Prediction of YouTube Video Trend for BigFame Co.

Team: Respect

Group 16

Qiaoling Chen: e0983208

Lyu Xinyu: e0983260

Xie Siteng: e0983376

Yashna Gogineni: e0983387

Zhao Rudan: e0983390

Content

Biz & Data

Models

Conclusion

Background

Data Processing

Data Insight

Topic Modeling

Clustering

Time Series

Sentiment Analysis

Limitations

Business Insight

Project Management

Background

BIGFAME — Company

- **BigFame:**

  Multi-Channel

  Network (MCN) Co.

- **Org. Structure:**

1 **Marketing Department**

Through various channels, activities, to promote videos and activities, business negotiations and cooperation with advertisers and so on

2 **Content Department**

Responsible for the creative design and development of short video and live content. Frequently analyze and study popular video works on various platforms, and keep up with the hot spots.

3 **Operation Department**

Responsible for the overall operation planning of the company, analyze and summarize all operational indicators, and formulate reasonable operational objectives and plans.

4 **Host Management Department**

Be responsible for the discovery, recruitment and training of influencers to create star Internet celebrity anchors.

5 **HR Department**

6 **Administration Department**

7 **Financial Department**

8 **Technology Department**

Background

Challenges

- **Revenue generate from Key Opinion Leader(KOL) in different categories**

- **Challenges we can deal with by this data analysis project:**

Host Management

Lack of quality talent.

Content Creation

Lack of quality content ideas.
The content is seriously homogenized

Competitive Pressures

KOL erosion.
The problem of professionalization and contract.

Platform Pressure

Content operation investment in different categories of platforms

Background

RESPECT — Team

- **Respect:**
  - **Data analysis team for Content Operations Department of BigFame Co.**
- **Team members:**
  - **Qiaoling Chen**
  - **Lyu Xinyu**
  - **Xie Siteng**
  - **Yashna Gogineni**
  - **Zhao Rudan**

Background

Business Objectives

Become a much more strategic generator of value and potentially also a powerful differentiator

Looking at video popularity trends and make well preparation for the hot category next period

Make agile, flexible, and resilient operational decisions based on full range of clean data

Background

Technical Objectives

- Classifying the sentiment
- Machine learning model such as clustering to identify different viewers group

- Mathematical model such as TOPSIS to calculate videos' popular
- Time series model to predict the future popularity trend for all categories

- Preparing data including integration
- Generating tags for each video based on the title and comments
- Building an interactive dashboard

Data Pre-pocessing

Data Pre-processing

Data

Remove

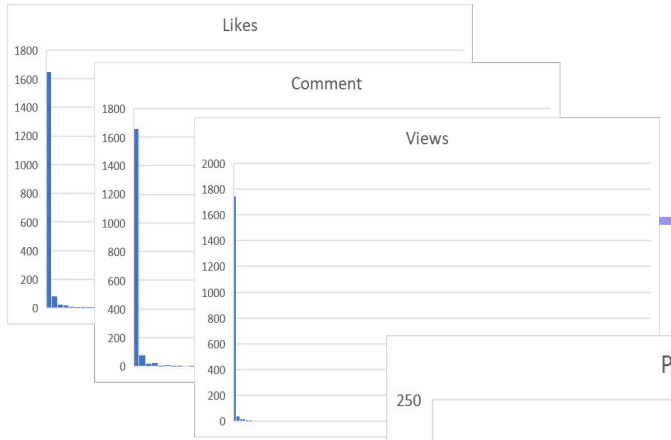
- Missing value
- Duplicated value
- Outliers

Add Feature

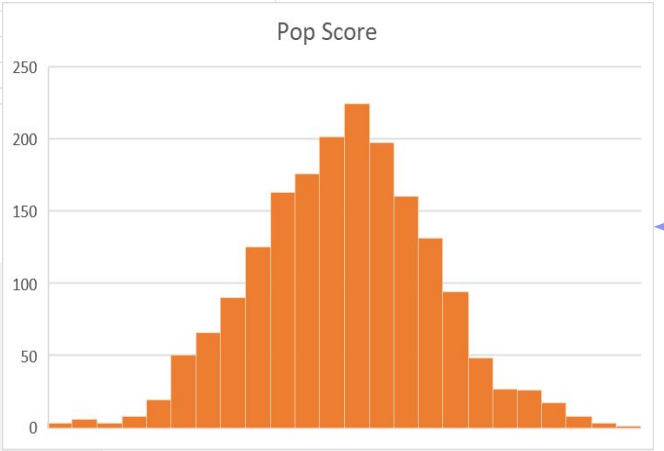
- Generate Pop Score:

  TOPSIS Method Based on Entropy Weight

- Normalization



Likes	0.301295
Comments	0.294011
Views	0.404695




```python
# Def: Standardize the matrix
def Y_ij(data1):
    for i in data1.columns:
        for j in range(n+1):
            if i == str(f'X{j}negative'):  # negative
                data1[i]=(np.max(data1[i])-data1[i])/(np.max(data1[i])-np.min(data1[i]))
            else:   # positive
                data1[i]=(data1[i]-np.min(data1[i]))/(np.max(data1[i])-np.min(data1[i]))
    return data1
```

```python
# Def: Caculate the entropy
# Build empty matrix
None_ij = [[None] * n for i in range(m)]
def E_j(data2):
    data2 = np.array(data2)
    E = np.array(None_ij)
    for i in range(m):
        for j in range(n):
            if data2[i][j] == 0:
                e_ij = 0.0
            else:
                p_ij = data2[i][j] / data2.sum(axis=0)[j]
                e_ij = (-1 / np.log(m)) * p_ij * np.log(p_ij)
            E[i][j] = e_ij
    E_j=E.sum(axis=0)
    return E_j
```

```python
# Calculate the optimum and the worst
Imax_j = Z_ij.max(axis=0)
Imin_j = Z_ij.min(axis=0)
Dmax_ij = np.array(None_ij)
Dmin_ij = np.array(None_ij)
for i in range(m):
    for j in range(n):
        Dmax_ij[i][j] = (Imax_j[j] - Z_ij[i][j]) ** 2
        Dmin_ij[i][j] = (Imin_j[j] - Z_ij[i][j]) ** 2
```

```python
# Calculate the pop score
Dmax_i = Dmax_ij.sum(axis=1)**0.5
Dmin_i = Dmin_ij.sum(axis=1)**0.5
S_i = Dmin_i/(Dmin_i + Dmax_i)
S_i= pd.Series(S_i, index=video_data.index, name='pop_score')
```

Data Pre-processing

Text

Replace

- Contractions
- Capital → Lowercase

Remove

- Non ASCII Words
- Punctuation
- Stopwords

Lemmatize

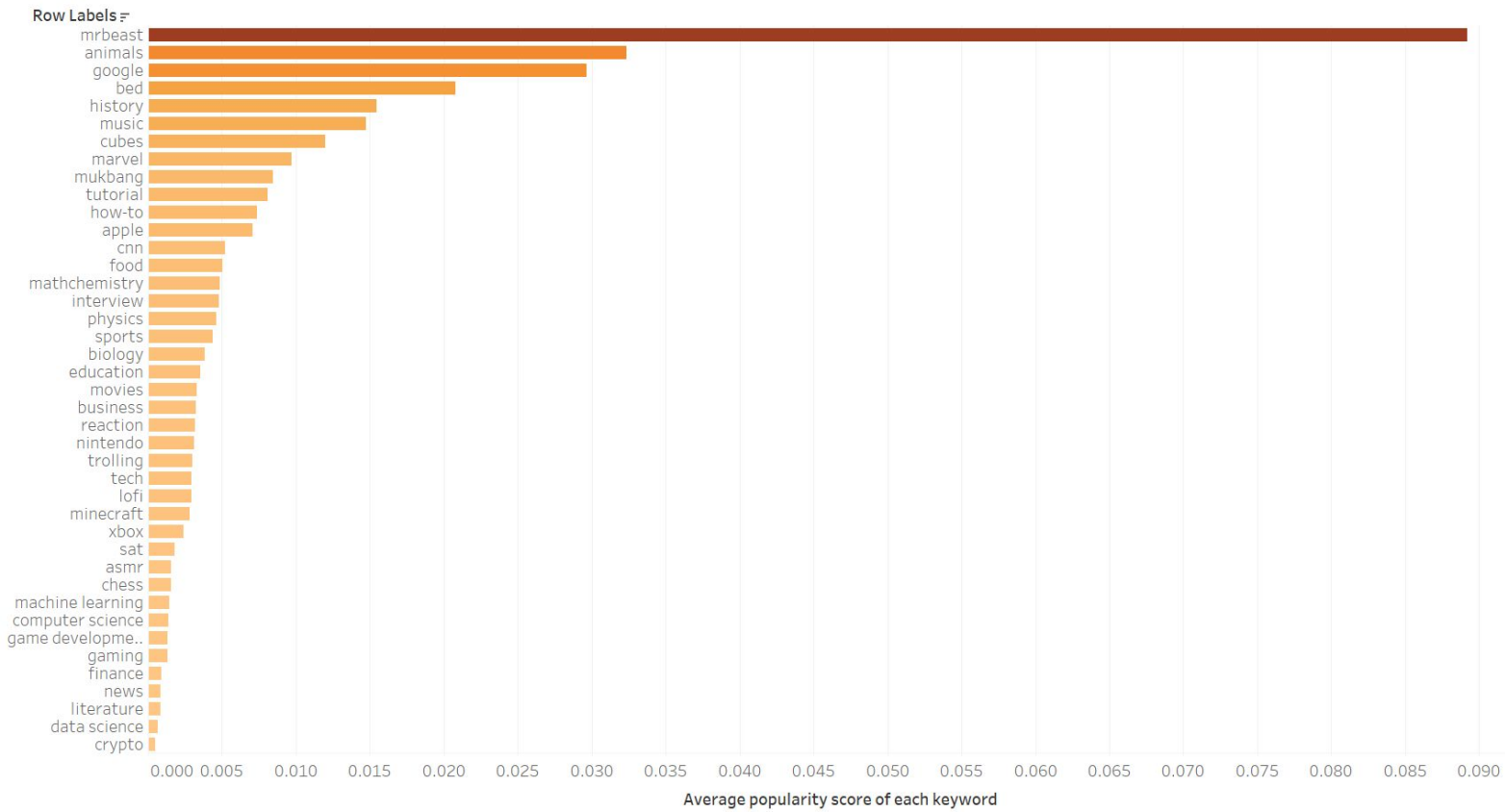
Tokens

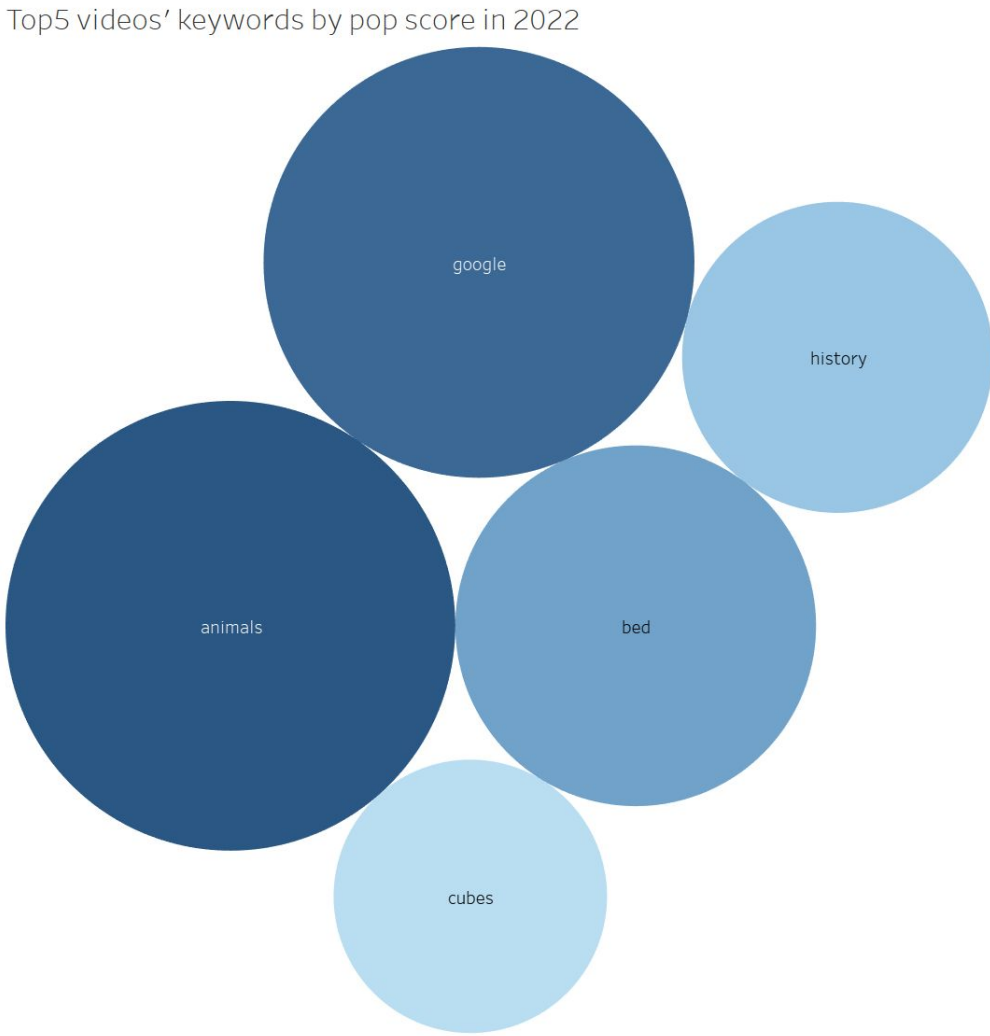
```
def replace_contractions(text):
    """Replace contractions in string of text"""
    return contractions.fix(text)


def remove_non_ascii(words):
    """Remove non-ASCII characters from list of tokenized words"""
    new_words = []
    for word in words:
        new_word = unicodedata.normalize('NFKD', word).encode('ascii', 'ignore').decode('utf-8', 'ignore')
        new_words.append(new_word)
    return new_words

def to_lowercase(words):
    """Convert all characters to lowercase from list of to
    new_words = []
    for word in words:
        new_word = word.lower()
        new_words.append(new_word)
    return new_words
```

```
def remove_punctuation(words):
    """Remove punctuation from list of tokenized words"""
    new_words = []
    for word in words:
        new_word = re.sub(r'[^\w\s]', '', word)
        new_word = re.sub(r'(.)\1+', r'\1\1', new_word)
        if new_word != '':
            new_words.append(new_word)
    return new_words

def remove_stopwords(words):
    """Remove stop words from list of tokenized words"""
    new_words = []
    sw = stopwords.words('english')
    for word in words:
        if word not in sw:
            new_words.append(word)
    return new_words

def lemmatize_verbs(words):
    """Lemmatize verbs in list of tokenized words"""
    lemmatizer = WordNetLemmatizer()
    lemmas = []
    for word in words:
        lemma = lemmatizer.lemmatize(word, pos='v')
        lemmas.append(lemma)
    return lemmas
```

Data Insight

Data Insight



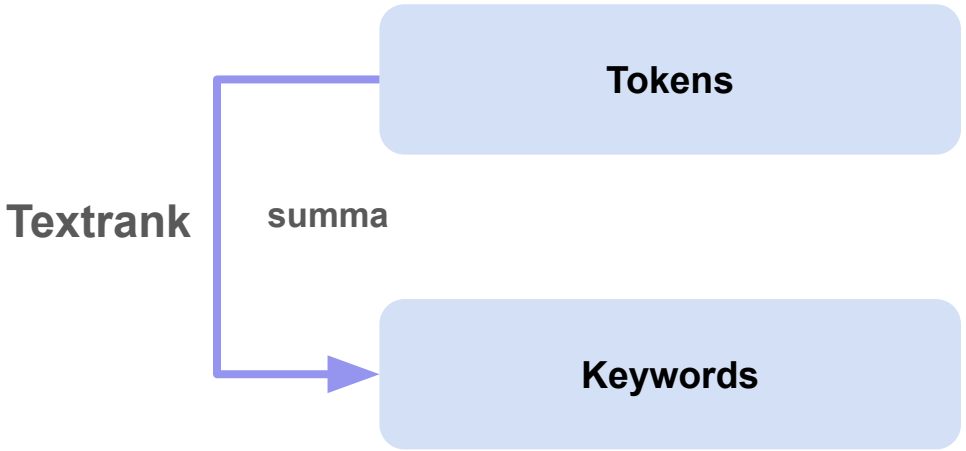


Topic Modeling

Topic Modeling

Preparation for Clustering & Sentiment Analysis

- Each video has 10 comments
- Topic Extraction for Each Video



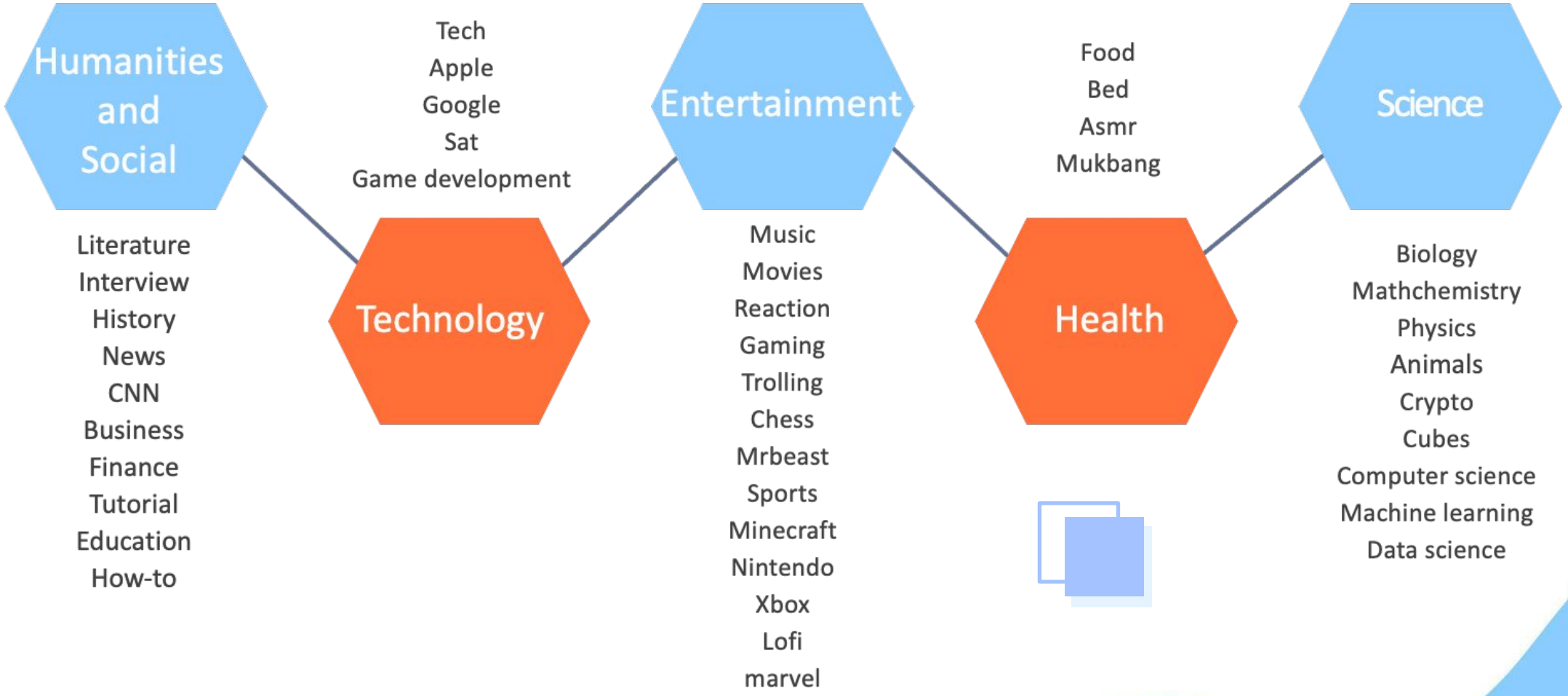

```python
from summa import keywords

def generate_key(comm):

    TR_keywords = keywords.keywords(comm, scores=True)
    com_key=[]
    for x,y in TR_keywords:
        com_key.append(x)
    return com_key


df_comment['Comment_keyword']=df_comment['Comment_tag'].apply(generate_key)
# for i range(len(df_comment)):
#     df_comment['Comment'][i]=df_comment['Comment'][i]
```

```python
df_comment.to_csv('data/comment_keyword_true_tag.csv')
df_comment['Comment_keyword'][0:10]
```

```
Video ID
--ZI0dSbbNU    [eating, eat, eats, fry, feel, smile, man, suf...
--hxd1Cr0qg    [russia, til, n, wo, realize, ukraine, analyza...
--ixiTypG8g    [college, colleges, tuition, regulated, regula...
-64r1hcxtV4    [life, inspiration, inspiring, inspire, health...
-6IgkG5yZfo    [thankful, thank, thanks, time, help, helped, ...
-7hzaGya86g    [good, like, mikey, dinner, food, eating, dump...
-8TnsjDRXUE    [kid, amazing, know, knowing, guy, time, work,...
-9hjdvULDyc    [reading, prediction, th, nd, wa, writing, tes...
-Cr69sGnrk8    [doe, u, fan, thank, thanks, sccoby, support, ...
-D4S6Tpn044                                          [getbonus]
Name: Comment_keyword, dtype: object
```

Time Series Analysis

Time Series Analysis



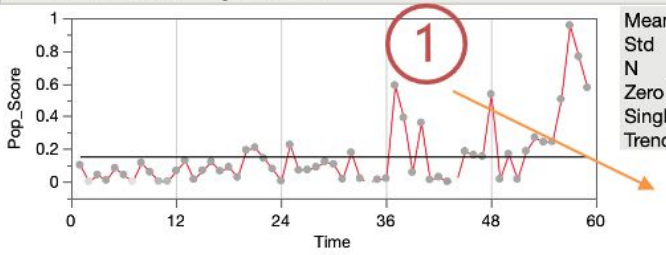
ENTERTAINMANT

Time Series Analysis

Non- stationary process

Stationary process

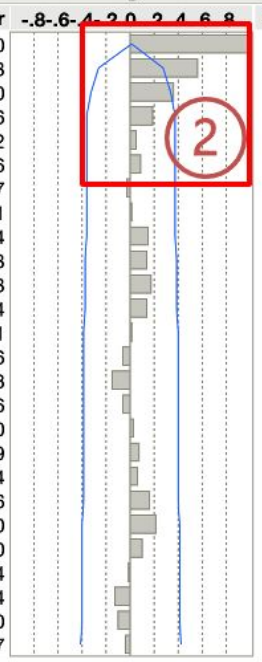
Time Series Pop_Score



Mean	0.1551984
Std	0.1968848
N	59
Zero Mean ADF	-2.384636
Single Mean ADF	-3.439207
Trend ADF	-4.541994

An Increasing Trend

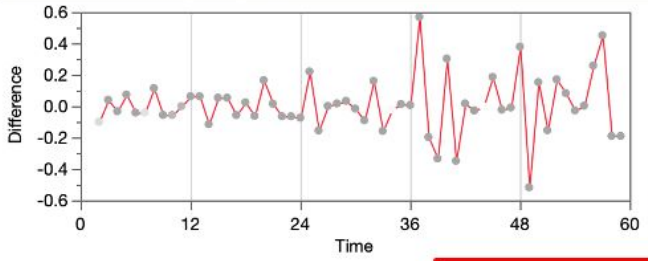
Time Series Basic Diagnostics



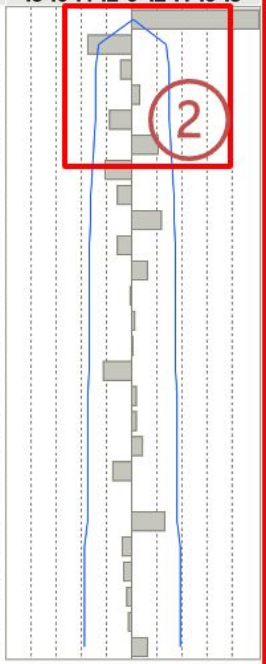
Lag	AutoCorr	Ljung-Box Q	p-Value	Lag	Partial
0	1.0000	.	.	0	1.0000
1	0.5643	19.7583	<.0001*	1	0.5643
2	0.3640	28.1252	<.0001*	2	0.0669
3	0.1906	30.4595	<.0001*	3	-0.0572
4	0.0462	30.5993	<.0001*	4	-0.0856
5	0.0926	31.1704	<.0001*	5	0.1502
6	-0.0267	31.2187	<.0001*	6	-0.1540
7	0.0101	31.2257	<.0001*	7	0.0768
8	0.1524	32.8651	<.0001*	8	0.2232
9	0.1363	34.2014	<.0001*	9	-0.0310
10	0.1733	36.4069	<.0001*	10	0.0102
11	0.1354	37.7809	<.0001*	11	0.0434
12	0.0141	37.7962	0.0002*	12	-0.1325
13	-0.0576	38.0558	0.0003*	13	-0.1243
14	-0.1448	39.7333	0.0003*	14	-0.0072
15	-0.0626	40.0537	0.0004*	15	0.1410
16	0.0230	40.0979	0.0008*	16	0.0297
17	0.0789	40.6307	0.0011*	17	0.0674
18	0.0644	40.9944	0.0015*	18	-0.0846
19	0.1656	43.4621	0.0011*	19	0.1735
20	0.2080	47.4554	0.0005*	20	0.0552
21	0.0960	48.3285	0.0006*	21	-0.1550
22	-0.0184	48.3615	0.0010*	22	-0.0697
23	-0.1204	49.8097	0.0010*	23	0.0073
24	-0.0940	50.7186	0.0011*	24	0.0082
25	-0.0377	50.8689	0.0017*	25	0.0168

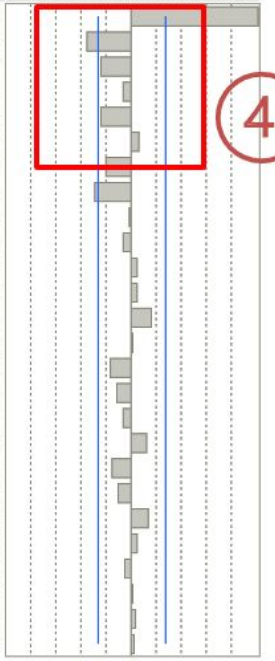
One Regular Differencing

Difference: (1-B)^1



Mean	0.0081648
Std	0.1765736
N	58
Zero Mean ADF	-10.71045
Single Mean ADF	-10.67164
Trend ADF	-10.6278





Lag	AutoCorr	Ljung-Box Q	p-Value	Lag	Partial
0	1.0000	.	.	0	1.0000
1	-0.3484	7.4128	0.0065*	1	-0.3484
2	-0.0813	7.8234	0.0200*	2	-0.2307
3	0.0653	8.0929	0.0441*	3	-0.0601
4	-0.1796	10.1725	0.0376*	4	-0.2323
5	0.2079	13.0096	0.0233*	5	0.0667
6	-0.2062	15.8562	0.0145*	6	-0.1964
7	-0.1115	16.7046	0.0194*	7	-0.2819
8	0.2387	20.6689	0.0081*	8	-0.0128
9	-0.1058	21.4636	0.0107*	9	-0.0622
10	0.1300	22.6890	0.0120*	10	0.0511
11	-0.0161	22.7082	0.0194*	11	0.0519
12	0.0271	22.7636	0.0298*	12	0.1665
13	0.0133	22.7772	0.0444*	13	0.0179
14	-0.2190	26.5721	0.0219*	14	-0.1574
15	0.0408	26.7068	0.0312*	15	-0.1146
16	0.0352	26.8095	0.0436*	16	-0.0581
17	0.0828	27.3909	0.0526	17	0.1263
18	-0.1447	29.2131	0.0458*	18	-0.1543
19	-0.0037	29.2143	0.0627	19	-0.1029
20	0.2664	35.7154	0.0166*	20	0.1394
21	-0.0786	36.2968	0.0202*	21	0.0464
22	-0.0668	36.7277	0.0253*	22	-0.0469
23	-0.0326	36.8333	0.0338*	23	0.0067
24	-0.0300	36.9253	0.0445*	24	0.0381
25	0.1218	38.4904	0.0414*	25	0.0292

ENTERTAINMANT

Time Series Analysis

Model	DF	Variance	AIC	SBC	RSquare	-2LogLH	MAPE	MAE
ARIMA(1, 1, 1)	**55**	**0.024106**	**-45.44431**	**-39.26298**	**0.404**	**-51.44431**	**692.7482**	**0.105634**
AR(1)	57	0.026241	-44.93511	-40.78003	0.345	-48.93511	564.0437	0.10742
IMA(1, 1)	56	0.02596	-44.82172	-40.70083	0.362	-48.82172	616.0846	0.108038
ARMA(1, 1)	56	0.025985	-44.41087	-38.17826	0.362	-50.41087	557.495	0.105578
AR(2)	56	0.026079	-44.25493	-38.02232	0.36	-50.25493	556.4575	0.105483
IMA(1, 2)	55	0.02572	-44.20452	-38.02319	0.379	-50.20452	651.6136	0.104614
ARIMA(2, 1, 1)	54	0.024432	-43.93216	-35.69039	0.407	-51.93216	679.9363	0.105215
Seasonal ARIMA(1, 1, 1)(0, 0, 1)12	54	0.024561	-43.45321	-35.21144	0.404	-51.45321	693.1448	0.10542
Seasonal ARIMA(1, 1, 1)(1, 0, 0)12	54	0.024559	-43.45152	-35.20975	0.404	-51.45152	693.0725	0.105457
ARIMA(1, 1, 2)	54	0.026167	-42.35606	-34.11429	0.38	-50.35606	609.5492	0.105772

- **Diagnostic Checking---Model Selection Criteria**

✓ Akaike Information Criterion (AIC) = -2 ln(L) + 2k

✓ Schwartz Bayesian Criterion (SBC) = -2 ln(L) + k ln(n)

ENTERTAINMANT

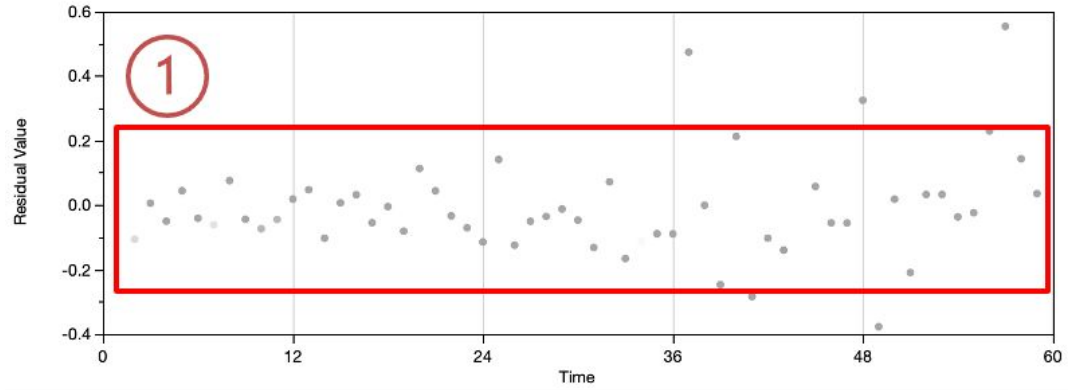
Time Series Analysis

- **Diagnostic Checking---Residual Analysis**

✓ Check the Residual Values

✓ Check the ACF and PACF

✓ Check the Result of Protmanteau Test

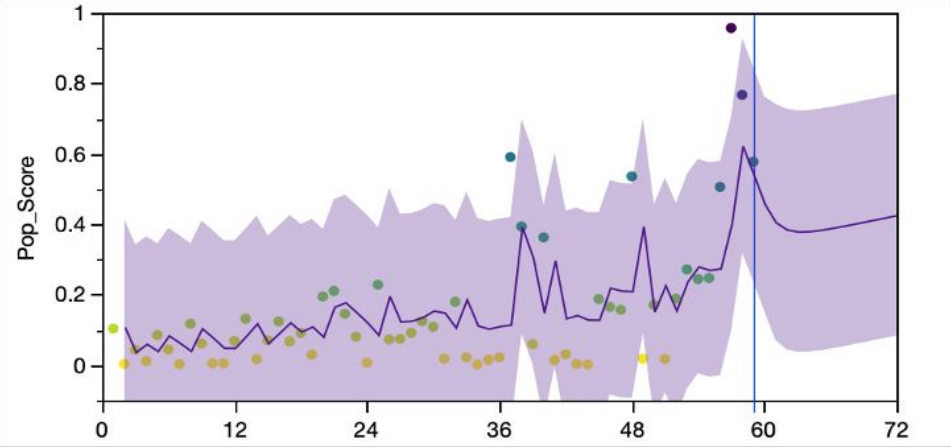


2 3 2

Lag	AutoCorr	-.8 -.6 -.4 -.2 0 .2 .4 .6 .8	Ljung-Box Q	p-Value	Lag	Partial	-.8 -.6 -.4 -.2 0 .2 .4 .6 .8
0	1.0000		.	.	0	1.0000	
1	-0.0509		0.1581	0.6909	1	-0.0509	
2	0.0509		0.3191	0.8525	2	0.0484	
3	0.0664		0.5981	0.8969	3	0.0717	
4	-0.1585		2.2169	0.6959	4	-0.1556	
5	0.1166		3.1092	0.6831	5	0.0983	
6	-0.2016		5.8305	0.4424	6	-0.1895	
7	-0.1184		6.7863	0.4515	7	-0.1266	
8	0.1705		8.8102	0.3586	8	0.1589	
9	-0.0509		8.9940	0.4378	9	0.0230	
10	0.1135		9.9272	0.4469	10	0.0475	
11	0.0229		9.9662	0.5334	11	0.0211	
12	-0.0080		9.9710	0.6185	12	0.0143	
13	-0.0503		10.1664	0.6803	13	-0.1628	
14	-0.2387		14.6735	0.4008	14	-0.2022	
15	-0.0373		14.7860	0.4669	15	-0.0160	
16	-0.0013		14.7861	0.5404	16	0.0373	
17	0.0593		15.0841	0.5894	17	0.1187	
18	-0.0986		15.9300	0.5974	18	-0.1528	
19	0.0542		16.1918	0.6444	19	0.0324	
20	0.2438		21.6332	0.3607	20	0.1826	
21	-0.0153		21.6552	0.4196	21	-0.0344	
22	-0.0409		21.8171	0.4709	22	-0.0902	
23	-0.0824		22.4918	0.4907	23	-0.0197	
24	-0.0656		22.9320	0.5238	24	-0.0073	
25	0.0571		23.2758	0.5615	25	-0.0089	

Time Series Analysis

ENTERTAINMANT



Model: ARIMA(1, 1, 1)

Model Summary

DF	55	Stable	Yes
Sum of Squared Errors	1.32581033	Invertible	Yes
Variance Estimate	0.02410564		
Standard Deviation	0.15525992		
Akaike's 'A' Information Criterion	-45.444309		
Schwarz's Bayesian Criterion	-39.26298		
RSquare	0.40369567		
RSquare Adj	0.38201188		
MAPE	692.74816		
MAE	0.10563359		
-2LogLikelihood	-51.444309		

1

2

Parameter Estimates

Term	Lag	Estimate	Std Error	t Ratio	Prob>|t|	Constant Estimate	Mu
AR1	1	0.4617515	0.1225077	3.77	0.0004*	0.00341949	0.006353
MA1	1	1.0000000	0.0735173	13.60	<.0001*		
Intercept	0	0.0063530	0.0020585	3.09	0.0032*		

- **ARIMA(1, 1, 1) Model Summary**

✓ Stable and Invertible

✓ Significance of Parameters (at 95% confidence level)

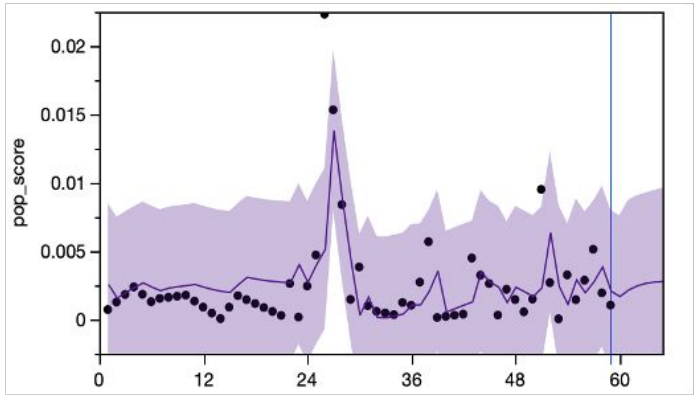
$$(1-\phi_1 B)(1-B)\,y_t = \phi_0 + (1-\theta_1 B)\,\varepsilon_t$$

- **Result Analysis**

✓ In general, the popularity of the topic of ENTERTAINMENT shows an upward trend.

Time Series Analysis

TECHNOLOGY

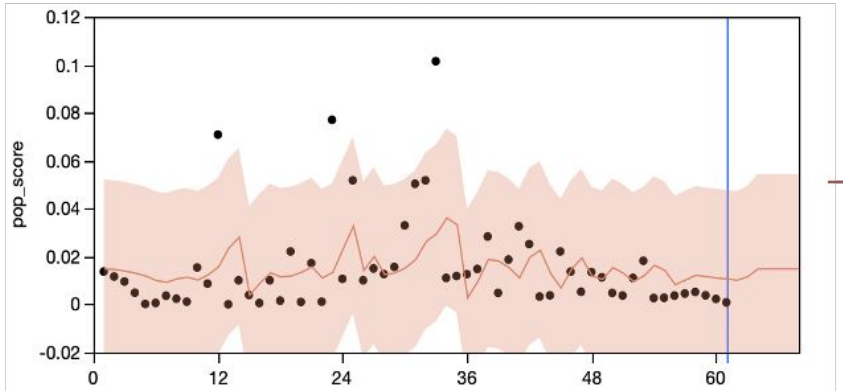


- Stationary Process
- ARMA(2, 1)

- Stable trend but with fluctuates in September and October from 2019.

- Focus on the technology innovation, especially topics like APPLE.

HEALTH



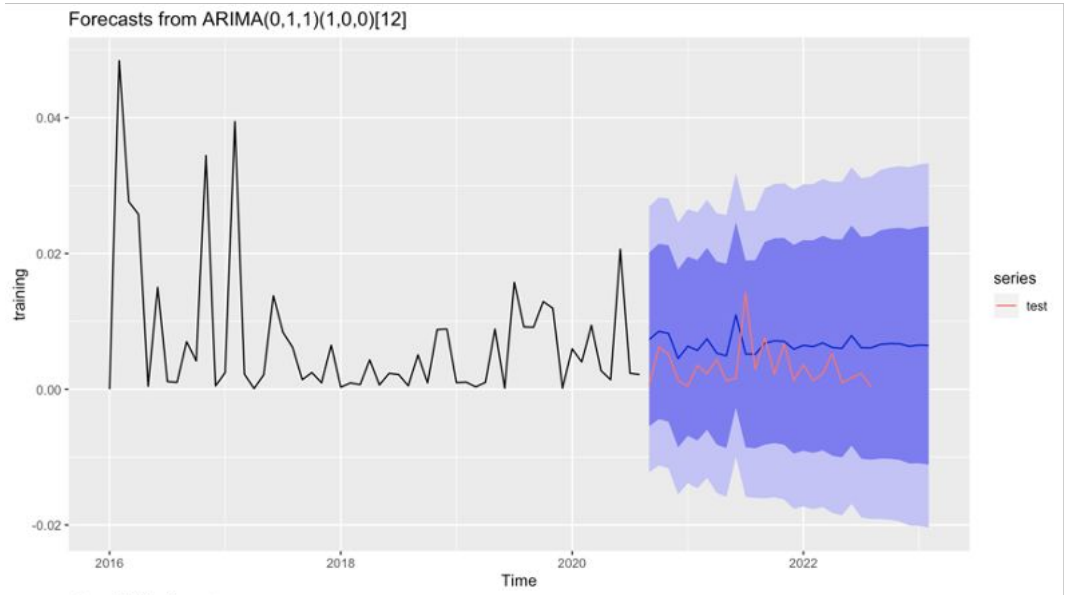
- Stationary Process
- MA(2)

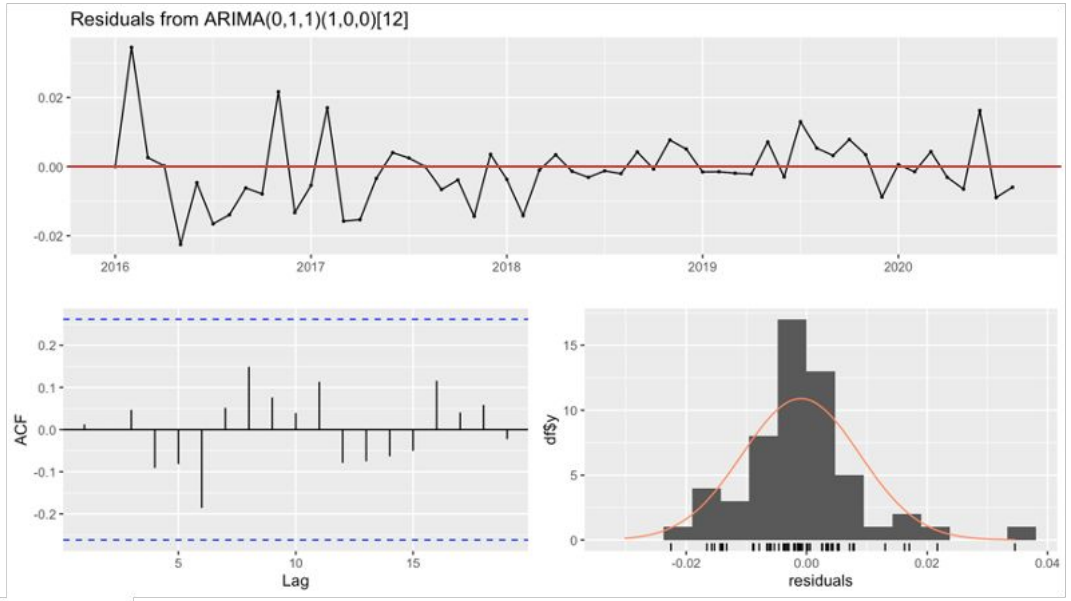
- Showing a stable trend without seasonal variation.

- Encourage innovation to attract more followers.

Time Series Analysis

SCIENCE



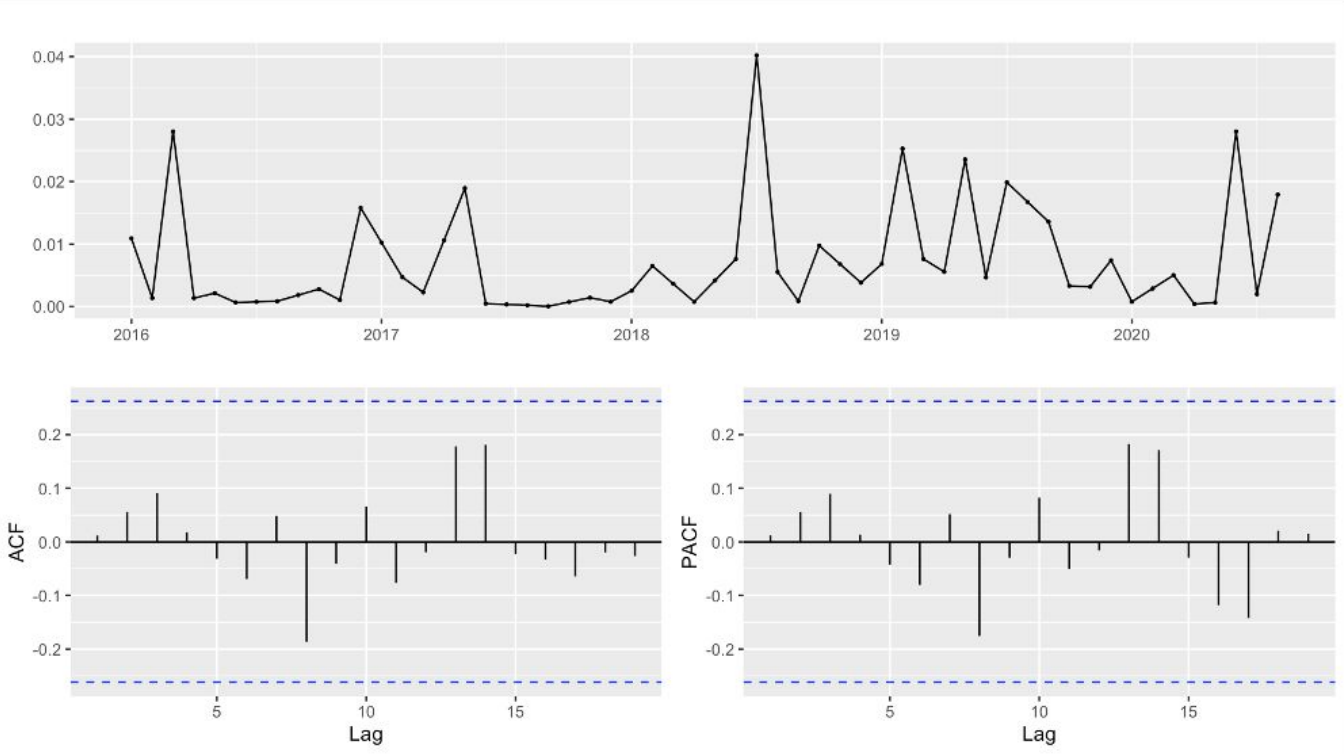



```
Coefficients:
          ma1     sar1
      -0.8757  0.3137
s.e.   0.0683  0.1695

sigma^2 = 9.978e-05:  log likelihood = 175.04
AIC=-344.08   AICc=-343.61   BIC=-338.06

Training set error measures:
                      ME        RMSE         MAE       MPE     MAPE      MASE       ACF1
Training set -0.0009385169 0.009717594 0.006951047 -764.5449 803.7078 0.8912433 0.01250267
```

```
        Ljung-Box test

data:  Residuals from ARIMA(0,1,1)(1,0,0)[12]
Q* = 6.4604, df = 9, p-value = 0.6931
```

Time Series Analysis

HUMANITIES AND SOCIAL




```
> fitAutoArima = auto.arima(training)
> fitAutoArima
Series: training
ARIMA(0,0,0) with non-zero mean

Coefficients:
        mean
      0.0072
s.e.  0.0012

sigma^2 = 7.667e-05:  log likelihood = 186.37
AIC=-368.74   AICc=-368.52   BIC=-364.69
```

Ljung-Box Test --- RANDOM

↓

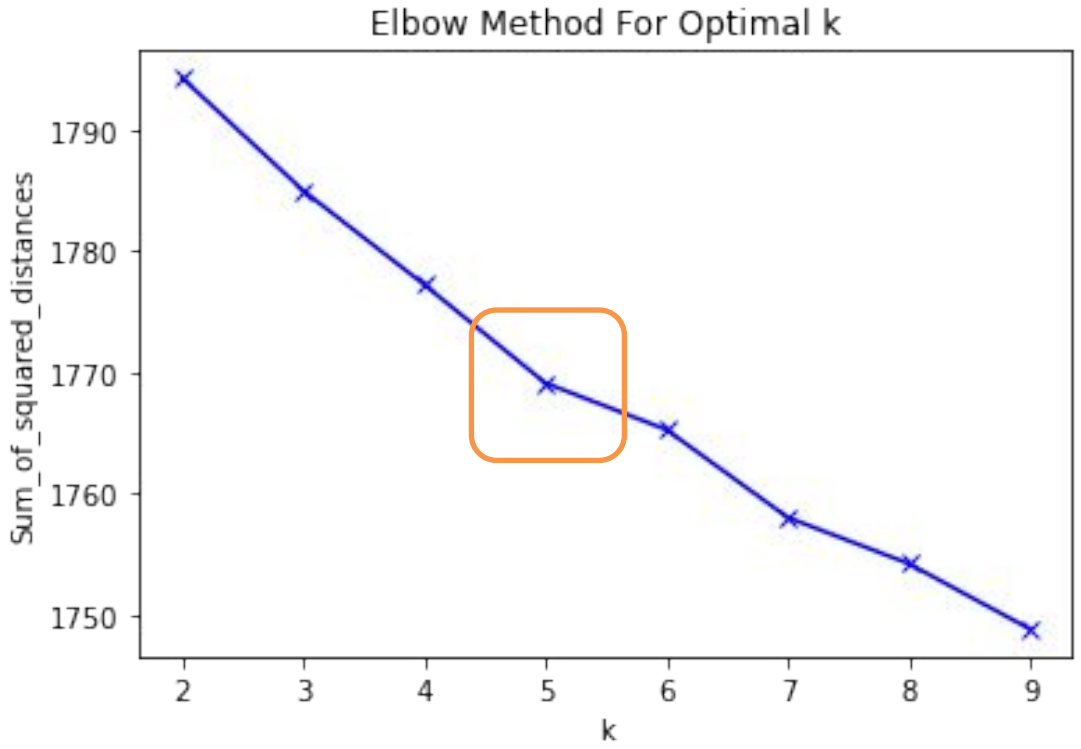
White Noise Series

Clustering

Clustering

Elbow Method

determining the number of clusters in a data set



Clustering

```
luster 0: game play playing wa like player gaming played look make
luster 1: song que music la love listening en se listen mix
luster 2: love loved watching like watch wa look make man great
luster 3: thank thanks learning learn wa help helpful work great like
luster 4: wa like people time ha thing make great know guy
```

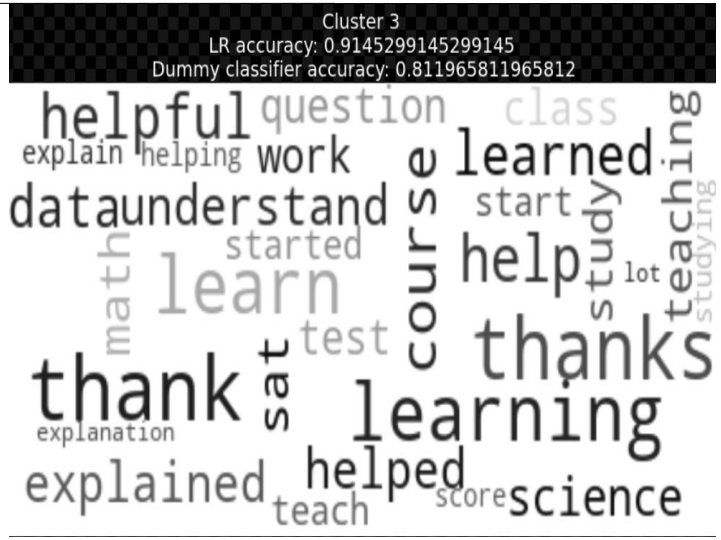
1. GAME
2. MUSIC
3. LEARNING
4. TECHNICAL
5. EATING

Clustering


Cluster 0
LR accuracy: 0.8888888888888888
Dummy classifier accuracy: 0.8739316239316239
tually
making
poi
new
team
played
console
hit
chess
game
looking
development
gaming
seri
better
dev
like
playing
magnus
devel
hardware
monitor
goal
excited


Cluster 1
LR accuracy: 0.9188034188034188
Dummy classifier accuracy: 0.8888888888888888
música
song
listen
que
son
listening
como
mais
esse
eu
music
com
amo
playlist
um
não
para
lofi
muito
tan


Cluster 2
LR accuracy: 0.8739316239316239
Dummy classifier accuracy: 0.8482905982905983
loving
watched
cooking
respect
asmr
feel
animal
food
eating
movie
eats
makeup
channel
man
watching
sub
look
absolutely
loved
talk
eat
love
good
sound
cook
cool
make
lovely
watch
beautiful


Cluster 3
LR accuracy: 0.9145299145299145
Dummy classifier accuracy: 0.811965811965812
helpful
question
class
explain
helping
work
learned
course
teaching
dataunderstand
start
study
started
help
learn
lot
math
test
thanks
thank
sat
learning
explanation
explained
helped
teach
score
science


Cluster 4
LR accuracy: 0.9294871794871795
Dummy classifier accuracy: 0.5769230769230769
trading
proud
live
pretty
guy
crypto
level
living
max
turn
phone
iphone
cubing
story
people
expert
country
cube
achievement
achieves
fact
apple
right
law
market
trade
google

Sentiment Analysis

Sentiment Analysis

- **Problem Statement**

The aim of the problem is to predict the sentiment of the comment

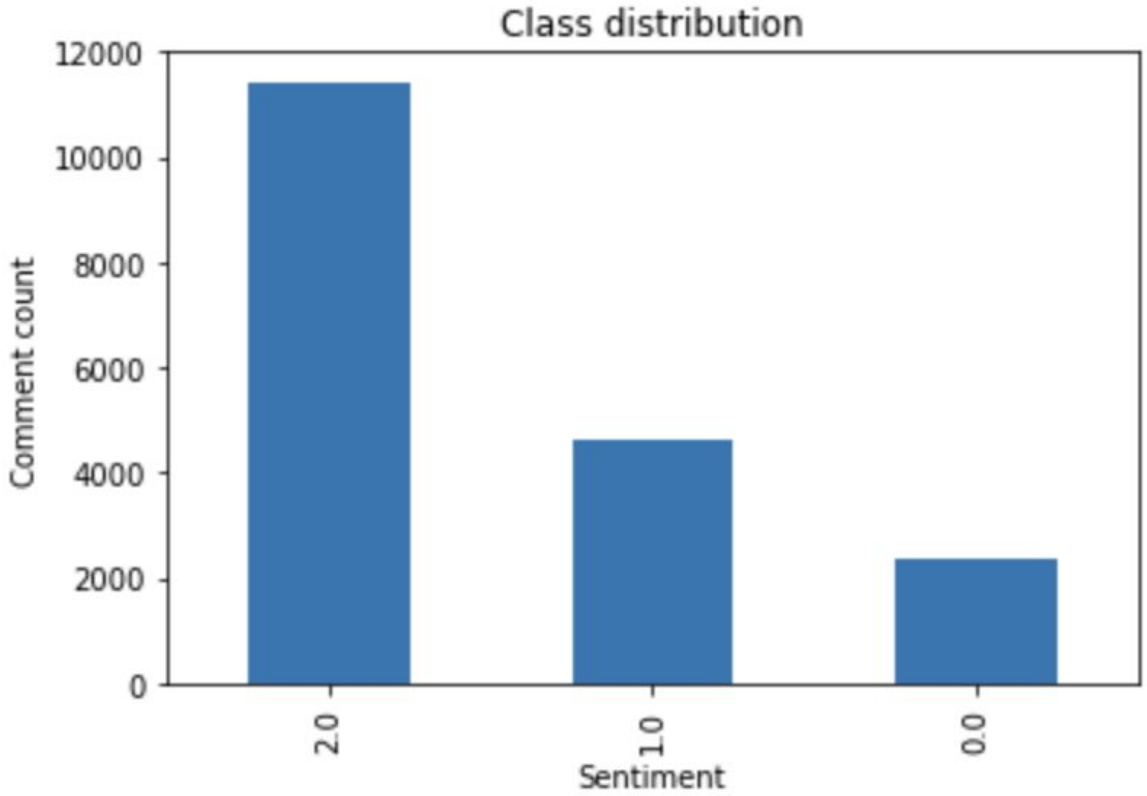
Positive - 2

Neutral - 1

Negative - 0

- **Imbalanced Class distribution**

The bar chart shows the class distribution of the comments in the data set



· **F1 score**

Due to the imbalanced distribution,we chose F1 score as a evaluation method

$$\frac{2(P*R)}{P+R}$$

Word Cloud

- Sentiment 2 / Positive Sentiment

This has words like ‘Thank’, ‘Love’, ‘Great’, ‘much’

- Sentiment 1 / Neutral Sentiment

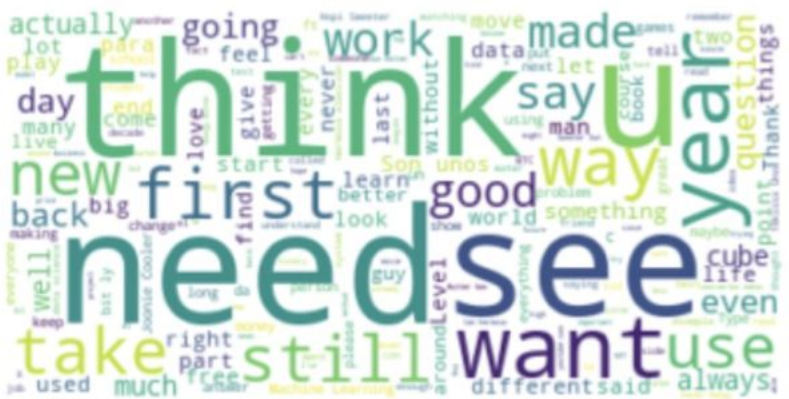
This has words like ‘see’, ‘think’

- Sentiment 0 / Negative Sentiment

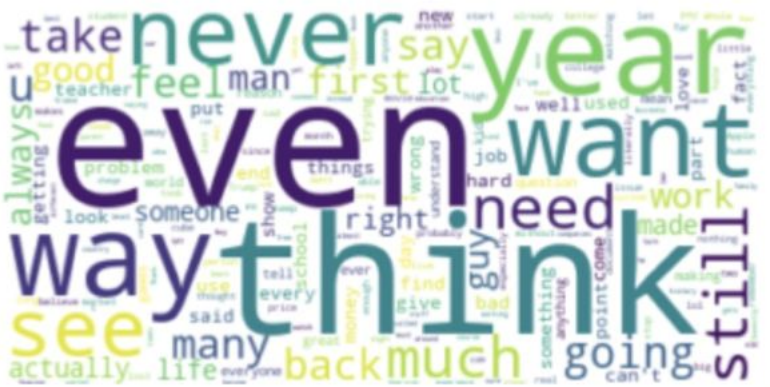
This has words like ‘never’, ’need’, ’even’

SENTIMENT: 2.0

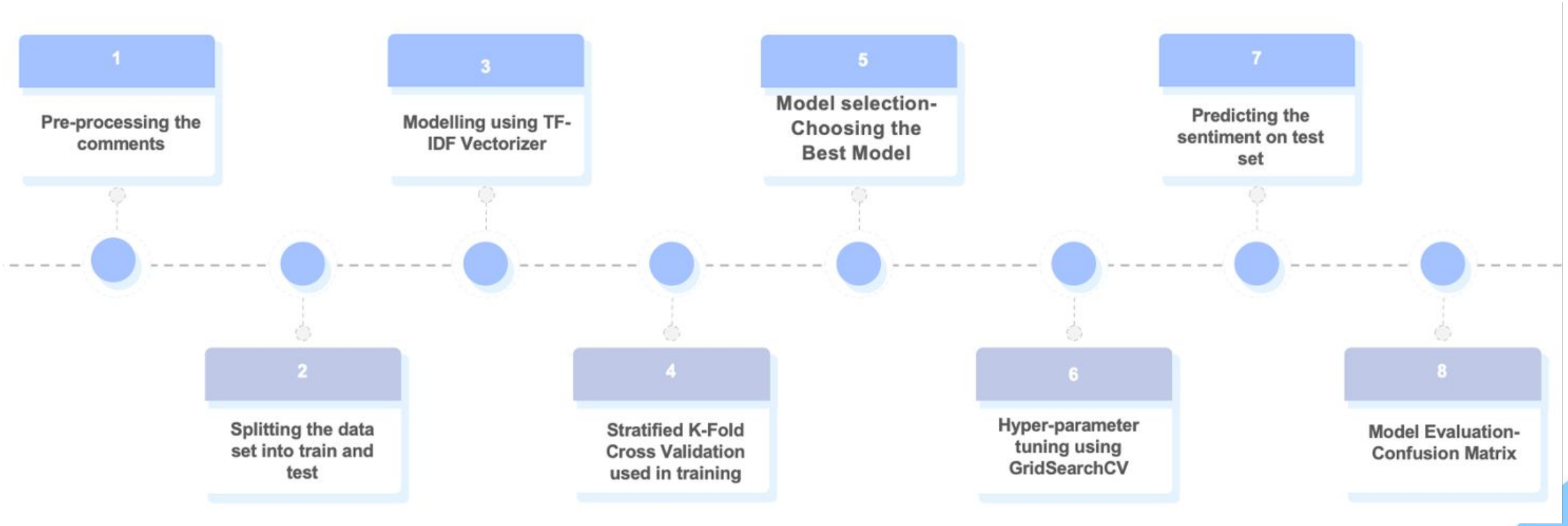
SENTIMENT: 1.0



SENTIMENT: 0.0



Steps followed for Predictive Analytics Sentiment Prediction



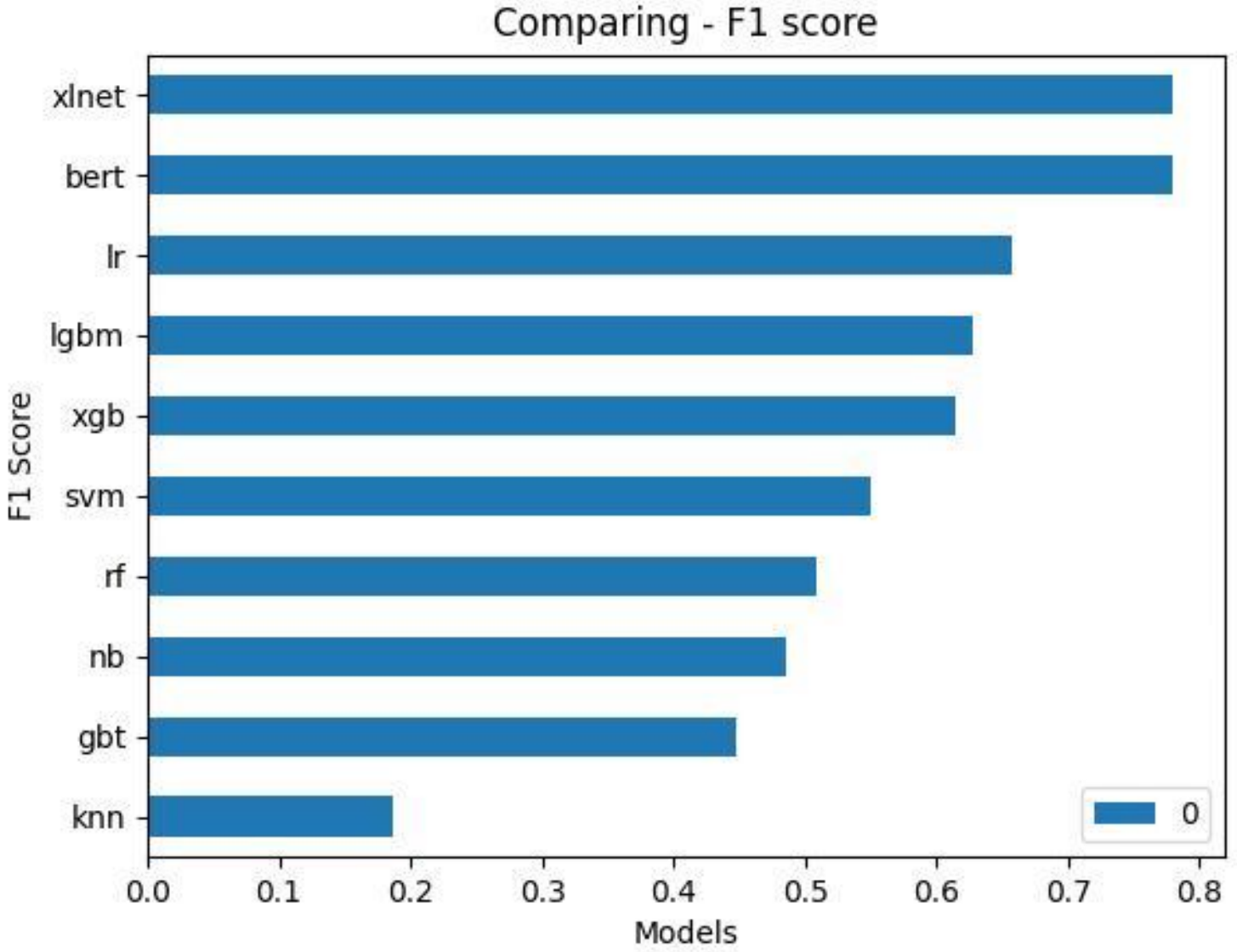
Sentiment Analysis

Machine Learning:

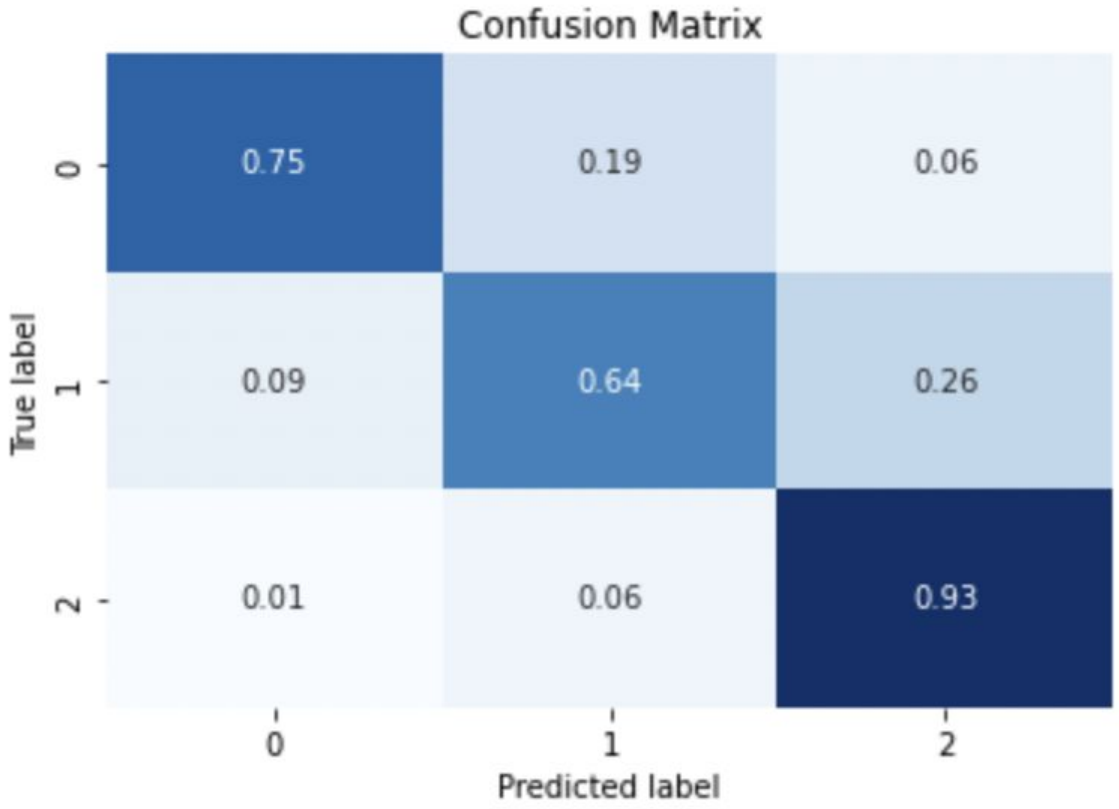
Best: 65%

DeepLearning :

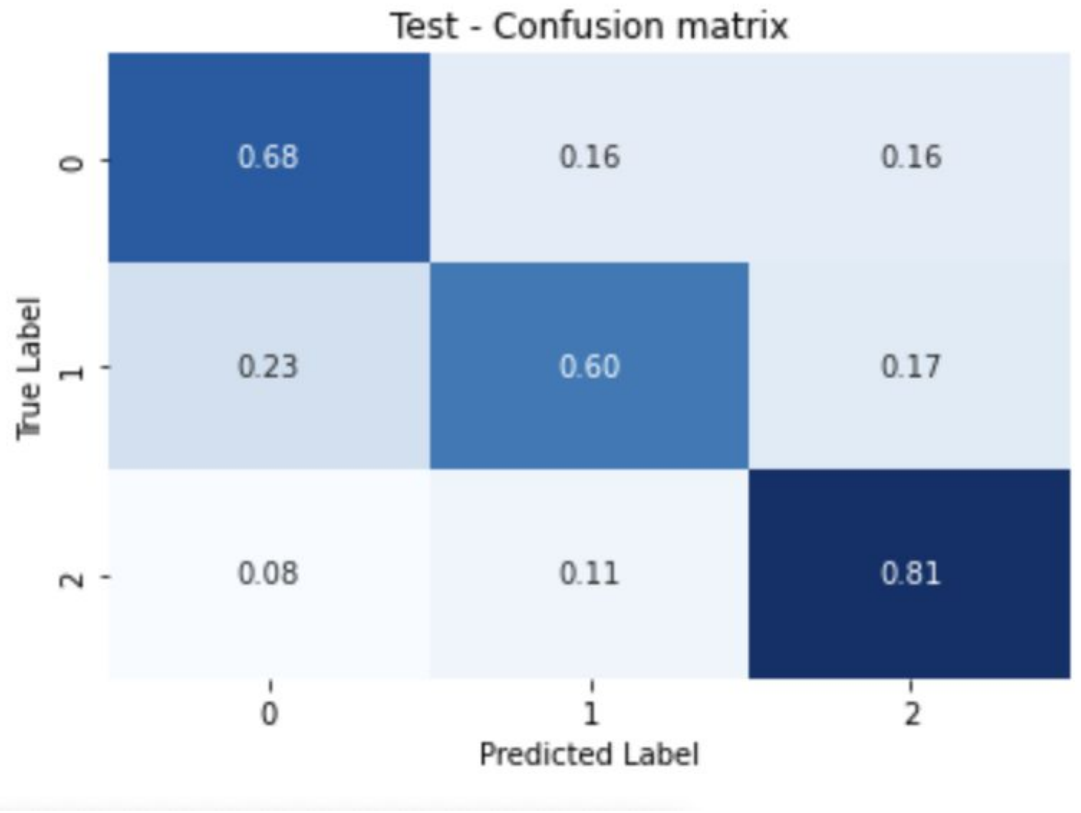
Best:78%



Sentiment Analysis



Deep Learning Model



Logistic Regression

Sentiment Analysis

Cost= Device_price * time_consuming

	BERT	LR
Device_price	GPU= $15/h	CPU= $0.15/h
Time_consuming	60s	32ms
Cost	**0.25**	**0.000001**
F1-score(Reward)	**0.78**	**0.67**

Sentiment Analysis

- The classification Report on Train and Test set using Logistic Regression model

- The model performs well both on training and testing data

```
Classification report on train set
              precision    recall  f1-score   support

         0.0       0.79      0.99      0.88      1753
         1.0       0.75      0.93      0.83      3479
         2.0       0.98      0.84      0.91      8574

    accuracy                           0.88     13806
   macro avg       0.84      0.92      0.87     13806
weighted avg       0.90      0.88      0.88     13806

Classification report on test set
              precision    recall  f1-score   support

         0.0       0.45      0.68      0.54       585
         1.0       0.62      0.60      0.61      1159
         2.0       0.89      0.81      0.84      2858

    accuracy                           0.74      4602
   macro avg       0.65      0.70      0.67      4602
weighted avg       0.76      0.74      0.75      4602

CPU times: user 18 s, sys: 3.73 s, total: 21.7 s
Wall time: 3.58 s
```

Limitation

1. Sentiment analysis have long tail class distribution

2. Some results of ARIMA have extremely low AIC or SBC

3. The iterative method of clustering makes results only locally optimal.

Business Insight

1. Business Suggestions
2. Business Value

Business Suggestion

Time Series Model

Focus on vloggers in Entertainment and Technology area.

- Help them catch the headlines
- Invest more on generating more creative and technical videos

Clustering Model

Differentiate viewers and catch up the attractive point they value on

- Games - Experience improvement
- Music - Relax
- Learning - Math & Science
- Tech - Financial Technique, Tech Market
- Eating - Kind; animal

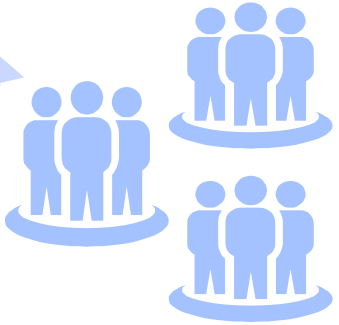
Business Value

Sentiment Analysis Model — LR Model

- Lower Cost
- Higher Accuracy
- Strong Explanatory

Large Scale of Comments

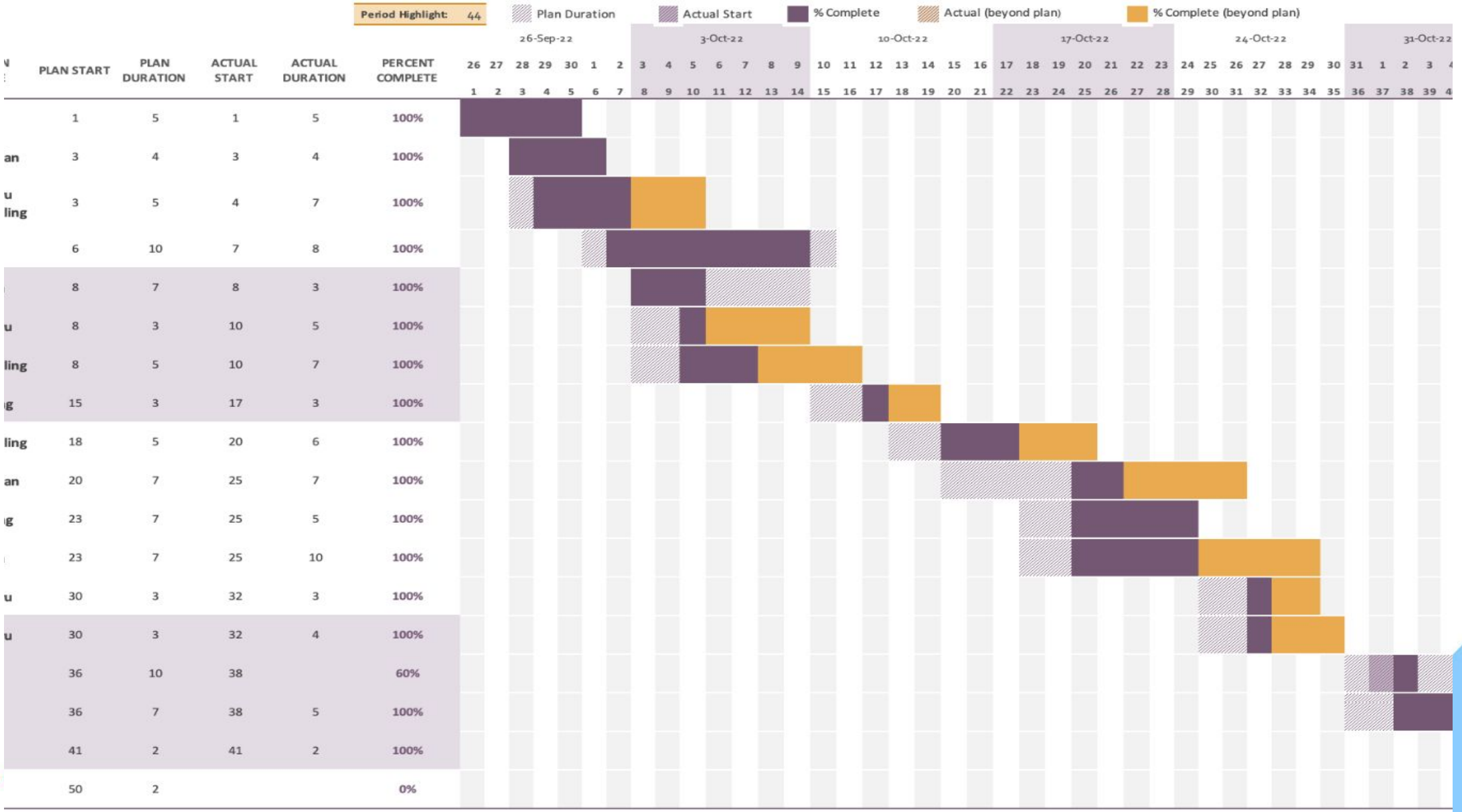
Differentiate the KOLs



Follow the trend

Catch up the hottest topic

Project Management



	PLAN START	PLAN DURATION	ACTUAL START	ACTUAL DURATION	PERCENT COMPLETE
	1	5	1	5	100%
an	3	4	3	4	100%
u ling	3	5	4	7	100%
	6	10	7	8	100%
	8	7	8	3	100%
u	8	3	10	5	100%
ling	8	5	10	7	100%
g	15	3	17	3	100%
ling	18	5	20	6	100%
an	20	7	25	7	100%
g	23	7	25	5	100%
	23	7	25	10	100%
u	30	3	32	3	100%
u	30	3	32	4	100%
	36	10	38		60%
	36	7	38	5	100%
	41	2	41	2	100%
	50	2			0%

Thank You