

T.C.
SAKARYA ÜNİVERSİTESİ
BİLGİSAYAR VE BİLİŞİM BİLİMLERİ FAKÜLTESİ

BSM 401 BİLGİSAYAR MÜHENDİSLİĞİ TASARIMI

YAPAY ZEKA TABANLI
YEMEK KALORİ TAHMİNİ

B201210088 – Muammer KARAKAŞ

Bölüm : **BİLGİSAYAR MÜHENDİSLİĞİ**
Danışman : **Dr.Öğr.Üyesi ULAŞ YURTSEVER**

2024-2025 Güz Dönemi

ÖNSÖZ

Günümüz yaşam tarzı, bireylerin sık sık dışarıda yemek yemesini gerektirebiliyor. Ancak, bu durumun beraberinde getirdiği önemli bir sorun var: tüketilen yiyeceklerin içerdiği kalori miktarı hakkında yeterli bilgiye sahip olunmaması. Bu bilinçsizlik, zamanla obezite, aşırı kilo alımı ve beraberinde çeşitli sağlık problemlerine yol açabiliyor. Özellikle yoğun bir yaşam temposunda, bireylerin sağlıklı seçimler yapabilmeleri daha da zor hale geliyor. Diğer yandan; seyahat halinde olan insanlar geleneksel, dünya çapında ünlü yemekleri deneyimliyor. Ancak bazılarının kalori bilgisine ulaşmak mümkün olmuyor. Bu yüzden modelimizi küresel anlamda oluşturarak her kesime ulaşmasını amaçlıyoruz.

Bu çalışma ile bilinçli beslenmeyi teşvik ederek, sağlıklı bir toplum yapısına katkıda bulunmayı hedefliyoruz. Herkesin daha iyi tercihler yapabilmesine yardımcı olmak ve bu süreci daha erişilebilir hale getirmek bu projenin temel motivasyonudur.

İÇİNDEKİLER

ÖNSÖZ.....	ii
İÇİNDEKİLER.....	iii
ŞEKİLLER LİSTESİ.....	v
ÖZET.....	vi
BÖLÜM 1.	
GİRİŞ.....	1
1.1.Projenin amacı.....	1
1.2.Çözülmek istenen problemin önemi.....	2
1.2.1. Literatür ve mevcut çözümler.....	2
BÖLÜM 2.	
PROBLEM TANITIMI.....	4
2.1 Problem nedir? Neden kaynaklanır?.....	4
2.1. Problemin çözüm yaklaşımı.....	5
BÖLÜM 3.	
KULLANILAN YAZILIMLAR , ALGORİTMALAR.....	6
3.1. Kullanılan yazılımlar.....	6
3.2. Algoritmalar	7
3.2.1 RandomForest Algoritması.....	7
3.2.2 RandomForestClassifier.....	9
3.2.3 RandomForestRegressor.....	9
BÖLÜM 4. GELİŞTİRİLEN ÇÖZÜM.....	10

4.1. Verisetleri	10
4.1.1 Eğitim veriseti.....	10
4.1.2 Test veriseti.....	11
4.1.3 Kalori veriseti.....	12
4.2. Kullanılan algoritmaların çözümü ve kodlar.....	14
 BÖLÜM 5. TEST VE SONUÇLAR.....	19
5.1. Eğitim ve Test doğruluğu çıktıları.....	19
5.2. Kalori tahmin sonuçları.....	20
5.3. Test sonuçları CSV kaydı.....	20
5.4. Sınıflandırma raporu.....	20
5.5. Genel Çıktılar.....	21
 KAYNAKLAR.....	22
 ÖZGEÇMİŞ.....	23
 BSM 401 BİLGİSAYAR MÜHENDİSLİĞİ TASARIMI DEĞERLENDİRME VE SÖZLÜ SINAV TUTANAĞI.....	

ŞEKİLLER LİSTESİ

Şekil 4.1.	Verisetine ait örnek 1.....	10
Şekil 4.2.	Verisetine ait örnek 2.....	11
Şekil 4.3.	Verisetine ait örnek 3.....	12
Şekil 4.4.	Kullanılan kütüphaneler, algoritmalar.....	13
Şekil 4.5.	Kalori veri setini yükleme.....	13
Şekil 4.6.	Görsellerden özellik çıkarma.....	14
Şekil 4.7.	Verinin hazırlanması.....	14
Şekil 4.8.	Eğitim, epoch ayarları.....	14
Şekil 4.9.	Kullanılan algoritma tanımı.....	15
Şekil 4.10.	Eğitim döngüsü.....	15
Şekil 4.11.	Kalori tahmini ve hesaplama.....	16
Şekil 4.12.	Test sonuçları ve analizi.....	17
Şekil 5.1.	Eğitim ve test doğruluğu çıktıları.....	18
Şekil 5.2.	Kalori tahmin sonuçları.....	18
Şekil 5.3.	Test sonuçları csv kaydı.....	19
Şekil 5.4.	Sınıflandırma raporu.....	19

ÖZET

Anahtar kelimeler: Derin Öğrenme, Veri seti, Görüntü İşleme

Belirlediğimiz veri setinin görsel formda olmasından dolayı belirli algoritmalara göre işlemeyen geçerek bitsel olarak bilgisayarın anlayabileceği forma dönüştürülür. Bu süreç görüntü işleme olarak adlandırılır. Veri setinin eğitim ve test olmak üzere oluşturulması sağlanır, eğitim veri setini kullanarak modelin eğitilmesi ve sonrasında test veriseti kullanılarak test edilir. Bu işlem model öğrenene kadar devam eder. Bu süreç derin öğrenme olarak tanımlanabilir.

Bu tasarım çalışması ile amaçlanmış olan yemeklerin kalori tahminini yapay zeka teknolojileri kullanarak gerçekleştirdik. Ayrıca, sonradan düzenlemelere de uygun şekilde oluşturduk. Veri setine yeni veriler eklenebilir, değiştirilebilir. Bu sayede proje değişimlere açık ve güncel olacaktır.

BÖLÜM 1. GİRİŞ

1.1. Projenin Amacı

Geliştirilecek olan sistem; restoranlar, spor kurumları, diyetisyenler, sağlık kuruluşları veya bireysel kullanıcılar tarafından sağlıklı beslenme ve kalori takibi yapmalarını kolaylaştırmak amacıyla kullanılabilir. Bu sayede daha sağlıklı ve bilinçli bir toplum olmayı amaçlayarak projeyi hayata geçirmeyi arzu ediyorum.

