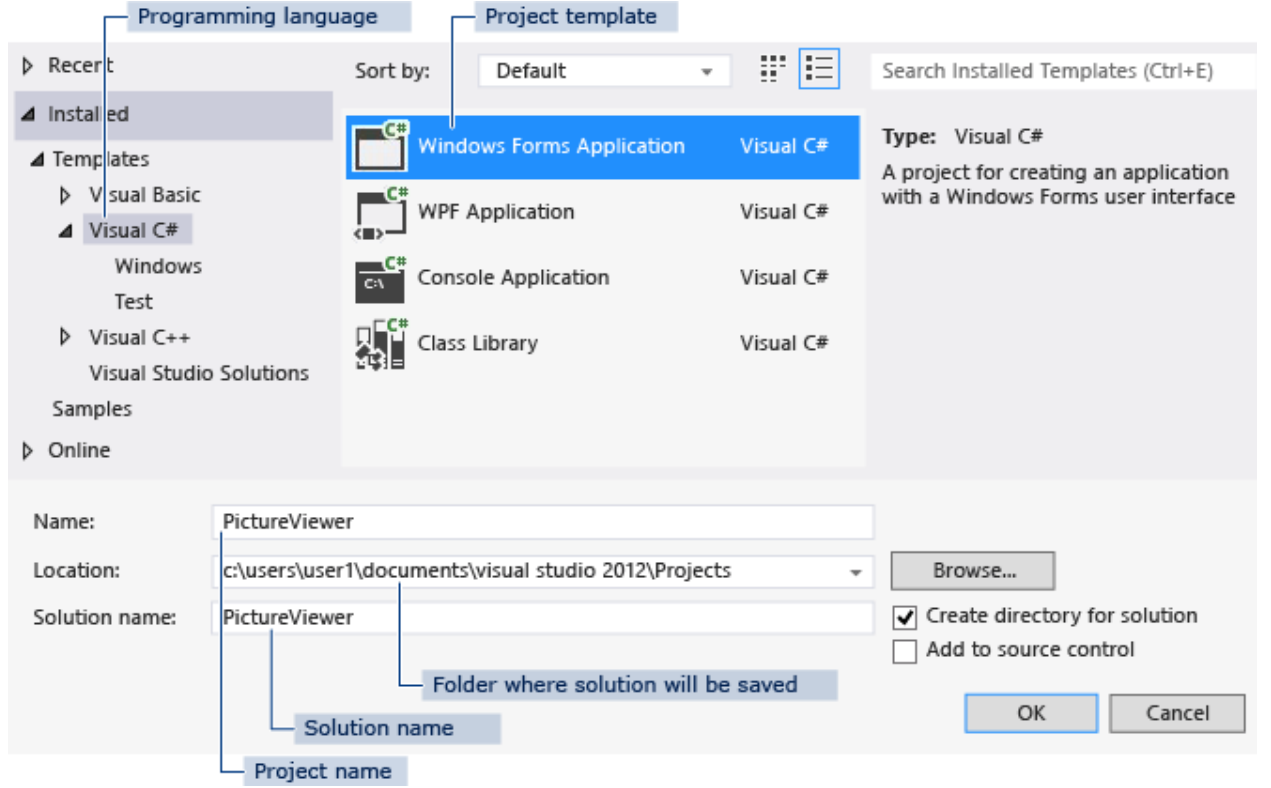
Şekil 7.1 Proje Solution Görseli

Proje Solution genel görünümü şekil 7.1 'de gösterilmiştir.

7.2.1 Windows forms application teknolojisi

Windows Forms GUI grafiksel kullanıcı arabirimi sayesinde yazılımcıları metin tabanlı programlama araçlarından kurtaran bir teknolojidir. Kütüphaneleri ile geliştiricilere görsel anlamda üst düzeyde kolaylık sağlamaktadır. Geliştiricinin kendi başına oluşturacağı görsel nesnelere Windows Forms teknolojisi ile kütüphanelerinde hazır gelmektedir. Bize kalan ise nesnelere özelliklerini ve metodlarını tanımdır (Windows Forms, 2016). Visual Studio 2015 için Şekil 7.2 'de Windows Forms Application oluşturabilecek ekran görseli görülmektedir.

Microsoft görselliğinde öneminin farkına vardığında görsel birimlere daha da önem vererek farklı alanlarda ataklara kalkmıştır. Bu hareketle yola çıkarak Silverlight 04-09-2007 tarihinde ilk sürümü ile yazılım dünyasına büyük bir yankı ile girmiştir. Beklenen talebi görmediği gibi dünya çapında ses getiren bir proje ile Silverlight terimleri bir arada duyulmamıştır.



Şekil 7.2 Windows Forms Application Oluşturma Örneği

7.3 Veritabanı Oluşturma Aşaması

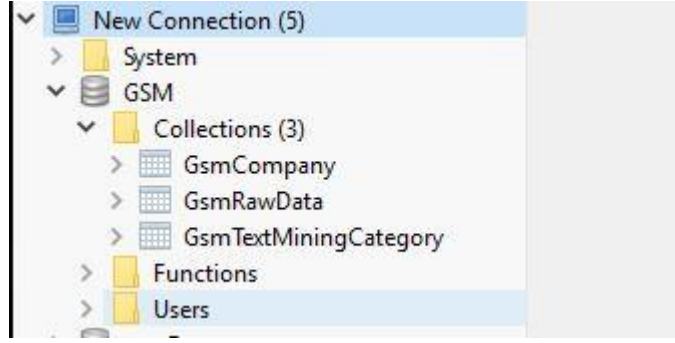
7.3.1 NoSQL mongoDB veri vabanı oluşturma

Visual Studio 'da el ile oluşturulan projedeki sınıfların özellikleri ile serileştirilen veri aynı şekilde veri tabanına eklenir. Şekil 7.3 'de Veri tabanı olarak MongoDB 'de ki Clollections görülmektedir. Entities projesinde bulunan sınıflarla aynı tipte oluşan collection 'ları görmekteyiz.

Ayrı bir yazılım ile toplanan verinin veya kod tarafında oluşan verileri MongoDB 'ye sadece eklemek ile ilgili collection 'ın oluşmasını ve verilerle birlikte tutulmasını sağlamaktadır

Şekil 7.5 'de ise GSMRawData sınıfımızın özelliklerine uygun olarak oluşan Collection örneği görülmektedir.

Şekil 7.4 'de ise GSMRawData Collection içerisinde bulunan HtmlData item 'ının içeriği görülmektedir.



Şekil 7.3 Veri Tabanı MongoDB'de Barınan Collections Görseli

```

/* 1 */
{
  "_id" : ObjectId("5732e46c6f42414d88895f10"),
  "Company" : "Vodafone",
  "WebSite" : "www.sikayetvar.com",
  "Url" : "https://www.sikayetvar.com/vodafone?page=",
  "PageID" : "?page=1",
  "Type" : 3,
  "HtmlData" : "\n<li class=\"sikayet clearfix\" data-id=\"5475832\">\n<a cla
}

/* 2 */
{
  "_id" : ObjectId("5732e46c6f42414d88895f11"),
  "Company" : "Vodafone",
  "WebSite" : "www.sikayetvar.com",
  "Url" : "https://www.sikayetvar.com/vodafone?page=1",
  "PageID" : "?page=0",
  "Type" : 3,
  "HtmlData" : "\n<li class=\"sikayet clearfix\" data-id=\"5473777\">\n<a cla
}

/* 3 */
{
  "_id" : ObjectId("5732e46c6f42414d88895f12"),
  "Company" : "Vodafone",
  "WebSite" : "www.sikayetvar.com",
  "Url" : "https://www.sikayetvar.com/vodafone?page=2",
  "PageID" : "?page=3",
  "Type" : 3,
  "HtmlData" : "\n<li class=\"sikayet clearfix\" data-id=\"5473441\">\n<a cla
}

/* 4 */
{
  "_id" : ObjectId("5732e46c6f42414d88895f13"),
  "Company" : "Vodafone",
  "WebSite" : "www.sikayetvar.com",
  "Url" : "https://www.sikayetvar.com/vodafone?page=3",
  "PageID" : "?page=4",
  "Type" : 3,
  "HtmlData" : "\n<li class=\"sikayet clearfix\" data-id=\"5473033\">\n<a cla
}

