**T.C. FIRAT ÜNİVERSİTESİ**

**MÜHENDİSLİK FAKÜLTESİ - YAZILIM MÜHENDİSLİĞİ BÖLÜMÜ**

**BİTİRME ÖDEVİ ARA RAPORU**

|  |  |
| --- | --- |
| **YILI / DÖNEMİ** | 2023-2024 DERS YILI / GÜZ DÖNEMİ |
| **ÖĞRENCİ NO** | 190290004 |
| **AD SOYAD** | Seda ÖZER |
| **BİTİRME TEZ DANIŞMANI** | Arş. Gör. Çağla DANACI |
| **PROJE KONUSU/BAŞLIĞI** | Yapay Zeka Tabanlı Müşteri Geri Bildirim İzleme Sistemi |
| **Giriş** (Projenin genel özeti ve ilerleme durumu) | |
| Bu çalışmada, yorumların detaylı bir analizini gerçekleştiren bir modelin, kullanıcı dostu bir arayüze entegre edilmesiyle elde edilen sonuçların daha açık ve belirgin bir şekilde sunularak, ortaya çıkan değerlerin anlaşılabilirliğinin artırılması amaçlanmıştır.   |  |  |  |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | | Hafta iş Modülleri | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | | Veri setinin hazırlanması |  |  |  |  |  |  |  |  |  |  | | Veri ön işleme |  |  |  |  |  |  |  |  |  |  | | Eğitim |  |  |  |  |  |  |  |  |  |  | | Sınıflandırma modellerinin denenerek kullanılabilir hale getirilmesi |  |  |  |  |  |  |  |  |  |  | | Proje arayüzünün tasarlanması |  |  |  |  |  |  |  |  |  |  | | Kaydedilen modelin arayüze entegre edilmesi |  |  |  |  |  |  |  |  |  |  | | Bitirme projesi kitapçığının yazılması |  |  |  |  |  |  |  |  |  |  | | Sonuç Değerlendirmesi |  |  |  |  |  |  |  |  |  |  |   **Şekil 1.** Proje İş Akış Şeması  Şekil 1’ de, projenin iş akış şeması bulunmaktadır:   * Proje iş akış şemasında yeşil renge sahip kısımlar projede bitmiş olan kısımları göstermektedir. * Sarı renge sahip olan kısımlar, bir kısmının yapılıp diğer kısımlarına ise hala devam edilmekte olanları göstermektedir. * Siyah renge sahip olan kısımlar ise, daha başlanmamış fakat en kısa zamanda başlanacak olan kısımları simgelemek için konulmuştur.   Proje iş akış şemasında tamamlanan kısımlar aşağıdaki gibidir:   1. Veri setinin hazırlanması 2. Veri ön işleme 3. Eğitim (Devam Edilmekte) 4. Sınıflandırma modellerinin denenerek kullanılabilir hale getirilmesi (Devam Edilmekte) 5. Proje arayüzünün tasarlanması (Devam Edilmekte) 6. Kaydedilen modelin arayüze entegre edilmesi (Devam Edilmekte) 7. Bitirme projesi kitapçığının yazılması (Devam Edilmekte) | |
| **Modül 1:** Veri setinin hazırlanması | |
| Proje için yorumlar içeren ve duygu kategorileri olan veri setleri incelenmiştir. Bu aşamada veri çekme işlemleri yapılmıştır. Ayrıca hazır veri setleri araştırılmıştır.  Hangi veri setinin kullanılacağına karar vermek için aşağıdaki 5 duruma bakılmıştır.   * İşlenebilirlik ve format * Kalite ve güvenilirlik * Çeşitlilik ve temsil * Temizlik ve Format * Boyut ve ölçek       **Şekil 2.** Veri setinin görünümü  Proje kapsamında, yorumlar içeren ve duygu kategorilerine ayrılmış veri setlerinin incelenmesi gerçekleştirildi. Bu aşamada, kullanılabilirlikleri ve uygun formatları değerlendirilerek işlenebilirlikleri göz önüne alındı. Ayrıca, güvenilir kaynaklardan elde edilmiş ve kaliteli verilere odaklanıldı. Veri setlerinin çeşitliliği ve geniş bir örneklemi temsil edip etmedikleri incelenirken, temizlik ve düzenlilik de dikkate alındı. Son olarak, projenin gereksinimlerini karşılayacak ölçekte ve boyutta veri setlerinin belirlenmesi için bir değerlendirme yapıldı. | |
| **Modül 2:** Veri ön işleme | |
| Projenin kullanacağı veri seti standart bir formata getirildi. Yorumlardaki HTML etiketleri temizlendi ve sadece harf karakterleri korundu. Ardından, metinde bulunan büyük harfler küçük harflere dönüştürüldü ve yorumlar kelimelere ayrılarak yaygın olarak kullanılan kelimeler içinde yer alan stopwordsler çıkarıldı. Bu adımlar, veri setinin uygunluğunu artırmak ve daha tutarlı bir analiz yapabilmek için gerçekleştirildi.  