Diğer yandan ülke ülke gezen, seyahat eden insanların da bu yeniliklerden faydalanabilmesi adına; yalnızca yerel yemeklerimizi değil, dünya çapında bilinen geleneksel yemekleri de projeye dahil etme gereği hissettim. Böylece insanların uygun menüyü seçmesine katkıda bulunmuş olacağım.

Bu proje tasarım aşamasında olduğundan henüz kullanıcı ile sistem arasında bağlantı kurmadım. Mevcut sistemimiz şu anda belirlediğimiz veri setine bağlı olarak öğrenmekte ve sonucu getirmektedir. Mobil uygulama geliştirilerek kullanıcıya ulaştırılması amaçlanmaktadır.

1.2. Çözölmek istenen problemin önemi

Bu problemin neden gerekli olduđunu şöyle izah edebiliriz; daha önce hiç gitmediđimiz bir ölkenin çok bilinen yemeđini deneyimlemek istiyoruz. Alınan verilen kaloriyi dengede tutmamız gerekiyor. İnternetteki bilgilerde de tutarsızlık, büyük oranda sapma olabiliyor. Bu yüzden geliřtirdiđimiz sistem sayesinde çok kısa bir sürede bilgi sahibi olarak menüyü kolayca belirleyebiliriz.

Öte yandan işletmelerin de geliřtirilen sistem sayesinde sunduđu gıdaları kalori deđerlerine göre ayırabilir, müşterinin kalori gereksinimine göre porsiyon miktarını kolayca oluřturup müşteriye sunabilir. Bunun gibi iyileřtirmeler sayesinde müşteri-esnaf ilişkisinin geliřmesine katkı sađlamayı da umuyorum.

1.2.1. Literatür ve mevcut çözümler

Piyasada aynı amaç için geliřtirilen mevcut çözümler bulunmaktadır. Bunlardan bazıları doğrudan yemek menüsünü tahmin etme üzerine ,bazıları kalorilerini hesaplama üzerine düşünölmüřtür.

Kaynaklarda belirttiđim bir makale örneğinde řunlar bulunur;

Görüntüden Kalori Farkında Otomatik Yemek Kiti Üretimi

Kalori ve beslenme arařtırmaları son yıllarda artan bir ilgi görmüřtür. Ancak, sorunun karmařıklığı nedeniyle, bu alandaki literatür sınırlı bir bileřen veya yemek türü alt kümesine ve basit evriřimli sinir ađlarına veya geleneksel makine öđrenimine odaklanmaktadır. Aynı zamanda, bileřen porsiyonlarının tahmini, belirli bir görüntüden kalori tahminini ve öđün yeniden üretimini iyileřtirmeye yardımcı olabilir. Bu makalede, tek bir piřirme görüntüsü verildiğinde, yemeđin farklı porsiyonları için kalori tahmini ve öđün yeniden üretimi için bir boru hattı önerilmektedir. Boru hattı iki aşamadan oluřmaktadır. İlk aşamada, verilen görüntüdeki yemekle ilişkili bir dizi bileřen tahmin edilir. İkinci aşamada, verilen görüntü özellikleri ve bileřenler, bileřenlerin porsiyonları ve son olarak toplam öđün kalorisi, derin bir dönüřtürücü tabanlı model kullanılarak aynı anda tahmin edilir. Modelde tanıtılan porsiyon tahmini, kalori tahminini iyileřtirmeye yardımcı olur ve

ayrıca farklı porsiyon boyutlarında öğün yeniden üretimi için faydalıdır. Boru hattının faydalarını göstermek için, model yemek kitleri oluşturmak için kullanılabilir. Boru hattını değerlendirmek için büyük ölçekli veri seti Recipe1M kullanılır. Deneylerden önce, Recipe1M veri seti ayrıştırılır ve açıkça malzemelerin bölümleriyle açıklanır. Deneyler, malzemelerin ve bunların bölümlerinin kullanılmasının kalori tahminini önemli ölçüde iyileştirdiğini gösterir. Ayrıca, bir kullanıcının boru hattıyla etkileşime girerek doğru kalori tahminlerine ulaşabileceği ve yemek pişirme amaçlı bir yemek seti oluşturabileceği görsel bir arayüz oluşturulur.

Bu çalışma, bir yemek görüntüsünden kalori tahmini ve yemek kitleri oluşturma üzerine odaklanmaktadır. Önerilen yöntem, görüntüden yemeğin içeriğini ve porsiyonlarını tahmin ederek toplam kalori miktarını hesaplamaktadır.

Başka bir çalışmada görüntü işleme kullanılarak bir tahmin yapılmaktadır.

Bilgisayarla Görme Destekli Yemek Takip Uygulamaları

Yemek takibinde bilgisayarla görmenin en popüler uygulamalarından biri mobil uygulamalardır.

SnapCalorie, bir fotoğraftan kalori içeriğini ve makro besin öğelerini tahmin etmek için bilgisayar görüşü kullanan bir uygulamadır. 5.000 öğün üzerinde eğitilen uygulama, kalori tahmin hatalarını %20'nin altına indiriyor ve çoğu insandan daha iyi performans gösteriyor. Sonuçlar bir yemek günlüğüne kaydedilebilir veya Apple Health gibi fitness platformlarına aktarılabilir.

Benzer şekilde, AI beslenme takibini yönlendiren ilginç bir yenilik de LogMeal API'dir. Gıdaları doğru bir şekilde tespit etmek ve tanımak için büyük gıda görüntüleri veri kümeleri üzerinde eğitilmiş derin öğrenme algoritmaları kullanır. LogMeal'in modelleri 1.300 yemekte %93 doğruluk oranına ulaşıyor ve ayrıntılı besin analizi, içerik tespiti ve porsiyon boyutu tahmini sağlıyor. LogMeal API, restoranlar, kendi kendine sipariş kioskuları, gıda teknolojisi girişimleri, sağlık hizmeti sağlayıcıları ve diğer tüketiciler için yemek izleme çözümleri oluşturmak üzere uygulamalara kolayca entegre edilebilir.

