T.C
KONYA TEKNİ ÜNİVERSİTESİ BİLGİSAYAR MÜHENDİSLİĞİ
BİLGİSAYAR MÜHENDİSLİĞİ UYGULAMASI – 2
(BİTİRME PROJESİ-2) FİNAL RAPOR FORMU

Öğrencinin Adı-Soyadı:	Dilemre Ülkü
Numarası:	171213055
Danışmanın Adı-Soyadı:	Sedat Korkmaz
Sınav Tarihi:	17.06.2022
Projenin Konusu: Sayısal Üslup Analizi (Stilometri) Yöntemleri ve Yapay Zekâ Kullanılarak Eser –Yazar Eşleştirme	

Yapılan Çalışmaların Özeti:

Projeye başlanmadan önce üslup analizi, metin madenciliği, doğal dil işleme ve yazarlık atfı üzerine literatür taraması yapılmıştır. Elde edilen bilgiler ile bir izlenecek yol belirlenmeye çalışılmıştır. Bu bağlamda proje geliştirilirken şu adımlar izlenmiştir:

1. Veri toplanması
2. Verilerin ön işlemden geçmesi
3. Ön işlemden geçen verilerden öznitelik çıkartılması
4. Ön işlenmiş veriler ile makine öğrenmesi modelinden sonuç elde edilmesi
5. Diğer makine öğrenmesi modelleri ile karşılaştırılma yapılması
6. Modelin hiper parametre ayarlaması yapılması
7. Kullanım kolaylığı için arayüz geliştirilmesi

Veri toplanırken web kazıma yöntemi ile toplam 204 yazarın 2415 köşe yazısına ait; köşe yazısının yazıldığı tarih, yazarın ismi, köşe yazısının başlığı, köşe yazısının web sitesi bağlantısı ve köşe yazısının metni elde edilerek veri seti hazırlanmıştır.

Metinden uygulanan ön işlemler şunlardır:

1. Metinin tamamını küçük harfe çevirme
2. Metindeki noktalama işaretlerini kaldırma
3. Metindeki satır boşluklarını kaldırma
4. Metindeki stopword kelimeleri kaldırma
5. Metini kelime belirteçlerine ayırma
6. Metinicümle belirteçlerine ayırma
7. Kelime belirteçlerinin eklerini kaldırıp köklerini elde etme

Ön işlemeden sonra her bir metin için elde edilen öznitelikler şunlardır:

1. Kök zenginlik oranı (type token ratio)
2. Ortalama kelime uzunluğu (harf olarak)
3. Ortalama cümle uzunluğu (kelime olarak)
4. Noktalama işareti sayısı
5. Stopword sayısı
6. Tamamı büyük harf ile yazılmış kelime sayısı
7. Kelime uzunluk histogram vektörü (harf olarak)
8. Cümle uzunluk histogram vektörü (kelime olarak)
9. Kelime çantası vektörü (word of bag)
10. TF-IDF vektörü

Elde edilen öznitelik vektörleri asgari-azami normalizasyon (min-max normalisation) yöntemi ile normalleştirilmiştir. Veriler %77'si eğitim (1398 metin) ve %33'ü test (689 metin) olarak ayrılmıştır. Varsayılan parametreli SVM modelinde %58 kesinlik (accuracy) değeri elde edilmiştir. Varsayılan parametreli MLP modeli ile %?? Kesinlik değeri elde edilmiştir. Hiper parametre C parametresi 216 olan linear çekirdekli SVM modelinde %80 kesinlik değeri elde edilmiştir. Edilen vektör modelleri, SVM modeli ve yazarların sayısal temsillerini içeren authorMap sözlüğü kaydedilmiştir.

Uygulamanın arayüzünde 2 metin 1 metin kutusu ve 1 buton bulunmaktadır. Metin kutusundan alınan köşe yazısı ön işleme ve öznitelik çıkarma adımlarından sonra kaydedilen vektör modelleri ile vektöre dönüştürülmüştür ve daha önce kaydedilmiş SVM modeli ile tahmin elde edilmiştir. Elde edilen tahmin daha önceden kaydedilen authorMap ile yazar ismine dönüştürüldükten sonra bir metin kutusunda gösterilmiştir.

