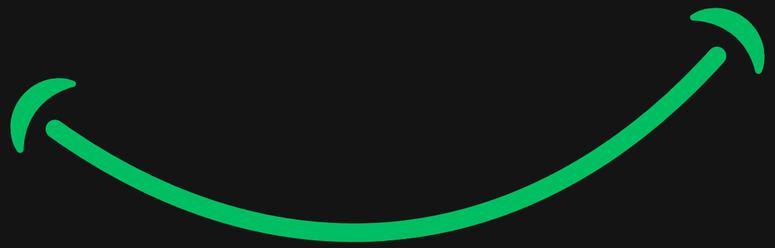
CS 210

SENTIMENT ANALYSIS



MUSTAFA TOPCU

HASAN YILMAZ

ANIL ŞEN

ORHUN ÖZPAY

MERT KULECİ

FIVE HORSEMEN OF THE APOCALYPSE

LIST OF CONTENTS

01 INTRODUCTION

CONTEXT AND THE OBJECTIVES OF THE PROJECT

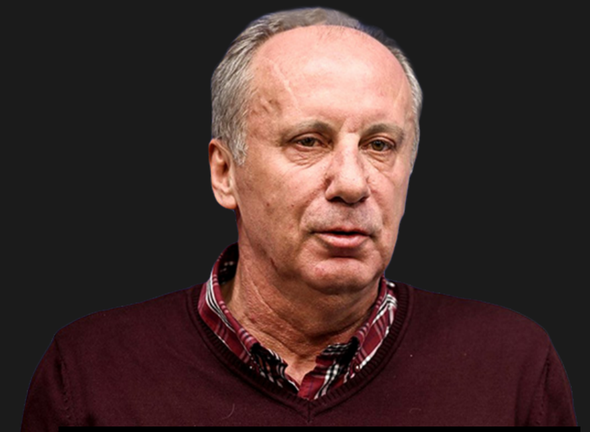
2023 Türkiye Presidental and General Elections

Sentiment Analysis on Youtube comments

Some predictions about future using ML

RECEP TAYYİP ERDOĞAN

KEMAL KILIÇDAROĞLU



MUHARREM İNCE

SİNAN OĞAN

OUR HYPOTHESIS

As the 2023 elections approach in Turkey, the average sentiment scores decrease and the appearance frequency of candidates in comments increases over time due to the escalating social tension towards the end of the elections

02 METHODOLOGY

GATHERING AND PREPARING THE DATASET

The necessary data is obtained using the YouTube API

We have specified a cutoff date (February 1, 2023) to gather data between the cuttof date and the date that data was gathered

The dataset includes pre-specified channels, video titles from the cutoff date, comments published under those videos, and the publication dates of those comments.

```
API_KEY="AIzaSyD1UOcSN6iwihCm__KcEPp128TPDNsWDNY"

# It is redefined (must redefined) when new channels are added to the list. Do not forget to redefine in the writing csv part too!!
channel_ids=[["Babala TV", "UCZ5aOEWFOopXLeiIUd2mfJw", 34],
             ["Medyal1 TV", "UCtTHvHWkbCO2H9Ua8HA8ekA", 293],
             ["Kendine Muhabir", "UCapDJ1RRsp5cNB-PMdPzVyw", 115],
              ["Sokaktan Al Haberi", "UCVSgtPIie4rli7dysK0tjTQ", 113],
              ["Özlem Gürses","UCojOP7HHZvM2nZz4Rwnd6-Q",362],
              ["Nevşin Mengü","UCrG27KDq7eW4YoEOYsalU9g", 145],
              ["Cüneyt Özdemir", "UCkwHQ7DWv9aqEtvAOSO74dQ", 1245],
              ["Mevzular Açık Mikrofon", "UCWl9g7avNGKdJPEzsMQOnKw", 116],
              ["Fatih Portalal TV", "UCTRxpG0DLS9eNmeeqTsz_jQ", 567],
              ["Yeni Şafak", "UCClO1RgRkaOcC9cLj-bLuEw", 231],
              ["SÖZCÜ Televizyonu", "UCOulx_rep5O4i9y6AyDqVvw", 3642]]

youtube=build("youtube","v3",developerKey=API_KEY)
```

```python
import random

def get_100_random_comments(youtube, video_id):
    commentList = {}
    nextPageToken = None

    while len(commentList) < 100:
        request = youtube.commentThreads().list(
            part="snippet",
            videoId=video_id,
            maxResults=100,
            pageToken=nextPageToken
        )
        response = request.execute()

        for item in response["items"]:
            comment = item["snippet"]["topLevelComment"]["snippet"]["textDisplay"]
            publishDate = item["snippet"]["topLevelComment"]["snippet"]["publishedAt"]
            commentList[comment] = publishDate

            if len(commentList) >= 100:
                break

        nextPageToken = response.get("nextPageToken")

        if not nextPageToken or len(commentList) >= 100:
            break

    if len(commentList) < 100:
        print("There are fewer than 100 comments available.")

    random_comments = random.sample(list(commentList.items()), min(100, len(commentList)))
    random_comment_dict = dict(random_comments)

    return random_comment_dict
```

GATHERING AND PREPARING THE DATASET

GENERATING LIST OF VIDEOIDS

EXTRACTING THE COMMENTS FOR EACH VIDEO, RANDOMLY SELECTING 100 COMMENTS AMONG THEM.

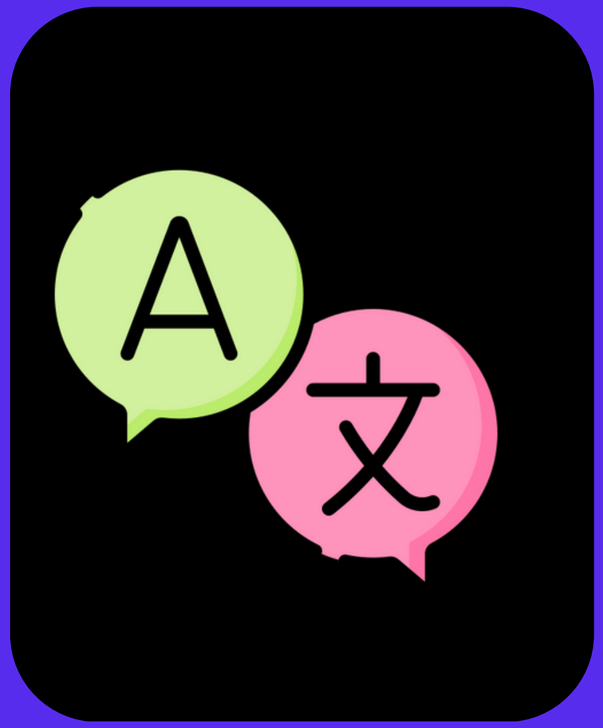
REASON FOR SELECTING 100 RANDOM COMMENTS:

ENSURE DATASET MANAGEABILITY FOR PROCESSING. RANDOM SELECTION MITIGATES POTENTIAL BIAS.

THE RESULTING VALUES WERE STORED WHERE THE CHANNEL NAMES SERVED AS KEYS FOR EASY RETRIEVAL.

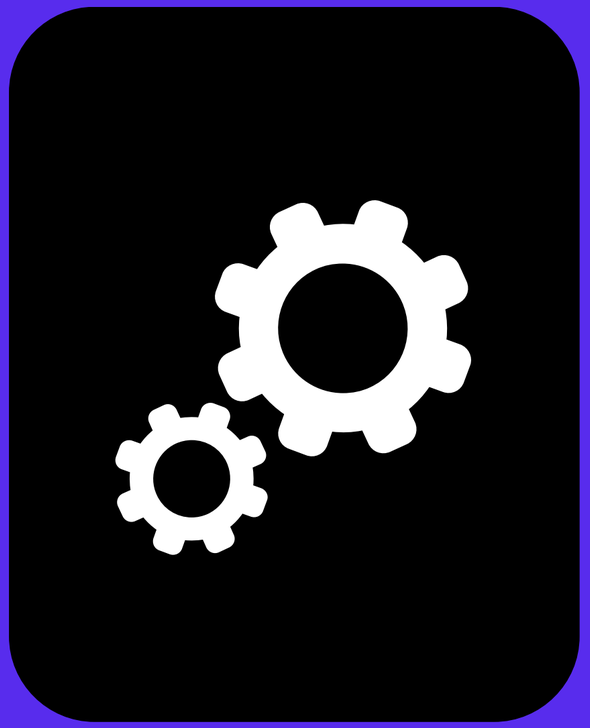
SAVED INTO DIFFERENT CSV FILES, CORRESPONDING WITH THE CHANNEL NAMES

HOW DID WE DEAL WITH COMMENTS?