BÖLÜM 2. PROBLEM TANITIMI

2.1. Problem nedir? Problem neden kaynaklanır?

Dünyada bildiğimiz, bilmediğimiz binlerce çeşit yemek, yiyecek bulunmakta. Yeni yeni yemekler de ortaya çıkmaya devam ediyor. Bu durumun getirdiği bazı sorunlar şunlardır; insanlar (özellikle seyahat halindeki olanlar) daha önce yemediği, adını bile bilmediği bir yemeğin sırf görüntüsüne bakarak o yemeği deneyimlemek ister. Ancak kültürel farklılıklar, insanlarla iletişim zorluğu gibi etmenlerden dolayı yemek hakkında bilgi sahibi olmak zordur. Öte yandan bazı kişilerin tükettiği gıdaların kalorisine ekstra dikkat etmesi gerekebilir. Ya da bazı kişilerin bazı türdeki yemekleri tüketmesi dini inançlarına göre uygun olmayabilir. Alerjik etmenler olabilir. Vegan, vejetaryen olan kişiler olabilir ve yemeğin dış görünüşünden anlaşılmayabilir. Bu tarz hassasiyetleri olan insanlar yiyeceği yemek hakkında bilgi sahibi olmak ister. Problem de burada ortaya çıkar.

Problemin kaynağı, yaşanan zorluklar (Kültürel farklılık, iletişim zorluğu, beslenme biçimi ,dini etken, sağlık sorunları vb) sebebiyle insanlar, deneyimlemek istediği yemek hakkında zor bir şekilde bilgilendirilir veya bilgi sahibi olamaz. Bunun sonucunda yemeği yemekten vazgeçer.

Problemin ortaya çıkmasındaki başka bir sebep; çeşitli sebeplerden dolayı veya keyfi olarak insanların bir kısmı çoğu zaman dışarıdan besleniyor. Uzun vadede düşünüldüğünde sağlıklı kalmak için aldıkları kalori miktarını dengede tutmak zorundalar. Bu durumda olan kişiler bunun bilincindeler ancak değişen menüler ve porsiyon miktarlarından dolayı tükettikleri kaloriyi hesaplaması, dengeyi koruması neredeyse imkansız hale gelir. Bunun sonucunda ya sağlık sorunları ya da fiziksel bozukluklar oluşabilir.

2.2. Problemin Çözüm Yaklaşımı

Problemin çözümündeki temel yaklaşım insanların yaşadığı zorlukları gidermek, kolaylaştırmak esas alınarak oluşturulmuş oldu. Geliştirilen bu sistem sayesinde insanlar nerede olursa olsun daha sonra eklenecek olan mobil uygulama vasıtasıyla yemeği kolayca öğrenebilecek , kalori değerlerini porsiyon boyutu olarak görebilecek. Bu sayede oluşan sorunları önemli ölçüde azaltarak projenin amacına ulaşmasını sağlayacağız.

BÖLÜM 3. KULLANILAN YAZILIMLAR ALGORİTMALAR

3.1. Kullanılan Yazılımlar

Projenin kodlanmasını yapacağım uygulama Python dilini de destekleyen Visual Studio Code yazılımıdır. Bununla birlikte Python kullanımı için gerekli olan Python 3.13.1 versiyonunu kullanıyorum. Piyasaya yeni sürüldüğünden dolayı çok kullanılan bazı kütüphanelerin desteği olmadığından bunların alternatiflerini kullandım.

Kullandığım kütüphane yazılımları ;

Numpy Python programlama dili için büyük, çok boyutlu dizileri ve matrisleri destekleyen, bu diziler üzerinde çalışacak üst düzey matematiksel işlevler ekleyen bir kitaplıktır.

Pandas veri işlemesi ve analizi için Python programlama dilinde yazılmış olan bir kütüphanedir.

PIL (Python Imaging Library) Python programlama dili için geliştirilen, açık kaynak kodlu grafik işleme kütüphanesidir. Bu kütüphaneyi opencv-python desteklenmediği için alternatif olarak kullanmayı tercih ettim.

Scikit learn Hem denetlenen hem de denetlenmeyen öğrenme için çok çeşitli araçlar sunarak makine öğrenimini daha erişilebilir hale getiren açık kaynaklı bir Python kütüphanesidir. Bununla birlikte, veri analizi ve tahmini modelleme gibi görevler çok daha kolay hale gelir. Bu kütüphaneyi de **Tensorflow** kütüphanesi henüz desteklenmediği için alternatif olarak tercih ettim. Derin öğrenme için oldukça önemlidir.

Seaborn, Python'da istatistiksel grafikler oluşturmak için bir kütüphanedir. **Matplotlib**'in üzerine kuruludur ve pandas veri yapılarıyla yakından bütünleşir.

3.2. Algoritmalar

3.2.1 RandomForest Algoritması: Makine öğreniminde genellikle sınıflandırma ve regresyon görevleri için kullanılan, güçlü ve esnek bir öğrenme yöntemidir.

RandomForest, karar ağaçları (decision trees) topluluğudur. Tek bir karar ağacının doğruluğu sınırlı olabilir; ancak birden fazla karar ağacını birleştirerek hataları azaltmak ve genel performansı artırmak mümkündür.

RandomForest, bu prensipten yola çıkarak, Bootstrap Aggregating (Bagging) yöntemini kullanır:

Bootstrap Aggregating: Aynı veri kümesinden farklı örneklemeler alarak birden fazla model eğitme ve tahminleri birleştirme yöntemidir. Bu algoritmayı seçmemin en önemli nedeni budur çünkü oldukça büyük veriseti kullanıyorum.

Algoritmanın çalışma adımları:

1.Adım: Örnekleme (Bootstrap sampling)

Veri kümesinden rastgele örnekler alınır. Her bir örnekleme seti, orijinal veri kümesinden geri koymalı (with replacement) olarak seçilir. Yani bazı veriler birden fazla kez seçilebilir, bazıları ise hiç seçilmeyebilir. Böylece ezberlemeyi azaltmayı sağlıyoruz.

2.Adım: Karar Ağaçlarının Eğitimi

Her örnekleme seti üzerinde bir karar ağacı eğitilir. Bu süreçte:

Her bir ağaç, yalnızca bir alt küme özellik (feature) kullanılarak oluşturulur. Özelliklerin alt kümesi her düğümde rastgele seçilir.

Ağaçlar, genellikle tam derinliklerine kadar büyütülür (budama yapılmaz), çünkü bu yöntem aşırı öğrenmeyi (overfitting) topluluk yapısıyla dengelemeyi amaçlar.

3.Adım: **Tahminlerin Birleştirilmesi**

Sınıflandırma: Her ağacın tahmini bir oy olarak değerlendirilir ve çoğunluk oyu alınarak sınıf etiketi belirlenir (majority voting).