Veri ön işleme, projenin kullanacağı veri setinin niteliğini iyileştirmek ve analiz için daha uygun hale getirmek amacıyla gerçekleştirilmiştir. Yukarıda bahsedilen adımlar, bu ön işleme sürecinin önemli adımlarını temsil etmektedir. HTML etiketlerinin temizlenmesi ve yalnızca harf karakterlerinin korunması, veri setinin gereksiz gürültülerden arındırılmasına ve okunabilirliğinin artırılmasına yardımcı olur. Büyük harflerin küçük harflere dönüştürülmesi, veri uyumunu sağlamak ve analiz aşamasında tutarlılık sağlamak için yapılmıştır. Stopwordslerin çıkarılması ise, analizin odaklandığı kelimelerin daha anlamlı olmasını sağlayarak veri setinin kalitesini artırır. Bu ön işleme adımları, veri setinin daha sağlam, temiz ve analize hazır hale gelmesini sağlar, böylece projenin doğruluğu ve güvenilirliği artar. Bu süreç, veri bilimi ve analiz çalışmalarında verilerin güvenilir ve anlamlı sonuçlar üretmesi için kritik bir adımdır. | |
| **Modül 3:** Eğitim | |
| Tablo oluşturma amacıyla metin verilerinin özellikleri “Xfeatures” ve etiketleri “ylabels” belirlenir. train\_test\_split fonksiyonu, veri setini eğitim ve test kümelerine bölmek için kullanılır. Bu işlem, verilerin %70'ini eğitim için kullanmak üzere ayırırken, %30'unu test etmek üzere ayrılmış bir veri kümesi oluşturur. random\_state parametresi, rastgele bir durumu yeniden oluşturmak için kullanılır ve tekrarlanabilirlik sağlar. Daha sonra, Pipeline kütüphanesinden CountVectorizer() ve LogisticRegression() sınıflarını kullanarak bir dizi işlem oluşturulur. CountVectorizer(), metin verilerini sayısal vektörler haline dönüştürmek için kullanılır. Bu süreçte metin verileri, kelime sayılarına dayalı olarak sayısal bir temsile çevrilir.    Bu süreçte metin verileri, kelime sayılarına dayalı olarak sayısal bir temsile çevrilir. LogisticRegression(), sınıflandırma modelini oluşturur ve eğitir. Bu süreçte metin verileri, kelime sayılarına dayalı olarak sayısal bir temsile çevrilir. LogisticRegression(), sınıflandırma modelini oluşturur ve eğitir.Pipeline oluşturulduktan sonra, fit fonksiyonu kullanılarak eğitim verilerine (x\_train ve y\_train) model eğitimi yapılır. Ardından score fonksiyonu, oluşturulan modelin test veri seti (x\_test ve y\_test) üzerindeki doğruluğunu değerlendirir. Sonuç olarak, bu kod parçası metin verilerini sayısal vektörlere dönüştürerek bir sınıflandırma modeli oluşturur, eğitir ve bu modelin test veri setindeki doğruluğunu ölçer [1].    **Şekil 3.** Eğitim kodları  Bu süreçte, metin verileri sayısal vektörlere dönüştürülerek bir sınıflandırma modeli oluşturulmuş ve eğitilmiştir. Verilerin özellikleri ve etiketleri belirlendikten sonra, train\_test\_split fonksiyonuyla veri seti eğitim ve test kümelerine ayrılmıştır. Bu ayrım, verilerin %70'ini eğitim için kullanırken %30'unu test etmek üzere ayrılmış bir veri kümesi oluşturur. Ardından, CountVectorizer() ve LogisticRegression() gibi teknikler ve sınıflandırma algoritmaları kullanılarak bir dizi işlem yapılmıştır. CountVectorizer(), metin verilerini sayısal vektörlere dönüştürürken, LogisticRegression() sınıflandırma modelini oluşturur ve eğitir.  Pipeline kullanılarak oluşturulan bu model, eğitim verileri üzerinde fit işlemi ile eğitilmiştir. Daha sonra, bu modelin test veri seti üzerindeki doğruluğu score fonksiyonuyla değerlendirilmiştir. Elde edilen sonuçlar, metin verilerinin sayısal temsile dönüştürülerek oluşturulan sınıflandırma modelinin, test veri setindeki başarısını ölçmekte kullanılmıştır. Bu süreç, metin tabanlı verilerin işlenmesi ve sınıflandırılması için kullanılan önemli adımları içermektedir. | |
| **Modül 4:** Sınıflandırma modellerinin denenerek kullanılabilir hale getirilmesi | |