TRANSLATOR

X



TEXT PROCESSING

X

SENTIMENT MODEL

TEXT PROCESSING

1

```python
import pickle
import re

with open('scripts/Emoji_Dict.p', 'rb') as fp:
    Emoji_Dict = pickle.load(fp)
    Emoji_Dict = {v: k for k, v in Emoji_Dict.items()}

def convert_emojis_to_word(text):
    for emot in Emoji_Dict:
        converted_word = "[" + "_".join(Emoji_Dict[emot].replace(",", "").replace(":", "").split()) + "]"
        pattern = re.escape(emot)
        text = re.sub(pattern, converted_word, text)

    text=text.replace("["," ").replace("],"  ").replace("_"," ")
    return text
```

2

```python
def lemmatize(list_of_words):

    py_lemmatizer = WordNetLemmatizer()

    pos_tags = pos_tag(list_of_words)

    py_lemword = []
    for word, tag in pos_tags:

        wn_tag = wordnet.NOUN
        if tag.startswith('V'):
            wn_tag = wordnet.VERB
        elif tag.startswith('J'):
            wn_tag = wordnet.ADJ
        elif tag.startswith('R'):
            wn_tag = wordnet.ADV

        lem_word = py_lemmatizer.lemmatize(word, pos=wn_tag)
        py_lemword.append(lem_word)

    return py_lemword
```

THROUGH THE USE OF EMOJI DICTIONARY FOUND ONLINE

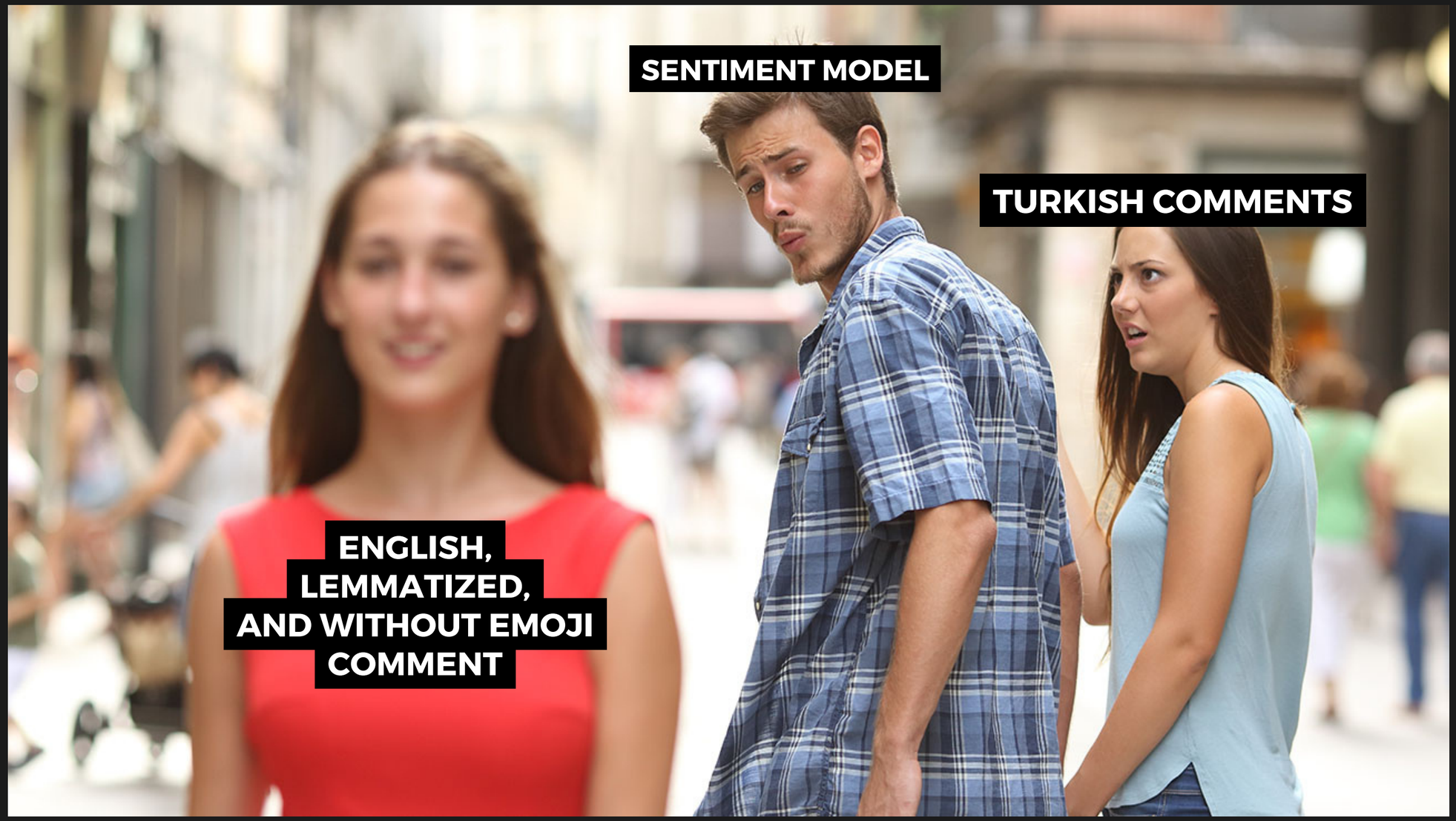
❤ -> "RED HEART"

BY USING NLTK LIBRARY OUR LEMMATIZER FUNCTION IS READY!

```python
from transformers import pipeline

sentiment_classifier = pipeline("sentiment-analysis", model="nlptown/bert-base-multilingual-uncased-sentiment")

# Analyze the sentiment of a sentence
#sentence = 'i believed once, he said that the nation alliance will win, it was a lie, i will not believe it again'
```

```python
def analyze_sentiment(sentence):

    result = sentiment_classifier(sentence)[0]

    return result["label"]
```

OUR SENTIMENT MODEL IS FROM
NLPTOWN/BERT-BASE-MULTILINGUAL-UNCASED-SENTIMENT


SENTIMENT MODEL
TURKISH COMMENTS
ENGLISH, LEMMATIZED, AND WITHOUT EMOJI COMMENT

POORLY WRITTEN TURKISH COMMENTS

ENGLISH COMMENTS

"CCCCUUUUSSSSSS ADELET GELECEK OYLEMİ"

"BU INSANLAR SEKSEN HARFI DEYIŞİMLE CALIY BIRAKILDI SEKSEN YILDIR BAŞ ORTUSURLE UGRAŞDIRLAR BU KADAR CAHIL INSANLAR HELE VAR LAYIK INSANLAE SIPIKER SEN MUSLUMN DEYILSINKI MUSSLUMANA SAYGIN OLSUNMEDIYA"

"TERORİSTMİ DENĀNĀSİN TURK MİLLĀTİ AYIB YA"

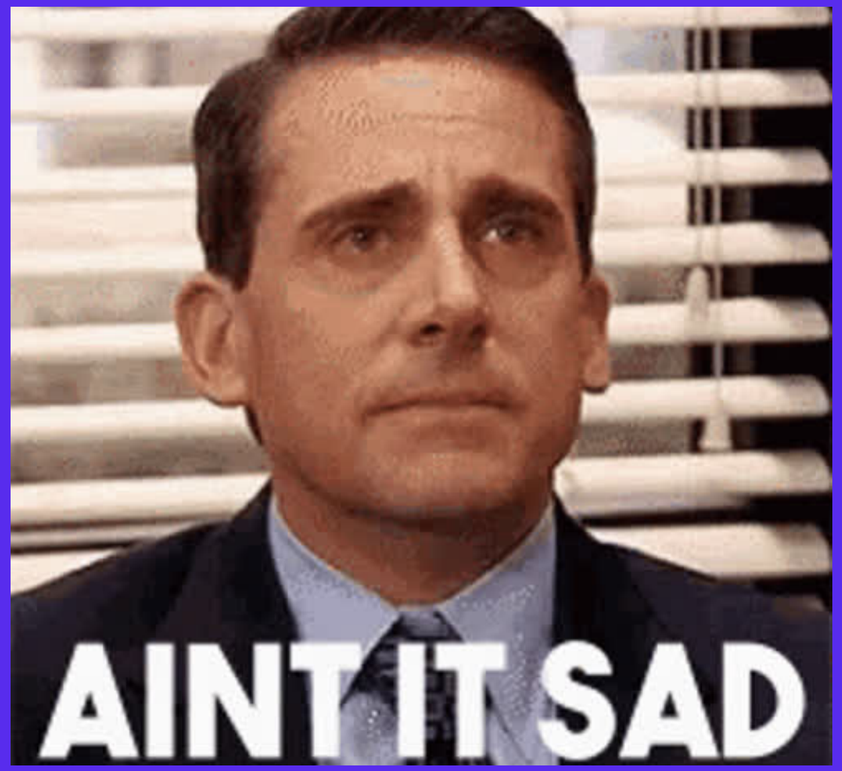
1

```
import html
import re
def textSentiment(text):
    #for sentiment analysis , 512 max is suppported
    text=text[:512]

    #translate text
    text_translated=translate_to_english(text)

    #handling emojis
    text_translated=convert_emojis_to_word(text)



    #text is from now on a list of words

    text_list = re.findall(r'\b\w+\b', text_translated)
    text_list=lemmatize(text_list)


    text_manipulated=" ".join(text_list)

    text_manipulated=text_manipulated[:512]
    #now text is a sentence again
    #ready for sentiment analysis without stopwords,handled emojis,lemmatized

    text_sentiment_score=analyze_sentiment(text_manipulated)
    text_sentiment_score=int(text_sentiment_score[0])

    return text_sentiment_score
```