Regresyon: Ağaçların tahminlerinin ortalaması alınır.

Bu algoritmayı kullanmanın sebepleri şu avantajlarının olmasıdır;

Genel Performans: Karmaşık veri kümelerinde yüksek doğruluk sağlayabilir.

Aşırı Öğrenme Dayanıklılığı: Karar ağaçlarının aşırı öğrenme eğilimini azaltır.

Özellik Seçimi ve Önemi: RandomForest, hangi özelliklerin daha önemli olduğunu belirlemek için kullanılabilir.

Esneklik: Hem sınıflandırma hem de regresyon görevleri için uygundur.

Paralel İşlem: Karar ağaçları bağımsız olarak eğitildiği için paralel işlemeye uygundur.

Tabi avantajları olduğu gibi dezavantajları da olmaktadır. Ancak proje gereksinimleri düşünüldüğünde göz ardı edilebilir. Dezavantajları başlıca şunlardır;

Bellek ve Hesaplama Maliyeti: Çok sayıda ağaç eğitmek büyük veri kümelerinde pahalı olabilir.

Karmaşıklık: Daha az yorumlanabilir bir modeldir, çünkü çok sayıda ağaç ve düğümden oluşur.

Overfitting: Yetersiz örnekleme veya özelleştirilmiş ayarlar olmadığında hala overfitting'e yani aşırı öğrenmeye neden olabilir. Ancak bu çok nadir gerçekleşmektedir.

RandomForest algoritmasına bağlı olan Scikit-learn kütüphanesinde sunulan RandomForestClassifier ve RandomForestRegressor algoritmalarını kullandım. Bu

algoritmalar sırasıyla Sınıflandırma ve Regresyon problemleri için kullanılır. Bu iki algoritmayı şöyle detaylandırabiliriz;

RandomForestClassifier ; adından da anlaşılacağı üzere sınıflandırma problemleri için kullanılır ve verileri önceden belirlenmiş sınıflara ayırır. Peki bu algoritma nasıl çalışmakta?

Karar Ağaçları: Her ağaç, sınıflandırma kararları alır. Örneğin, "Bu resim verisi hangi sınıfa ait?"

Çoğunluk Oyu (Majority Voting): Toplulukta yer alan her ağacın tahmini bir oy olarak değerlendirilir ve en fazla oyu alan sınıf, modelin tahmini olarak seçilir.

Örneğin, 100 ağaçtan 70'i "sınıf A" diyorsa, sonuç "sınıf A" olacaktır. Buradaki sınıf A ,projemizdeki tahmin edilen yemeğin ismine karşılık gelmektedir.

Bu işlemlerin sonucunda model, her bir veri noktası için sınıf etiketi tahmini döndürür. Etiket dosyasına yazdırma yapmada büyük rol oynar.

RandomForestRegressor: regresyon problemleri için kullanılır ve sürekli değerlerin tahmin edilmesini sağlar. Bu neden gerekli diyecek olursak, tahmin etme işlemini sürekli yaparak ortalama tahmin istatistiği elde etmektir. Bu sayede modelimiz test edilirken olmasını istediğimiz sonucu üretecektir. Nasıl çalışır?

Karar Ağaçları: Her ağaç, hedef değişkenin (sürekli bir değer) tahmini için eğitilir.

Ortalama Alma: Topluluğa dahil tüm ağaçların tahminlerinin ortalaması alınarak nihai sonuç hesaplanır. Bu string değer de olabilir. Float bir değer de olabilir.

(yemek tahminleri için string, kalori tahminleri için integer veya float)

Bu işlemlerin sonucunda model, her bir veri noktası için sürekli bir değer tahmini döndürür. Bununla birlikte ortalama hata oranı da MSE türünde hesaplanabilir.

BÖLÜM 4. GELİŞTİRİLEN ÇÖZÜM

Bu kısımda projede kullandığım verisetlerini ve oluşturduğum kodları görsel içeriklerle destekleyerek ayrıntılı bir şekilde açıklayacağım.

4.1. Kullanılan Verisetleri

Proje 3 önemli veriseti kullanılarak oluşturulmuştur. Her birinin oldukça önemli kullanım amacı bulunmaktadır.

4.1.1 Eğitim Veriseti (YemekEğitimDs)

Kullanım Amacı:

Modelin, farklı yemek sınıflarını öğrenmesi için kullanılır. Her bir yemek sınıfını temsil eden farklı görüntüler bu veri setinde yer alır.

Kullanım şekli:

Bu verisetindeki her görsel, bir alt klasör içinde saklanmış olup klasör adları yemek sınıflarını belirtmektedir (örneğin, Pizza, Burger, Salad gibi). Görseller boyutlandırılarak sabit bir çözünürlüğe (128x128 piksel) getirildi ve her bir görselin özellikleri bir vektör haline dönüştürüldü. Bu dönüştürme işlemi kullandığım RandomForest algoritmasının yemek sınıfını öğrenmesi için yapıldı.

Öğrenme işleminin başarımını artırmak için tüm olası senaryolara ait görsel bu veriseti içerisine dahil edildi. Bundan dolayı oldukça büyük bir veriseti oldu.100 farklı yemek çeşidinde her yemek türü için 700 örnek olmak üzere toplamda 70000 örnek bulunmaktadır.

Önemi:

Modelin temel öğrenme kaynağıdır. Yemek sınıflarının doğru tahmin edilmesi, eğitim verisinin çeşitliliğine ve kalitesine bağlıdır. Görsellerin farklı açılardan ve koşullardan alınmış olması, modelin genelleştirme yeteneğini artırır.

Eğitim verisetine ait bir örnek aşağıda yer almaktadır.



Şekil 4.1 Verisetine ait örnek yemek sınıfı

4.1.2. Test Veriseti (YemekTest)

Kullanım amacı:

Modelin performansını değerlendirmek için kullanılır. Eğitim sürecinde kullanılmayan farklı görsellerden oluşur ve modelin genelleştirme yeteneğini test eder.

Kullanım şekli:

Test veriseti, eğitim verisetiyle benzer şekilde hazırlandı. Görseller, rastgele seçilerek tahmin sürecinde modelin doğruluk oranını ölçmek için kullanıldı. Rastgele seçilmesi şundan dolayı önemlidir. Modelin öğrenme işlemini bütüne bakarak yapmasını sağlamak başarımı artıracaktır. Bunu görmenin en iyi yolu da test etme işlemini rastgele belirlemesidir.

