大數據與商業分析期中報告

第一組

林軒逸 張睿君 單開民 余采庭 李沛璇 黃心柔 何勁儀

一、分析對象選擇：鴻海、友達、聯發科

本組在考量「新聞量多寡」以及「股價對新聞消息的敏感度」後，決定選擇新聞量多、股價波動大者作為分析對象，分別為鴻海、友達、聯發科。

二、建立訓練文件集，分出 GOUP 及 GODOWN 兩種文件類別

1. 用標題或內容篩選有提到鴻海的文章，對每日、每篇文，以及對應到該日七天後的均價做比對。若該日無股價資料，則標示為 NO DATA。
2. 價差：|Price(D+7)-Price(D)| / Price(D) 小於 5%者，視為不顯著，不予分類。
3. 若價差顯著，且 Price(D+7)>Price(D)，則歸類為 GOUP；

   Price(D+7)<Price(D)，則歸類為 GODOWN。

類別	日期	新聞
nodata	2016/1/1 13:00	全球最大手機用戶市場在中國大陸，其次是哪一個市場呢？答
nodata	2016/1/3 0:03	你知道只逛街、不買東西，也可以免費累積點數嗎？而且累積
nodata	2016/1/3 19:12	雖然中韓自由貿易協定生效，南韓面板銷陸並非立即零關稅，
common	2016/1/4 10:54	難忍低薪！根據yes123求職網調查，逾7成上班族今年首季想跳
common	2016/1/4 9:43	台股迎接2016新春開紅盤，但受到休市期間美股連二黑下跌逾
common	2016/1/4 14:52	去年前11月營收已達4兆734.41億 全年成長1成可達陣…… 鴻海
common	2016/1/4 12:45	新的一年到來，鴻海下午2時將在土城總部舉辦記者會，預料鴻
common	2016/1/4 13:08	由於4K電視價格下滑，逐漸成為市場主流，根據日本電子情報
common	2016/1/4 13:57	台北股市今天開低走低，收盤跌223.8點，為8114.26點，跌幅2.6

如 1/1 股市休市，該日不會擁有股價資料，分類標示為 nodata。

三、由步驟二所得之兩類文章中切出詞，挑選代表「漲」的詞及代表「跌」的詞

1. 將 GOUP 及 GODOWN 類文章中各自切出 2 gram 的詞。
2. 透過 Excel 去掉 GOUP 文章與 GODOWN 文章中共同出現的字詞，以幫助篩除「鴻海」、「今天」等中性字詞。
3. 合併字詞，如「聯發」、「發科」合併成「聯發科」。合併方式依據所計算的 tf-idf 值，再用人工看看法。
4. GOUP 類篩出的關鍵詞即為代表「上漲」的詞，GODOWN 篩出的關鍵詞即為代表「下跌」的詞。

聯發科goup	聯發科godown
英特爾	董事長
IC設計	美國
面板	台幣
行動	設計
連結	人才
開放	蘋果
裝置	獲利
震盪	公布
服務	毛利
大漲	領域
整體	創新
開發	員工
處理	走勢
方案	美股
業者	利率
部分	IC
油價	系統
終場	漲幅
缺貨	曾銘宗
相對	下跌
VR	平均
維持	上漲
行情	新高

友達godown	友達goup
台股漲	台股
億元	成交
金融股	蘋果
盤中	營業
夏普	金融
全球	申請
廠商	淨利
高點	減少
聯電	包括
競爭	全年
未來	顯示
A股	利率
出現震盪	張數
資金	指出
早盤	可望
權息	新高
低點	預期
資訊	銷售
紡織	昆山
華映	目前
發展	半年
食品股	維持
塑化股	液晶

鴻海goup	鴻海godown
日圓	iphone
銀行	台灣
指出	產品
簽約	印度
鴻夏戀	營收
雙方	權證
消息	開盤
負債	法人
收購	預期
財務	員工
INCJ	OLED
國際	大立光
方案	富士康
總部	布局
希望	中國
是否	銷售
月底	新台幣
以及	除權息
台幣	戴正吳
達成	方面
併購	跌幅
引述	權值
對於	漲幅