2

```
from tqdm import tqdm

for index, row in tqdm(df.iterrows(), total=len(df)):
    sentence = row["sentence"]
    sentiment_score = textSentiment(sentence)
    df.at[index, "sentiment_score"] = sentiment_score
```

NOW TEXT SENTIMENT FUNCTION USES OUR FUNCTIONS DEFINED SO FAR AND RETURNS SENTIMENT SCORES AS DESIRED



SENTIMENT MODEL WORKING WITH 245.000 ROWS

SAMPLING 28.000 COMMENTS AMONG 245K

```
KKList= ["kemal","kılıçdar","gılıçdar","kemalkkılıçdaroğlu",
         "kiliçdar","kilicdar","kılıçtar","kilicdarogllu","piro", " kk "]

RTEList=["rte","reis","erdogan","tayyip","recep","receb","erduvan","erdoğan","tayyib","erdogaan","erdogann","tayyyip"]

SOList=["sinan","ogan","s.ogan","s.oğan""sinanogan","oğan","ssogan"
    "sinanogann",
    "sinanogğan",
    "sinanoğgan",
    "sinanoğaann",
    "sinanogaan",
    "sinanoğgaan",
    "sinanogaann",
    "sinanogğaan",
    "sinanoğaan",]

MIList= [
    "muharrem ince",
    "muharem ince",
    "muharrem inçe",
    "muharem inçe",
    "muharrem ince",
    "muharem ince",
    "muharrem inçe",
    "muharem inçe",
    "muharremince",
    "muharemince",
    "muharremінçe",
    "muharemінçe",
    "muharremince",
    "muharremінçe",
    "marrem","muharrem","mince"
]
```

GIVING THE CANDIDATE SCORES

```python
def whichPolitician(sentence, KKList, RTEList, SOList, MIList):
    try:
        sentence_lower = sentence.lower()
        KKscore = sum([1 for word in KKList if word in sentence_lower])
        RTEscore = sum([1 for word in RTEList if word in sentence_lower])
        SOscore = sum([1 for word in SOList if word in sentence_lower])
        MIscore = sum([1 for word in MIList if word in sentence_lower])

        return KKscore, RTEscore, SOscore, MIscore
    except Exception as e:
        print("An error occurred during processing:", str(e))
        return 0, 0, 0, 0
```

EACH COMMENT GETS 1 SCORE FOR EACH OCCURENCES OF WORDS DEFINED IN OUR LIST

04 RESULTS AND FINDINGS

SENTIMENT ANALYSIS

28000 COMMENTS

14 DIFFERENT CHANNEL

SENTIMENT AVERAGE: 2.01/5

RECEP TAYYİP ERDOĞAN

SENTIMENT AVERAGE: 1.86/5

MENTİONED: 3061

KEMAL KILIÇDAROĞLU

SENTIMENT AVERAGE: 1.82/5

MENTİONED: 1917

SİNAN OĞAN

SENTIMENT AVERAGE: 1.82/5

MENTİONED: 1001

MUHARREM İNCE

SENTIMENT AVERAGE: 1.57/5

MENTİONED: 535

SENTIMENT AVERAGES OF EACH CHANNEL:

14 DIFFERENT CHANNEL

SENTIMENT AVERAGE: 2.01/5

BABALA TV: 2.1165

AHSEN TV: 2.2985

KANAL TURKİYE: 2.133

KENDİNE MUHABİR: 1.837

OZLEMGURSES: 2.0795

NEVSİN MENGU: 2.0035

ORTAYA KARİSİK: 1.985

PURPLEBİXİ: 1.866

FATİH PORTAKAL: 1.8895

YENİSAFAK: 2.231

SOKAKTAN AL HABERİ: 1.808

ACİK MİKROFON: 2.0615

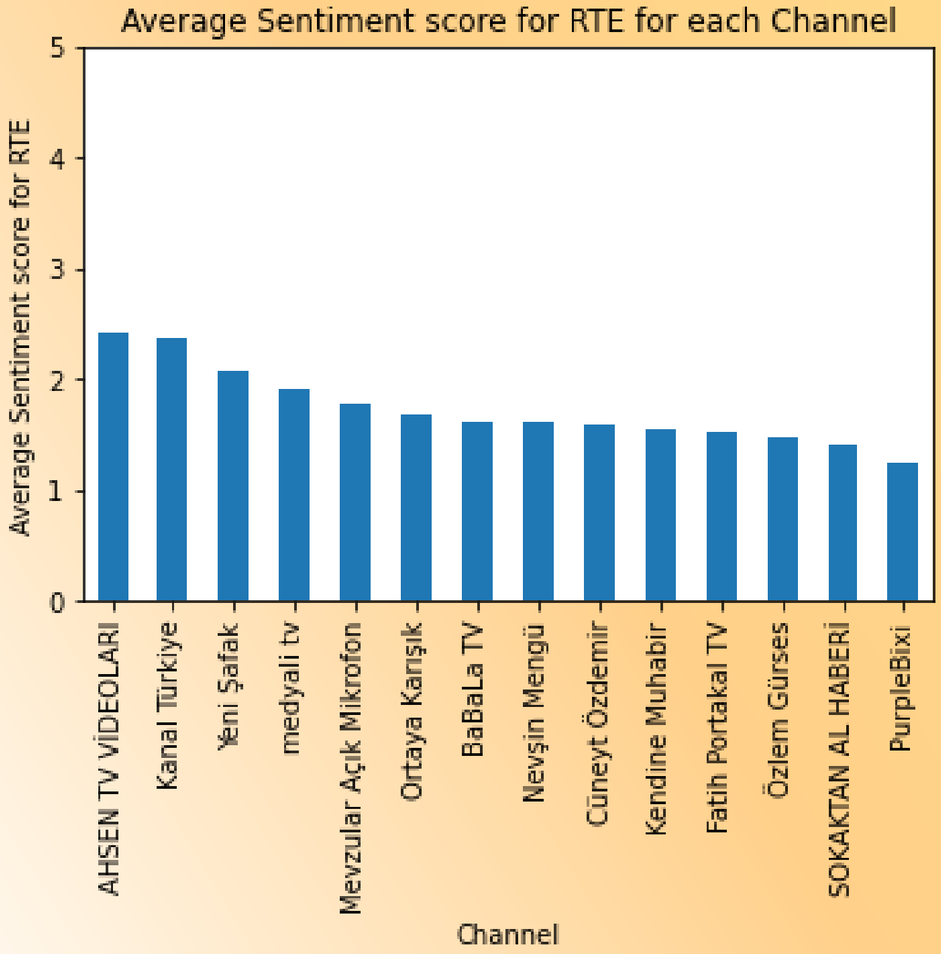
CUNEYT OZDEMİR: 1.9335

MEDYALİ TV: 1.985