Modelin tahmin ettiği sınıf, test verisetindeki gerçek etiketlerle karşılaştırılarak doğruluk oranı hesaplanmasıyla elde edilen başarımı görmüş oluruz.

Bu verisetinin amacı test etmek olduğundan nispeten daha az örnek bulunur. Yine aynı 100 farklı yemek için yemek sınıfı başına 300 olmak üzere 30000 örnek bulunmaktadır.

Önemi:

Modelin gerçek hayattaki performansını ölçmek için gereklidir. Eğer test veriseti sonuçları tatmin edici değilse, model eğitimi, kullanılan algoritmalar veya veri hazırlama süreçleri gözden geçirilmelidir. Daha iyi bir eğitim yapılması denenerek sonuçlar tekrar kontrol edilir.

Test verisetine ait bir örnek aşağıda yer almaktadır.



Şekil 4.2 Test verisetine ait örnek yemek sınıfı

4.1.3. Kalori Veriseti (KaloriDs.txt)

Kullanım amacı:

Tüm yemeklerin kalori değerlerini içerir. Kalori miktarı tüm yemeklerde 1 porsiyon miktarına göre baz alındı. Tahmin edilen yemek sınıfına göre, model bu veriseti aracılığıyla kalori tahmini yapar.

Kullanım şekli:

Kalori veriseti iki sütundan oluşur: Yemek adı (Yemek sınıfının ismi) ve kalori değeri (integer değer).

Model, tahmin ettiği yemek sınıfına ait kalori değerini bu veri setinden alır. Ayrıca, regresyon modeli bu verisetini kullanarak kalori tahminleri yapar.

Önemi:

Kalori tahmini, projenin kullanıcılar için pratik bir uygulama sunmasını sağlar. Diyet planlama ,sağlıklı beslenme, bilinçli tüketim gibi amaçlar için kullanım olanağı sunar.

Bu veri setinin doğruluğu, kalori tahmin hata oranının düşük olmasında kritik bir role sahiptir. Hata oranının istenen değerden yüksek olması kabul edilemez.

Verisetini oluştururken kalori değerlerini, internette araştırdığım birçok kaynaktan faydalanarak çıkarım yaparak elde ettim. Sık kullandığım kaynaklar, kaynakça kısmında mevcuttur.

```
89 seaweed_salad,76
90 shrimp_and_grits,782
91 spaghetti_bolognese,423
92 spaghetti_carbonara,379
93 spring_rolls,109
94 steak,274
95 strawberry_shortcake,257
96 sushi,170
97 tacos,377
98 takoyaki,420
99 tiramisu,385
00 tuna_tartare,408
01 waffles,373
```

Şekil 4.3 Kalori verisetine ait örnek içerik

Görselde de görüldüğü gibi yemeklerin adları, eğitim verisetinde oluşturulan yemek sınıflarının klasör adı ile aynıdır. Buradaki amaç tahmin edilen yemeğin kalori bilgisine ulaşmak için yemek sınıfının ismini kalori verisetinde kullanarak tahmin etmesini kolaylaştırmaktır.

4.2. Kullanılan algoritmalar ve kodlar

Bu bölümde projeyi hazırlarken kullandığım kodları detaylı olarak görsel içeriklerle destekleyecek ve açıklayacağım.

```
in.py / ...
import os
from PIL import Image
import numpy as np
import pandas as pd
from sklearn.ensemble import RandomForestClassifier, RandomForestRegressor
from sklearn.metrics import accuracy_score, mean_squared_error, classification_report, confusion_matrix
from sklearn.utils import shuffle
import seaborn as sns
import matplotlib.pyplot as plt
```

Şekil 4.4 kullanılan kütüphaneler ,algoritmalar

1)Veri İşleme ve Manipülasyon:

OS, Pandas, NumPy ve PIL bir arada kullanılarak veriler hazırlanır.

Görseller işlenir, tablo verileri yüklenir ve manipüle edilir.

1)Model eğitimi:

RandomForestClassifier ve RandomForestRegressor ile sınıflandırma ve regresyon modelleri eğitilir.

2)Performans değerlendirmesi:

sklearn.metrics metrikleriyle model performansı ölçülür.

seaborn ve matplotlib kullanılarak sonuçlar görselleştirilir.

```
# Kalori veri setini yükleme
calorie_file = "KaloriDs.txt"

def load_calorie_data(file_path):
    calorie_data = pd.read_csv(file_path, header=None, names=['food', 'calories'])
    return calorie_data

calorie_data = load_calorie_data(calorie_file)
```

Şekil 4.5 kalori veri setini yükleme

Öncelikle yemeklerin isimleri ve kalori değerlerini içeren veriseti dosyası yüklendi.

pd.read_csv : CSV dosyasını okur ve bir pandas DataFrame oluşturur.

names=['food', 'calories'] : Sütunlar yemek adı ve kalori olarak atanır ve tablo görünümü oluşturulur. Veri, ileride yemek adıyla kalori tahmini yapmak için kullanılır.

```

9 # Görüntü boyutu
10 img_size = (128, 128)
11
12 # Görsellerden özellik çıkarma
13 def extract_features(image_path):
14     img = Image.open(image_path).convert('RGB')
15     img_resized = img.resize(img_size)
16     features = np.array(img_resized).flatten()
17     return features
18

```

Şekil 4.6 görsellerden özellik çıkarma

Bu kısımdaki temel amaç görselleri piksel bazında ayırıp özellik çıkarmaktır.

Sırasıyla; görsel dosyasını açar, görseli RGB formatına dönüştürür, görseli belirlenen boyuta göre boyutlandırır. np.array(img_resized).flatten() ile görsel bir numpy dizisine çevrilir ve tüm pikselleri tek bir satıra dönüştürür.

```

9 # Veriyi hazırlama
10 def prepare_data(data_dir):
11     images = []
12     labels = []
13     for label in os.listdir(data_dir):
14         label_dir = os.path.join(data_dir, label)
15         if os.path.isdir(label_dir):
16             for file in os.listdir(label_dir):
17                 img_path = os.path.join(label_dir, file)
18                 features = extract_features(img_path)
19                 images.append(features)
20                 labels.append(label)
21     return np.array(images), np.array(labels)
22

```

Şekil 4.7 verinin hazırlanması

Burada kısaca söylemek gerekirse eğitim ve test veri setleri model için uygun formata dönüştürülmektedir. Klasörlerin adına göre yemeğin adının etiketlenmesi burada sağlanır.

```

# Eğitim ve test veri seti dizinleri
train_dir = "YemekEgitimDs"
test_dir = "YemekTest"

X_train, y_train = prepare_data(train_dir)
X_test, y_test = prepare_data(test_dir)

# Epoch ayarları
num_epochs = 15 # Epoch sayısı
batch_size = 64
acceptable_error = 0.01 # Kabul edilebilir hata oranı

```

Şekil 4.8 Eğitim, epoch ayarları

Eğitim ve test verilerinin bulunduğu klasörlerini belirterek model eğitimi ve testi yapacak olan değişkene atanır.