```

Şekil 7.4 Veri Tabanı MongoDB Collection Örneği

```

"WebSite" : "www.sikayetvar.com",
"Url" : "https://www.sikayetvar.com/vodafone?page=",
"PageID" : "?page=1",
"Type" : 3,
"HtmlData" : "\n<li class=\"sikayet clearfix\" data-id=\"5475832\"\>\n<a class=\"uyeAvatarCol\" href=\"/uye/detay/1200921\"
title=\"Yalçın S.\"\>\n<img class=\"uyeAvatar\" style=\"background:url( //www.sikayetvar.com/images/images_yeni/avатар_yok_erkek_1.jpg ) 7px
7px no-repeat;\" src=\"//www.sikayetvar.com/images/images_yeni/sikayet_liste_satir_avatar_golge.png\" title=\"Yalçın S.\"\>\n</a>\n<div
class=\"sikayetBaslik user \"\>\n<div>\n<h3>\n<a title=\"Vodafone Fahiş Paket Aşım Ücretlendirmesi!\"
href=\"/vodafone-fahis-paket-asim-ucretlendirmesi\"\>Vodafone Fahiş Paket Aşım Ücretlendirmesi!</a>\n</h3>\n<span class=\"sikayetBilgi\"\>\n<a
href=\"/uye/detay/1200921\"\>Yalçın S.</a> | \r\n<abbr class=\"timeago\" title=\"2016-05-11T10:41:38+03:00\"\>11 Mayıs</abbr> |
5475832\r\n</span>\n</div>\n<div>\n<div>\n<p class=\"sikayetOzet high clearfix\"\>\r\n
3 ay önce Vodafone'a
geçtim gu an çok pişmanım. İnternet paketim bitti, kullandığım 3 gün için 7x3 21 TL ek fatura ödemek zorunda kalıyorum. Bunun yanı sıra
İnternette ve müşteri hizmetlerinden ne hikmetse ek paket ala\r\n
... <a
href=\"/vodafone-fahis-paket-asim-ucretlendirmesi\" class=\"sikayetlerAnaListeSatirDevam sp\"\>\n</a>\n<br>\n</p>\n<ul class=\"ozetBilgiler\"
data-href=\"/vodafone-fahis-paket-asim-ucretlendirmesi\"\>\n<li class=\"islemButon\"\>\n<a class=\"userView\"\>\n<li class=\"spNew listeGoruldu
active\"\></li>\n<span class=\"listeOkunma\"\><span class=\"listCounter\"\>7 <em>Kişi Gördü</em></span>\n</span>\n</a>\n</li>\n<li
class=\"islemButon\"\>\n<a onclick=\"ga('send', 'event', 'New_Framework', 'Sikayet_Listesi', 'Takip');\" class=\"followComplaintList \"
complaintid=\"5475832\" title=\"Takip Et\" href=\"#javascript;\"\>\n<li class=\"spNew listeTakipEt\"\></li> \r\n</li>\n</ul>
Takip Et
\r\n</a>\n</li>\n<li class=\"islemButon\"\>\n<a class=\"sikayetPaylasButon\" rel=\"5475832\" href=\"#\"\>\n<li class=\"spNew listePaylas\"\></li>
Paylaş\r\n</a>\n</li>\n<li class=\"sikayetDurumBilgisi\"\>\n<a
href=\"/sikayet/detay/5475832/vodafone-fahis-paket-asim-ucretlendirmesi#gorusmeler\" class=\"cevapDurumuBekleniyor\"\><span
class=\"sp\"\></span></a>\n</li>\n<ul>\n<li class=\"socialShare socialShare5475832\" role=\"menu\"\>\n<li role=\"presentation\"\><a
class=\"btn2 btn-block btn-social btn-gplus\" onclick=\"ga('send', 'event', 'New_Framework', 'Sikayet_Listesi', 'GPlus');\"
href=\"https://plus.google.com/share?url=http://www.sikayetvar.com/sikayet/detay/5475832/vodafone-fahis-paket-asim-ucretlendirmesi\"\><i
class=\"fa fa-google-plus\"\></i>Google'da paylaş</a></li>\n<li role=\"presentation\"\><a class=\"btn2 btn-block btn-social btn-xs
btn-facebook\" onclick=\"ga('send', 'event', 'New_Framework', 'Sikayet_Listesi', 'Facebook');\"
href=\"https://www.facebook.com/sharer/sharer.php?u=http://www.sikayetvar.com/sikayet/detay/5475832/vodafone-fahis-paket-asim-ucretlendirmesi\"
\><i class=\"fa fa-facebook\"\></i>Facebook'ta paylaş</a></li>\n<li role=\"presentation\"\><a class=\"btn2 btn-block btn-social btn-xs
btn-twitter\" onclick=\"ga('send', 'event', 'New_Framework', 'Sikayet_Listesi', 'Twitter');\"
href=\"https://twitter.com/share?url=http://www.sikayetvar.com/sikayet/detay/5475832/vodafone-fahis-paket-asim-ucretlendirmesi&via=sikayetvar&
om&text=Vodafone Fahiş Paket Aşım Ücretlendirmesi!\"\><i class=\"fa fa-twitter\"\></i>Twitter'da paylaş</a></li>\n</ul>\n</li>\n<li
class=\"sikayet clearfix\" data-id=\"5476069\"\>\n<a class=\"uyeAvatarCol\" href=\"/uye/detay/1748872\" title=\"Samet E.\"\>\n<img
class=\"uyeAvatar\" style=\"background:url( //www.sikayetvar.com/images/images_yeni/avатар_yok_erkek_1.jpg ) 7px 7px no-repeat;\"
src=\"//www.sikayetvar.com/images/images_yeni/sikayet_liste_satir_avatar_golge.png\" title=\"Samet E.\"\>\n</a>\n<div class=\"sikayetBaslik
user \"\>\n<div>\n<h3>\n<a title=\"Vodafone'dan Aldığım Samsung S7'de Konuşma Sorunu!\"
href=\"/vodafonedan-aldigim-samsung-s7de-konusma-sorunu\"\>Vodafone'dan Aldığım Samsung S7'de Konuşma Sorunu!</a>\n</h3>\n<span
class=\"sikayetBilgi\"\>\n<a href=\"/uye/detay/1748872\"\>Samet E.</a> | \r\n<abbr class=\"timeago\" title=\"2016-05-11T10:40:56+03:00\"\>11
Mayıs</abbr> | 5476069\r\n</span>\n</div>\n<div>\n<div>\n<p class=\"sikayetOzet high clearfix\"\>\r\n
Yaklaşık
olarak 1 ay önce Vodafone İnternet sitesinden almış olduğum telefonumda yaklaşık 2 haftadır konuşmalarda problem yaşıyorum. Aradığım kişi
tekrar tekrar arıyorum ancak konuşabiliyoruz. Müşteri hizmetlerini aradığ\r\n
... <a
href=\"/vodafonedan-aldigim-samsung-s7de-konusma-sorunu\" class=\"sikayetlerAnaListeSatirDevam sp\"\>\n</a>\n<br>\n</p>\n<ul
class=\"ozetBilgiler\" data-href=\"/vodafonedan-aldigim-samsung-s7de-konusma-sorunu\"\>\n<li class=\"islemButon\"\>\n<a
class=\"userView\"\>\n<li class=\"spNew listeGoruldu active\"\></li>\n<span class=\"listeOkunma\"\><span class=\"listCounter\"\>14 <em>Kişi
Gördü</em></span>\n</span>\n</a>\n</li>\n<li class=\"islemButon\"\>\n<a onclick=\"ga('send', 'event', 'New_Framework', 'Sikayet_Listesi',
'Takip');\" class=\"followComplaintList \" complaintid=\"5476069\" title=\"Takip Et\" href=\"#javascript;\"\>\n<li class=\"spNew

```

Şekil 7.5 Veri Tabanı MongoDB Veri HTML Data Örneği

Veri tabanımızda **GsmTextMiningCategory** adındaki collection ‘da sözlük kelimelerimizi barındırmaktadır. **GsmRawData** adındaki collection ‘da ham veriyi barındırmaktadır. **GsmCompany** adındaki collection ‘da GSM firmalarının isimlerini barındırmaktadır. Şekil 7.3 ‘de görülmektedir.

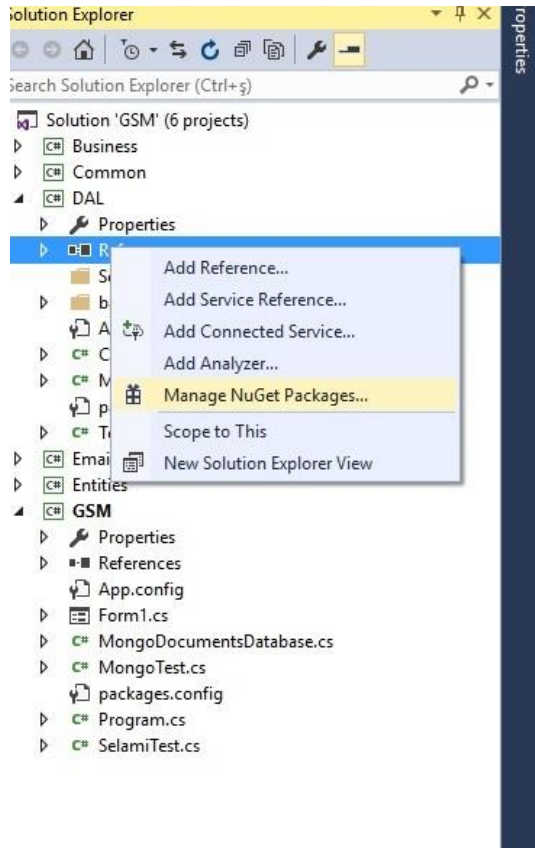
Collections : **GsmTextMiningCategory**, **GsmRawData**, **GsmCompany**

7.3.2 MongoDB Kullanımı

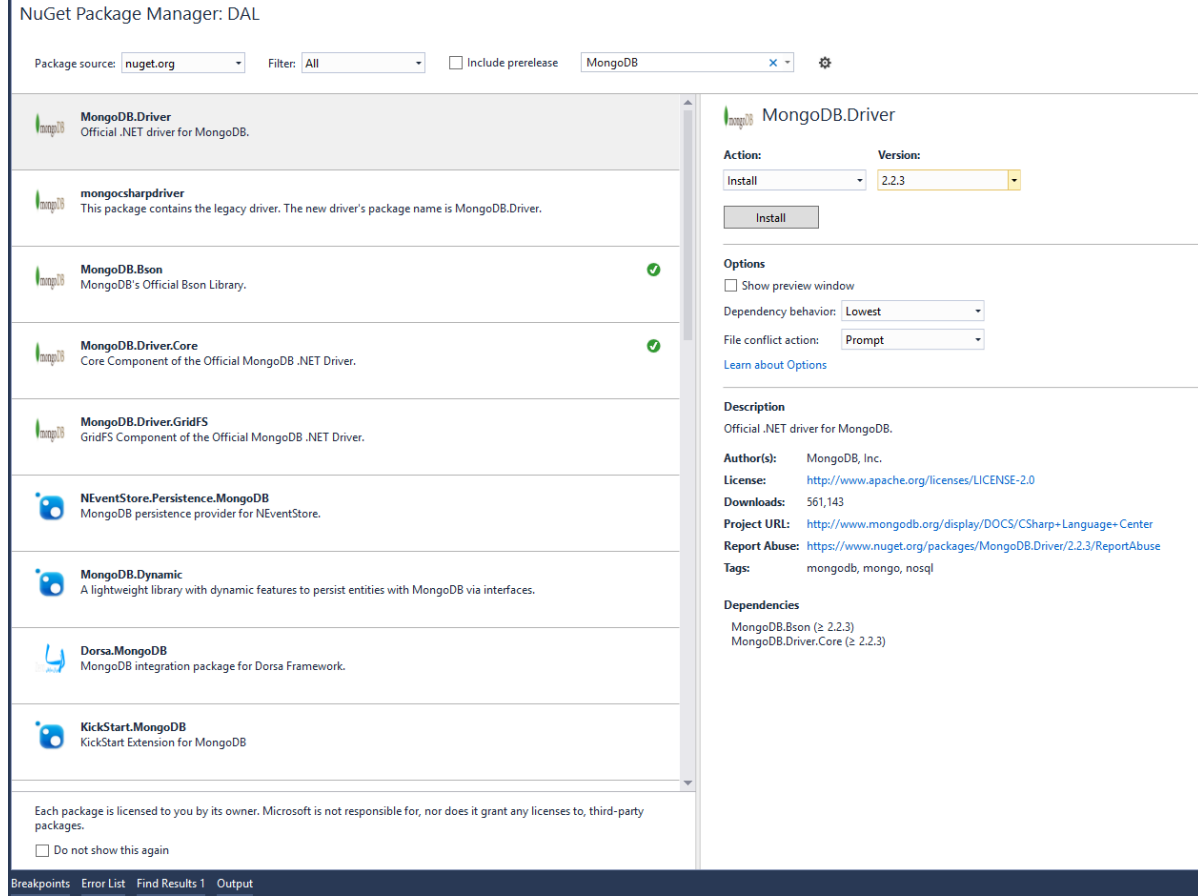
MongoDB 'yi kullanabilmek için C# projemize (projesine) öncelikle MongoDB C# driver 'ni yüklememiz gerekir. İlgili katmanlarda proje References üzerinde sağ tıklayıp **Manage NuGet Packages** tıklıyoruz. Şekil 7.6 'de gösterilmiştir.

