

PROGRAMLAMA UYGULAMALARI
YAŞAM KOÇU UYGULAMASI GELİŞTİRME ÖDEVİ

Teslim Tarihi:18 Kasım 2021

--Bengü YURDAKUL 19401831
--Habibe Çisem KAPLAN 19401836

Ders: Programlama Uygulamaları/Dr.Öğr.Üyesi Uzay Çetin

İÇERİK

1-GİRİŞ.....	3
-Proje Amacı.....	3
-Gerçek Hayata Uygulaması.....	3
2-PROJE MİMARİSİ.....	4
-Kod Yapısı.....	4
-Kod Nasıl Çalışır.....	7
-Projenin Yapım Süreci.....	8
3- SONUÇLAR.....	9
4-KAPANIŞ&FUTURE WORK.....	10

GİRİŞ

Proje amacı

Projemizin amacı istenilen temaya göre kullanıcıya şarkı sözü ve şarkıyı seslendiren kişinin de adını sunmaktır. Böylelikle kullanıcı şarkı sözünü beğendiği takdirde sanatçı bilgisine sahip olacağı için diğer parçalarını da araştırabilir.

Gerçek Hayata Uygulaması

Üzgün, mutlu, heyecanlı, endişeli, kızgın hissettiğimizde, iyi ya da kötü bir haber aldığımızda, ders çalıştığımızda, iş yaptığımızda, boş bir anımızda, yürürken, otururken, uyumaya çalışırken, uyanmaya çalışırken bütün insanlığın ortak bir sığınağı vardır: Müzik.

Günümüzde birçok insanın yaşamın koşturmasından veya insanlardan dolayı problemleri olabiliyor. Böyle zamanlarda da derdini söyleyebileceği ve ya yol gösterecek ya da içini rahatlatacak şeyler duymak istiyor. Ama insanlar her zaman dinleme ve öğüt verme konusunda güvenilir olamayabiliyor. Ya da o kişi sıkıntısını başkalarıyla paylaşmak konusunda kendisini rahat hissetmiyor ve sorunu kendi içinde çözmeye çalışıyor.

İşte böyle bir anda hem sizin derdinizi dinleyecek hem de size Türkçe pop kültürünün önemli şarkılarından parçalar sunacak olan programımız devreye giriyor. Ayrıca sadece dertleşmeye değil aynı zamanda dinlemek istediğiniz temaya göre şarkı sözleri ve şarkıların sahibini de sizlere sunar.

PROJE MİMARİSİ

Kod Yapısı

Programımızda kullanılan programlama dili Python'dır. GitHub'dan [yaşam koçu](#) uygulamasının kodunu temel aldık. Projemizde

- dataframe üzerinde kolayca işlem yapabilmek ve anlaşılır kılabilmek için Pandas,
- Matrisleri kolayca kullanabilmek için Numpy,
- Metni düzenlemek ve metinden alt parçaları elde etmek için Re (Regular Expressions),
- encoding binary verileri metne dönüştürmek için Base64,
- cosine-similarity fonksiyonunu kullanabilmek için Scikit-Learn
- Word2Vec için [Gensim](#) ve internet sitesine taşıyabilmek için de [streamlit](#)

kullandık.

Kütüphaneleri bu şekilde import ettik:

```
import re

import streamlit as st

import pandas as pd

import numpy as np

import base64

from gensim.models import KeyedVectors

from sklearn.metrics.pairwise import cosine_similarity
```

Topladığımız bazı şarkı sözlerimiz fazla uzun olduğu için alt alta yazmamız gerekiyordu. Bu da alt satıra geçme ile bölemeyeceğimiz anlamına geliyordu. Bu sebepten dolayı her sözün sonuna “!” işareti koyduk. Böylece her bir sözü birbirinden ayırıp gerekli anahtar kelimelerle eşleştirdik.

```
categories = [Hayat, Aşk, Kızgınlık, Üzgünlük, Rastgele]

categories_name = ["Hayat", "Aşk", "Kızgınlık", "Üzgünlük", "Rastgele"]

for i in range(len(categories)):

    categories[i] = categories[i].split("!" )
```

Pandas kullanarak dataframe oluşturduk ve Re kullanarak temizledik.

```
veri = pd.DataFrame(columns=['Şarkı', 'Kategori'])

for i in range(len(veri)):

    veri["cleaned_sentence"][i] = re.sub('[!@#\'`?.,\'$]', '',
    veri["cleaned_sentence"][i])

    veri["cleaned_sentence"][i] = veri["cleaned_sentence"][i].lower()
```

Word2Vec/Gensim kullanarak modeli eğitiyoruz. Bunu da [Github'dan](#) bulduk.

```
word_vectors = KeyedVectors.load_word2vec_format('trmodel', binary=True)
```

Düzenli ve okunabilir olması için bütün fonksiyonlarımızı kuzuKuzu adlı class yapımızın altında tanımladık.

jaccard_similarity(str1, str2) □ Parametre olarak aldığı string yapıları arasında jaccard benzerliği formülünü uygular.

recomendation_random_jaccard(cevap) □ Eğer kullanıcı temalar arasında seçim yaparak şarkı sözü önerisi almak istiyorsa bu fonksiyon çağrılır. Parametre olarak kullanıcının seçtiği anahtar kelimeyi alır. Seçilen anahtar kelimeyle eşleştirilmiş şarkı sözlerden birini geri döndürür. Bu aşamada jaccard_similarity() fonksiyonu çağrılır. Ekrana şarkı sözünü bastırır.

construct_wv_matrices() □ Parametre olarak dataframe'i ve kullanıcının yazdığı metni alır. Bunların üzerinde gensim kütüphanesinden word2vec algoritmasını kullanır. Class yapısının hemen üstünde eğitilen model burada kullanılır.

recommendation_wv() □ Kullanıcıya hangi yoldan ilerlemek istediğini sorar ve aldığı sonuca göre ya recomendation_random_jaccard() ya da construct_wv_matrices() fonksiyonunu çağırır.

Class yapısından çıktıktan sonra streamlit kullanılarak websitesi oluşturulup detaylar eklenir.

```
st.title("Kuzu Kuzu Gel Ruh Halini Şarkı Sözleri İle Yansıtayım!")

st.markdown('Tarkan sunar...')

st.caption('made by Çisem Kaplan & Bengü Yurdakul ')

st.sidebar.header("KUZU KUZU")

st.sidebar.write("İster at, ister öp beni. Ama önce dinle bak gözlerime!")
```

Oluşturulan sidebar'da kullanıcıdan programlama 0-100 değerlendirmesi isteyen ve eğer değerlendirme 100 ise ekrana balonlar çıkaran, 100'da farklı ise seçtiği değer 100'den farkını hesaplayıp bu kadar puanı neden kırdığını soran kod parçası:

```
result = st.sidebar.slider("Kuzu Kuzu tahminimi nasıl buldun? ", 0,100)
if (result == 100):
    st.balloons()
elif (result < 100):
    st.sidebar.write("Nerden kırdın " , int(100-result) , "puanı kuzum?" )
```

Arkaplandaki videonun ve sidebar'daki gifin kendi kendine oynayıp durmasını sağlayan kod parçası:

```
file_ = open("kuzu.mp4", "rb")
contents = file_.read()
data_url = base64.b64encode(contents).decode("utf-8")
file_.close()
display_bg_video = f"""
    <!-- The video -->
    <video autoplay="autoplay" muted loop
src="data:video/mp4;base64,{data_url}" type="video/mp4"
    id="myVideo" mimeType="application/mp4"></video>
    """
st.markdown(display_bg_video, unsafe_allow_html=True)
```

```
file_ = open("tarkangif.gif", "rb")
contents = file_.read()
data_url = base64.b64encode(contents).decode("utf-8")
file_.close()
st.sidebar.markdown(
    f'',
    unsafe_allow_html=True,
)
```

Son olarak ise kuzuKuzu.recommendation_wv() fonksiyonu çağrılarak kod çalıştırılmış olur.

Kod nasıl çalıştırılır?

Projenin çalışması için gerekli olanlar:

- Streamlit kütüphanesinin indirilmiş olması

```
pip install streamlit
```

- Word2Vec'in istediğimiz gibi çalışması için Gensim kütüphanesinin indirilmiş olması

```
pip install --upgrade gensim
```

- requirements dosyasının içinde bulunan kuzu.mp4 ve tarkangif.gif dosyaları

Eğer kodu indirip başka bir ortamda çalıştırmayı düşünüyorsanız arkaplanda oynatılan Tarkan - Kuzu Kuzu MV'sinin ve yanda (sidebar) bulunan gifin PATH'ine (konumu) dikkat edip gerekirse kodu değiştirmeyi unutmayın.

Projenin çalışması için Anaconda Prompt'ta streamlit'i indirdikten sonra çalıştırılacak kod dosyasının bulunduğu dizine ilerlenilerek

```
streamlit run dosya_adı.py
```

yazarak internet sitesine rahatlıkla ulaşılabilir.

İnternet sitesine ulaşım sağlanınca ekranda kullanıcının nasıl ilerlemek istediği soruluyor. Tema seçmek ve temaya uygun söz almak istiyorsanız

“Seçenekler arasından seçim yapacağım.”

Yazması yeterlidir. Ekranı seçilebilir seçenekler çıkacaktır. Seçenekler:

- Hayat
- Aşk
- Kızgınlık
- Üzgünlük
- Rastgele

Kullanıcı sunulan seçeneklerden birini seçmelidir. Program, girilen anahtar kelime ile eşleştirilmiş şarkı sözlerinden rastgele birini alıp geri verecektir.

Eğer yazarak duygularına göre cevap almak istiyorsa

“İçimi dökeceğim.”

Yazması yeterlidir. Word2Vec ile girilen kelimeleri işleyen algoritma, uygun bulunduğu şarkı sözünü geri döndürecektir.

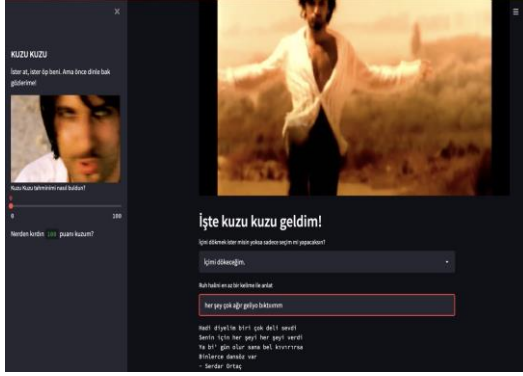
Projenin Yapım Süreci

Projemizin yapımı yaklaşık 2 hafta sürdü. Süreç kullanmayı planladığımız data biriktirme işlemiyle başlayıp ilk hafta kullanıcıdan sorununa cevap vermemiz için tek kelimeli girişler alınacak şekilde yapıldı. İlk hafta sadece streamlit ile siteyi oluşturup bazı basit düzenlemeleri yaptık. Anahtar kelimelerden birini seçen kullanıcı, o anahtar kelime ile eşleştirdiğimiz şarkı sözlerinden birini alıyordu. İkinci hafta ise Gensim kütüphanesi ve Scikit-Learn kullanarak model eğittik. Böylece kullanıcıdan alınan metini işleyerek öneri şarkı sözü de çıkarabilen bir program haline getirdik. Ayrıca sidebar'a bir değerlendirme ekledik.

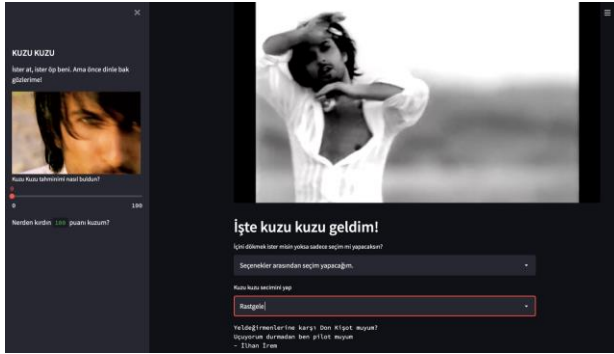
Projemiz için megastar Tarkan'ın ikonikleşmiş şarkısı Kuzu Kuzu'dan esinlendik ve tamamımızı Kuzu Kuzu seçtik. Arkada ilgili şarkının klibi dönüp durmakta, yanda Tarkan'ın gifi oynamakta ve kodumuz kullanıcıya “Kuzu Kuzu” diye seslenmektedir.

Kafamızda tasarladığımız birçok şeyi yaptık. Planlanan şeylerden biri de her cevabı içeren şarkının klibini o sözle birlikte oynatmaktı. Ama çok fazla yer kaplayacağından dolayı bu plan yerine sadece bir tane video kullanıldı.

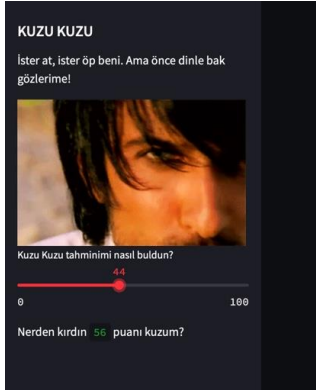
SONUÇLAR



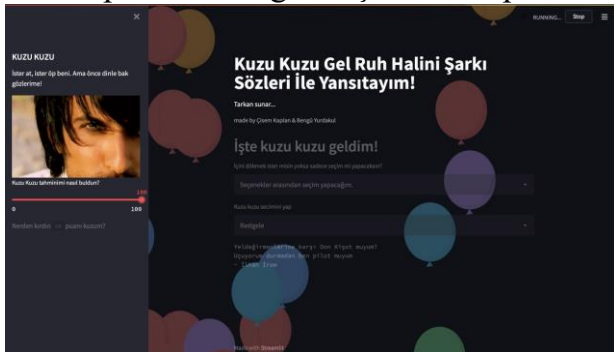
-İçimi dökceğim şıkkı tıklandığında



-Seçenekler arasından seçeceğim şıkkı tıklandığında



-Eksik puan verildiğinde çıkan cevap



-Uygulamadan %100 memnuniyet cevabı aldığıında küçük bir kutlama

KAPANIŞ & FUTURE WORK

Yapılan bu uygulamanın amacı basitçe kullanıcının içinde olduğu duruma ya da dinlemek istediğı temaya göre karşısına şarkı sözü çıkarmaktır. Aynı zamanda geliştirilmesi çok kolay ve önü açık bir uygulamadır.

Tespit edilen buglar:

- Program ilk açıldığında (kullanıcı henüz bir input vermemişken) ekranda rastgele şarkı sözü yazılı duruyor. Bu kaldırılabilir.
- Video'nun yüklenmesi uzun sürüyor, kısaltılabilir.
- Yanda, değerlendirme isteyen slider, henüz değerlendirme yapılmamışken kullanıcıya neden puan kırdığını soruyor. Kullanıcıdan herhangi bir input almamışken çalışmasının önüne geçilebilir.

İlerleyen zamanlarda kod şu şekilde geliştirilebilir:

- Videonun yanında Kuzu Kuzu şarkısı da otomatik olarak arka planda çalabilir.
- Şarkı sözleri ve sanatçının yanı sıra, şarkının adı verilebilir.
- Şarkı sözüne uygun benzer şarkı önerileri çıkabilir.
- Her çıkan sonuca göre ekranda sanatçının fotoğrafı veya şarkının MV'si çıkabilir.