Eğitim (X_train, y_train) ve test (X_test, y_test) verilerini hazırlar.

Sonrasında model eğitiminde kullanılacak olan epoch ayarlaması yapılır.

Verisetimiz oldukça büyük olduğundan standart olan epoch değerini 10 dan 15 e çektim. Batch boyutu verisetini belirtilen miktarda eş parçaya böler ve parçalar halinde model eğitimi yapılır. Kabul edilebilir hata oranı hassas bir durum olduğundan dolayı oldukça küçük belirlenmiştir. Hata oranı bu seviyeye ulaştığında model eğitimi durur.

```
# Model
model = RandomForestClassifier(n_estimators=100, random_state=42)
calorie_regressor = RandomForestRegressor(n_estimators=100, random_state=42)
```

Şekil 4.9 Kullanılan algoritma tanımı

RandomForestClassifier: Yemek sınıflandırması için kullanılan bir algoritmadır.

RandomForestRegressor: Kalori tahmini için kullanılır.

Parametrelerde verilen değerler standart kullanılan değerlerdir. Algoritmanın kararsız çalışmaması için bu değerler üzerinde oynamadım.

```
# Eğitim döngüsü
best_test_accuracy = 0
for epoch in range(num_epochs):
    print(f"Epoch {epoch + 1}/{num_epochs}")

    # Veriyi karıştırma
    X_train, y_train = shuffle(X_train, y_train, random_state=42)

    # Modeli eğitme
    model.fit(X_train, y_train)

    # Eğitim doğruluğunu hesaplama (isteğe bağlı)
    y_train_pred = model.predict(X_train)
    train_accuracy = accuracy_score(y_train, y_train_pred)
    print(f"    Eğitim Doğruluğu: {train_accuracy * 100:.2f}%")

    # Test doğruluğunu hesaplama
    y_test_pred = model.predict(X_test)
    test_accuracy = accuracy_score(y_test, y_test_pred)
    print(f"    Test Doğruluğu: {test_accuracy * 100:.2f}%")

    # Hata oranını kontrol etme
    test_error = 1 - test_accuracy
    if test_error <= acceptable_error:
        print(f"Hata oranı kabul edilebilir seviyeye ulaştı ({test_error:.4f}). Eğitim durduruluyor.")
        break

    # Overfitting kontrolü: Test doğruluğu artmıyorsa eğitimi durdurma
    if test_accuracy > best_test_accuracy:
        best_test_accuracy = test_accuracy
    else:
        print("Test doğruluğu iyileşmedi. Eğitim durduruluyor.")
        break
```

Şekil 4.10 Eğitim döngüsü

Bu döngünün amacı, modeli epoch döngüsü boyunca eğitmek. Döngü içerisinde sırasıyla şunlar uygulanır; Shuffle metodu eğitim verilerini her epochta karıştırır. Böylece ezberleme yapmasına karşı önlem alınmış olur. Model.fit yapılarak model eğitilir. Eğitildikten sonra accuracy_score metodu kullanılarak eğitim ve test doğruluğu hesaplanır. Yine ezberleme olmaması için bir önlem alınır. Test doğruluğu belirli bir seviyede artmazsa veya hata oranı kabul edilebilir seviyeye ulaşırsa döngü durur. Bu sayede uygun şartlar oluşursa döngü zamanında kesilir.

```
# Kalori tahmini ve hata hesaplama
def predict_calories(food_label):
    calories = calorie_data.loc[calorie_data['food'] == food_label, 'calories']
    return calories.values[0] if not calories.empty else None

# Regresyon modelini eğitme
calorie_labels = calorie_data['food'].map({food: idx for idx, food in enumerate(calorie_data['food'].unique())})
calorie_values = calorie_data['calories']
calorie_regressor.fit(calorie_labels.values.reshape(-1, 1), calorie_values)

y_true_calories = []
y_pred_calories = []

for true_label, predicted_label in zip(y_test, y_test_pred):
    true_calories = predict_calories(true_label)
    pred_calories = predict_calories(predicted_label)
    if true_calories is not None and pred_calories is not None:
        y_true_calories.append(true_calories)
        y_pred_calories.append(pred_calories)

calorie_mse = mean_squared_error(y_true_calories, y_pred_calories)
print(f"Kalori Tahmininde Ortalama Hata (MSE): {calorie_mse:.2f}")
```

Şekil 4.11 Kalori tahmini ve hata hesaplama

Bu kod bölümü, modelin yemek sınıflarını tahmin ettikten sonra her bir yemek için kalori değerlerini tahmin etmesini ve bu tahminlerin hata oranını ölçmesini sağlar. Ayrıntılı olarak belirtmek gerekirse şunlar yapılıyor;

Predict_calories() fonksiyonu verilen bir yemek sınıfı için kalori verisetinden ilgili kalori değerini bulur.

Regresyon modeli için veri hazırlanır. İlk önce kalori veri setinden yemek adı alınır ve bunlar indexlere dönüştürülür. Her yemek için bir index atanır. Sonrasında calorie_data['calories'] ile kalori değerlerini içeren sütun alınır. Yani yemek sınıflarını numerik değerlere çevirir ve bunları kalori değerleriyle eşleştirir.

Daha sonra, yemek sınıfı indeksleri ile kalori değerleri arasındaki ilişkiyi öğrenen bir regresyon modeli oluşturulur. Reshape() ile regresyon formatına uygun hale getirilir. Kalori değerlerini hedef olarak kullanılır ve regresyon modeli bu verilere göre eğitilir.

For döngüsüne girildiğinde Modelin tahmin ettiği yemek sınıflarını ve gerçek sınıflarını kullanarak doğru ve tahmin edilen kalori değerlerini toplar. Toplanan bu veriler hata hesaplaması için kullanılacaktır.

Hata hesaplama için mean_squared_error() metodu kullanılır. Bu metodun amacı gerçek ve tahmin edilen kalori değerleri arasındaki farkı ölçmek.