Şekil 7.6 sonrasında NuGet Package Manager açılacaktır. Search kısmına MongoDB yazdığımızda sıralanan seçenekler arasından Official .NET driver for MongoDB açıklaması olan satıra tıkladığımızda sağda **install** ibaresi tıklamamız ile projemize MongoDB dahil olacaktır. Projede MongoDB Driver Version olarak 2.2.3 eklenmiştir.

Şekil 7.7 görülmektedir.



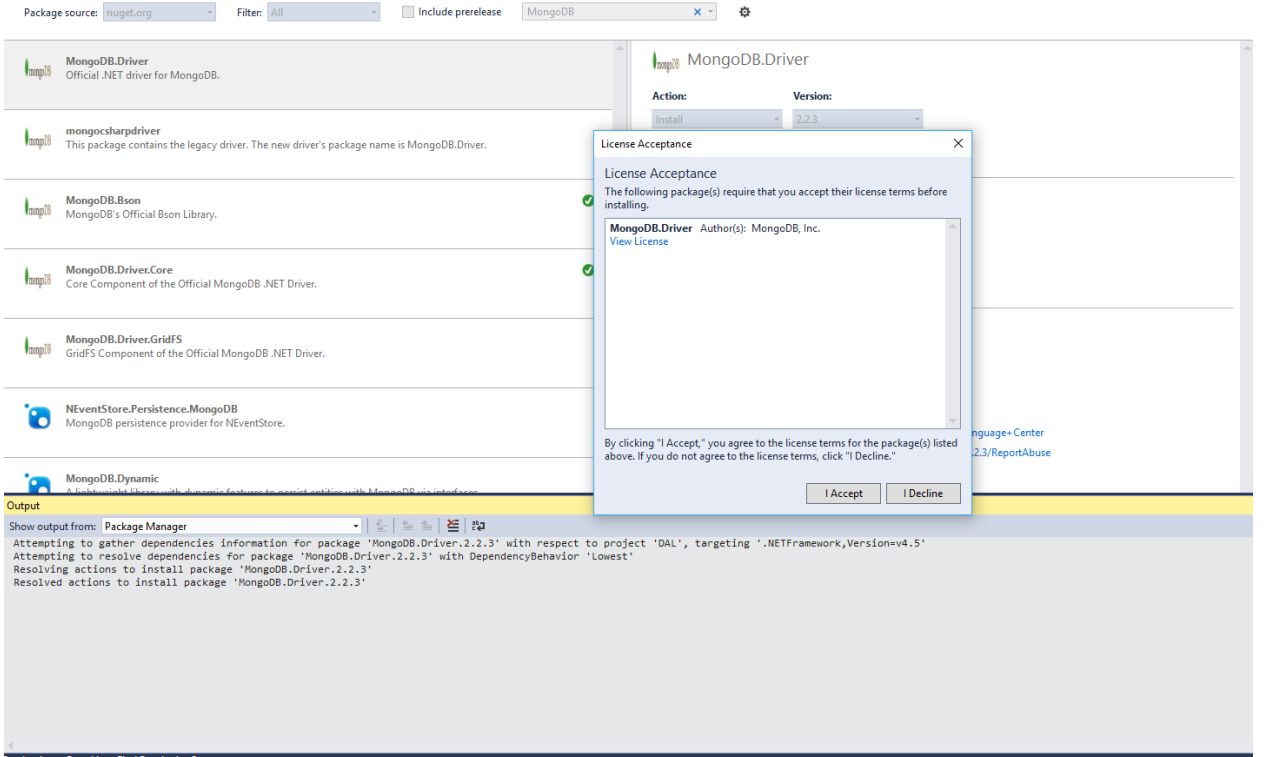
Şekil 7.6 References NuGet



Şekil 7.7 MongoDB C# Driver Ekleme

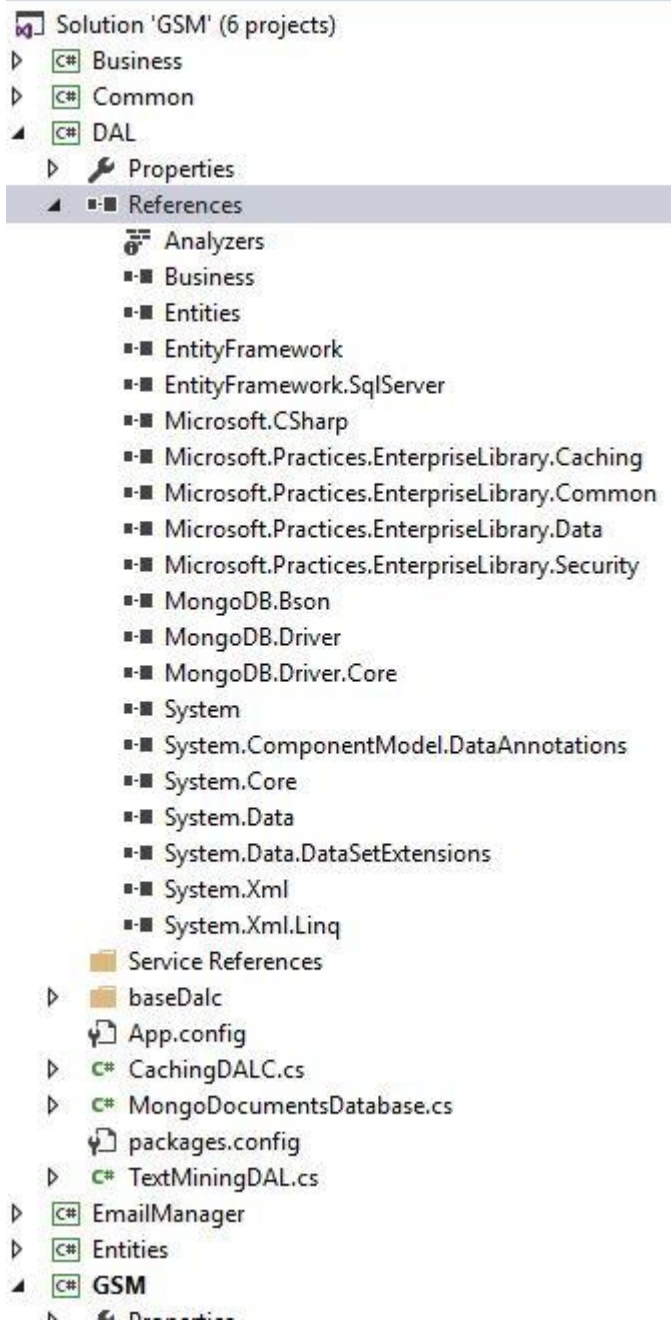
Install butonuna bastıktan sonra lisans sözleşmesini kabul ediyoruz. Şekil 7.8 'de görülmektedir.

NuGet Package Manager: DAL



Şekil 7.8 MongoDB.Driver lisans sözleşme onayı

I Accept butonuna bastıktan sonra Şekil 7.9 'de görüldüğü üzere MongoDB.Driver kütüphanelerinin projemizin References' ları içerisinde görüyoruz.