SENTIMENT AVERAGE: 1.86/5

MENTIONED: 3061

RECEP TAYYİP ERDOĞAN



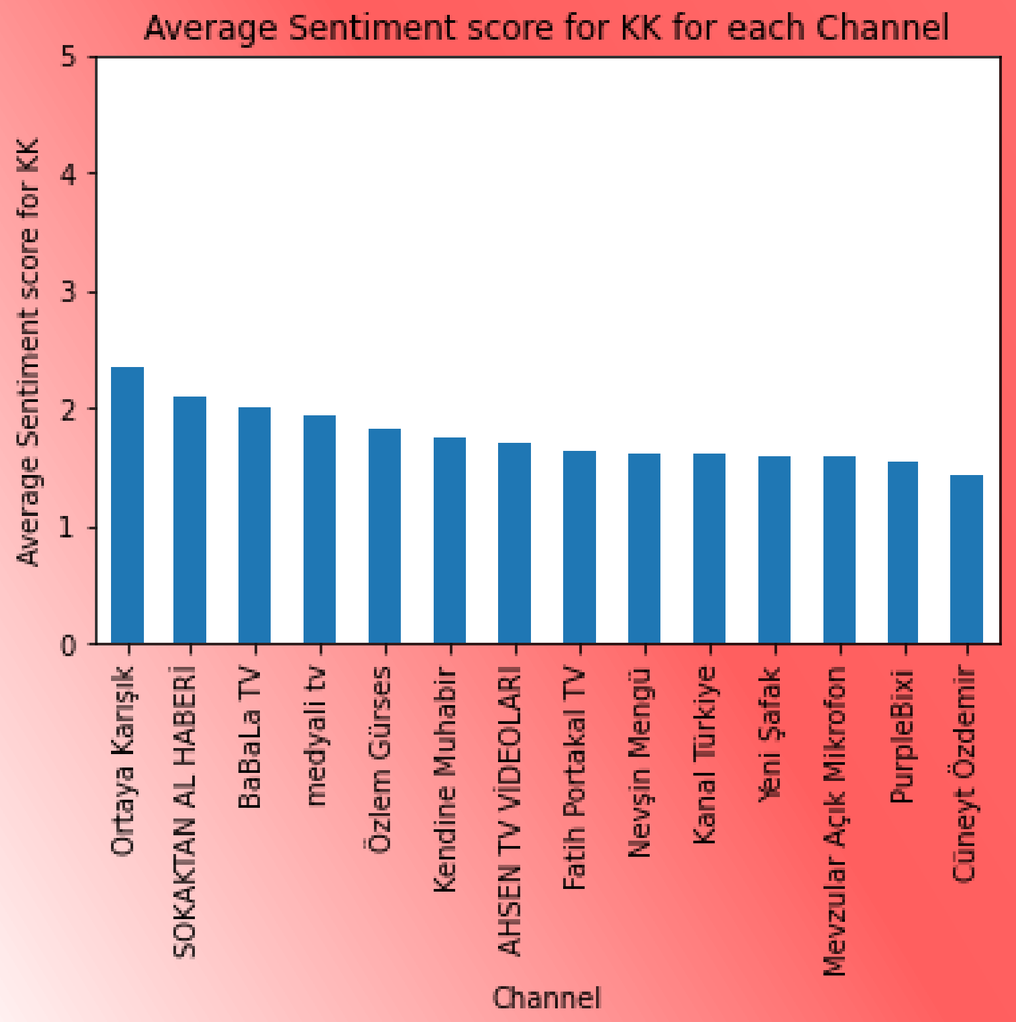
CHANNEL NAME WITH THE HIGHEST SCORE

AHSEN TV VİDEOLARI

SENTIMENT AVERAGE: 1.82/5

MENTIONED: 1917

KEMAL KILIÇDAROĞLU



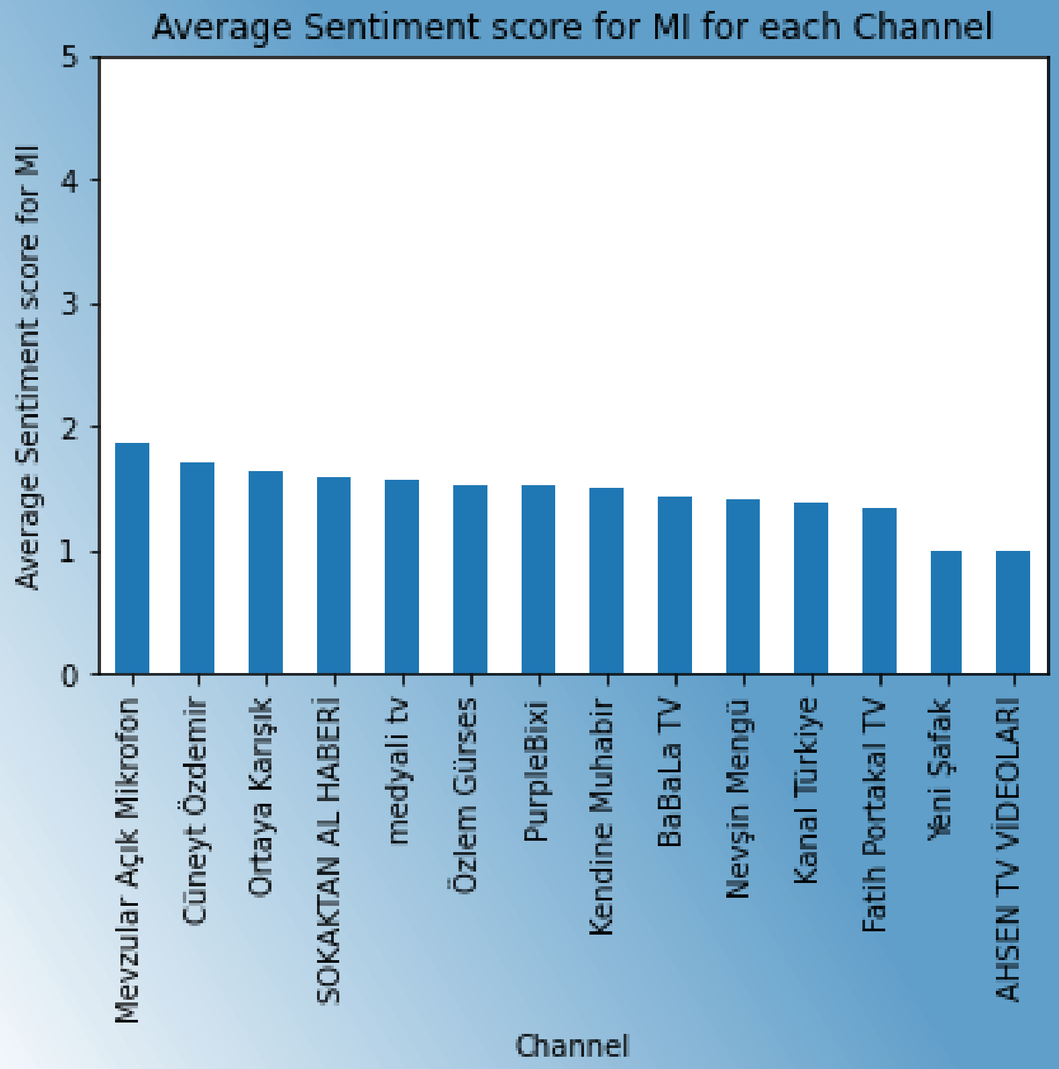
CHANNEL NAME WITH THE HIGHEST SCORE

ORTAYA KARIŞIK

SENTIMENT AVERAGE: 1.57/5

MENTIONED: 535

MUHARREM İNCE