四、建立向量空間（以測試股之外的其中一股建立向量空間）

利用 Python 建立維度為 dim{GOUP+GODOWN} 的向量空間。

1. 去除標示為 NO DATA 的新聞，因為無從得知結果正確與否。
2. 將 GOUP 類與 GODOWN 類字詞合併成 List。
3. 尋找每篇新聞的向量。
4. 第四步驟共有三種方法：
    a. 0/1 法：有該字詞為 1，沒有為 0
    b. 次數法：計算該字詞在該新聞出現次數
    c. tf-idf 法：利用該字詞「總出現篇數(df)」與「該字詞在該新聞出現次數(tf)」計算 tf-idf 值
5. 計算訓練集向量，記錄於向量空間。
6. 挑選較簡單且準確的 kNN 法進行評估。

五、計算某新聞與他點的距離

1. 計算每篇欲歸類新聞與訓練集各點距離。
2. 預留前四個最小距離者，在一般情形以前三個鄰居進行投票，但若有三種分類（GOUP、GODOWN、COMMON）各一的情形，則以第四名的歸類為最終分類。

六、Inside Test 確認準確率

以非 NO DATA 的新聞作為訓練集，進行上述的計算，最後將模型預測結果與真實答案進行比較，驗證準確率。在這邊，比較三種方法的結果：

	聯發科	友達	鴻海
0/1 法	80.71%	73.91%	90.5%
次數法	80.71%	74.69%	90.02%
tf-idf 法	82.77%	74.69%	90.66%

聯發科

實際	預測
godown	common
godown	godown
godown	godown
godown	godown
godown	common
godown	godown
godown	godown
godown	common
godown	common
godown	godown
godown	common
godown	common
common	common
common	common
common	common
common	common
common	common
common	common
common	common
common	common
common	common
common	common
common	goup

友達

實際	預測
common	common
common	common
common	common
common	common
common	common
common	common
common	common
godown	godown
godown	common
godown	godown
godown	godown
godown	common
godown	common
godown	common
godown	godown
godown	godown
godown	common
godown	common
godown	common
common	common
common	common
common	common
common	common

鴻海

實際	預測
common	common
common	common
common	common
common	common
common	common
common	common
common	common
common	common
common	common
common	common
common	common
common	common
common	common
common	common
common	common
common	common
common	common
common	common
common	common
common	common
common	common
common	common
common	common
common	common
common	common
common	common
common	common
common	common
common	common

七、結論

tf-idf 法整體準確率較高，但與其他兩種發法相比，增加幅度並不高。

0/1 法與數數法各有高低準確率之公司。友達使用數數法準確率較高，而鴻海在使用 0/1 法時相較於數數法來得高，而聯發科的準確率不會有差別。

八、程式截圖

Step1.建造訓練集

```
F1=open("fox_data.txt",'r')
F2=open("fox_date.txt",'r')
F3=open("foxnews.txt",'r',encoding="UTF-8")
Output=open("7days.txt",'w')
Os=open("typ_fox.txt",'w')
Output1=open("fox_godown.txt",'w',encoding="UTF-8")
Output2=open("fox_goup.txt",'w',encoding="UTF-8")

b=F1.read().splitlines()
day=F2.read().splitlines()
new=F3.read().splitlines()
Out=[0]*len(b)
Out2=[0]*len(b)
newout=[0]*len(new)
newout2=[0]*len(new)
print(len(new))
#print(len(b))
for i in range(0,len(b)):
    j=len(b)-1-i
    if(b[j]=="" or b[j-6]==""):
        Out2[j]="nodata"
        Out[j]=""
    else:
        if(j-6> 0):
            #print(b[j-6])
            x1 = b[j]
            y1 = float(x1)
            x2 = b[j-6]
            y2 = float(x2)
            y = (y2-y1)/y1
            Out2[j]=y
        if(y > 0.05):
            Out[j] = "goup"
        elif(y < -0.05):
            Out[j] = "godown"
        else:
            Out[j] = "common"
```

```
for i in range(0,len(new)):
    for j in range(0,len(b)):
        if(day[j] in new[i]):
            Os.write(Out[j])
            Os.write("\n")
            break
        if(j==len(b)-1):
            Os.write("nodata\n")
for l in range(0,len(b)):
    m=len(b)-1-l
    if(Out[m] == "godown"):
        for n in range(0,len(new)):
            if(day[m] in new[n]):
                newout[n]=new[n]
    if(Out[m] == "goup"):
        for n in range(0,len(new)):
            if(day[m] in new[n]):
                newout2[n]=new[n]
for k in range(0,len(b)):
    Output.write(str(day[k]))
    Output.write(" ")
    #Output.write(str(Out2[k]))
    #Output.write(" ")
    Output.write(str(Out[k]))
    Output.write("\n")
for newn in range(0,len(new)):
    if(newout[newn]!=0):
        Output1.write(str(newout[newn]))
        Output1.write("\n")
for newn in range(0,len(new)):
    if(newout2[newn]!=0):
        Output2.write(str(newout2[newn]))
        Output2.write("\n")
```