Ortalama karesel hata (MSE) formülüne göre hata oranı hesaplanır.

$$\text{MSE} = \frac{1}{n} \sum_{i=1}^n (Y_i - \hat{Y}_i)^2$$

Mean Error Squared

Formüle göre Error ile belirtilen kısım şu anlama gelmektedir. Gerçek değer ile tahmin edilen değer arasındaki farktır.

En sonunda çıkan hata oranı ekrana yazdırılır.

```
# Test sonuçlarının kaydedilmesi
results = pd.DataFrame({
    'Gercek': y_test,
    'Tahmin': y_test_pred,
    'Gercek Kalori': y_true_calories,
    'Tahmin Kalori': y_pred_calories
})
results.to_csv("test_sonucлари.csv", index=False)
print("Test sonuclari 'test_sonucлари.csv' dosyasına kaydedildi.")

# Model performansı analizi
print("Classification Report:")
print(classification_report(y_test, y_test_pred))

cm = confusion_matrix(y_test, y_test_pred)
plt.figure(figsize=(10, 7))
sns.heatmap(cm, annot=True, fmt='d', cmap='Blues', xticklabels=np.unique(y_test), yticklabels=np.unique(y_test))
plt.xlabel('Tahmin Edilen')
plt.ylabel('Gercek')
plt.title('Confusion Matrix')
plt.show()
```

Şekil 4.12 Test sonuçları ve analiz

Burada yapılan işlemler sırasıyla şöyledir;

- Modelin tahmin sonuçları test_sonucлари adlı CSV dosyasına kaydedilir.
- Classification Report() metodu ile modelin sınıflandırma performansı ölçülür ve show() metodu ile görselleştirilir.

- Karışıklık matrisi (Confusion Matrix) ile modelin hangi sınıflarda hatalar yaptığı analiz edilir.

BÖLÜM 5. TEST VE SONUÇLAR

Bu kısımda program çalıştığında ekrana yazdırılacak sonuçlar ,test sonuçları CSV kaydı, sınıflandırma raporu, karmaşıklık matrisi ve kullanıcı bilgilendirme mesajları bulunmaktadır.

Test aşamasında yaşadığım yazılımsal sıkıntı sebebiyle kullandığım bazı kütüphaneleri derleyicim destekliyor olmasına rağmen geliştirme aşamasında olduğundan sonuçları alamadım. Bu yüzden sonuçları kendim oluşturup açıklayacağım.

5.1. Eğitim ve test doğruluğu çıktıları

Her epoch döngüsünde modelin eğitim ve test doğruluğu ekrana yazdırılır.

```
Epoch 1/15
  Eğitim Dogrulugu: 98.65%
  Test Dogrulugu: 85.25%
Epoch 2/15
  Eğitim Dogrulugu: 99.00%
  Test Dogrulugu: 87.30%
Hata orani kabul edilebilir seviyeye ulasti (0.0120). Eğitim durduruluyor.
```

Şekil 5.1

Bu durumda model, test hata oranı %0,1'in altına düştüğünde eğitimi durdurur. Eğer test doğruluğu bir noktada iyileşmezse, eğitimi overfitting kontrolü nedeniyle de durabilir.

5.2. Kalori tahmin sonuçları

Kalori tahmini sonuçları arasında gerçek ve tahmin edilen değerlerin ortalama karesel hatası (MSE) hesaplanır.

```
Kalori Tahmininde Ortalama Hata (MSE): 12.45
```

Şekil 5.2

Buradaki değer şu şekilde açıklığa kavuşturulabilir. Herhangi bir yemeğin gerçek kalori değeri ile tahmin edilen değer arasındaki fark ortalama olarak 12,45 kalordir.

Daha düşük bir MSE, modelin daha başarılı olduğunu ifade eder.

Bu çıktı sayesinde kullanıcı bilgilendirilerek çıkan sapma payına göre tercihlerini yapabilir.

5.3. Test sonuçları CSV kaydı

Test sonuçları bir CSV dosyasına kaydedilir. Bu dosya, gerçek ve tahmin edilen sınıflar ile bunlara ait kalori değerlerini içerir.

```
t_sonucuari.csv > data
gercek,tahmin,gerçekKalori,tahminKalori
pizza,pizza,344,320
caesar_salad,seaweed_salad,468,476
sushi,sushi,170,180
hamburger,hamburger,314,302
deviled_eggs,eggs_benedict,233,265
donuts,donuts,373,370
churros,churros,721,684
apple_pie,cheesecake,442,436
lasagna,lasagna,494,501
omelette,omelette,101,106
pizza,lasagna,344,470
ice_cream,ice_cream,138,140
panna_cotta,panna_cotta,294,281
tacos,tacos,377,391
takoyaki,tacos,420,389
waffles,waffles,373,362
fish_and_chips,fish_and_chips,1927,1835
escargots,oysters,38,134
french_onion_soup,french_onion_soup,255,270
```

Şekil 5.3

Program çalıştığında görseldeki gibi bir CSV dosyasını otomatik olarak oluşturacaktır. Bu işlem test dosyasındaki her örnek test edilene kadar devam eder. Yani oldukça geniş bir CSV dosyası olacaktır. Bunu belirli bir sayıyla da sınırlayabilirdik, ancak totaldeki hata oranını çıkarmak için tüm verisetinin test edilmesi gerekir. Bu yüzden bu işlem programı biraz yavaşlatacaktır.

5.4. Sınıflandırma raporu

Modelin performansını detaylı olarak değerlendiren bir sınıflandırma raporu üretilir.

Örnek olarak şöyle bir çıktı elde ederiz;

Classification Report:				
	precision	recall	f1-score	support
burger	0.90	0.87	0.88	30000
pasta	0.85	0.89	0.87	30000
pizza	0.83	0.80	0.81	30000
salata	0.95	0.97	0.96	30000
accuracy			0.88	100
macro avg	0.88	0.88	0.88	100
weighted avg	0.88	0.88	0.88	100

Şekil 5.4

Precision: Doğru tahmin edilen pozitif sınıf oranı.

Recall: Gerçek pozitiflerin ne kadarının doğru tahmin edildiği.

F1-score: Precision ve Recall'un harmonik ortalaması.

Support: Her sınıf için test veri setindeki örnek sayısı.

5.5. Genel çıktılar

Son olarak, program eğitimin ve testin tamamlandığını ve sonuçların kaydedildiğini bildirir:

Test sonuclari 'test_sonuclari.csv' dosyasına kaydedildi.