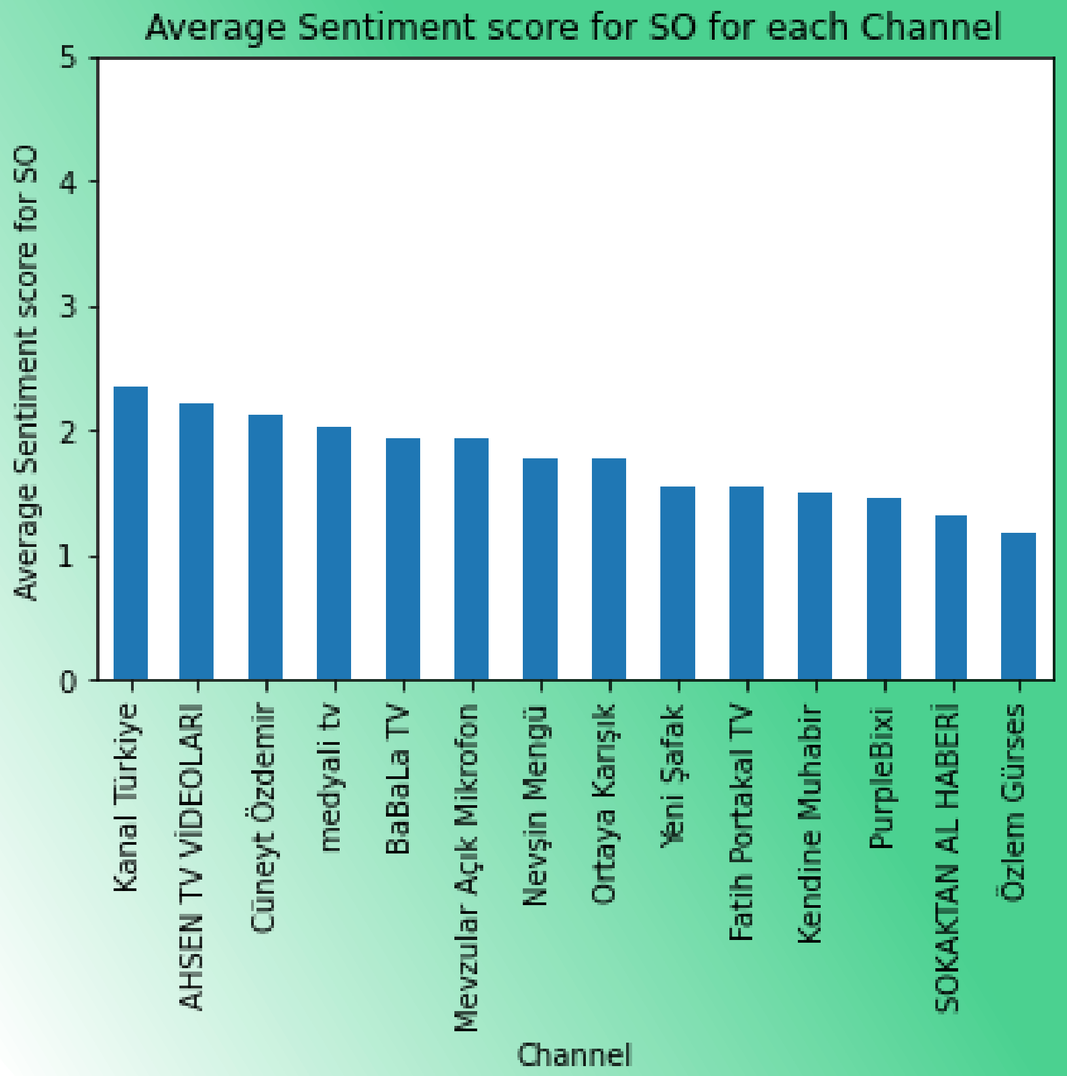
CHANNEL NAME WITH THE HIGHEST SCORE

AÇIK MİKROFON

SENTIMENT AVERAGE: 1.82/5

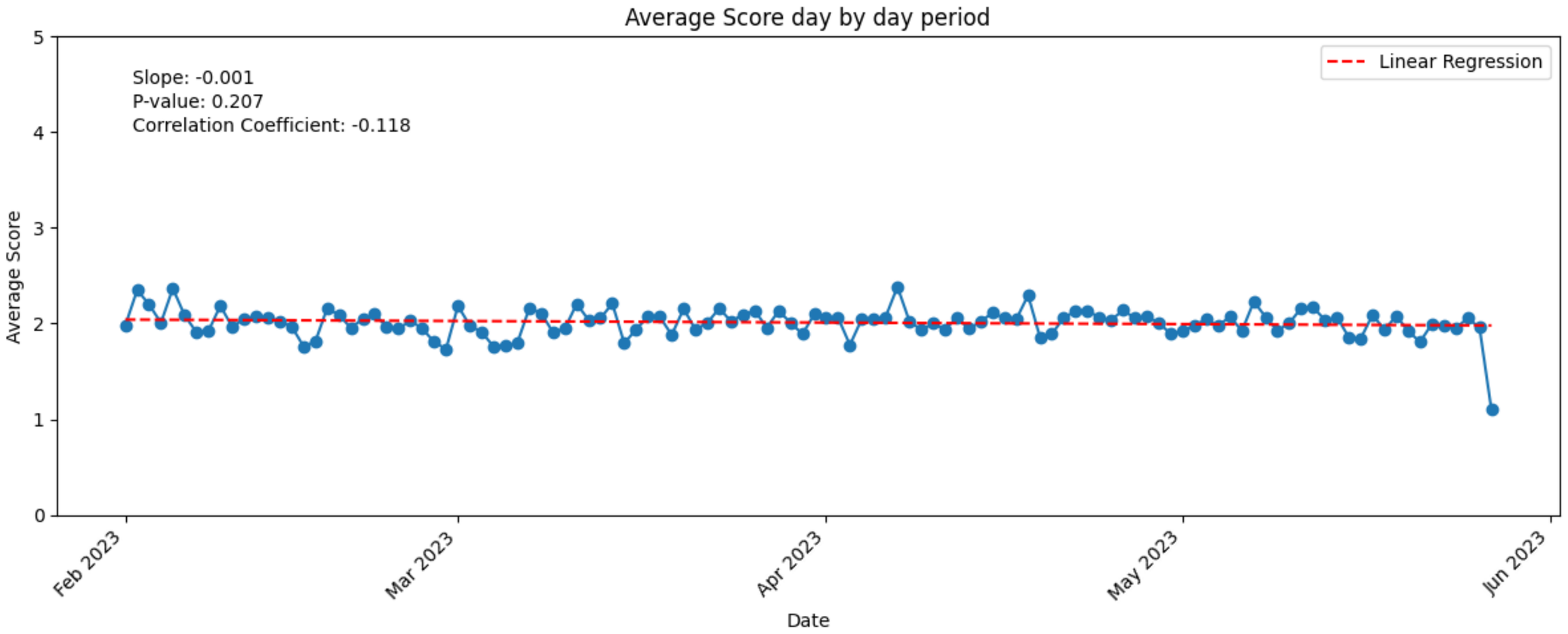
MENTIONED: 1001

SİNAN OĞAN



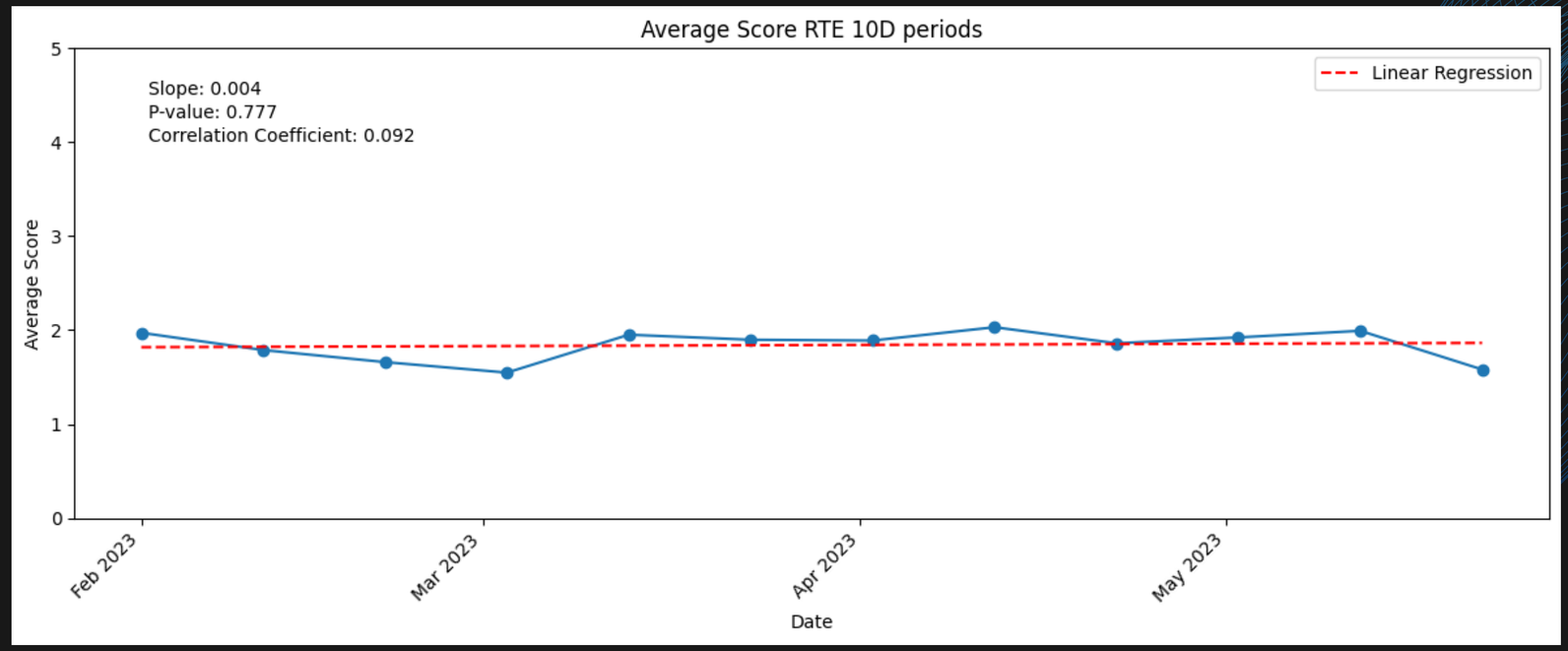
CHANNEL NAME WITH THE HIGHEST SCORE

KANAL TÜRKİYE

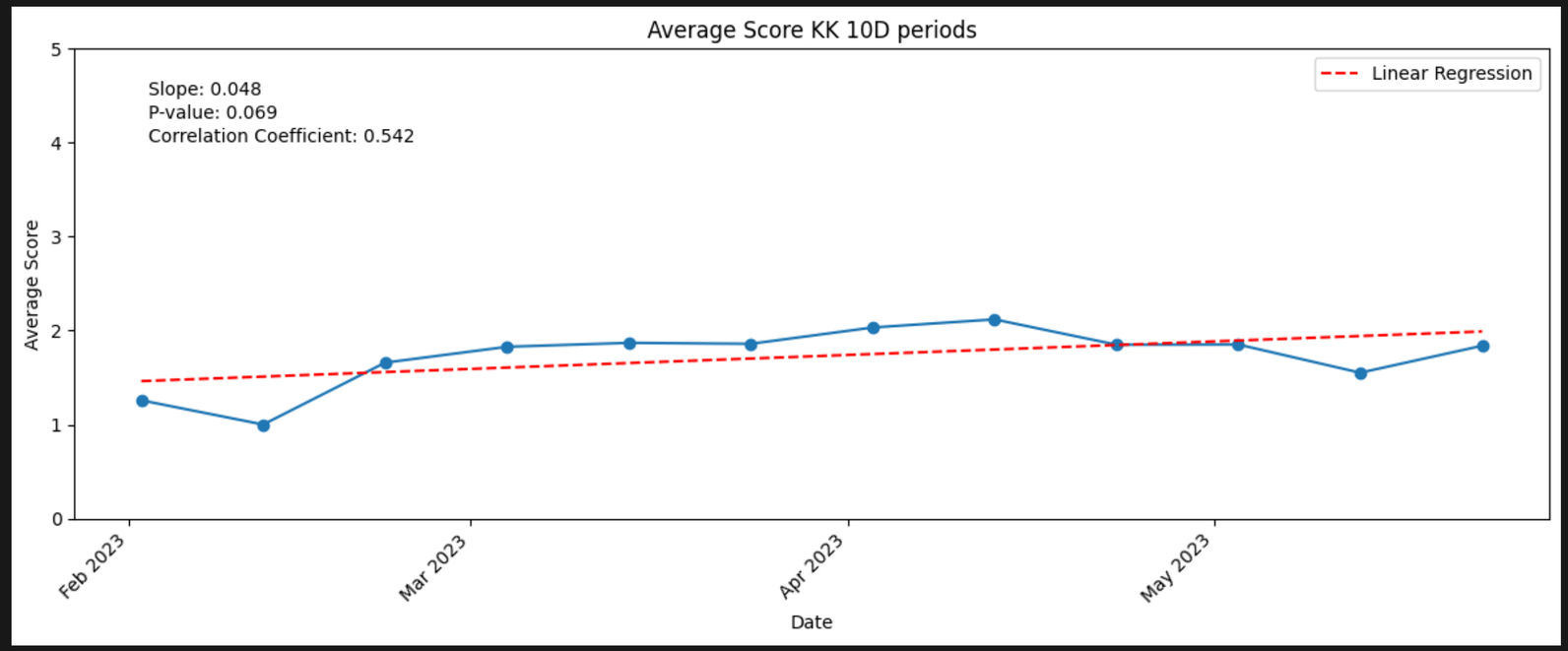


AVERAGE SENTIMENT SCORE GRAPH

RECEP TAYYİP ERDOĞAN

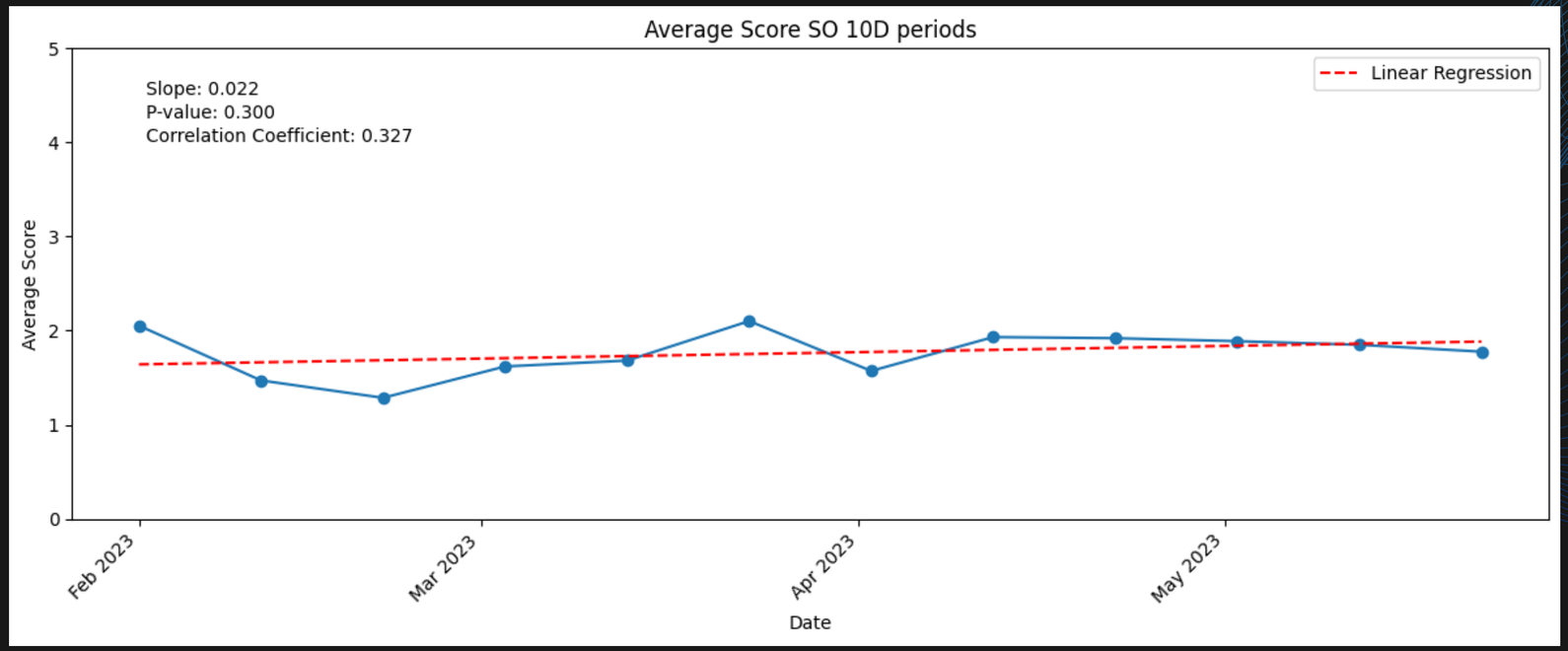


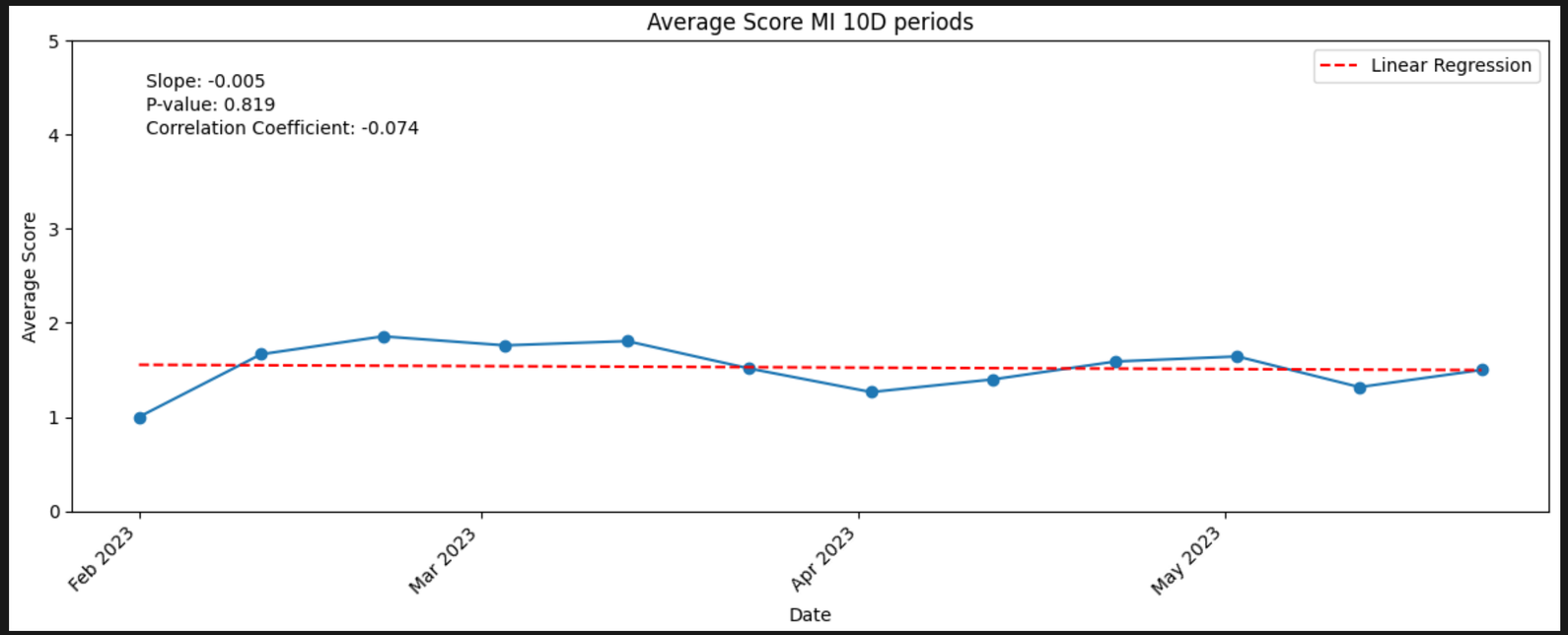
KEMAL KILIÇDAROĞLU

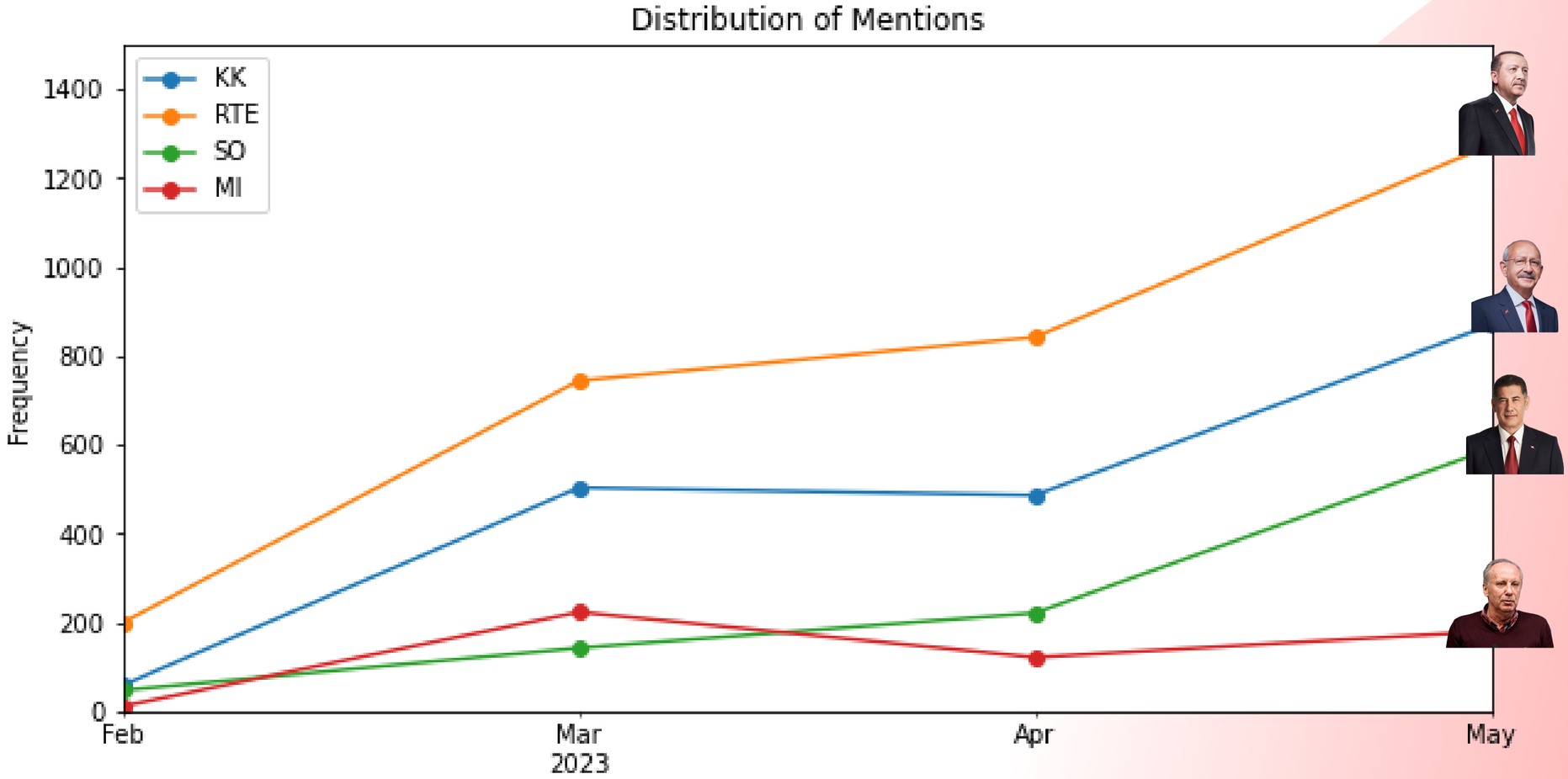


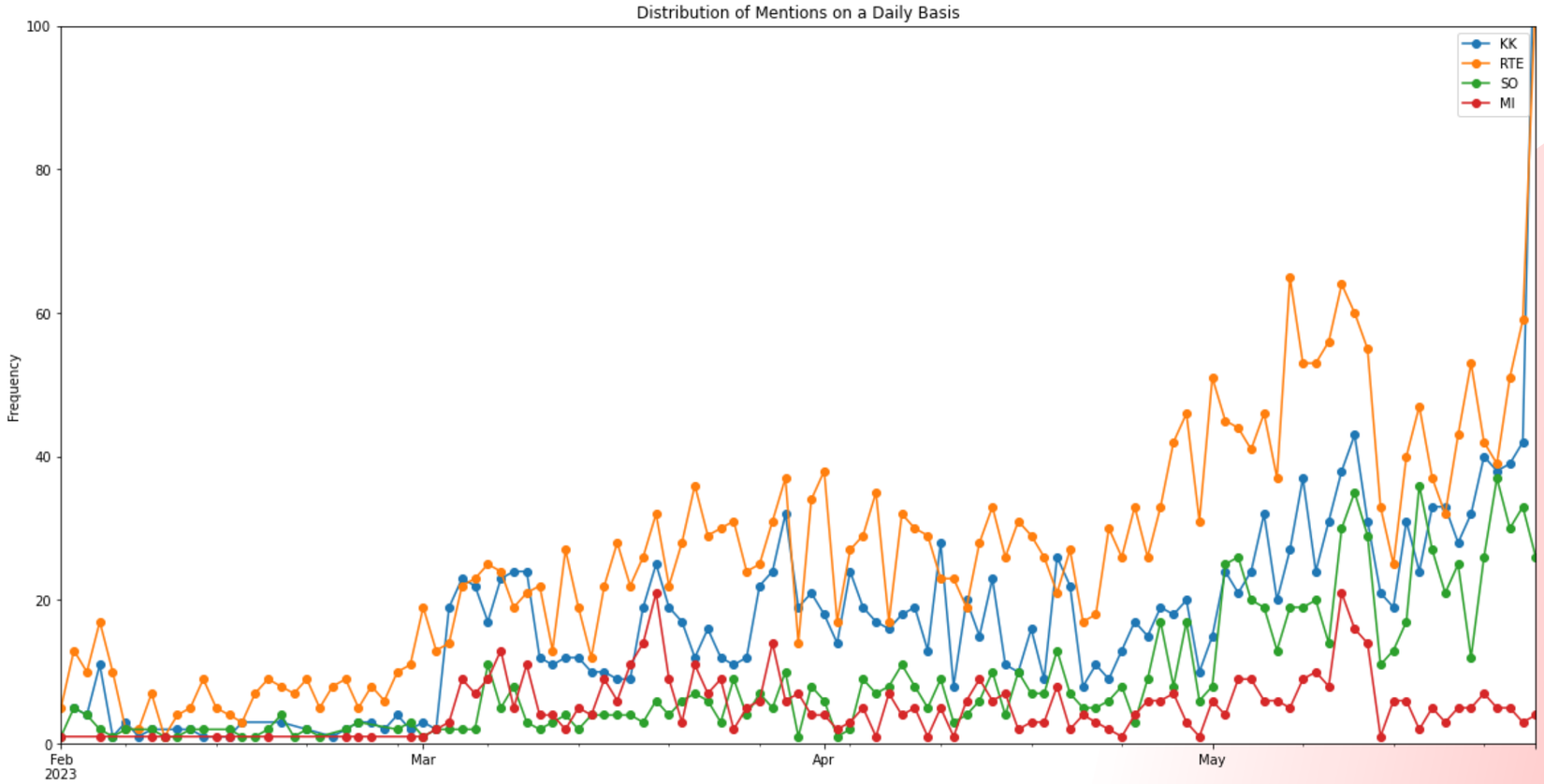
SİNAN OĞAN

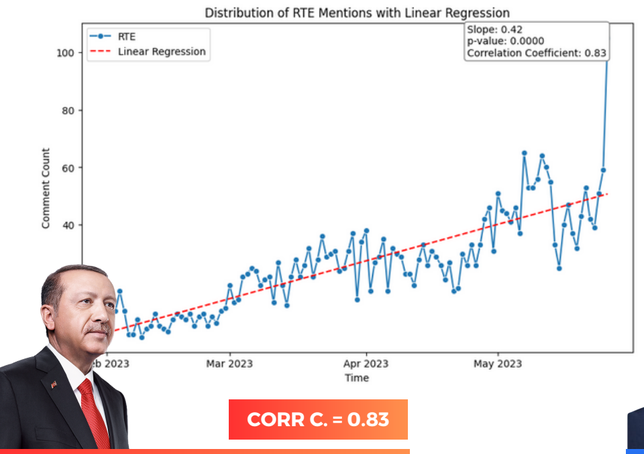
MUHARREM İNCE





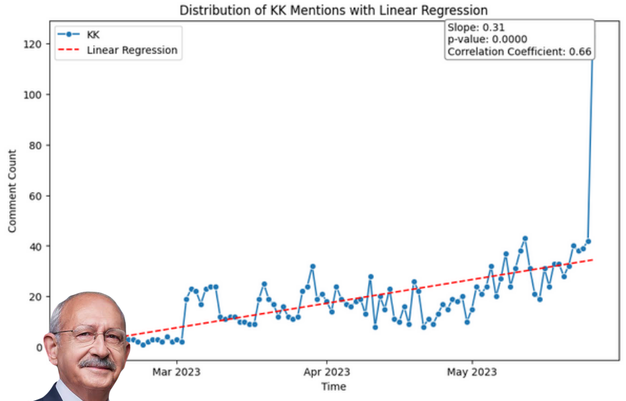

Distribution of Mentions
KK
RTE
SO
MI
Frequency
0
200
400
600
800
1000
1200
1400
Feb
Mar
2023
Apr
May





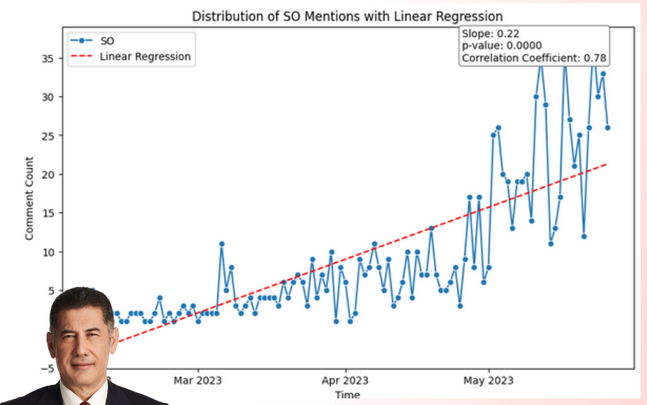
CORR C. = 0.83

RECEP TAYYİP ERDOĞAN



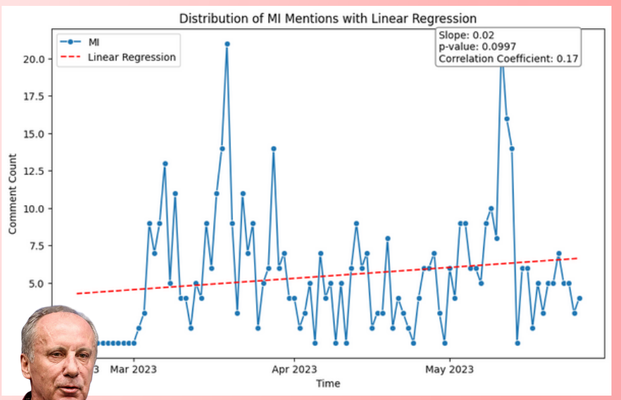
CORR C. = 0.66

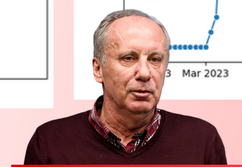
KEMAL KILIÇDAROĞLU



CORR C. = 0.78

SİNAN OĞAN





CORR C. = 0.17

MUHARREM İNCE

FINDINGS & CONCLUSION

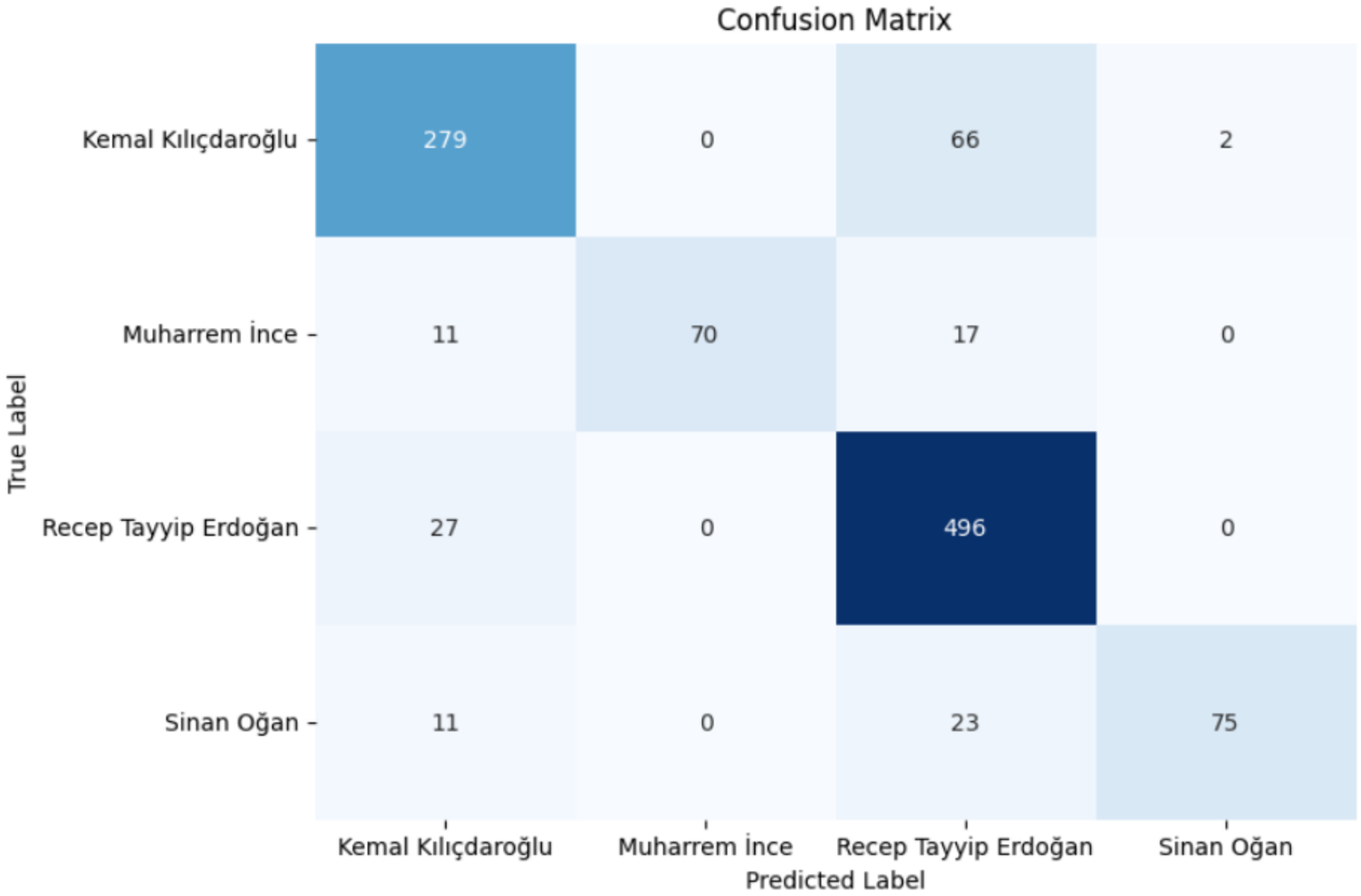
05 MACHINE LEARNING

	channelName	video_title	sentence	publishDate	sentiment_score	KKscore	RTEscore	SOscore	MIscore	candidateLabel
7	Cüneyt Özdemir	YILMAZ ÖZDİL SÖZCÜDEN NEDEN AYRILDI? #shorts	Bunlar da akp gibi olacak diye korkuyorum insa...	2023-03-08 11:24:50	1	1	0	0	0	Kemal Kılıçdaroğlu
12	Cüneyt Özdemir	KILIÇDAROĞLU'NUN 2. TUR STRATEJİSİ NE OLDU?	şuanda ali davutoğlunun keyfini hayal ediyorum...	2023-05-20 10:51:21	1	1	0	0	0	Kemal Kılıçdaroğlu
18	Cüneyt Özdemir	JAHREIN, SİYASİ ATMOSFERİ NASIL DEĞERLENDİRİYOR?	Jahrein sen dürüst bir insan degilsin. Muharre...	2023-05-12 20:53:51	1	0	0	0	2	Muharrem İnce
30	Cüneyt Özdemir	TOKİ BİNALARI NASIL AYAKTA KALDI?	Akıl bilim eğitim liyakat ahlak vicdan şart.zi...	2023-03-09 16:12:34	1	1	0	0	0	Kemal Kılıçdaroğlu
31	Cüneyt Özdemir	ANKETLERDE KİM ÖNDE? CÜNEYT ÖZDEMİR NE DEDİ NE...	Muharrem ince olmasaydı rte seçimi önde kazanı...	2023-04-27 06:41:06	1	0	1	0	2	Muharrem İnce

LABELING THE DATA

MACHINE LEARNING

SUPERVISED LEARNING



ACCURACY = 0.8542

F1 SCORE = 0.8520

LET'S SEE SOME OF IT'S PREDICTIONS

HDP

KEMAL KILIÇDAROĞLU:161
RECEP TAYYİP ERDOĞAN: 138

DİN

RECEP TAYYİP ERDOĞAN: 2187
KEMAL KILIÇDAROĞLU:161
MUHARREM İNCE: 1

TOTAL NUMBER OF PREDICTIONS

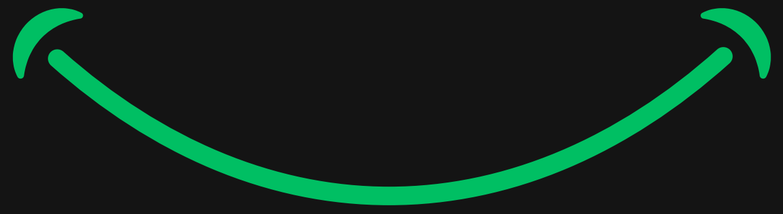
RECEP TAYYİP ERDOĞAN: 20354

KEMAL KILIÇDAROĞLU: 2203

SİNAN OĞAN: 14

MUHARREM İNCE: 47

THANKS FOR LISTENING



MUSTAFA TOPCU HASAN YILMAZ ANIL ŞEN ORHUN ÖZPAY MERT KULECİ