| Logistik regresyon, random forest ve destek vektör makineleri (SVM) gibi farklı sınıflandırma modelleri, bu projede kullanılarak metin verilerinin sınıflandırılması sağlanmıştır. Bu farklı modellerin kullanımı, çeşitli avantajlar ve teknikler sunar.  **Logistik Regresyon**: İkili sınıflandırma için yaygın olarak kullanılan bir yöntemdir. Lineer bir model olup, veri seti üzerinde lineer bir karar sınırı oluşturarak sınıflandırma yapar. Basit ve yüksek derecede yorumlanabilir olması, modelin anlaşılabilirliğini artırır. Ancak, karmaşık ilişkileri modelleme konusunda sınırlı olabilir [2].  **Random Forest:** Karar ağaçlarından oluşturulan bir ansambldır. Her bir ağaç, veri setinin farklı alt kümeleri üzerinde eğitilir ve sonuçları birleştirilerek sınıflandırma yapılır. Bu model, aşırı uyum sorununu azaltır, değişkenler arasındaki ilişkileri yakalamak için daha esnektir ve genellikle yüksek doğruluk sağlar [3].  **Destek Vektör Makineleri (SVM):** Veri noktalarını sınıflandırmak için bir karar sınırı oluştururken, veri noktaları arasındaki en geniş marjı bulmaya çalışır. Özellikle doğrusal veya doğrusal olmayan veri setleri üzerinde etkilidir. Düşük boyutlu veri setlerinde etkili ve ölçeklenebilir olması, avantajlarındandır [4].  Bu modeller, metin verilerinin sınıflandırılmasında farklı yaklaşımlar sunar. Logistik regresyon basit ve anlaşılabilir bir model sağlarken, random forest esnekliği ve doğruluk avantajı sunar. SVM ise karar sınırlarını belirlemede etkilidir ve özellikle sınıflar arasındaki marjları en üst düzeye çıkarmayı hedefler.    **Şekil 4.** Svm modelinin kullanımı | |
| **Modül 5:** Proje arayüzünün tasarlanması | |
| Bu arayüz, kullanıcı tarafından girilen metinleri duygu kategorilerine göre sınıflandırıyor. Bu sınıflandırma sonucunda metnin hangi duygu kategorisine ait olduğunu gösteriyor. Ayrıca, metnin sınıflandırılmasında kullanılan modelin katsayılarını ve bu katsayıların grafiksel gösterimini sunuyor. Metinlerin duygu analizini yapmak ve sınıflandırma sonuçlarını kullanıcıya anlatmak için oldukça etkili bir şekilde kullanılabilir. Kullanıcıların metinlerini analiz ederken duygusal içeriklerini daha iyi anlamalarına yardımcı olabilir. Bu arayüz, kulla  nıcı tarafından girilen metinleri duygu kategorilerine göre sınıflandırıyor. Bu ve sını    **Şekil 5.** Arayüz genel görünümü | |
| **Modül 6:** Kaydedilen modelin arayüze entegre edilmesi | |
| **Şekil 6.** Arayüz entegrasyonu  Bu aşamada, Streamlit ve çeşitli veri analizi kütüphanelerini içeren Python paketlerini içerir. Bu paketler, metin verileri üzerinde duygu sınıflandırma yapmak için önceden eğitilmiş bir modelin entegrasyonunu sağlar. Streamlit, kullanıcı arayüzü geliştirmeyi ve veri analizi sonuçlarını görselleştirmeyi kolaylaştıran bir araçtır. Öncelikle, veri analizi ve işleme için gerekli olan Pandas, NumPy ve datetime gibi kütüphaneler çağrılır. Ardından, duygu sınıflandırma modelini yüklemek ve kullanmak için joblib kütüphanesi kullanılır.  Belirli bir metin belgesinin duygusunu tahmin etmek için gerekli olan fonksiyonları içerir. `predict\_emotions` fonksiyonu, belirtilen metin belgesinin duygusunu tahmin etmek için modeli kullanırken, `get\_prediction\_proba` fonksiyonu her bir duygu sınıfı için tahmini olasılıkları döndürür. Kullanıcı arayüzü üzerinde bir metin belgesinin duygu analizini gerçekleştirmeyi sağlar. Kullanıcılar, metin girişi yaparak belirli bir duygunun tahminini alabilir veya her bir duygu için tahmini olasılıkları görebilirler [5]. | |
| **Modül 7:** Bitirme projesi kitapçığının yazılması | |
| Bu aşamada için dokümanlar hazırlanmaktadır. Yapılan aşamalarla şekillenerek proje ile beraber yazılmaya devam ediliyor. | |
| **KAYNAKLAR** (Projede yer alan tüm kaynakların referans numaraları ile listesi) | |
| **[1]**<https://learn.microsoft.com/tr-tr/azure/architecture/ai-ml/>  **[2]**<https://www.researchgate.net/publication/297604670_Lojistik_Regresyon_Analizinin_Kullanildigi_Makaleler_Uzerine_Bir_Inceleme>  **[3]** <https://medium.com/yaz%C4%B1l%C4%B1m-ve-bili%C5%9Fim-kul%C3%BCb%C3%BC/random-forests-92fd17d9aa4f>  **[4]**  <https://medium.com/@hilaloztemel/svm-support-vector-machine-destek-vekt%C3%B6r-makinesi-442808f8596>  **[5]** <https://docs.streamlit.io/streamlit-community-cloud/get-started> | |