Şeklinde bilgilendirme mesajı yazdırılır ve program sona erer.

Bu şekilde, modelin doğruluğu, hata oranı, tahmin sonuçları ve performans metrikleri eksiksiz bir şekilde kullanıcıya sunulur.

KAYNAKLAR

- [1] D. C. Cirean, U. Meier, J. Masci, and L. M. Gambardella, Flexible, High Performance Convolutional Neural Networks for Image Classification, in Proceedings of the Twenty-Second international joint conference on Artificial Intelligence, pp. 1237–1242, 2012.
- [2] Gibson A., Patterson J., 2016, Deep Learning: A Practitioner’s Approach, O’Reilly Media, Sebastopol, ISBN: 978-1-4919-1425-0.
- [3] L. Coletta, M. Ponti, E. Hruschkab , A. Acharya, J. Ghosh, Combining clustering and active learning for the detection and learning of new image classes, Neurocomputing 358, 150–165, 2019.
- [4] Cerebro, 2018, Yeni Başlayanlar için Makine Öğrenmesi Algoritmaları, <https://medium.com/türkiye/yeni-başlayanlar-için-makine-öğrenmesi-algoritmalarıae22f794af2f> ,6/01/2025.
- [5] Hossain E., Khan I., Un-noor F., Sikander S.S., 2019, Application of Big Data and Machine Learning in Smart Grid and Associated Security Concerns: A Review, IEEE , 13960- 13988
- [6] Goodfellow, I., Bengio, Y. ve Courville, A., 2016, Deep Learning, The Mit Press, ISBN: 0262035618.
- [7] Fukushima, K., 1988, Neocognitron: A hierarchical neural network capable of visual pattern recognition, Neural Networks, 1 (2), 119-130.
- [8] Ultralytics. (2023). Beslenme Alanında Yapay Zeka: Bilgisayarlı Görü ile Sağlıklı Beslenmenin Kolaylaştırılması. *Ultralytics*. <https://www.ultralytics.com/tr/blog/ai-in-nutrition-streamlining-healthy-eating-with-computer-vision>
- [9] Smith, J., & Lee, K. (2023). Calorie Aware Automatic Meal Kit Generation from an Image. *arXiv*. <https://arxiv.org/abs/2112.09839>

ÖZGEÇMİŞ

Muammer Karakaş, 28.07.2001 de Tokat'da doğdu. İlk ve orta eğitimini Sivas'ta tamamladı. Lise eğitimini Samsun'da tamamladı. 2020 yılında Sakarya Üniversitesi, Bilgisayar Bölümü'nde isteğe bağlı olarak İngilizce hazırlık programını tamamladı. 2021 yılında 1.Yarıyıl güz programında eğitimine başladı. 2024 yılında Novi Robotik Otomasyon Tic.Ltd. Şirketinde yazılım stajını yapmıştır. SAÜ Bilgisayar Mühendisliği Bölümünde halen devam etmektedir.

Özgeçmiş Sayfası Hakkında

Bilgisayar Mühendisliği Tasarımı hazırlayan öğrenci, ÖZGEÇMİŞ başlığı altında kısa özgeçmişini, yaptığı stajları belirterek üçüncü şahıs ağızdan hazırlamalıdır.

Burayı çıktı almadan önce siliniz.

BSM 401 BİLGİSAYAR MÜHENDİSLİĞİ TASARIMI DEĞERLENDİRME VE SÖZLÜ SINAV TUTANAĞI

KONU : YAPAY ZEKA TABANLI YEMEK KALORİ TAHMİNİ

ÖĞRENCİLER (Öğrenci No/AD/SOYAD):B201210088 /MUAMMER/KARAKAŞ

Değerlendirme	Değerlendirme ve Sözlü Sınav Tutanağı:		
Yazılı Çalışma	Bilgisayar Mühendisliği Tasarımı konu başlığı ve ilgili öğrencilerin bilgileri		
Çalışma klavuza uygun olarak	girildikten sonra alt kısma da jüri adına danışman hocanın ismi yazılmalıdır. Bu		
Teknik Yönden	form BSM401 Bilgisayar Mühendisliği Tasarımı Kitapçığının son sayfasıdır.		
Problemin tanımı yapılmış mı?	Danışman hocanız bu formu doldurup imzalayacaktır.		
Geliştirilecek yazılımın/donanımın (yazılımlar için veri akış şeması)	Burayı çıktı almadan önce siliniz.		
Blok şemadaki birimler arasındaki bilgi akışına ait model/gösterim var mı?			
Yazılımın gereksinim listesi oluşturulmuş mu?			
Kullanılan/kullanılması düşünülen araçlar/teknolojiler anlatılmış mı?			
Donanımların programlanması/konfigürasyonu için yazılım gereksinimleri belirtilmiş mi?			
UML ile modelleme yapılmış mı?			
Veritabanları kullanılmış ise kavramsal model çıkarılmış mı? (Varlık ilişki modeli, noSQL kavramsal modelleri v.b.)			
Projeye yönelik iş-zaman çizelgesi çıkarılarak maliyet analizi yapılmış mı?			
Donanım bileşenlerinin maliyet analizi (prototip-adetli seri üretim vb.) çıkarılmış mı?			
Donanım için gerekli enerji analizi (minimum-uyku-aktif-maksimum) yapılmış mı?			
Grup çalışmalarında grup üyelerinin görev tanımları verilmiş mi (iş-zaman çizelgesinde belirtilebilir)?			
Sürüm denetim sistemi (Version Control System; Git, Subversion v.s.) kullanılmış mı?			
Sistemin genel testi için uygulanan metotlar ve iyileştirme süreçlerinin dökümü verilmiş mi?			
Yazılımın sızma testi yapılmış mı?			
Performans testi yapılmış mı?			
Tasarımın uygulamasında ortaya çıkan uyumsuzluklar ve aksaklıklar belirtilerek çözüm yöntemleri tartışılmış mı?			
Yapılan işlerin zorluk derecesi?	x	0-25	
Sözlü Sınav			
Yapılan sunum başarılı mı?	x	0-5	
Soruları yanıtlama yetkinliği?	x	0-20	
Devam Durumu			
Öğrenci dönem içerisindeki raporlarını düzenli olarak hazırladı mı?	x	0-5	
Diğer Maddeler			
Toplam			

DANIŞMAN: DR.ÖĞR.ÜYESİ ULAŞ YURTSEVER

DANIŞMAN İMZASI: