Group5

Leo Shr

NTHU ISS 110

110078509

Karen Chiu

NTHU ISS 110

110078506

Joanne Lee

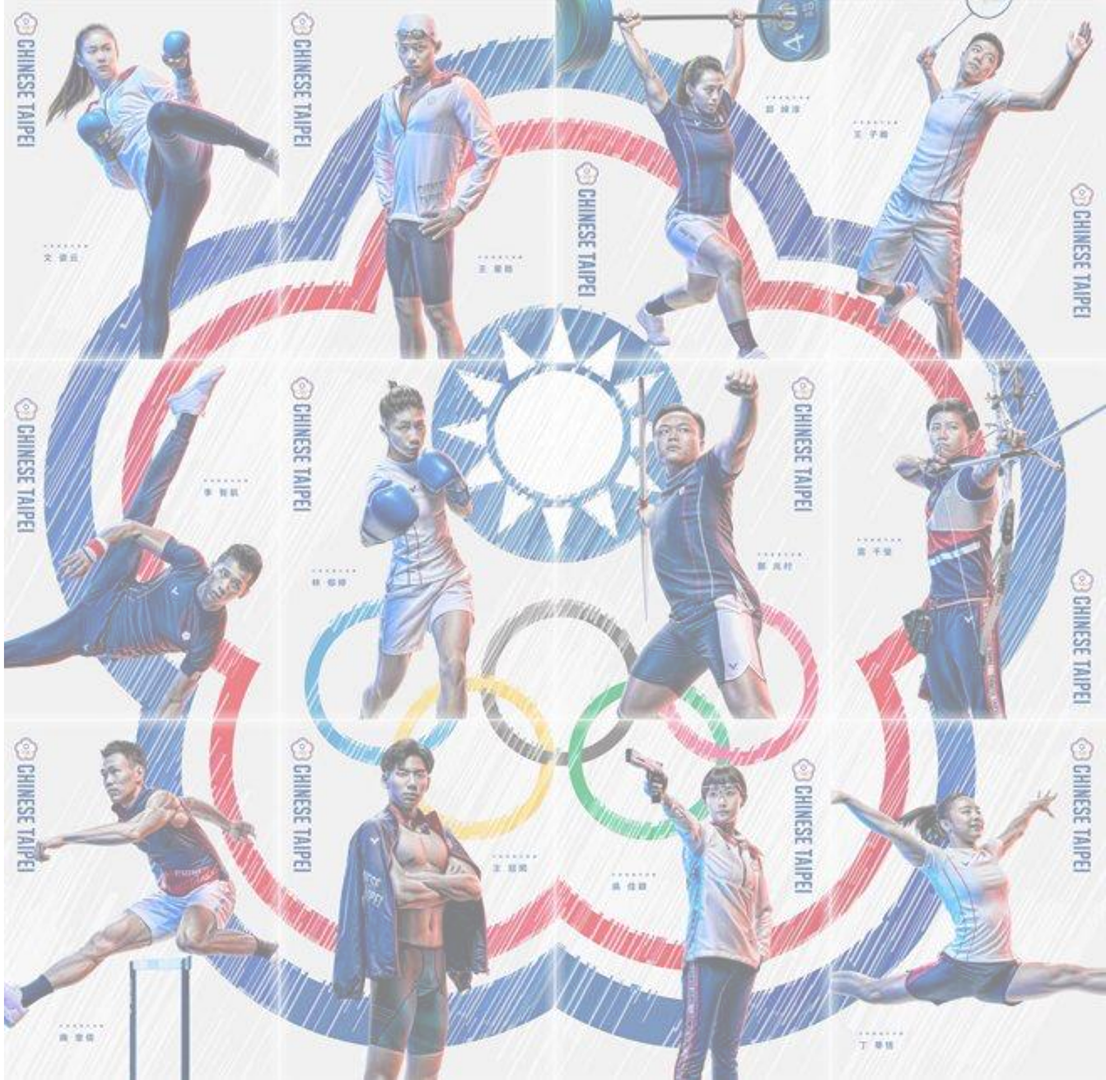
NTHU ISS 110

110078501

Background

In Tokyo Olympics, Chinese Taipei won a total of 12 Olympic medals, including 2 gold medals, 4 silver medals, and 6 bronze medals. That was the best record since the 1984 Los Angeles Summer Olympics.

Based on our mid-term project, we want to dig deeper into the documents of athletes.



Motivation

Are you curious how many people will reply to you after writing the article?


批踢踢實業坊 › 看板 Gossiping
作者 dzwei (Args&&... args)
標題 [問卦] 當年同意東奧正名看到郭婞淳奪金在想啥？
時間 Fri Jul 30 07:45:44 2021

[新聞] 東奧正名沒過 郭婞淳放心了
https://tinyurl.com/54ew7fvz

底下噓聲一片

中國台北一片

結果人家打破世界紀錄奪金

那些噓聲一片中國台北一片的人

也跟著The蹭郭婞淳台灣之光了嘛？

有沒有八卦？


批踢踢實業坊 › 看板 Gossiping
作者 s66449 (老實樣)
標題 [新聞] 李智凱戰勝心魔奪銀！成功拿出壓箱寶
時間 Sun Aug 1 19:04:05 2021

https://tinyurl.com/3bkaeehd

李智凱戰勝心魔奪銀！成功拿出壓箱寶　完美落地激動落淚

2021年08月01日 18:38

記者王真魚／日本東京報導

「鞍馬王子」李智凱在2016年里約奧運不慎落馬，儘管仍有心魔，但化悲憤為力量，2017年台北世大運奪金，2018年雅加達亞運再奪隊史首金；今年二度闖奧運，資格賽鞍馬第一晉級，1日決賽成功挑戰難度6.700，僅次於英國惠特洛克（Max Whitlock ），最終奪下銀牌，成為台灣體操第一人。獲獎那一刻，他如釋重負，激動落下淚水，這一次，他真的突破里約奧運心魔。


批踢踢實業坊 › 看板 Gossiping
作者 postar (郵星)
標題 [新聞] 中國玻璃心碎!林昀儒鄭怡靜奪冠國旗風波
時間 Sun Mar 27 21:30:40 2022

1.媒體來源:自由時報

2.記者署名:梁偉銘

3.完整新聞標題:
桌球》中國玻璃心碎滿地！林昀儒／鄭怡靜奪冠國旗風波

4.完整新聞內文:
https://i.imgur.com/dBB2UBb.jpg

Objective

We want to predict how many comments an article will have.

And further help those who want to be a topic of conversation on PTT.


批踢踢實業坊 › 看板 Gossiping

作者 dzwei (Args&&... args)
標題 [問卦] 當年同意東奧正名看到郭婞淳奪金在想啥？
時間 Fri Jul 30 07:45:44 2021

[新聞] 東奧正名沒過 郭婞淳放心了
https://tinyurl.com/54ew7fvz

底下噓聲一片

中國台北一片

結果人家打破世界紀錄奪金

那些噓聲一片中國台北一片的人

也跟著The蹭郭婞淳台灣之光了嘛？

有沒有八卦？


批踢踢實業坊 › 看板 Gossiping

作者 s66449 (老實樣)
標題 [新聞] 李智凱戰勝心魔奪銀！成功拿出壓箱寶
時間 Sun Aug 1 19:04:05 2021

https://tinyurl.com/3bkaeehd

李智凱戰勝心魔奪銀！成功拿出壓箱寶　完美落地激動落淚

2021年08月01日 18:38

記者王真魚／日本東京報導

「鞍馬王子」李智凱在2016年里約奧運不慎落馬，儘管仍有心魔，但化悲憤為力量，2017年台北世大運奪金，2018年雅加達亞運再奪隊史首金；今年二度闖奧運，資格賽鞍馬第一晉級，1日決賽成功挑戰難度6.700，僅次於英國惠特洛克（Max Whitlock ），最終奪下銀牌，成為台灣體操第一人。獲獎那一刻，他如釋重負，激動落下淚水，這一次，他真的突破里約奧運心魔。


批踢踢實業坊 › 看板 Gossiping

作者 postar (郵星)
標題 [新聞] 中國玻璃心碎!林昀儒鄭怡靜奪冠國旗風波
時間 Sun Mar 27 21:30:40 2022

1.媒體來源:自由時報

2.記者署名:梁偉銘

3.完整新聞標題:
桌球》中國玻璃心碎滿地！林昀儒／鄭怡靜奪冠國旗風波

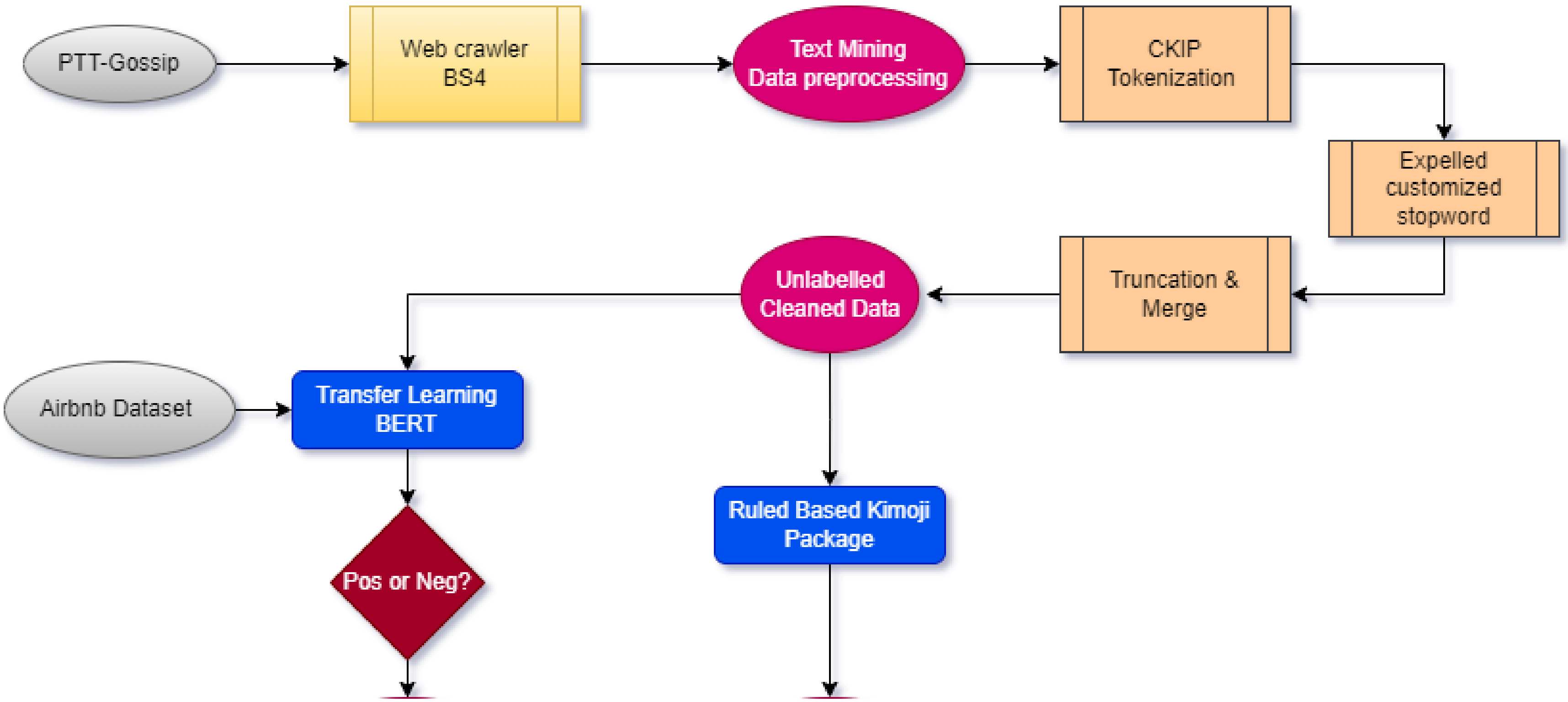
4.完整新聞內文:
https://i.imgur.com/dBB2UBb.jpg

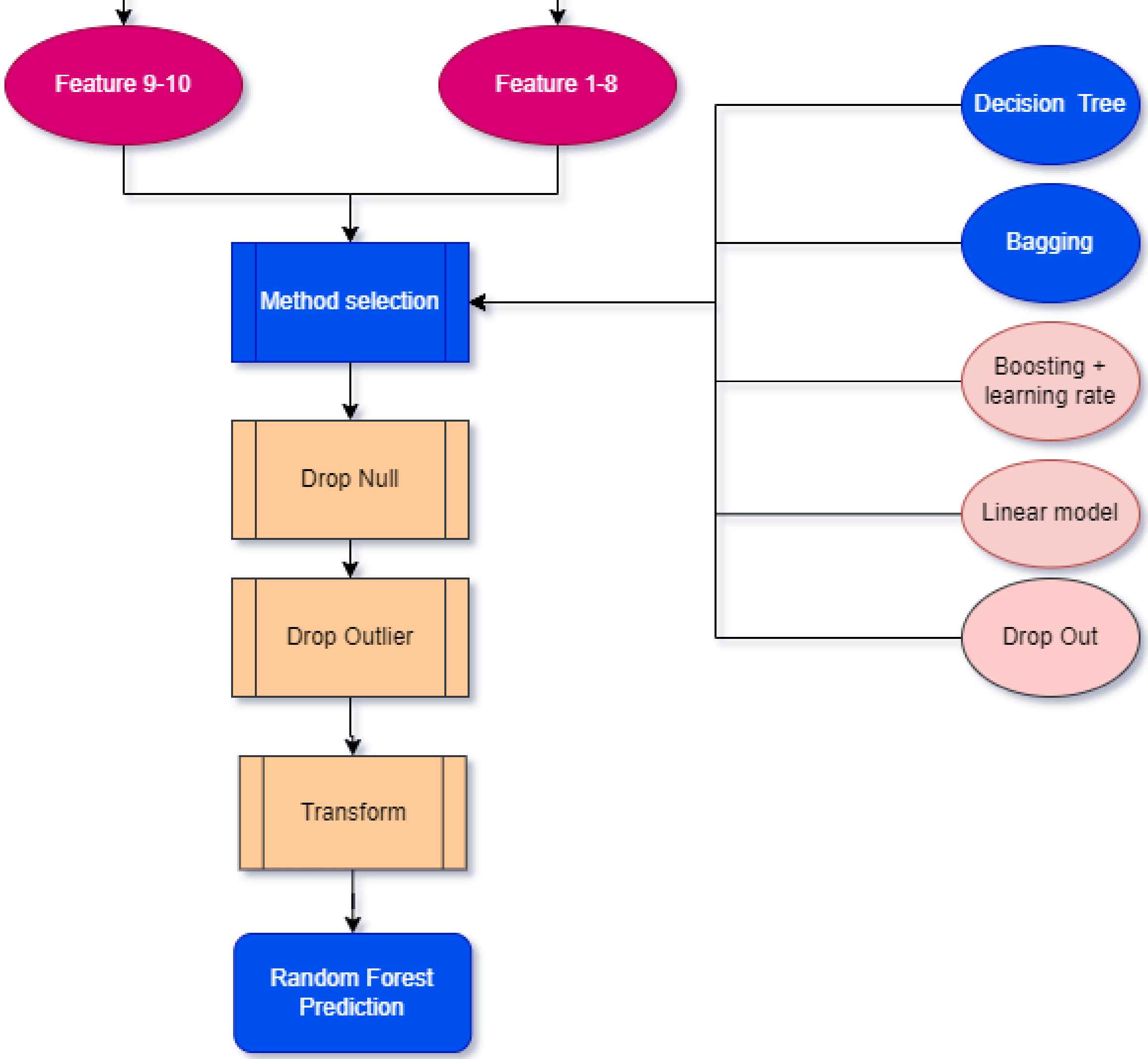
Topic

Using BERT and Ruled Based Sentiment Analysis to Predict PTT Public Opinion Responding Level

透過文章內文進行多標籤極性分析以預測輿論影響程度

Flow Chart


PTT-Gossip
Web crawler BS4
Text Mining Data preprocessing
CKIP Tokenization
Expelled customized stopword
Truncation & Merge
Unlabelled Cleaned Data
Airbnb Dataset
Transfer Learning BERT
Pos or Neg?
Ruled Based Kimoji Package



Feature 9-10
Feature 1-8
Method selection
Drop Null
Drop Outlier
Transform
Random Forest Prediction
Decision Tree
Bagging
Boosting + learning rate
Linear model
Drop Out

Data Source

PTT Gossiping

Name of athlete

Tai Tzu-Ying (戴資穎)

Lin Yun-Ju (林昀儒)

Kuo Hsing-Chun (郭婞淳)

Lee Chih-Kai (李智凱)

Yang Yung-wei (楊勇緯)

Data Structure

Athor / Title / Time

Content of article

Numer of Comments

Technique

```python
# 抓指定網址的文章內容和留言
def find_document(address, search):
  for i in range(start,end,+1):

    # 組成 正確 URL
    link = address+str(i)+search

    # 執行單頁面網頁爬蟲
    get_one_page(link)
    # 避免被太快被 PTT 封鎖請求
    time.sleep(1)

    article_href = find_link(link)

    for pcontent in range(len(article_href)):
      payload = {'form':article_href[pcontent],
            'yes':'yes'
            }

      rs = requests.session()

      res = rs.post('https://www.ptt.cc/ask/over18',verify=False,data=payload)

      res = rs.get('https://www.ptt.cc'+article_href[pcontent],verify=False,data=payload)

      soup = bs4.BeautifulSoup(res.text, "lxml")

      results = soup.select('span.article-meta-value')
```

Web scrapping

BeautifulSoup

- store to .csv files

Technique

Tokenization CKIP

Define our stop words

```python
## 自建停用字
my_stop_words = ['：','。','。','、','※','【','】','，','，','，','※', '」']
term_stopwords = ['什麼','可以','我們','挑選','篇文章', '備註', '是', '新聞網','或', '的', '某' , '分', '等', '才能','可以','我們','挑選','篇文章','你',
        '知道','任何','如果','而且','不會','。',
        '從來','外有','不論','記量','從來','只是','還是','这是','以便',
        '放在','問個','同時','需登','有些','不用','一定','什麼',''
        ,'運動給'
        ,'結果'
        ,'行為'
        ,'另一台'
        ,'这次','你','我','他','上次','']
special_sign = ['◆','■','▎','▌','▲','◤','▼','▲','▲','◣','▏','◢', '～', '○', '●', '▼', '▲', '->', '：',
                '.','～～','！','？', '.', '/', '\ ']
full_stopwords = list(STOPWORDS) + term_stopwords + special_sign + my_stop_words + content_sup
punctuation_str = punctuation
```

Technique - Data Cleaning

```
def data_cleaning(str_list):
  string_content = str_list
  article_1 = [article.replace('\n', '') for article in string_content]
  tokens_list = [ token_pack(article) for article in article_1]
  clean_sentences = []
  for article in tokens_list:
    clean_tokens = []
    for _ in article:
      for token in _:
        # 去除token所有半形空白
        token = token.replace(' ', '')
        # 前後去除空白 #去除中英夾雜 #去除純數字
        if token.strip() != "" and not token.encode().isalpha()  and token.isdigit() == False:
          # 去除自建停用字
          if not token in full_stopwords:
            # 去除英文半形標點符號  #去除中文之標點符號
            if  not token in punctuation and not token in punctuation_str:
              clean_tokens.append(token)
      if len(clean_tokens)>0:
        clean_sentences.append(clean_tokens)
  return clean_sentences
```

Technique - Ruled Based Sentiment Analysis Keymoji Package

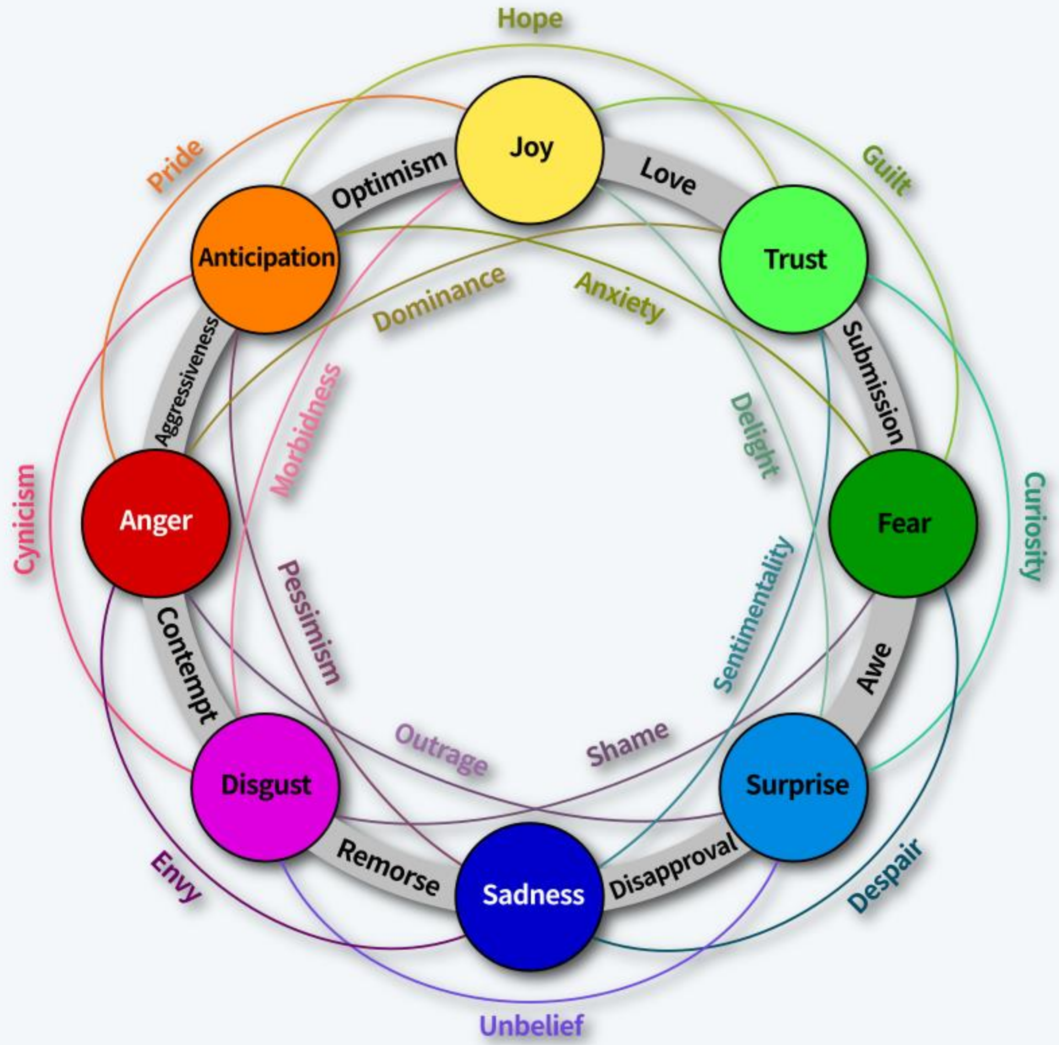
```
# from KeyMojiAPI import KeyMoji
# # 若您是使用 Docker 版本，無須填入 username, keymoji_key 參數
keymoji = KeyMoji(username="", keymojiKey="")
temp = TAI_title_df[0:100]
KeyMoji_score = []
for i in range(0, 100):
    KeyMoji_score.append(keymoji.sense8(temp[0][i]))

KeyMoji_score
```

```
[{'msg': 'Success!',
  'results': [{'Anger': 0.1526,
    'Anticipation': 6.9466,
    'Disgust': 0.0951,
    'Fear': 0.162,
    'Joy': 7.8578,
    'Sadness': 0.969,
    'Surprise': 6.892,
    'Trust': 7.1368,
    'input_str': '幻象伴飛戴資穎麟洋配蔡壁如發文誤新聞雲呂晏慈台北報導幻象伴飛戴資穎麟洋配蔡壁如發
  'sense': 'sense8',
  'status': True,
  'version': 'v102'}]
```

Keymoji package

- 8 Sentiment Score
  - Anger
  - Anticipation
  - Disgust
  - Fear
  - Joy
  - Sadness
  - Surprise
  - Trust



Hope
Joy
Optimism
Love
Pride
Guilt
Anticipation
Trust
Dominance
Anxiety
Aggressiveness
Submission
Morbidness
Delight
Cynicism
Anger
Fear
Curiosity
Pessimism
Sentimentality
Contempt
Awe
Outrage
Shame
Disgust
Surprise
Remorse
Sadness
Disapproval
Envy
Despair
Unbelief

Transfer learning BERT

Technique - Transfer learning BERT

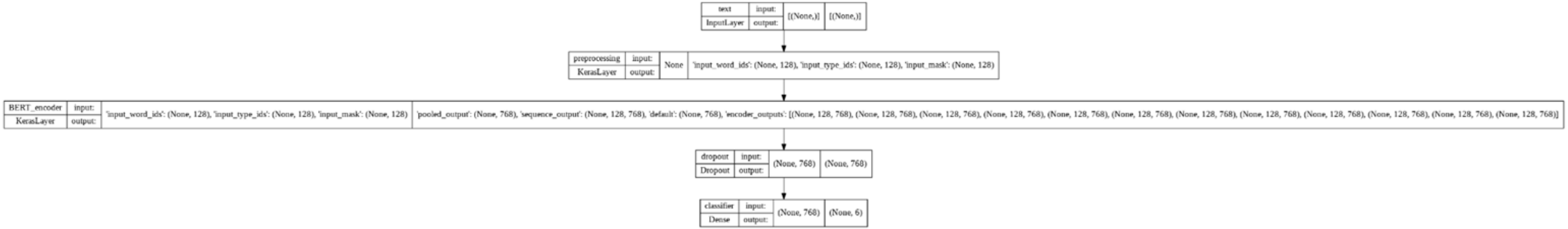
Problem: PTT reviews have no label required for classification

- Transfer Learning:

We use Airbnb reviews data (Mandarin data with labelled) to fit the BERT Pretrained model

- Classification task

Then, we applied fitted BERT model to the PTT Gossiping data to do the classification task




```
Epoch 10/10
100/100 [==============================] - 97s 975ms/step - loss: 0.0015 - sparse_categorical_accuracy: 0.9997 - val_loss: 0.8028 - val_sparse_categorical_accuracy: 0.9000
```

[Code Link]

Output Demo:

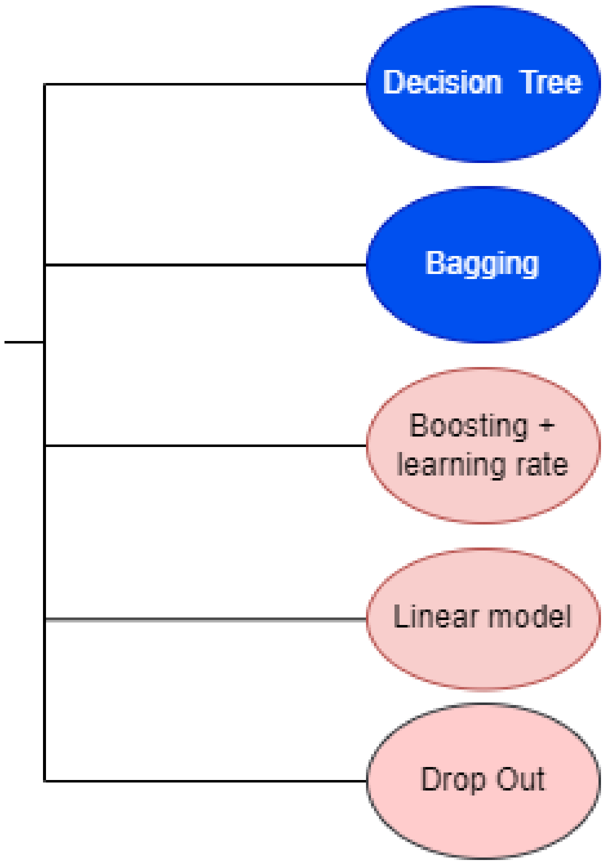
category	title	content	limited_Bert	mix_predict	title_predict
新聞	李智凱秀妻子送他歲生日豪禮要當爸	三立李鴻典台北報導恭喜李智凱秀妻子送他歲生日豪禮要當爸爸了完整新聞內文恭喜再恭喜歲體操國手鞍...	李智凱秀妻子送他歲生日豪禮要當爸:三立李鴻典台北報導恭喜李智凱秀妻子送他歲生日豪禮要當爸爸了...	pos	pos
新聞	台灣首位世界第「鞍馬王子」李智凱	林岳甫台灣首位世界第「鞍馬王子」李智凱我會繼續努力完整新聞內文在東京奧運勇奪鞍馬銀牌，為台灣...	台灣首位世界第「鞍馬王子」李智凱:林岳甫台灣首位世界第「鞍馬王子」李智凱我會繼續努力完整新聞...	pos	pos
新聞	李智凱登鞍馬世界第恩師林育信報喜	游郁香李智凱登鞍馬世界第恩師林育信報喜希望鼓勵小選手追夢完整新聞內文「鞍馬王子」李智凱以招牌...	李智凱登鞍馬世界第恩師林育信報喜:游郁香李智凱登鞍馬世界第恩師林育信報喜希望鼓勵小選手追夢完...	pos	pos
問卦	台灣李智凱奧運體操為什麼能贏歐美中國	歐美中國算是體操大國，歐美中國也投資很多資源在體操為什麼在台灣培訓，被台灣教練訓練李智凱能夠...	台灣李智凱奧運體操為什麼能贏歐美中國:歐美中國算是體操大國，歐美中國也投資很多資源在體操為什...	pos	pos
問卦	李智凱敢挑戰街健嗎	街健肌肉應用最高形式完美控制身體藝術相比之下體操簡直是平淡創意卻企圖以華麗表演來掩飾李智凱雖...	李智凱敢挑戰街健嗎:街健肌肉應用最高形式完美控制身體藝術相比之下體操簡直是平淡創意卻企圖以華...	pos	pos
問卦	李智凱沒拍過武打片?	李智凱好像是上屆奧運就很有名選手他顏值和身手尤其是掃堂腿應該很適合拍武打功夫片吧怎麼沒成為男主?	李智凱沒拍過武打片?:李智凱好像是上屆奧運就很有名選手他顏值和身手尤其是掃堂腿應該很適合拍武...	pos	neg

In order to avoid the problem of a fake title,

1. Concated the **title** and **content** together
   (Truncated the string under 123 + 3 (CLS, SEP, UNK) to form the second prediction of classification prediction.)
2. Also, it improved the following rule-based algorithm result.

It can alleviate the fear of judging a book by its cover.

Method Selection & Tuning Skill



Technique - Predictive Model Selection

1. **Improve linearity:**

- Log Transform the continuous variables

1. **Comparison the RMSEs:**

- K-fold Algorithm (LOOCV)

- Tree-based Model v.s. Linear Model?

K-fold Validation (k = n)	In-sample RMSE	Out-sample RMSE
Simple OLS Model	0.8172109	0.8520548
Simple Decision Tree	0.7562913	0.8667383

Very Closed

Technique - Method Selection

- Bagging

K-fold Validation (k = n)	In-sample RMSE	Out-sample RMSE
Bagged OLS Regression	0.8172109	0.8520548
Bagged Decision Tree	0.7495758	0.7558893

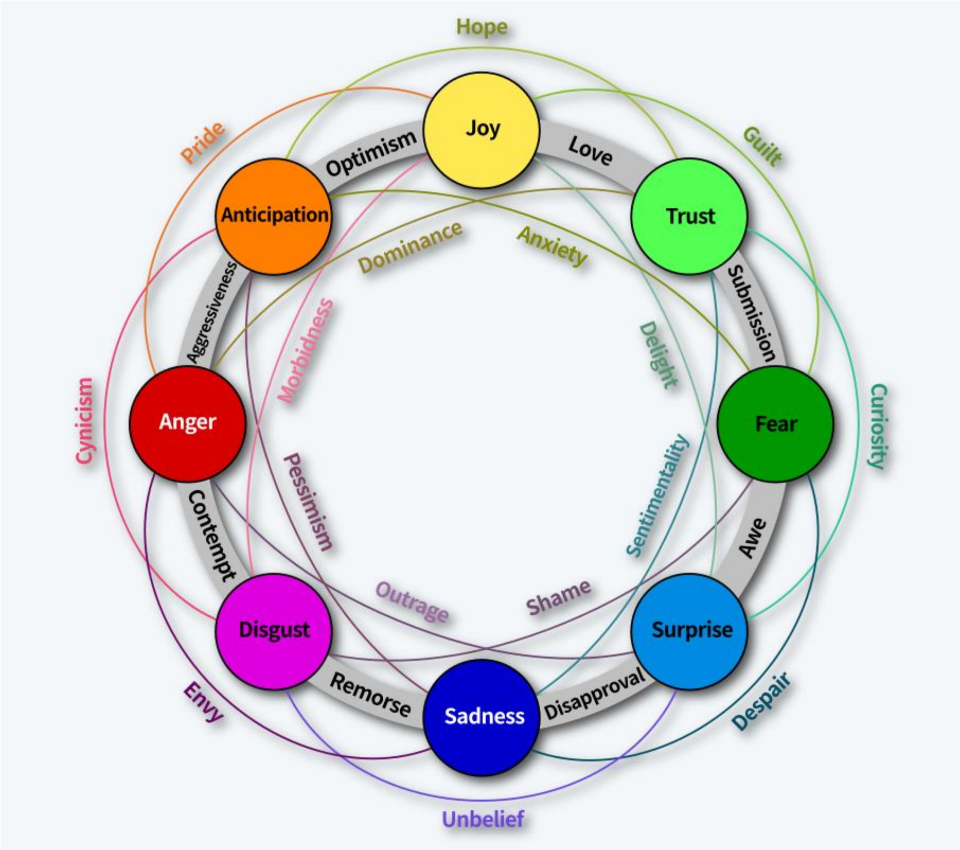
- Boosted

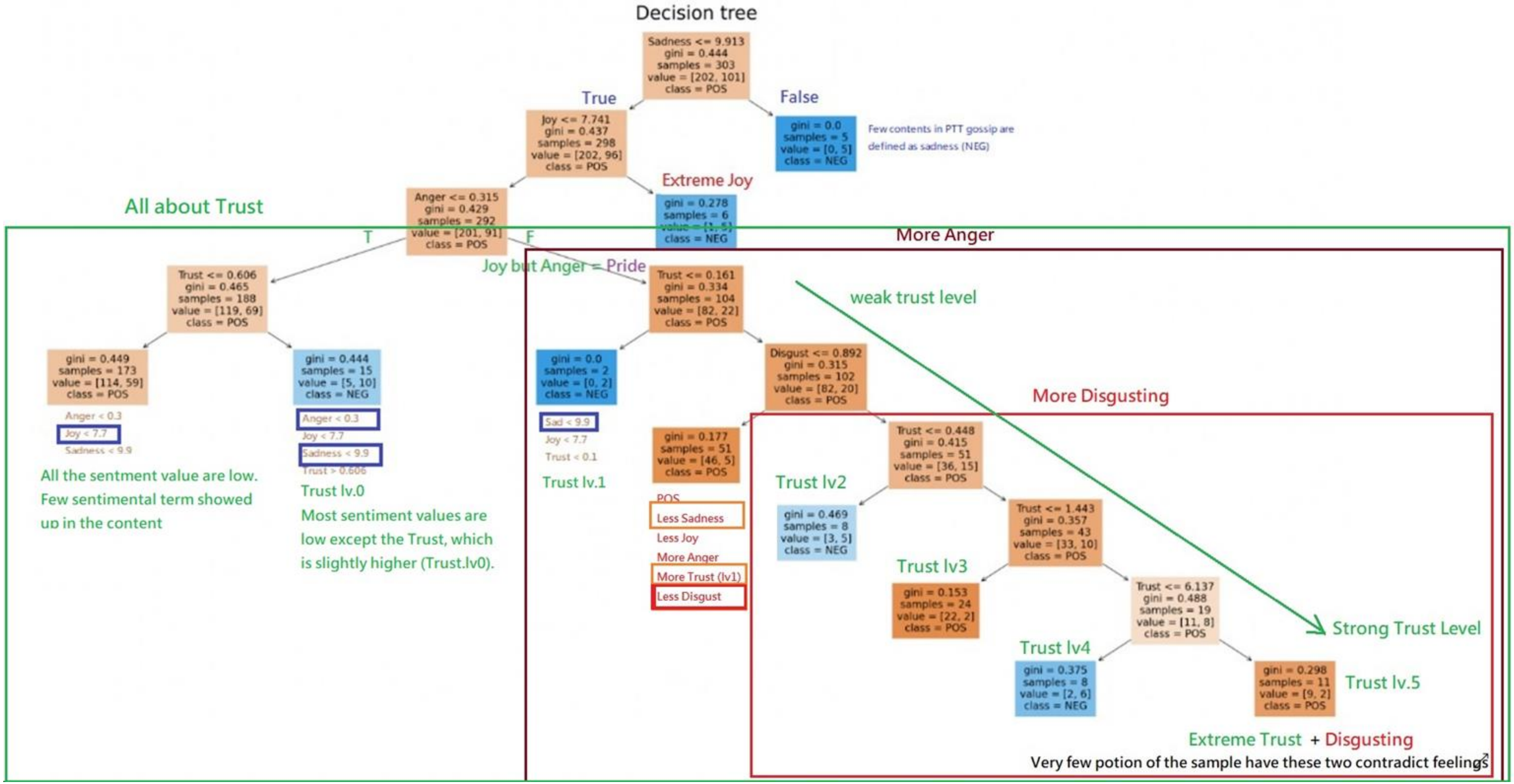
K-fold Validation (k = n)	In-sample RMSE	Out-sample RMSE
Adaboost OLS Regression	0.844557	0.8646697
Adaboost Decision Tree	0.854412	0.8779588

Appearently, “Bagged Decision Tree” won!

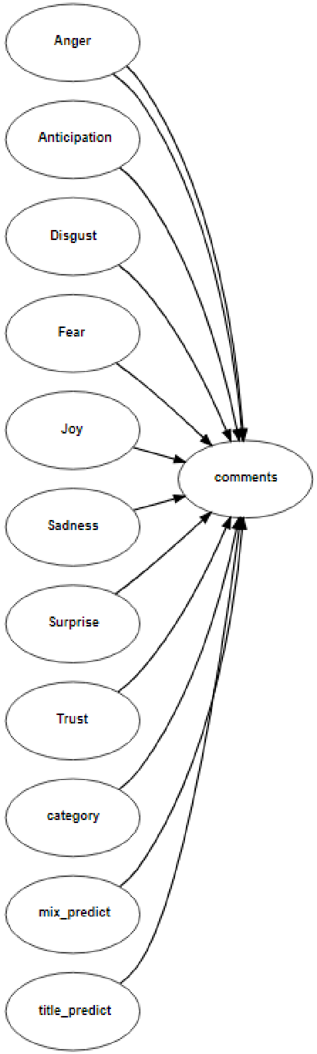
But it’s too stable. Therefore, we applied RandomForest (Bagged Decision Tree + Drop out)

Decision Tree + Kimoji




Decision tree
Sadness <= 9.913
gini = 0.444
samples = 303
value = [202, 101]
class = POS
True
False
Joy <= 7.741
gini = 0.437
samples = 298
value = [202, 96]
class = POS
gini = 0.0
samples = 5
value = [0, 5]
class = NEG
Few contents in PTT gossip are defined as sadness (NEG)
Extreme Joy
gini = 0.278
samples = 6
value = [1, 5]
class = NEG
Anger <= 0.315
gini = 0.429
samples = 292
value = [201, 91]
class = POS
All about Trust
T
F
More Anger
Joy but Anger = Pride
Trust <= 0.606
gini = 0.465
samples = 188
value = [119, 69]
class = POS
Trust <= 0.161
gini = 0.334
samples = 104
value = [82, 22]
class = POS
weak trust level
gini = 0.449
samples = 173
value = [114, 59]
class = POS
gini = 0.444
samples = 15
value = [5, 10]
class = NEG
gini = 0.0
samples = 2
value = [0, 2]
class = NEG
Disgust <= 0.892
gini = 0.315
samples = 102
value = [82, 20]
class = POS
More Disgusting
Anger < 0.3
Joy < 7.7
Sadness < 9.9
Anger < 0.3
Joy < 7.7
Sadness < 9.9
Trust > 0.606
Sad < 9.9
Joy < 7.7
Trust < 0.1
gini = 0.177
samples = 51
value = [46, 5]
class = POS
Trust <= 0.448
gini = 0.415
samples = 51
value = [36, 15]
class = POS
All the sentment value are low.
Few sentimental term showed up in the content
Trust lv.0
Most sentiment values are low except the Trust, which is slightly higher (Trust.lv0).
Trust lv.1
Trust lv2
POS
Less Sadness
Less Joy
More Anger
More Trust (lv1)
Less Disgust
gini = 0.469
samples = 8
value = [3, 5]
class = NEG
Trust <= 1.443
gini = 0.357
samples = 43
value = [33, 10]
class = POS
Trust lv3
gini = 0.153
samples = 24
value = [22, 2]
class = POS
Trust <= 6.137
gini = 0.488
samples = 19
value = [11, 8]
class = POS
Strong Trust Level
Trust lv4
gini = 0.375
samples = 8
value = [2, 6]
class = NEG
gini = 0.298
samples = 11
value = [9, 2]
class = POS
Trust lv.5
Extreme Trust + Disgusting
Very few potion of the sample have these two contradict feelings

Random Forest



Random Forest - Original Dataset (n = 430)

	category	mix_predict	title_predict	Trust	Joy	Surprise	Anticipation	Fear	Anger	Disgust	Sadness	comments
0	新聞	pos	pos	5.2276	6.0957	5.8269	5.4967	0.4347	0.7251	0.4470	1.3124	17
1	新聞	pos	pos	6.7392	7.2615	6.1154	7.0486	0.3577	0.4990	0.5004	0.8641	11
2	新聞	pos	pos	6.5321	7.2815	7.0499	7.6680	0.7563	0.8893	1.0379	1.3384	264
3	問卦	pos	pos	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	13
4	問卦	pos	pos	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	37
...	...	...	...	...	...	...	...	...	...	...	...	...
425	新聞	neg	neg	2.0367	1.4839	2.2353	0.7314	8.0992	8.4973	8.4190	8.5586	101
426	新聞	pos	neg	0.7393	0.9027	0.8875	0.9404	1.1693	1.2011	0.9429	2.3210	285
427	問卦	neg	pos	0.6784	1.7261	2.2559	0.2382	6.6203	6.6478	6.1981	6.9905	77
428	新聞	neg	neg	0.5312	0.1788	0.5290	0.2730	0.7389	1.0391	0.8457	1.0394	212
429	問卦	neg	neg	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	10

430 rows × 12 columns

Moverover, we take categories (新聞 / 問卦...) into account.

Random Forest - Remove Keymoji = 0

	category	mix_predict	title_predict	Trust	Joy	Surprise	Anticipation	Fear	Anger	Disgust	Sadness	comments
0	新聞	pos	pos	5.2276	6.0957	5.8269	5.4967	0.4347	0.7251	0.4470	1.3124	17
1	新聞	pos	pos	6.7392	7.2615	6.1154	7.0486	0.3577	0.4990	0.5004	0.8641	11
2	新聞	pos	pos	6.5321	7.2815	7.0499	7.6680	0.7563	0.8893	1.0379	1.3384	264
5	問卦	pos	neg	0.7988	0.7348	0.1838	0.0977	5.7180	5.2093	5.5036	5.8055	1
6	問卦	neg	neg	1.9442	0.7051	1.3637	0.4363	7.9398	7.6130	8.1667	7.5377	11
...	...	...	...	...	...	...	...	...	...	...	...	...
423	問卦	neg	neg	1.4795	0.3989	0.9883	0.4617	7.2876	6.5310	6.8515	7.3672	1
425	新聞	neg	neg	2.0367	1.4839	2.2353	0.7314	8.0992	8.4973	8.4190	8.5586	101
426	新聞	pos	neg	0.7393	0.9027	0.8875	0.9404	1.1693	1.2011	0.9429	2.3210	285
427	問卦	neg	pos	0.6784	1.7261	2.2559	0.2382	6.6203	6.6478	6.1981	6.9905	77
428	新聞	neg	neg	0.5312	0.1788	0.5290	0.2730	0.7389	1.0391	0.8457	1.0394	212

245 rows × 12 columns

Random Forest - Remove Outliers

	category	mix_predict	title_predict	Trust	Joy	Surprise	Anticipation	Fear	Anger	Disgust	Sadness	comments
0	新聞	pos	pos	5.2276	6.0957	5.8269	5.4967	0.4347	0.7251	0.4470	1.3124	17
1	新聞	pos	pos	6.7392	7.2615	6.1154	7.0486	0.3577	0.4990	0.5004	0.8641	11
2	新聞	pos	pos	6.5321	7.2815	7.0499	7.6680	0.7563	0.8893	1.0379	1.3384	264
5	問卦	pos	neg	0.7988	0.7348	0.1838	0.0977	5.7180	5.2093	5.5036	5.8055	1
9	新聞	pos	pos	6.2177	7.1949	6.9580	7.0330	0.2766	0.7458	0.9264	1.4924	13
...	...	...	...	...	...	...	...	...	...	...	...	...
420	新聞	pos	neg	6.1145	7.2552	6.9558	6.7165	0.5074	0.9019	0.5813	1.2807	31
423	問卦	neg	neg	1.4795	0.3989	0.9883	0.4617	7.2876	6.5310	6.8515	7.3672	1
426	新聞	pos	neg	0.7393	0.9027	0.8875	0.9404	1.1693	1.2011	0.9429	2.3210	285
427	問卦	neg	pos	0.6784	1.7261	2.2559	0.2382	6.6203	6.6478	6.1981	6.9905	77
428	新聞	neg	neg	0.5312	0.1788	0.5290	0.2730	0.7389	1.0391	0.8457	1.0394	212

230 rows × 12 columns

Random Forest - Split the training (70%) and testing set (30%)

Train X

Bert results + Keymoji scores

	category	mix_predict	title_predict	Trust	Joy	Surprise	Anticipation	Fear	Anger	Disgust	Sadness
411	2	0	0	1.4171	0.9691	0.8949	0.0952	5.8422	5.5582	5.7069	6.1785
392	1	0	1	0.6158	0.0411	0.3109	0.0975	5.6267	5.1936	6.0025	5.5343
288	1	0	0	0.9431	1.1344	1.2162	0.0975	6.5368	5.6930	6.1549	7.0323
9	2	1	1	6.2177	7.1949	6.9580	7.0330	0.2766	0.7458	0.9264	1.4924
79	2	0	1	6.9949	7.7016	6.7549	6.8085	0.1656	0.1559	0.0972	0.9903
...	...	...	...	...	...	...	...	...	...	...	...
167	1	0	0	1.1717	1.6687	1.3896	0.0987	6.1207	5.3274	5.9014	7.0811
20	2	1	0	0.2189	0.1782	0.2017	0.0446	0.2184	0.2885	0.1569	0.5334
141	2	1	1	0.3602	0.3749	0.3515	0.2760	0.2995	0.3007	0.2923	0.3677
285	2	1	1	0.4291	0.3849	0.4213	0.4350	0.2330	0.2800	0.3003	0.2763
162	2	0	0	1.7041	1.7783	1.7205	1.6202	3.4890	2.0821	1.8353	1.9892

Train Y

Comments

	comments
411	26
392	5
288	47
9	13
79	17
...	...
167	26
20	94
141	8
285	184
162	4

Accuracy

```
from sklearn import metrics
print('Mean Absolute Error:', metrics.mean_absolute_error(new_test_y, new_pred_y))
print('Mean Squared Error:', metrics.mean_squared_error(new_test_y, new_pred_y))
print('Root Mean Squared Error:', np.sqrt(metrics.mean_squared_error(new_test_y, new_pred_y)))
```

```
Mean Absolute Error: 1.5904733711081305
Mean Squared Error: 3.6530239281796746
Root Mean Squared Error: 1.9112885517837632
```

```
# Calculate the absolute errors
errors = abs(new_pred_y - new_test_y)
# Print out the mean absolute error (mae)
print('Mean Absolute Error:', round(np.mean(errors), 2), 'degrees.')

# Calculate mean absolute percentage error (MAPE)
mape = 100 * (errors / new_test_y)

# Calculate and display accuracy
accuracy = 100 - np.mean(mape)
print('Accuracy:', round(accuracy, 2), '%.')
```

```
Mean Absolute Error: 1.59 degrees.
Accuracy: 58.06 %.
```

Result

Actual comment number of test set

vs.

Prediction

	Actual	Predicted
397	5.584963	5.307983
101	2.000000	6.149640
15	2.000000	4.658203
343	6.321928	4.050191
22	5.169925	4.650202
...	...	...
192	4.459432	4.893522
41	2.584963	4.619536
261	4.247928	5.786965
104	4.643856	5.015419
123	3.459432	5.928810

69 rows × 2 columns

Result

批踢踢實業坊 › 看板 Gossiping
作者 dzwei (Args&&... args)
標題 [問卦] 當年同意東奧正名看到郭婞淳奪金在想啥？
時間 Fri Jul 30 07:45:44 2021

[新聞] 東奧正名沒過 郭婞淳放心了
https://tinyurl.com/54ew7fvz

底下噓聲一片

中國台北一片

結果人家打破世界紀錄奪金

那些噓聲一片中國台北一片的人

也跟著The蹭郭婞淳台灣之光了嘛？

有沒有八卦？

Actual number of comments: 48
Our prediction: 40

批踢踢實業坊 › 看板 Gossiping
作者 s66449 (老實樣)
標題 [新聞] 李智凱戰勝心魔奪銀！成功拿出壓箱寶
時間 Sun Aug 1 19:04:05 2021

https://tinyurl.com/3bkaeehd

李智凱戰勝心魔奪銀！成功拿出壓箱寶　完美落地激動落淚

2021年08月01日 18:38

記者王真魚／日本東京報導

「鞍馬王子」李智凱在2016年里約奧運不慎落馬，儘管仍有心魔，但化悲憤為力量，2017年台北世大運奪金，2018年雅加達亞運再奪隊史首金；今年二度闖奧運，資格賽鞍馬第一晉級，1日決賽成功挑戰難度6.700，僅次於英國惠特洛克（Max Whitlock ），最終奪下銀牌，成為台灣體操第一人。獲獎那一刻，他如釋重負，激動落下淚水，這一次，他真的突破里約奧運心魔。

Actual number of comments: 36
Our prediction: 18

批踢踢實業坊 › 看板 Gossiping
作者 postar (郵星)
標題 [新聞] 中國玻璃心碎!林昀儒鄭怡靜奪冠國旗風波
時間 Sun Mar 27 21:30:40 2022

1.媒體來源:自由時報

2.記者署名:梁偉銘

3.完整新聞標題:
桌球》中國玻璃心碎滿地！林昀儒／鄭怡靜奪冠國旗風波

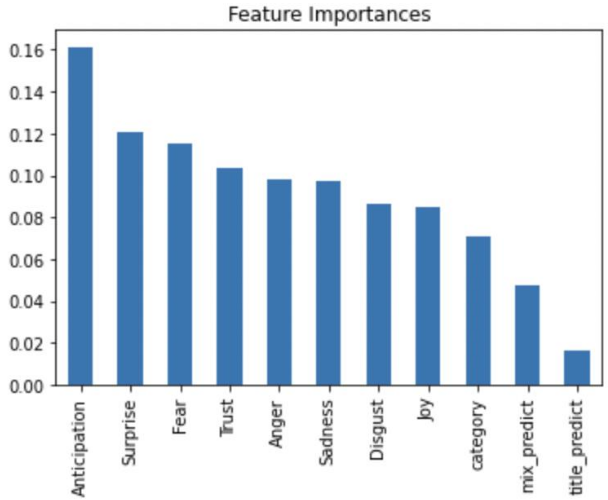
4.完整新聞內文:
https://i.imgur.com/dBB2UBb.jpg

Actual number of comments: 37
Our prediction: 27

Feature Importance

```
features = train_x.columns.to_list()
feature_importances = randomForestModel.feature_importances_
forest_importances = pd.Series(feature_importances, index=features).sort_values(ascending=False)
print("The firstly important features is: ", forest_importances.index[0])
print("The secondly important features is: ", forest_importances.index[1])
print("The thirdly important features is: ", forest_importances.index[2])
```

```
The firstly important features is:  Anticipation
The secondly important features is:  Surprise
The thirdly important features is:  Fear
```



Outcome - Kuo Hsing-Chun (郭婞淳)

Title Only

Title + Content

Outcome - Lee Chih-Kai (李智凱)

Title Only

Title + Content

Outcome - Lin Yun-Ju (林昀儒)

Title Only

Title + Content

Outcome - Tai Tzu-Ying (戴資穎)

Title Only

Title + Content

Outcome - Yang Yung-wei (楊勇緯)

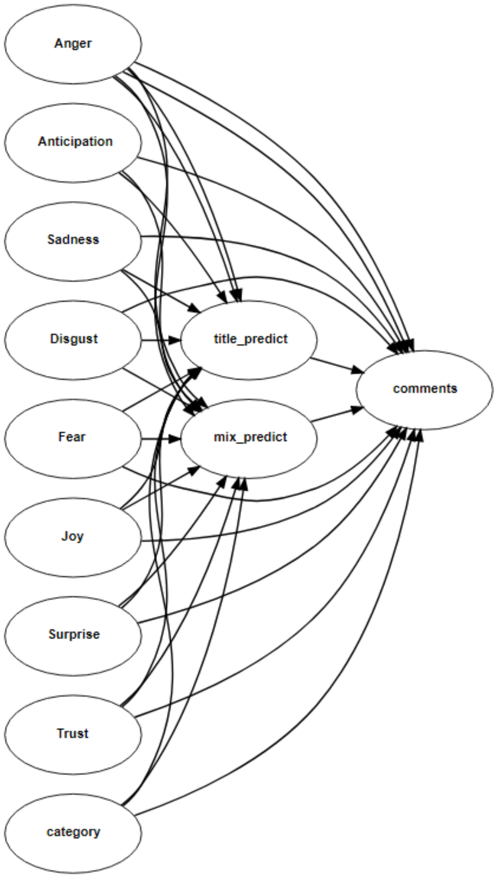
Title Only

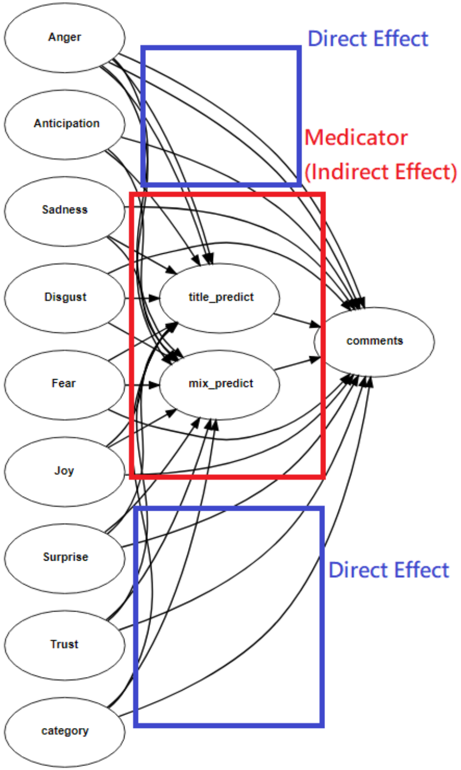
Title + Content

Compare Title Sentiment

Future Potential

Composite Model VS. Factor Model





Group 5

hank You

See you 6/21 !