Projede Kullanılan Materyal ve Metotlar:

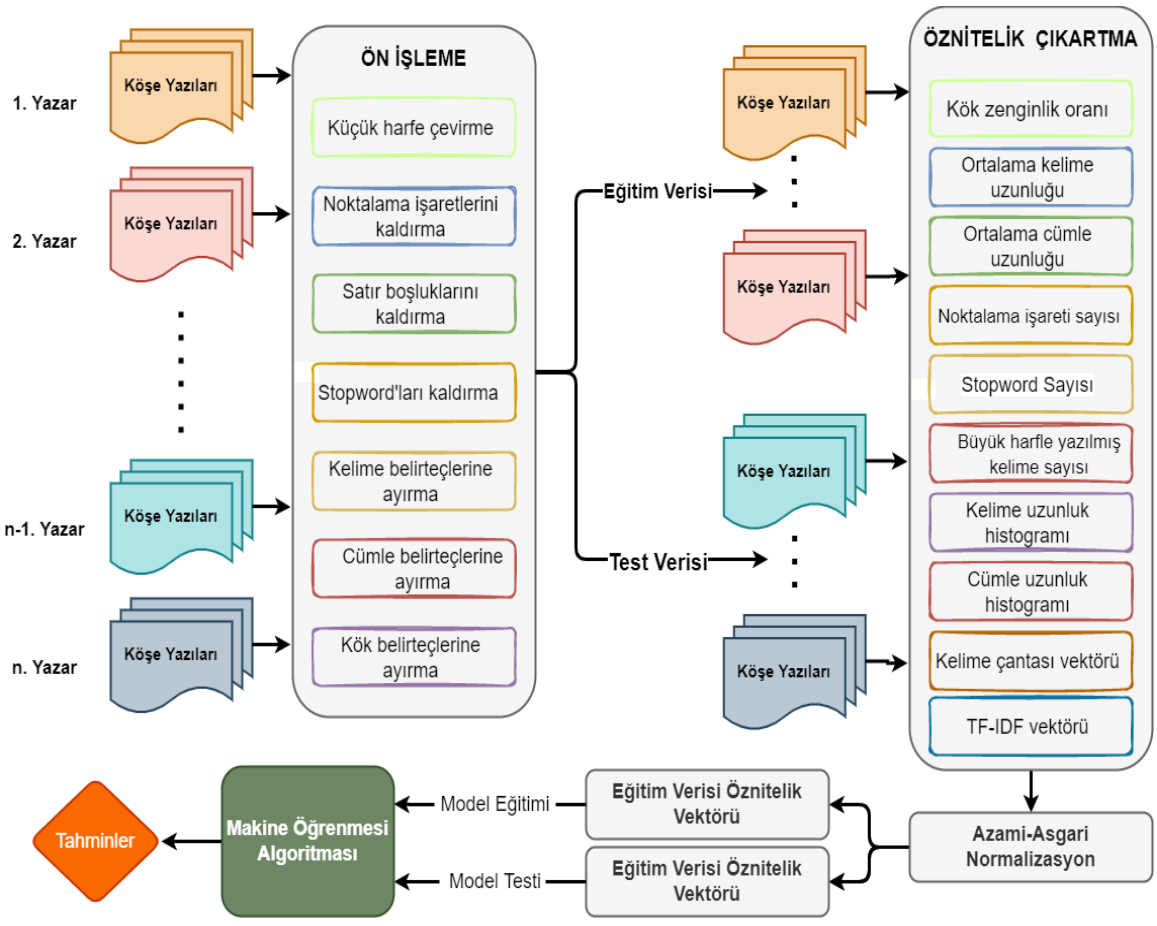
a) Materyal listesi:

Projede materyal olarak www.sabah.com.tr sitesinde yayımlanış 204 yazarın 2415 tane köşe yazısı kullanılmıştır. Siteden elde edilen veriler ile bir veri seti hazırlanmıştır. Hazırlanan veri setine ait veri sözlüğü aşağıdaki tablodadır.

Sütun	Açıklama	Veri tipi
info	Köşe yazısının yayımlanma tarihi	string
title	Köşe yazısının yazarı	string
caption	Köşe yazısının başlığı	string
link	Köşe yazısının web adresi	string
text	Köşe yazısının metni	string

Tablo 1. Veri sözlüğü

b) Metoda ilişkin akış şeması:



Şekil 1: Metot akış şeması

Yapılan Çalışmaya Ait Genel Bilgiler:

Teknolojiler	Açıklama
Anaconda 2.14	Paket yönetimini ve dağıtımını kolaylaştıran Python dağıtımı.
Jupyter Notebook 6.4.5	Web tabanlı etkileşimli hesaplama ortamı.
Spider 5.1.5	IDE. Arayüz geliştirmek için kullanıldı.
Python 3.9.7	Programlama dili.
beautifulsoup4	HTML dosyalarından veri çekmek için Python kütüphanesi. Web sitelerinde ilgilenilen kısımları ayırmak için kullanıldı.
requests	Python HTTP kütüphanesi. Web kazıma sırasında HTTP isteklerini yönetmek için kullanıldı.
Zemberek 0.17.1	Java tabanlı Türkçe dil işleme kütüphanesi. Ön işleme yaparken
zipfile	Python'da sıkıştırılmış dosyaları yönetmek için Python kütüphanesi. Zemberek modüllerini listelemek için kullanıldı.
Jpype 1.3	Python içinden Java'ya tam erişim sağlayan bir kütüphane. Zemberek kütüphanesini Python'da kullanmayı sağlamıştır.
Pandas 1.4.1	Python'da veri manipülasyon ve analiz kütüphanesi.
Numpy 1.20.3	Python'da bilimsel hesaplama kütüphanesi. Numpy dizileri ve üst düzey matematiksel fonksiyonları kullanılmıştır.
Scikit-learn 1.0.2	Python'da makine öğrenmesi kütüphanesi. Öznitelik çıkartma, hiper parametre arama gibi işlemlerde modülleri ve makine öğrenmesi modelleri kullanıldı.
TensorFlow	LSTM modeli oluşturulmak için kullanıldı.
pickle	Elde edilen modelleri kaydetmek ve tekrar yüklemek için kullanıldı.
PySimpleGUI	Arayüz geliştirmek için kullanılmıştır.

Tablo 2: Kullanılan teknolojiler

Veri olarak elde edilmesi kolay olan köşe yazıları tercih edilmiştir. Veriler web kazıması (web scraping) adı verilen teknik ile belirlenen bir haber sitesinden elde edilmiştir. Web kazıması yapılırken requests kütüphanesi ile www.sabah.com.tr/yazarlar/arşiv adresi HTML formatında indirilmiş ve beautifulsoup4 kütüphanesi ile HTML etiketleri ayrıştırılarak tüm yazarların isimleri ve yazarların tüm yazılarının bulunduğu bağlantıları elde edilmiştir. Ardından bir döngü ile her bir yazarın sayfasına girerek tüm yazılarının bağlantıları elde edilmeye çalışılmış fakat site sonsuz kaydırma (infinite scroll) sayfa yapısında olduğu için sadece en son yayımlanan 20 köşe yazısının bağlantısı elde edilmiştir. Ayrı bir iç içe döngü ile her bir yazarın elde edilen köşe yazılarına girerek köşe yazılarının yayımlanma tarihleri, yazar ismi, yazının başlığı ve yazı elde edilmiştir. Son olarak elde edilen bilgiler ile Pandas veri yapısı olan bir DataFrame oluşturulduktan sonra CSV formatında kaydedilmiştir.

Veri seti 14 Ocak 2017 tarihinden 19 Mart 2021 tarihleri arasında yazılmış 204 yazara ait 2415 köşe yazısından oluşmaktadır. Veri seti dengesizdir: 95 yazar 15 köşe yazısından az köşe yazısına sahiptir ve bu yazarların sahip olduğu toplam köşe yazısı sayısı ise 328'dir. Veri setindeki dengesizliği azaltmak için bu yazarlar ve köşe yazıları veri setinden çıkartılmıştır. Böylece toplam yazar sayımız 109, toplam köşe yazısı sayımız 2087 olmuştur.

CSV formatındaki veri seti Pandas kütüphanesi yardımı DataFrame veri yapısında tutularak ön işlemler gerçekleştirilmiştir. Proje ilk doğal dil işleme kütüphanesi olarak kullanılan NLTK kütüphanesi Türkçe'ye kapsamlı bir destekleme sağlamadığı için Zemberek kütüphanesi ile değiştirilmiştir. Java tabanlı Zemberek kütüphanesinde kullanacağımız modüler Jpype kütüphanesi yardımı ile Python projesinde kullanılmıştır. Zemberek'in TurkishMorphology modülü kelimelerin köklerini elde etmek için; TurkishSentenceExtractor modülü cümle belirteçleri elde etmek için kullanılmıştır. Ayrıca yine Zemberek içerisinde bulunan Türkçe stopword listesi txt olarak indirilip kullanılmıştır.

Sırası ile metine uygulanan ön işlemler ve açıklamaları şekildedir:

Ön işlem	Açıklama
Metini küçük harfe çevirme	Kelime çantası ve TF-IDF gibi öznitelik vektörleri oluşturulurken kelimelerin sayılmasını kolaylaştırır.
Metindeki noktalama işaretlerini kaldırma	Kelimelerin ve cümlelerin temsillerini kolaylaştırır ve öznitelik vektörünü azaltır. Buna rağmen bazı noktalama işaretlerinin varlığı bir yazar için tanımlayıcı olabilir.
Metindeki satır boşluklarını kaldırma	Öznitelik elde etmeyi kolaylaştırmak için kullanılmıştır.
Metindeki stopword kelimeleri kaldırma	Metindeki bütün kelimeler tanımlayıcı değillerdir. Türkçede sıklıkla kullanılan kelimeler çıkartılmıştır.
Metni kelime belirteçlerine ayırma	Metni kelime birim parçalarına ayırarak elde edilir (tokenization). Daha sonraki öznitelikleri elde ederken kolaylık sağlaması için kullanılmıştır.
Metini cümle belirteçlerine ayırma	Cümle uzunluk histogramı gibi öznitelikleri elde etmeyi kolaylaştırmak için kullanılmıştır.
Metini kök belirteçlerine ayırma	Türkçe sondan eklemeli bir dil olduğundan dolayı çekim ekleri ile beraber birçok kelime türetilebilmektedir. Kelime çantası gibi öznitelik vektörlerinin boyutunu azaltmak için kelimeleri köklerine ayrılmıştır.
Yazarları sayısal olarak kodlama	Makine öğrenmesi modeli için tahmin edeceği yazarlar sayısal ifade olarak (0-108) değiştirilir.

Tablo 3: Ön işleme yöntemleri

Ön işlemler sonucu elde edilen çıktılar DataFrame yeni satırlar olarak eklenmiştir. Eklenen yeni satırlar şunlardır:

Satır	Açıklama	Veri Tipi
clean_text	Satır boşlukları ve noktalama işaretleri kaldırılmış, tamamı küçük harfe çevrilmiş metin.	string
word_token	clean_text'in stopword'inin kaldılmış ve kelime belirteçlerine parçalanmış listesi	array
sent_token	Mein cümle belirteçlerine ayrıldıktan sonra satır boşlukları, noktalama işaretleri ve stopword'lar kaldırılmış ve tamamı küçük harfe çevrilmiş listesi	array
lemma_token	word_token'ların köklerinin listesi	array

Tablo 4: Metin ön işlendikten sonra elde edilen çıktılar

Kök elde etme sırasında Zemberek'in TurkishMorphology modülü yabancı kelimelerin (örneğin yabancı insan isimleri) morfolojik analizini yapamadığı için bu kelimelerin köklerini "UNK" olarak döndürür. Bu kelimeler UNK olarak bırakıldığında model performansı daha iyi olmuştur. Ayrıca UNK kökler sayılarak metinde geçen yabancı kelime sayısı da elde edilebilir.

Öznitelik çıkarma sırasında bazı çok kullanılan yöntemlerin yanında köşe yazıları incelendiği zaman yazarları tanımlayabileceği düşünülen farklı yöntemler de kullanılmıştır. Bu özniteliklerin bazılarının beraber kullanılmasının tek başına kullanılmasından daha iyi olduğuna karar verilmiş fakat tam olarak bu öznitelik kombinasyonlarının nasıl çalıştığı anlaşılamamıştır. Köşe yazıları daha dikkatli incelenerek yeni öznitelikler de eklenebilir.

Ön işlemeden sonra öznitelik çıkartmak için uygulanan yöntemler ve elde edilen öznitelikler şunlardır:

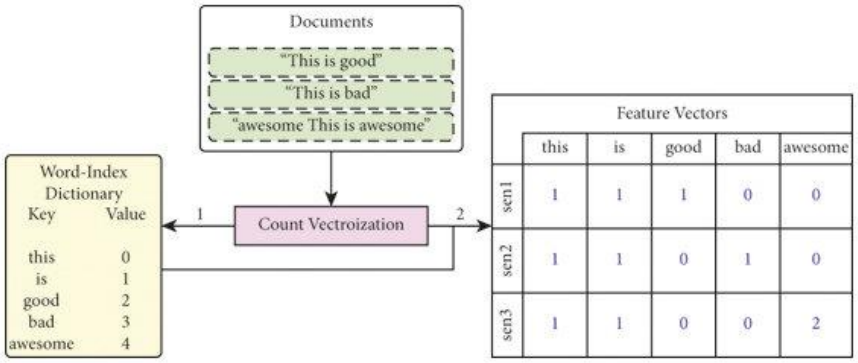
Öznitelik	Açıklama	Veri Tipi
Kelime Uzunluk Dağılımı	Kelime belirteç listesindeki kelimelerin harf olarak uzunluk dağılımıdır. Bu öznitelik yazarın kelime seçimlerine dair bilgi verebilir.	dict
Cümle Uzunluk Dağılımı	Cümle belirteç listesindeki cümlelerin kelime olarak uzunluk dağılımıdır. Yazarın uzun ya da kısa cümleleri ne sıklıkla tercih ettiği ile ilgili tanımlayıcı bilgiler verebilir.	dict
Kök uzunluk Dağılımı	Kelime Belirteç listesindeki kelimelerin köklerinin harf olarak uzunluk dağılımıdır. Kelime uzunluk dağılımı ile beraber yazarın kök kullanımına dair az da olsa bilgi verebilir.	dict
Kelime Zenginliği Ölçüsü	Kelime belirteç listesindeki benzersiz kelime sayısının toplam kelime sayına oranıdır. Yazarın metinlerinde ne kadar farklı kelimeler tercih ettiğini verebilir fakat köke eklenen her bir ek farklı birer kelime olarak algılanmaktadır.	float
Kök Zenginliği Ölçüsü	Kök belirteç listesindeki benzersiz kök sayısının toplam kök sayına oranıdır. Yazarın ne kadar çeşitli kelimeler kullandığı, kelime zenginliği ölçüsü ile beraber ortaya çıkabilir.	float
Ortalama Kelime Uzunluğu	Kelime belirteç listesindeki kelimelerin harf sayılarının ortalama uzunluğudur. Yazarın genellikle kısa kelimeler mi uzun kelimler mi tercih ettiğine dair bilgi verebilir.	float
Ortalama Cümle Uzunluğu	Cümle belirteç listesindeki her bir cümledeki kelime sayısının ortalamasıdır. Yazarın genellikle kaç kelimeden oluşan cümleleri tercih ettiğine dair bilgi verir.	float
Noktalama İşareti Sayısı	Ön işlenmemiş metindeki noktalama işaretlerinin sayısıdır. Uzun cümleler ile beraber noktalama işareti kullanma sayısının arttığı kabul edilebilir. Fakat bu gerçekleşmiyorsa yazar için tanımlayıcı bir bilgi olabilir.	int
Stopword Kelime Sayısı	Ön işlenmemiş metindeki Stopword kelimelerinin sayısıdır. Eğer yazar herkesin çokça kullandığı kelimeleri az sayıda kullanıyorsa tanımlayıcı bir bilgi olabilir. Fakat toplam kelime sayısına oranı daha doğru bir bilgi verebilir.	int
Tamamı Büyük Harfle Yazılmış Kelime Sayısı	Köşe yazıları incelendiğinde bazı yazarlar dikkat çekmek istedikleri fikirleri veya kelimeleri tamamen büyük harfte yazmaktadır. Büyük harfle yazılmış kelime sayısının bu sebepten ötürü tanımlayıcı bir öznitelik olduğu düşünülmektedir.	int

Tablo 5: Öznitelikler

Öznitelikler elde edildikten sonra veriler %77'si eğitim (1398 metin) ve %33'ü test (689 metin) olacak şekilde ikiye ayrılmıştır. Öznitelikleri makine öğrenmesi algoritmasına hazırlamak için Scikit-Learn kütüphanesindeki TfidfVectorizer, CountVectorizer, DictVectorizer modülleri kullanılmıştır.

DictVectorizer ile dict veri tipinde tutulan özniteliklerin vektör dönüşümünü yapmıştır. Bu modül liste içerisinde verilen sözlük benzeri yapılardaki verileri numpy veya scipy.sparse matrisine dönüştürür. Bu işlem sırasında liste içerisinde bulunan her bir sözlük bir satırı ifade eder ve bu sözlüklerdeki benzersiz her bir anahtar (key) bir sütunu oluşturur. Eğer bir sözlük içerisindeki değer kategorik liste olarak verilirse binary one-hot kodlama yapılır; integer ve float değerler olduğu gibi eklenir.

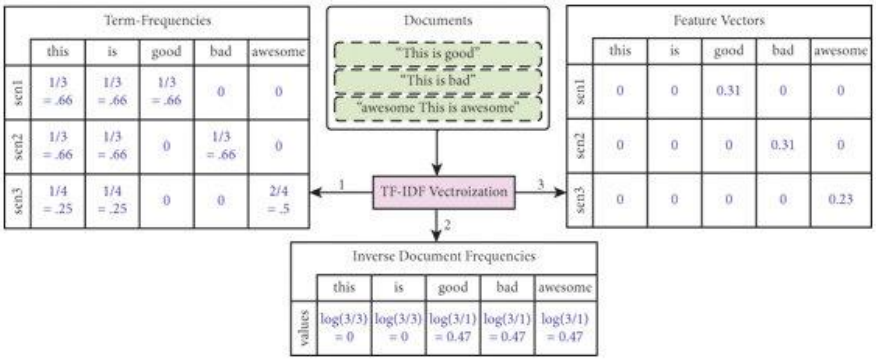
TfidfVectorizer ve CountVectorizer modüllerinin çıktıları yeni birer öznitelikdir. Bu modülleri kullanmak için eğitim verisindeki tüm metinlerin kök belirteç sözlükleri string listesine dönüştürülür ve derlem (külliyat, korpus) elde edilir. CountVectorizer derlemi belirteç sayısı matrisine (bag-of-word) dönüştürür.



Şekil 2: Count Vectorizer şeması
Kaynak: Khan, Nizami, Siddiqui, Wasi, Siyed, Qayoom (2021)

Belirteç sayısı matrisindeki sütunlar tüm derlemdeki benzersiz kökleri ifade eder. Satırlarda ise her bir metin için bu benzersiz köklerin kaç defa geçtiği bilgisi bulunmaktadır.

TfidfVectorizer derlemdeki her bir benzersiz kökün her bir metindeki TF-IDF (terim frekansı-ters metin frekansı (term frequency–inverse document frequency)) değerini hesaplar ve CountVectorizer'de olduğu gibi sütunlarında benzersiz kökler, satırlarında ise benzersiz köklerin her bir metin için değerlerinin bulunduğu bir matris oluşturur.



Şekil 3: TF-IDF Vectorizer şeması
Kaynak: Khan, vd. (2021)

TF-IDF değeri TF (terim frekansı) ve IDF (ters belge frekansı) olmak üzere iki istatistiki değerin çarpılması ile elde edilir. Bu hesaplama bir terimin bir derlemdeki bir metin için ne kadar önemli olduğunu göstermeyi amaçlamaktadır.

Terim Frekansı: d belgesindeki t teriminin göreceli frekansıdır ("tf-idf", 2022, Defination, paragraf 2):

$$tf(t,d) = \frac{f_{t,d}}{\Sigma_{t' \in d} f_{t',d}}$$

Ters Belge Frekansı: terimin ne kadar bilgi sağladığının bir ölçüsüdür, yani tüm belgelerde yaygın veya nadir olup olmadığı değerini verir ("tf-idf", 2022, Defination, paragraf 4):

$$idf(t,D) = \log \frac{N}{|\{d \in D : t \in d\}|} + 1$$

Burada N derlemdeki toplam belge sayısıdır, $|\{d \in D : t \in d\}|$ t'nin belge frekansıdır; belge kümesindeki t terimini içeren belgelerin sayısıdır. Logaritmaya 1 eklenmesi TfidfVectorizer'a özel bir hesaplamadır. Bu, tüm belgelerde görülen terimlerin göz ardı edilmesini engeller.

Elde edilen TfidfVectorizer, CountVectorizer, DictVectorizer modelleri, test verilerini de vektöre dönüştürmek için kullanılmış ve pickle kütüphanesi authorMap'te dahil olmak üzere kullanılarak .pkl uzantılı dosya olarak kaydedilmiştir. Bu modellerin ürettikleri matriksler test verisinde daha önce görmediği bir terim ile karşılaşırsa görmezden gelmektedir. Elde edilen tüm matrikslerin çıktı değerleri ayrı ayrı azami – asgari normalizasyon yöntemi ile normalize edilmiştir ve normalizasyon işlemi şu formüle göre yapılmıştır:

$$x' = \frac{x - \min(X)}{\max(X) - \min(X)}$$

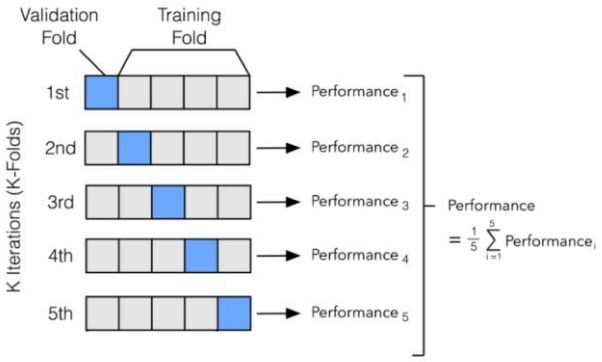
Burada x' normalize edilmiş değeri, x normalize edilmemiş değeri, X , x değerini içeren değer kümesini ifade etmektedir.