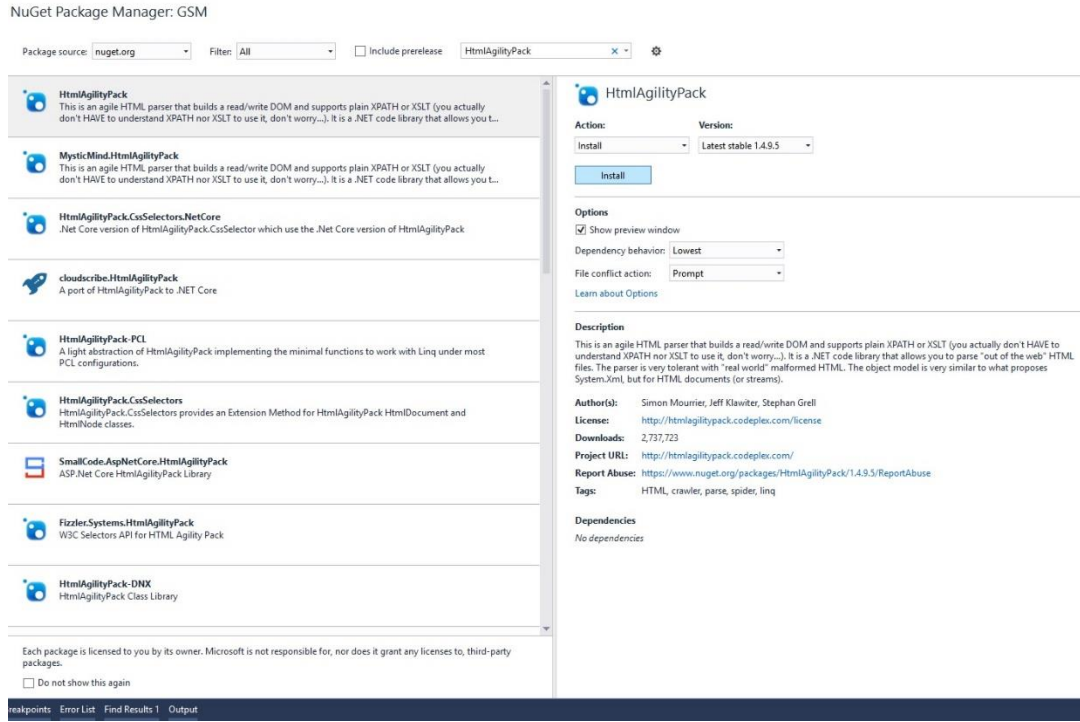


Şekil 7.9 References

7.4 Veri İşlemleri:

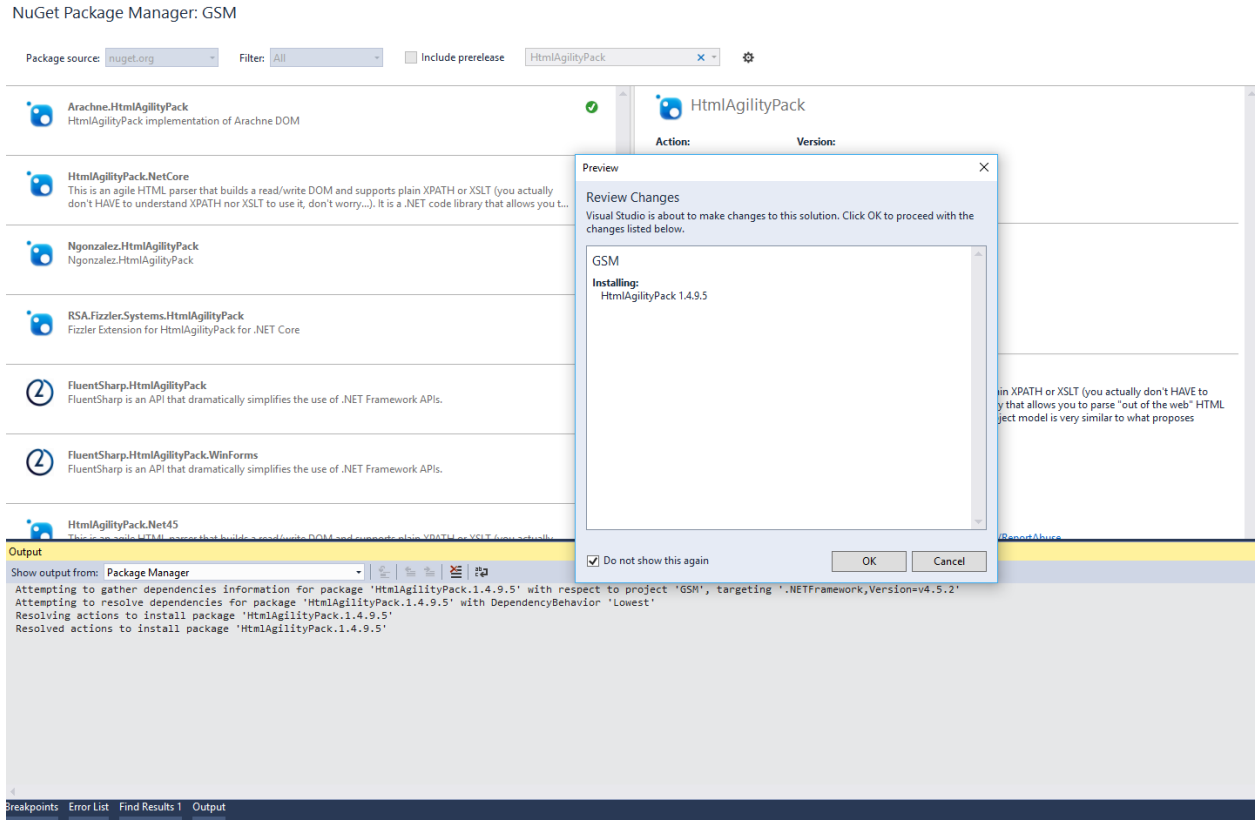
7.4.1 Veri toplama

Data toplama aşamasında BOT class'ı üzerinden sosyal paylaşım sitelerinden kullanıcı yorumlarını toplanacaktır. Windows projemizden online açık olan bir siteye ulaşım bilgi toplamak için HtmlAgilityPack.dll kütüphanesinden yararlanacağız. Projemize ekleyebilmek için Şekil 7.10 'de görüldüğü gibi yine **Manage NuGet Packages** uygulamasını kullanarak HtmlAgilityPack kütüphanesini yükleyeceğiz. Search kısmına **HtmlAgilityPack** yazıyoruz. Install butonuna tıklıyoruz.



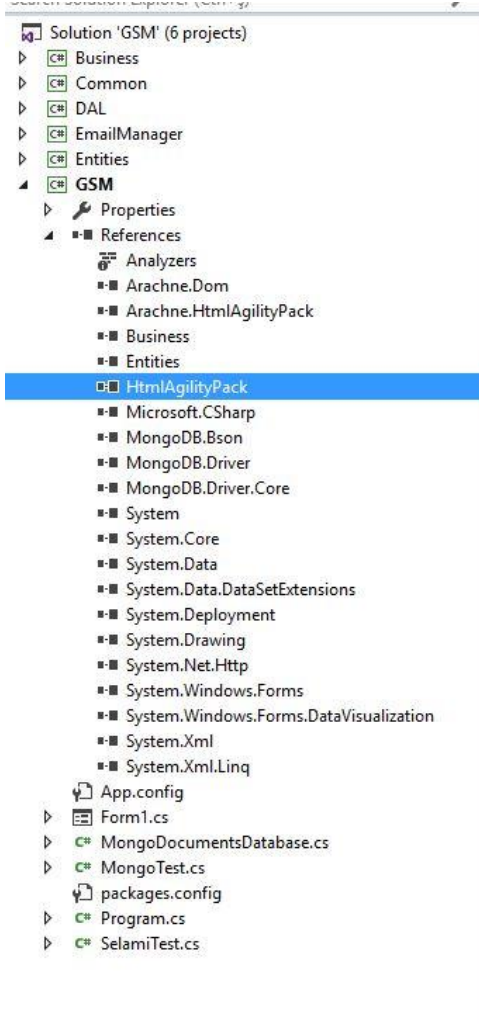
Şekil 7.10 HtmlAgilityPack kütüphanesinin eklenmesi

Daha sonra açılan ekranda ki değişiklik gereğini kabul ediyoruz. Ekran Şekil 7.11 'de görülmektedir.



Şekil 7.11 HtmlAgilityPack Ekleme Değişiklik Onayı

Projenin referans bölümünde HtmlAgilityPack kütüphanesinin eklendiğini Şekil 7.12 'de görüyoruz.



Şekil 7.12 GSM Projesinin Referans Listesi

BOT Class tetiklendiğinde; sikayetvar.com sitesinden GSM ile ilgili konuları okuyarak verileri ETL sürecine tabi tutarak MongoDB veri tabanımıza ham veri kayıt işlemi gerçekleşecektir. HTML sayfanın çağırısı metodu Şekil 7.13 görülmektedir.

```

public string GetHTMLonURL(string url)
{
    string result = string.Empty;
    WebClient client = new WebClient();
    client.Encoding = ASCIIEncoding.UTF8;
    client.Headers.Add("user-agent", "Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.2; .NET CLR 1.0.3705;)");
    result = client.DownloadString(url);
    return result;
}

```

Şekil 7.13 Html Read Metodu

Şekil 7.13 ‘de ki **GetHTMLonURL** metoduna **GetData** metodu çağrı yaparak Şekil 7.14 ‘da görüldüğü üzere TurkTelekom verileri toplanmaktadır. Recursive sorgu ile son sayfaya kadar tüm veri tek tek okunur ve veri tabanına gönderilir.

```

public class BOT
{
    private readonly string firstUrl = "https://www.sikayetvar.com/turk-telekom?page=";

    private static int page = 0;
    HtmlNode checkNextPage;
    1 reference
    public void GetData(string url, int? pageID)
    {
        if (String.IsNullOrEmpty(url) && pageID == null)
            url = firstUrl;
        else
            url = url + pageID;
        HtmlAgilityPack.HtmlDocument agi = new HtmlAgilityPack.HtmlDocument();
        agi.LoadHtml(GetHTMLonURL(url));
        var result = agi.DocumentNode.SelectSingleNode("/html/body/div[1]/div[2]/div[3]/ul[1]");
        var resultNextPageID = string.Empty;
        if (pageID < 2 || pageID == null)
            resultNextPageID = agi.DocumentNode.SelectSingleNode("/html/body/div[1]/div[2]/div[3]/div[4]/a[@href]").GetAttributeValue("href", "unknown");
        else if (pageID >= 2)
            checkNextPage = agi.DocumentNode.SelectSingleNode("/html/body/div[1]/div[2]/div[3]/div[4]/a[2][@href]");
        if (checkNextPage != null)
            resultNextPageID = agi.DocumentNode.SelectSingleNode("/html/body/div[1]/div[2]/div[3]/div[4]/a[2][@href]").GetAttributeValue("href", "unknown");
        if (result != null)
        {
            GsmRawData grd = new GsmRawData();

            grd.Company = "TurkTelekom";
            grd.WebSite = "www.sikayetvar.com";
            grd.Url = url;
            grd.PageID = resultNextPageID;
            grd.Type = GsmComplainingCategory.Hepsi;
            grd.HtmlData = result.InnerHtml;
            page++;
            if (!String.IsNullOrEmpty(resultNextPageID))
                GetData(firstUrl, page);
        }
    }
}

```

Şekil 7.14 Veri Toplama Metodu

Şekil 7.14 ‘de toplanan veri Şekil 7.15 ‘de veri tabanına aktarılır. Şekil 7.15 ‘de görülmektedir. MongoDB Driver ‘ın Insert metodu hile ham veri olduğu gibi veri tabanına eklenir. Şekil 7.15 ‘de görüldüğü üzere MongoDB driver Collection kütüphanesinin Insert metodudur.

```

if (result != null) {
    GsmRawData grd = new GsmRawData();

    grd.Company = "TurkTelekom";
    grd.WebSite = "www.sikayetvar.com";
    grd.Url = url;
    grd.PageID = resultNextPageID;
    grd.Type = GsmComplainingCategory.Hepsi;
    grd.HtmlData = result.InnerHtml;

    var Client = new MongoClient(connectionString);
    var MongoDB = Client.GetDatabase("GsmRawData");
    var Collec = MongoDB.GetCollection<GsmRawData>("GsmRawData");
    Collec.InsertOne(grd);
}

```

Şekil 7.15 Veri Insert Metodu

MongoDB ‘nin sağladığı driver’ın database insert ile veri hızla toplandığı gibi veri tabanına aktarılır.

7.4.2 Ön İşleme

Yukarıda ki işlemler sürecinde verileri topladıktan sonra uygulayacağımız mimariyi yoracak ve sonuçları etkileyecek veriler olduğu düşünülüyorsa bunları temizlememiz gerekir. Sözlükte kullandığımız kelimeler ile ilgisiz alandaki bir kelime benzerliği temizlemediğimiz takdirde sonuca etki edecektir.