Normalizasyon işlemi sonunda elde edilen tüm matriksler birleştirilmiştir ve Scikit-learn kütüphanesinde bulunan SVM modelinin varsayılan parametrelerinde %58 kesinlik değerine ulaşılmıştır. SVM iki ya da daha fazla sınıfı birbirinden ayıran hiper-düzlemin belirlenmesine dayalı bir sınıflandırma algoritmasıdır. Sınıfları birbirinden ayırmak için sonsuz adet düzlem belirlenebilmektedir (Şenel, 2020).

Sckit-learn kütüphanesinin SVM modelinin varsayılan parametreler şöyledir:
Kernel = RBF, C = 1.0, Gamma = $scale\left(\frac{1}{n_{features} \cdot var(x)}\right)$

Hiper-parametreler, makine öğrenmesi modellerinde doğrudan öğrenilmeyen parametrelerdir. Çapraz doğrulama ise daha güvenilir bir kesinlik düzeyi değerlendirmesini yapabilmek için kullanılır. Sonuçlar eğitim ve test veri setlerine bağımlı olabileceği için farklı veri setleri ile model denenerek elde edilen sonuçların ortalaması alınır.

Çapraz doğrulama olarak K katlamalı çapraz doğrulama kullanılmıştır. Veri kümesi k alt küme altına bölünür ve eğitim k kez tekrarlanır. Her defasında, k alt kümelerinden biri test kümesi olarak kullanılırken diğer k-1 alt kümeleri bir eğitim kümesi oluşturmak üzere bir araya getirilir. Ardından, tüm k denemelerindeki ortalama hatası hesaplanır.



Şekil 4: K Katlamalı Çapraz Doğrulama
Kaynak: Öğündür (2020)

En yüksek doğruluk puanını veren öznitelikler ile SVC için hiper-parametre ayarını çapraz doğrulama ile birlikte yapabilmek için Scikit-learn kütüphanesinde bulunan RandomizedSearchCV modülü kullanılmıştır. 30 iterasyonda, 10 katlamalı çapraz doğrulama ile şu parametreler ile yapılmıştır:

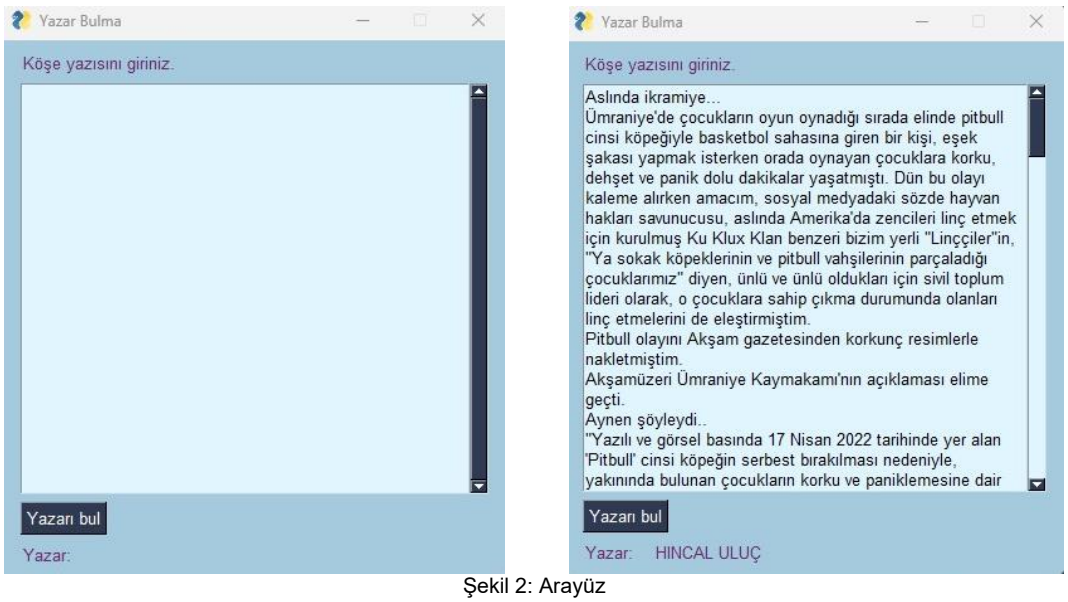
- kernel: rbf, C: loguniform(1e0, 1e3), gamma: loguniform(1e-4, 1e-2)
- kernel: linear, C: loguniform(1e0, 1e3)
- kernel: poly, gamma: loguniform(1e-4, 1e-2), degree : [2, 3, 4], C': loguniform(1e0, 1e3)
- kernel: sigmoid, gamma: loguniform(1e-4, 1e-2), C: loguniform(1e0, 1e3)

Burada loguniform scipy.stats modülünde bulunan olasılıksal yoğunluk oluşturan bir fonksiyondur. Değerler şu formülle hesaplanır:

$$f(a,b) = \frac{1}{\log\left(\frac{b}{a}\right)}$$

Yapılan işlem sonucu elde edilen en iyi sonuç, linear çekirdek ve C = 216 değeri ile %80 doğruluk puanıdır. Elde edilen en iyi parametreli model pickle kütüphanesi ile .pkl uzantılı olarak kaydedilmiştir.

Son arayüz için olarak PySimpleGUI kütüphanesi ile aşağıdaki arayüz geliştirilmiştir. Spyder'da ilk olarak arayüz oluşturulmuştur. Arayüzden alınan metin ön işlemden ve öznitelik elde etme işlemlerinden geçirildikten sonra kaydedilen modeller pickle ile yüklenmiş ve öznitelikler bu modellerden geçirilerek öznitelik matrisi elde edilmiştir. Bu matris yüklenen svm modeline girdi olarak verilerek bir tahmin elde edilmiş, elde edilen integer tahmin authorMap ile yazar ismine dönüştürülerek arayüzde gösterilmiştir. Aşağıdaki şekillerde Hınca Uluç'un 21 Nisan 2022 tarihli köşe yazısının sonuçları gösterilmektedir.



Şekil 2: Arayüz

Proje Hedeflerinin Değerlendirilmesi:

1. Veri seti oluşturma	Her bir yazara ait köşe yazısı sayısı geniş bir tarih aralığında arttırılmalı.
2. Ön işleme işlemlerini gerçekleştirme	Öznitelik çıkartılırken metindeki noktalama işaretleri tamamen kaldırılmıştır. Fakat kullanılan noktalama işaretleri yazar için tanımlayıcı olabilir. Ayıca metindeki rakamlar meta belirteç <num> olarak değiştirilebilir.
3. Öznitelik çıkartma işlemlerini gerçekleştirme	Daha fazla öznitelik çıkartılabilir. Örneğin kök türleri histogramı, ek türleri histogramı, n-gramlar...
4. Verilerin SVM modelini eğitecek şekilde düzenlenmesi	Farklı normalizasyon türleri ve farklı öznitelik vektörleştirme yöntemleri farklı sonuçlara sebep olabilir. Ayrıca farklı makine öğrenmesi modelleri ile daha başarılı sonuçlar elde edilebilir.
5. Yapay zekâ modelinde mevcut problem için en iyi parametrelerin belirlenmesi	Daha geniş aralıkta parametre aranabilir.
6. Arayüz geliştirme	Arayüz köşe yazısı metni yerine web sitesi bağlantısı alacak şekilde düzenlenirse web kazıma yöntemi ile kullanıcıların girdiği bağlantılardan yeni veriler elde edilebilir.

Kaynaklar:

Khan, Nizami, Siddiqui, Wasi, Siyed, Qayoom (2021). Automated Prediction of Good Dictionary EXamples (GDEX): A Comprehensive Experiment with Distant Supervision, Machine Learning, and Word Embedding-Based Deep Learning Techniques. Erişim adresi: https://doi.org/10.1155/2021/2553199

Öğündür (2020). Model Seçimi-K Fold Cross Validation. 15 Haziran 2022 tarihinde https://medium.com/@gulcanogundur/model-se%C3%A7imi-k-fold-cross-validation-4635b61f143c adresinden erişildi.

Tf-idf. *Wikipedia: The free encyclopedia içinde. (2022, 11 Haziran). 15 Haziran 2022 tarihinde* https://en.wikipedia.org/wiki/Tf%E2%80%93idf adresinden erişildi.

ÖĞRENCİNİN
Adı ve Soyadı : Dilemre ÜLKÜ
İmzası :

DANIŞMANIN
Adı ve Soyadı : Sedat KORKMAZ
İmzası :