7.4.3 Gövdeleme yöntemi

Projede Joker “Wildcard” gövdeleme yöntemi kullanıldı. Türkçe dili eklemeli dil olduğundan kelimedeki bir gövdenin sonuna farklı ekler alabilir. Bu sebeple gövdeleme uygulamadığımız takdirde metin içerisinde geçecek sözlük kelimesinin tüm ekli hallerini eklememiz gerekir. Buda hem hatalı sonuç üretebilir hem de işleme süreci olarak çok daha uzun bir zamana mal olur. Örnek olarak “haber” kelimesi ile “haberli”, “habersiz”, “haberde” ve “haberin” kelimelerini tek tek ve gelebilecek daha birçok ekide sözlüğe eklememiz gerekecekti. Haber kelimesi örneğinde ki gibi tüm eklere karşın “haber” gövdesini seçmek bir çok anlamda işimizi kolaylaştıracaktır.

Joker kelime seçiminin hedefi, aynı söz dizimi üzerine kurulmuş olan farklı eklere sahip kelimelerdir. Farklı ek almaları anlam değişikliğine sebebiyet vermemelidir. Gövdeleme yönteminde yapım ve çekim eklerinden sadeleştirilen kelimeler, ortak bir kelime köküne indirgenir. Ancak Joker kelimedeki kelimenin köküne indirgeme şartı yoktur.

Örnek joker kelime olarak “kalem*” ile (Joker olduğunu * işaretinden anlıyoruz). Süreç içerisinde “kalem” kelimesi farklı ekler olsa da ”kalem” kelimesi vurgulanmış olacaktır.

7.4.4 Sözlük oluşturma

Sözlük aşamasında seçilen kelimeler kategoriler altında toplanır. Bu sözcüklerin kategorilere uygun olmasına dikkat edilmelidir. Oluşturulan sözlük tablosunda gereksiz kelimelere yer verilmemesine ayrıca dikkat edilmelidir. Çünkü bu gereksiz kelimeler sonuçlardaki faydayı anlamsız kılabilir. GSM sektörünün şikâyetlerini temsil edebilecek dört kategori oluşturuyoruz.

Bunlar :

1. Kapsama
2. İnternet
3. Fatura
4. Hizmet

Bu kategoriler altında toplanan, kategoriyi simgeleyen sözlüklerden oluşmaktadır. Şekil 7.16 ‘da veri tabanında barınma şekli görünmektedir.

```
{
  "_id" : ObjectId("57d159ca3abf51278425371a"),
  "CategoryName" : "Kapsama",
  "CompanyName" : null,
  "WordList" : [
    {
      "Name" : "sokak",
      "WeightScore" : 0.1
    },
    {
      "Name" : "çek",
      "WeightScore" : 0.3
    },
    {
      "Name" : "sinyal",
      "WeightScore" : 0.1
    },
    {
      "Name" : "iletişim",
      "WeightScore" : 0.1
    },
    {
      "Name" : "göbeği",
      "WeightScore" : 0.3
    },
    {
      "Name" : "evim",
      "WeightScore" : 0.1
    }
  ]
}
```

Şekil 7.16 Proje Sözlük Yapısı

Şekil 7.16 'da görüldüğü gibi Kapsama kategorisinde bulunan her kelimenin ayrı bir ağırlık etkisi bulunmaktadır.

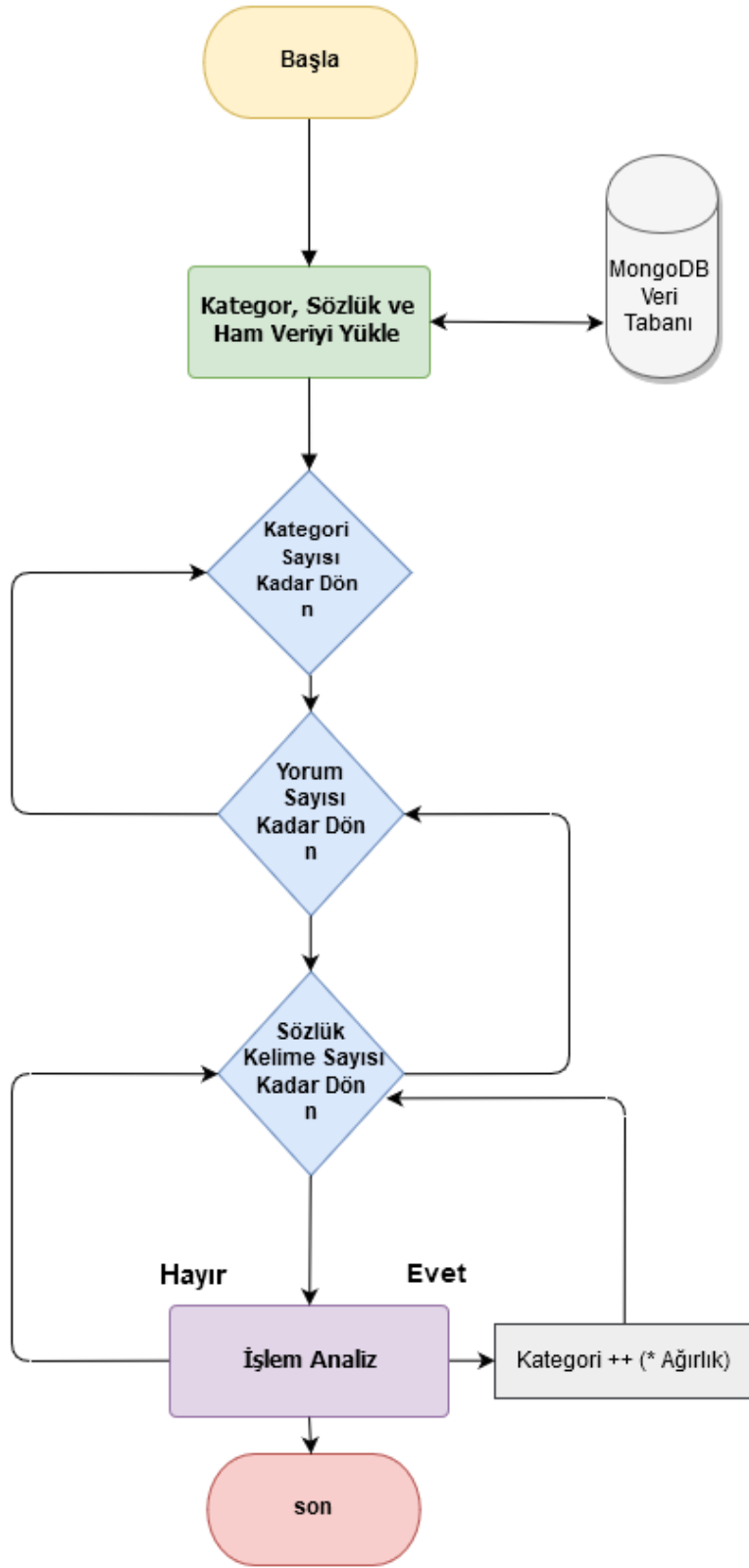
7.5 Algoritma Uygulama

Şekil 7.17 'de sözlük algoritmasının kodu görülmektedir. Veriler veri tabanımıza aktarıldıktan sonra oluşturulan sözlükle veriler üzerinde algoritma uygulamasını sağlayıp raporlama işlemleriyle beraber çalışmanın sonuçlarını gösteriyor olacağız.

Tüm veri öncelikle veri tabanın çekilmektedir. Sonrasında ise algoritmaya tabi tutularak raporlanacak halde RAM 'de bekletilmektedir. Algoritma kategorisinde kategori adedi kadar dönülürken her kategori başlığı için o firmaya yapılan yorum sayısı kadar dönülür. Her yorum için oluşturduğumuz sözlükteki kategoriye ait kelimelerin boyutu kadar dönülürken eşleşen kelimelerin ağırlık skorları toplanmaktadır. Bu skorlar analiz sonrasında GSM şirketlerinin rapor class 'ına yazılarak RAM'e aktarılır.

```
foreach (GsmTextMiningCategory item in gsmTextMiningCategory)
{
    GsmTextMiningResult _r = new GsmTextMiningResult();
    _r.CategoryName = item.CategoryName;
    _r.CompanyName = company;
    score = 0;
    foreach (GsmRawData data in grd)
    {
        foreach (var _w in item.WordList)
        {
            bool result =data.HtmlData.Contains(_w.Name);
            if (result)
                score= score+_w.WeightScore;
        }
    }
    _r.Result = score;
    _listResult.Add(_r);
}
```

Şekil 7.17 Algoritma Mimarisi



Şekil 7.18 Proje Algoritması

Şekil 7.18 'de projenin algoritma diyagramı görünmektedir. Veri tabanından çağrılan veriler ilk önce oldukları gibi RAM 'e aktarılırlar. RAM 'e verileri aktararak işlem süreleri kısaltılmış olur. RAM 'e yazılmış kategoriler sırası ile dönülürken bu kategoriye ait sözlük kelimeleri dikkate alınır. GSM şirketine ait her bir yorum için kategoriye ait sözlük elemanları dikkate alınarak kategori bazında puanlama yapılması sağlanır. Süreçte yoruma ait her kelime için kategoriye ait her bir sözlük kelitemizle eşleşme durumuna bakılır ve eğer eşleşme bulunursa; sözlük kelimesinin ilgili ağırlık puanı dikkate alınarak kategoride puan artışı sağlanır. Aynı yorum içerisindeki kelimeler dikkate alındığından ilgili yorum birden çok kategorinin puanlamasına etkide bulunabilir.

Örnek : “Vodafone'dan 24 ay taksitli Discovery Mobile GM5 Plus telefon aldım. Piyasada 1000 TL olan telefonu tarifesiyle birlikte 2400 TL'ye aldım farka bakın 1400 TL 400 TL telefon farkı ödüyorum zaten ama bir de Vodafone beni şu an piyasada diğer operatörlerinde sunduğu 20 TL'ye verilen tarifeyi 40 TL'ye verdi. 4.5G gibi bir teknolojinin olduğu dönemde 2 GB gibi küçük bir şey veriyor bana. Telefonda da memnun değilim kamerası çok kötü telefonun ya parasını verip tarifeden çıkmak istiyorum ya da tarifeyi 20 TL'ye düşürmek istiyorum Vodafone'u aradım ama 3. sıradasınız deyip 5 dakika sırada bekletip aramayı sonlandırdılar amaç arayanları yıldırmak yani hizmet değil. Koskoca şirketin büyük bir çağrı merkezinde yok bu şirket sizin onu aramanızı istemiyor zaten.”

Yorumunu ele alalım. Fatura, Hizmet, İnternet kategorilerimize ait sözlük kelimeleri barındırmaktadır. İlgili yorum hem Fatura hem İnternet hemde Hizmet kategorisinin puanını etkilemektedir. “çağrı” , “tarife” , ”4.5G” , “TL” , ”hizmet” kelimeleri farklı kategorilerinden oluşmasına karşın aynı yorum içerisinde yer almaktadır.

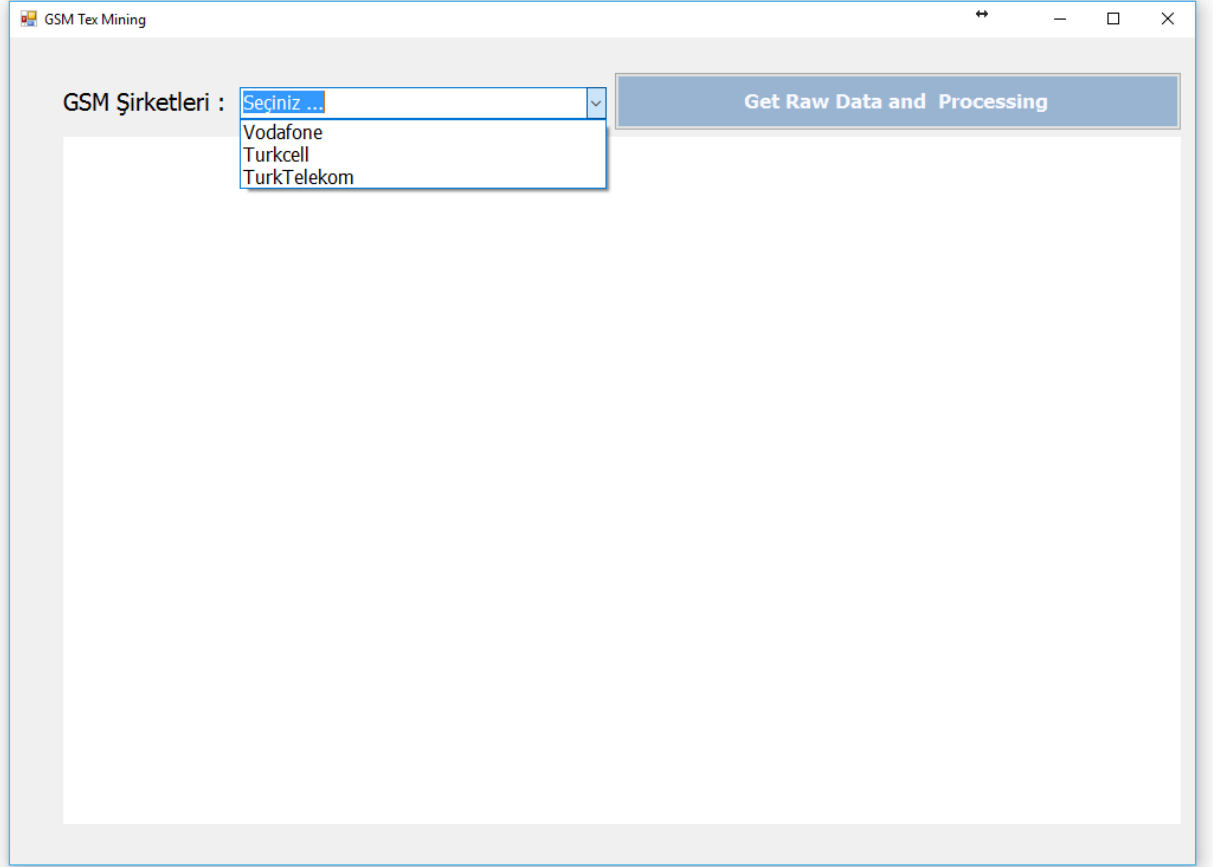
Fatura : Tarife,TL

Hizmet : Çağrı

İnternet : 4.5G, GB

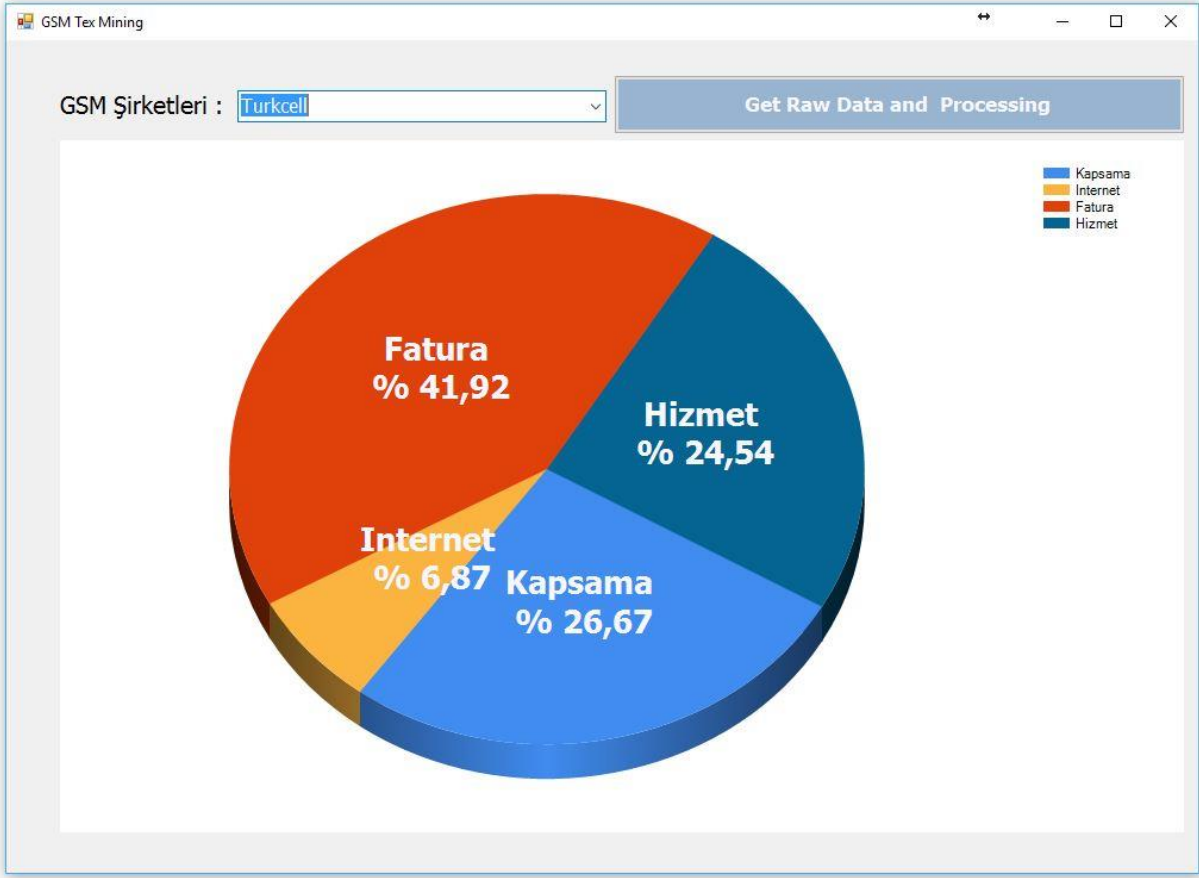
Yorum içerisinde geçen kelimeleri incelediğimizde yukarıda ifade edilen üç kategori altındaki sözlük kelimelerimiz görünmektedir. Örnek yorumumuz dört kategorimizden Fatura, Hizmet ve İnternet kategorilerini kapsamaktadır.

Algoritmaya tabi tutulan GSM şirketine ait her yorumun analiz sonrasında RAM ‘e aktarılan rapor verilerini programımızda seçilen şirket adı altında sunumu yapılmaktadır.



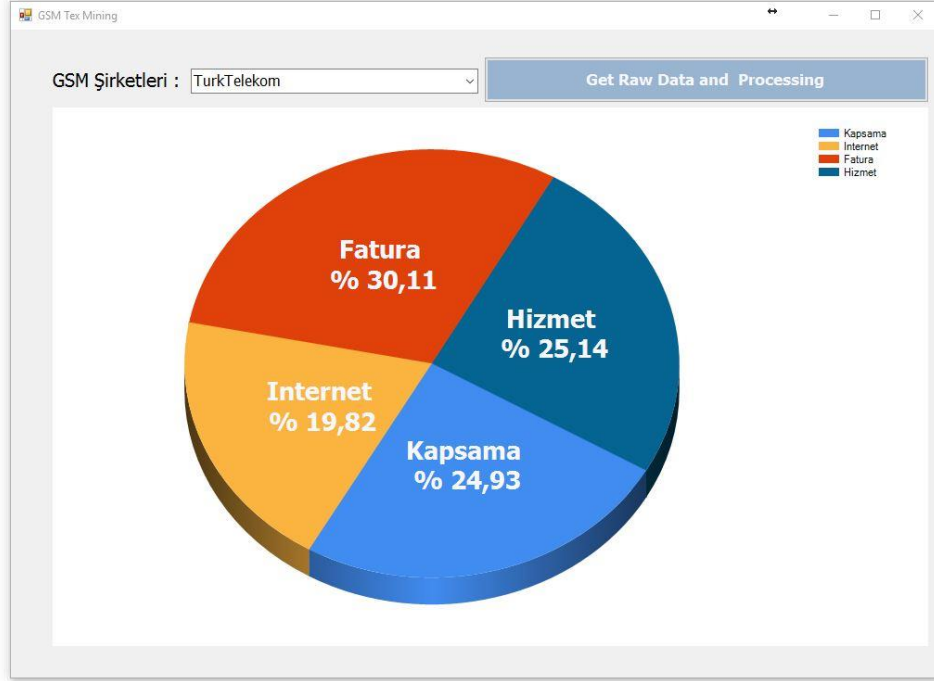
Şekil 7.19 Program Ara Yüzü

Şekil 7.19 ‘da programın ara yüzü görülmektedir. Öncelikle tüm datanın çağırılması ve işlenmesi gerekmektedir. Bu sebeple ilk önce “Get Raw Data and Processing” butonuna basmalıyız. Bu süreçte buton adından da anlaşılacağı gibi tüm veri, veri tabanından çekilerek algoritmaya tabi tutulmaktadır. Sonuçlar RAM alınarak kullanıcı için hazır bekletilmektedir. Sonrasında GSM şirketlerinin adının bulunduğu dropdownlist ‘den sonucunu görmek istediğimiz şirketi seçerek programın gövdesinde PIE CHART olarak raporun ekrana yansıtılması sağlanır.

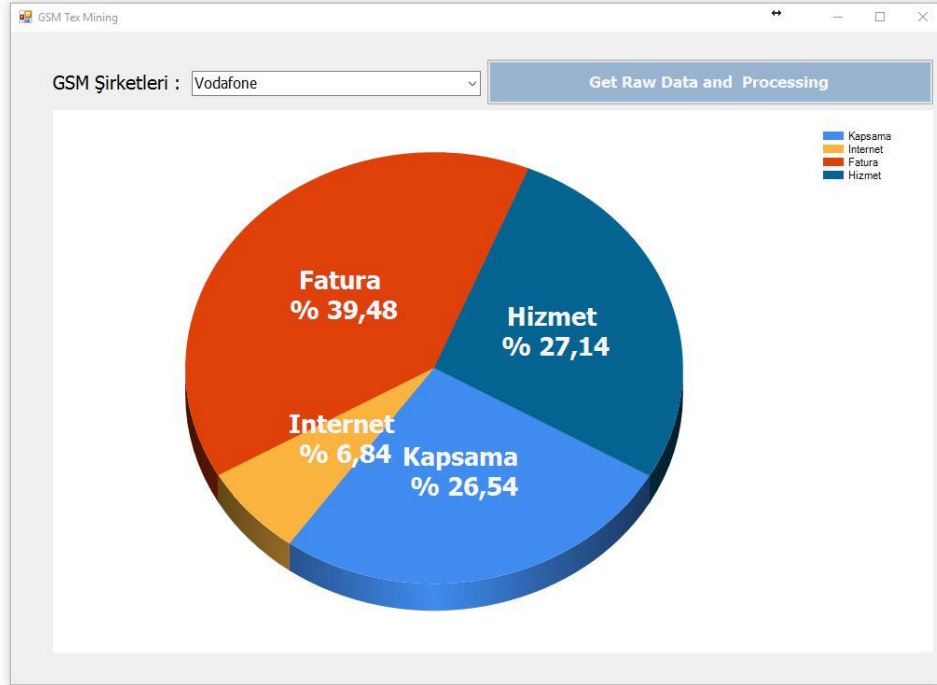


Şekil 7.20 Program Turkcell Sonuçları

Şekil 7.20 'de görüldüğü **kapsama, internet, fatura ve hizmet** kategorileri dikkate alınarak hazırlanan rapor PIE CHAR üzerinde yüzdelerle ekrana yansıtılmıştır. Şekil 7.20 'de Turkcell 'in aldığı yorumlar üzerinden yüzdelerle görünmektedir.



Şekil 7.21 Program Türk Telekom Sonuçları



Şekil 7.22 Program Vodafone Sonuçları

Şekil 7.20 , Şekil 7.21 ve Şekil 7.22 de tüm çalışmanın sonuçları görünmektedir. Bu raporlar üzerinden anlaşılacağı gibi toplam değerlendirmelerinde birbirlerinden farklılıkları görünmektedir.

8. SONUÇ

Sonuç olarak GSM şirketlerinin sunulan hizmet ve tüketici arasındaki ilişkileri üzerine yorum yapabiliriz. Firmalar bu doğrultuda pazarlama ya da iyileştirme adına stratejilerini belirleyebilirler. Turkcell ve Vodafone ‘nun birbirlerine işlem kalitesi olarak yakın oldukları görünmektedir. Fakat Türk Telekom ‘un müşterilerine “**internet**” kategorisinde daha çok sorun yaşattığı görülürken “**fatura**” kategorisinde ise diğer iki firmaya karşın çok daha avantajlı görünmektedir. Bu ekranlar sonucunda tüketicide hangi firmayı seçeceğine ya da seçeneklerinin neler olduğuna daha kolay verebilecektir.

Pazarlama stratejilerini düşündüğümüzde karar vericiler için en önemli unsur doğru bilgidir. Bu bilgiye ulaşmak ya da elde etmek çok maliyetli olabilmektedir. Bununla beraber bilgiye ulaşma anlamında yeteri kadar insan ile çözümlene yapılamadıysa elde edilen bilginin de doğruluğu azalacaktır. Buda stratejilerin yanlış temeller üzerinde kurgulanmasına sebep olacaktır.

Zamanımızın aracı olan sanal dünya ölçülebilir ve takip edilebilir bir yapıya sahiptir. Buna dikkat eden günümüz firmaları her sosyal mecrada yer alamaya çalışmaktadır. Bilinen hemen hemen her firma Twitter, Facebook, Instagram benzeri sosyal mecralarda kurumsal hesaplar açmaktalar. Firmalar sosyal mecraları aynı zamanda aktif olarak kullanmaktadırlar. Buda hem firmalar hem de insanların sosyal medyalar üzerinde bıraktıkları izlerin tarafımızdan kolayca izlenmesini sağlamaktadır. Bu büyük verinin farkında olan firmalar daha az enerji ile daha doğru sonuçlara bu sayede ulaşabilirler. Bu büyük veri üzerinde metin madenciliği işlemini yerine getirerek firmaların hizmetleri hakkında başarılı ya da başarısız oldukları ölçülebilmektedir. Bu sayede belirleyecekleri stratejileri hakkında karar verme süreçlerinde daha doğru bilgilere ulaşılmasını sağlanabilmektedir.

Projemizin sonuç ekranlarına bakıldığında GSM firmalarından müşterilerin beklentilerini veya firmaların birbirleri arasında ki farkı ya da benzerlikleri rahatlıkla görebiliyoruz. Böylelikle her GSM şirketi kendi eksiklerini belirleyerek müşteri memnuniyet kalitesinin artmasını sağlayabilirler. Alınan kararlar sonrasında ki iyileştirme süreçlerinin de ölçümleri yapılarak gelişmeleri de analiz edebilirler. Bu sayede metin madenciliğinin hızından faydalanarak her iyileştirme sonuçlarını ayrı ayrı analiz edebilirler. Örneğin Türk Telekom internet kategorisi sorunu % 19,82 iken Vodafone ve Turkcell 'de bu sorun % 6 seviyelerinde. Tüm GSM şirketlerinin kapsama ve hizmet yaklaşım yüzdeleri birbirine yakın. Fatura kategorisinde farklılık göstermektedirler. Fatura kategorisinde yaşanan sorunlarda Turkcell %41,92 , Vodafone %39,48 ve Türk Telekom %30,11 şeklinde sıralanmaktadır. Turkcell ve Vodafone 'nun internet kategorisinde daha az sorun yaşadıklarını da açıkça görebiliyoruz. . Bu rakamlara bakıldığında hangi firmanın hangi eksikliğini öncelikle gidermesi gerektiği açıkça görünmektedir.

Metin madenciliği burada kullanıldığı gibi birçok alanda kullanılmaktadır. Örneğin sağlık problemleri olarak alınan hasta şikâyetleri üzerinden yeni gelen bir hastanın ön tespiti metin madenciliği uygulaması ile çok daha hızlı ve başarılı yapılabilir Geliştirilecek uzman sistem ile insan kullanımı hata faktörü de ortadan kalkar. Hastayı karşılayan ilk birimi düşünürsek mesai bitimine doğru yorgunluk sebebi ile hata yapma oranı riski yüksektir. Metin madenciliği ile böyle bir uygulama geliştirilebilir. Böylelikle kişiye özel metin madenciliği uygulamasını da geliştirmiş olduk.

Yakın gelecekte metin madenciliği konusunun hayatımıza sağlayacağı hizmeti ve faydası büyük olacaktır.

KAYNAKLAR

- ÖZCAN C.** (2014) Veri Madenciliğinin Güvenlik Uygulama Alanları ve Veri Madenciliği ile Sahtekârlık Analizi, İstanbul Bilgi Üniversitesi Sosyal Bilimler Enstitüsü Bilişim Ve Teknoloji Hukuku Yüksek Lisans Programı
- MELEK C.** (2012) Metin Madenciliği Teknikleri Şirketlerin Vizyon İfadelerinin Analizi, Dokuz Eylül Üniversitesi Sosyal Bilimler Enstitüsü Ekonometri Anabilim Dalı Ekonometri Programı
- BEYHAN H. D.** (2014) Sosyal Medya Üzerinden Metin Madenciliği ve Duygu Analizi ile Pazar Değerlendirme, İstanbul Teknik Üniversitesi Fen Bilimleri Enstitüsü Endüstri Mühendisliği Anabilim Dalı Mühendislik Yönetimi Programı
- ARAVİ G.** (2014) Metin Madenciliği İle Sosyal Medya Analizi Yüksek Lisans Tezi, İstanbul Aydın Üniversitesi Fen Bilimleri Enstitüsü Bilgisayar Mühendisliği Anabilim Dalı Bilgisayar Mühendisliği
- KUZUCU K.** (2015) Müşteri Memnuniyeti Belirlemek İçin Metin Madenciliği Tabanlı Bir Yazılım Aracı, Fen Bilimleri Enstitüsü Bilgisayar Mühendisliği Anabilim Dalı
- DEMİREL A.** (2015) Metin Madenciliği Yöntemleri İle Sosyal Medyadan Toplanan Fotoğraflı Paylaşımların, Metin – Fotoğraf Eşleşmesinin İncelenmesi, Beykent Üniversitesi Fen Bilimleri Enstitüsü Matematik Bilgisayar Anabilim Dalı Bilgi Teknolojileri Bilim Dalı
- SANCAR Y.** (2016) Metin Madenciliği Kullanılarak Talep Tanıma Ve Yönlendirme Sistemi, Atatürk Üniversitesi Fen Bilimleri Enstitüsü Bilgisayar Mühendisliği Anabilim Dalı
- DÖVEN S.** (2013) Metin Madenciliği İle Dokümanlar Arasındaki Benzerliklerin Bulunması, Bahçeşehir Üniversitesi Fen Bilimleri Enstitüsü Bilgi Teknolojileri Anabilim Dalı
- DURMUŞ M.S.** (2005) Veri Kümeleme Algoritmalarının Performansları Üzerine Karşılaştırmalı Bir Çalışma, Pamukkale Üniversitesi Fen Bilimleri Enstitüsü Consumer Purchase Behavior İn An Online Shopping Context , International Journal Of Retail & Distribution Management, Vol. 31 Iss: 1, Pp.16 - 29
- SİLAHTAROĞLU G.** (2008), Veri Madenciliği, İstanbul: Papatya Yayınları

- OĞUZLAR A.** (2011), Temel Metin Madenciliği, Bursa : Dora Yayınları
- MELEK C.** (2012), Metin Madenciliği Teknikleri İle Şirketlerin Vizyon İfadelerinin Analizi, Dokuz Eylül Üniversitesi Sosyal Bilimler Enstitüsü Ekonometri Anabilim Dalı Ekonometri Programı
- OĞUZ B.** (2009), Kulak Burun Boğaz Epikriz Notlarından Birliktelik Kurallarının Çıkarılması, Akdeniz Üniversitesi
- BOZ E.** (2015) Kullanıcı Ve Sözlük İlişkisi, İç Anadolu Dilcileri Sürekli Çalıştayı-II
- ALTAN Z.** (2016) Dil Modellemede Belirsizlik Probleminin Etmenlenmiş Dilbilgisi İle Giderilmesi, İstanbul Üniversitesi Mühendislik Fakültesi Bilgisayar Mühendisliği Bölümü, Makale
- ŞENTÜRK F. & GÜNDÜZ G.** (2014), Arama Motoru Bing 'in Tekil-Çoğul Kelimeler İçin Gövdeleme Mekanizması, Pamukkale Üniversitesi Mühendislik Bilimleri Dergisi, Cilt 20, Sayı 2, 2014, Sayfalar 42-45
- ÖNER M.** (1999), Türkçede Edatlı (Sentaktik) İsim Çekimi , Türk Dili, sayı:565, Ocak-1999, 1018. s.
- ÇELİKİYAY E. K.** (2010), Metin Madenciliği Yöntemiyle Türkçede En Sık Kullanılan Ve Birbirini Takip Eden Harflerin Analizi Ve Birliktelik Kuralları, Beykent Üniversitesi Fen Bilimleri Enstitüsü Matematik Bilgisayar Anabilim Dalı Bilgisayar Ağları Ve İnternet Teknolojileri Bilim Dalı
- GÜRSOY U. T. Ş.** (2010),Uygulamalı Veri Madenciliği Sektörel Analizler, 2012 3. Baskı, ISBN: 978-605-364-040-0
- ŞENTÜRK A.** (2006), Veri Madenciliği Kavram ve Teknikleri, ISBN: 975-8768-24-7
- PEKTAŞ A. O.** (2013), SPSS ile Veri Madenciliği, ISBN: 978-605-86660-4-7

İNTERNET KAYNAKLARI

NoSQL, (2016) NoSQL , URL1 : <http://Nosql-Database.Org>

HuMONGOus,(2016) HuMONGOus, URL1: <https://www.mongodb.com/Faq?WhereDoesTheNameMongodbComeFrom?>

MongoDB, (2016) MongoDB, URL1: <https://Docs.Mongodb.Com/Manual>

Replication, (2016) Replication, URL1: <https://Docs.Mongodb.Com/Manual/Replication>

Sharding, (2016) Sharding, URL1: <https://Docs.Mongodb.Com/Manual/Sharding>

Map-Reduce, (2016) Map-Reduce, URL1: <https://Docs.Mongodb.Com/V3.2/Core/Map-Reduce>

MongoDB, (2016) MongoDB, URL1: <https://Docs.Mongodb.Com/Manual>

ACID, (2016) ACID, URL1: <https://msdn.microsoft.com/en-us/library/Aa480356.aspx>

Sciencedirect , (2016) Sciencedirect, Adres : <http://www.sciencedirect.com>

TDK, (2016) Türk Dil Kurumu,

URL1:http://www.tdk.gov.tr/index.php?option=com_gts&kelime=mobil

Windows Forms, (2016) Windows Forms, URL1: [https://msdn.microsoft.com/en-us/library/dd30h2yb\(v=vs.110\).aspx](https://msdn.microsoft.com/en-us/library/dd30h2yb(v=vs.110).aspx)

Digital 2016, (2016) Digital in 2016, URL1: <http://wearesocial.com/uk/special-reports/digital-in-2016>

ÖZGEÇMİŞ

Gürkan AYDIN

gurkan@gurkanaydin.com

+90 533 308 26 66

MCPD,MSSQL DBA,C#, Ado.Net, .Net,Oracle,MongoDB,ElasticSearch,Nginx

AMAÇ: İnternet sektöründeki iş tecrübesini, çalıştığı kurumun kültürüyle birleştirerek, kullanıcılara ve kurumuna katkı sağlamak.

Kişisel Bilgiler:

- Bilginin paylaşıldıkça değer kazandığına inanır.
- Çözüm odaklı çalışma prensipleri edinmiştir.
- Samimi iletişim tekniklerini geliştirmiştir.
- Yaratıcı ve yenilikçi eğitim teknolojilerine adaptasyon becerisi gelişmiştir.

Başarı için eğitimin sürekli olduğuna inanır. Gürkan AYDIN yazılım teknolojisini Microsoft .NET Framework ve MSSQL çatısı üzerine kurmuştur. Bu mimaride n katmanlı yazılımlar geliştirmektedir. Bu sayede Çalıştığı projelerde windows tabanlı ve web tabanlı projelerin yanı sıra web servisi uygulamaları ile üçüncü kişilerin sistemdeki verilerden yararlanmasını sağladı. Projelerinde bilgilerimize izin verdiğimiz ölçüde şirket içi çalışanlarımız bilgisayarlarından, sahada çalışan personelimiz laptop veya pda cihazlarından, müşterilerimiz web sayfamız üzerinden sistemimize erişebileceği yapılar geliştirdi.

ÇALIŞMA DENEYİMİ

2013- : ZUGO Bilgi Sistemleri A.Ş.

MSSQL DBA

Sorumlulukları:

Renault 'un kullandığı IMPETUS yazılımı: Araç Satış , Servis ve Servis Sonrası Yönetim Sistemleri üzerine çözüm sunmaktadır. Bununla beraber sorumluluk sahasında Türkiye genelinde bulunan 100 servis ve 100 bayi bulunmaktadır. Bu 200 işletmenin server dinamiklerinin yönetimi , veri tabanı performansının iyileştirilmesi, tek bir noktadan güncelleme , yedekleme , DataWareHouse işlemlerini planlarken sağlıklı bir şekilde yürütülmesini sağlamaktadır.

Başarıları:

- Türkiye genelinde dağınık olarak bulunan sistemlerin tek bir noktadan kontrol edilerek eş zamanlı olarak güncellenmesi ve yedeklenmesi.
- DataWareHouse yazılımında mimari yönetimi.
- Sahada çalışan canlı business datasının çalıştığı süreç dahil DataWareHouse 'da toplanarak raporların çıkarılmasının sağlanması.
- DataWareHouse ETL süreç yönetimi ve sürekli tutarlılık raporunun sağlanması.
- Performans iyileştirmeleri.

2011- : Turkuvaz Medya Grubu A.Ş.

Yazılım Uzmanı

Sorumlulukları:

-Proje yazılım

Başarıları:

Sektörde yeri olan Sabah, Fotomaç, Atv gibi kurumlar adına çalışma merakı ile işe girmiş olup, ATV 'nin video stream yapısını değiştirerek, VAST entegrasyonunu sağladı.

2010-2011 : B2B

Yazılım Koordinatörü

Sorumlulukları:

- NEOBET projesinin tasarlanıp tüm yönü ile hayat geçirilmesi.
- Kurumun tüm projelerinin sağlıklı yürümesini sağladı.
- Yeni projelerin mimarilerinin tasarlayarak ilgili ekiplerin yazılım sürecini takip etti.
- Kurumun içerisinde farklı ekipler kurdu ve bu ekiplerin çalışma sistemlerini organize etti.
-

Başarıları:

Çalışanların performanslarının artması sağlandı. NEOBET projesinin hayata geçirilerek aktif çalışması sağlandı. NEOBET,Admin,Destek sitelerinin ortak çalışması sağlanarak müşterilerine sorunsuz hizmet verilmesi sağlandı.

2009-2010 : Akıllı TV A.Ş.

Web Developer

Sorumlulukları:

-Akıllı TV 'ye ait www.akilli.tv web sitesinin yazılımının, yenilenmesi ve güncellenmesini gerçekleştirdi.

-Web Sitesinin 64 Bit 'e hazırlık ve MSSQL 2008 'e adaptasyonu için gerekli çalışmaları sürdürmektedir.

-Akıllı TV adına Facebook için application geliştirmektedir.

Başarıları:

Sitenin www.Akillioyun.tv ile adaptasyonunu sağladı.

2008-2010 : Başarisoft Bilgi Teknolojileri Akademisi

Web danışmanı ve eğitmen

Sorumlulukları:

- Başarisoft Bilgi Teknolojileri Akademisi 'ne ait www.basarisoft.com web sitesinin, arama motorlarına optimizasyonunun sağlanması ve web dünyasında kurmun tanıtılması organizasyonunu gerçekleştirdi.
- Başarisoft Bilgi Teknolojileri Akademisi 'nde C#, Ado.Net, .Net, SEO ve GAP eğitimleri ve seminerler düzenlemektedir.

Başarıları:

Başarisoft'un arama motorlarında istenilen kelimelerde ilk sayfada çıkmasını sağladı.

Web' ten gelen müşteri portföyünde %35 'lik bir artış sağladı.

Katıldığı Eğitim ve Seminerler:

Google Adwords Semineri	2009-2010
Google Adwords Webinerleri	2009-2010
Microsoft Webinerleri	2009-2010
İTÜ Ağ güvenliği	2009
Başarısoft Hizmet Kalite Yönetimi	2008

Öğrenim Durumu:

2017 Yüksek Lisans	İstanbul Aydın Üniversitesi Bil. Müh. Bölümü –
2004 Lisans diploması	Anadolu Üniversitesi İktisat Bölümü -4 yıl- mezun
1998 Ön Lisans	Kocaeli Üniversitesi Elektrik Terk
1992 Lise	Tuzla Teknik Lisesi (JICA) -4yıl –mezun

Kişisel Bilgi:

Doğum yeri ve tarihi	:	İstanbul, 05-05-1974
Medeni Hali	:	Evli, İki çocuk babası
Ehliyet	:	B sınıfı, Veriliş tarihi 1992