Step2.建造向量空間，計算鄰居並以去除 nodata 的資料計算準確率（以 tf-idf 法為例）

```
def df(FINDEE,FINDFROM):
    c=[0]*len(FINDEE)
    for i in range(0,len(FINDEE)):
        for j in range(0,len(FINDFROM)):
            if(FINDEE[i] in FINDFROM[j]):
                #print (FINDEE[i])
                c[i]+=1
    return c

def tf_idfs(tf,df,n):
    tf_idfa=[]
    for s in range(0,len(tf)):
        tf_idfa.append([])
    for i in range(0,len(tf)):
        for j in range(0,len(tf[i])):
            x=tf[i][j]
            #print(x)
            y=df[j]
            #print(y)
            z=n/y
            #print(z)
            if (x==0):
                tf_idfa[i].append(0)
            else:
                tf_idfa[i].append((1+log10(x))*log10(z))
    return tf_idfa


F1=open("foxup.txt",'r',encoding="UTF-8")
F2=open("foxdown.txt",'r',encoding="UTF-8")
F3=open("foxconn.txt",'r',encoding="UTF-8")
F10=open("typ_fox.txt",'r',encoding="UTF-8")

F7=open("result_fox","w",encoding="UTF-8")
F8=open("nonodata_typ_fox","w",encoding="UTF-8")
F9=open("nonodata_fox","w",encoding="UTF-8")
#IND=open("indcies.txt",'r',encoding="UTF-8")
#Output=open("df_台積電_6.txt",'w')
typ=[]
a1=F1.read().splitlines()
a2=F2.read().splitlines()
typ=F10.read().splitlines()

a=a1+a2
#print(a)
#print(typ)
b=F3.read().splitlines()
t_b=[]
t_typ=[]

for i in range(0,len(b)):
    if(typ[i]!="nodata"):
        t_b.append(b[i])
        t_typ.append(typ[i])
        F8.write(typ[i])
        F8.write("\n")

b=t_b
typ=t_typ
for i in range(0,len(b)):
    F9.write(b[i])
    F9.write("\n")

print(len(b))
print(len(typ))
#print(len(b))
#ind=IND.read().splitlines()
vector=[]
for s in range(0,len(b)):
    vector.append([])
#print(vector)

for i in range(0,len(b)):
    for j in range(0,len(a)):
        if(a[j] in b[i]):
            #print (a[i])
            vector[i].append(b[i].count(a[j]))
        else:
            vector[i].append(0)
```

```
target=b
print(len(target))
#print(target)
#print(len(target))
t_Vector=[]
for s in range(0,len(target)):
    t_Vector.append([])

for i in range(0,len(target)):
    for j in range(0,len(a)):
        if(a[j] in target[i]):
            #print (a[i])
            t_Vector[i].append(target[i].count(a[j]))
        else:
            t_Vector[i].append(0)

dfs= df(a,b)
print(dfs)

vector=tf_idfs(vector,dfs,len(b))
t_Vector=tf_idfs(t_Vector,dfs,len(b))

vector_nor=vector
target_nor=t_Vector
result_arr=[]
vec_result=[0]*len(vector_nor)
for k in range(0,len(target_nor)):
    target_numpy=np.array(target_nor[k])
    #print(target_numpy)
    for l in range(0,len(vector_nor)):
        try_numpy=np.array(vector_nor[l])
        vec_result[l]=np.linalg.norm(try_numpy-target_numpy)
    arr=np.array(vec_result)
    index=arr.argsort()[:3]
    index_pre=arr.argsort()[4]
    up=0
    down=0
    common=0
    for i in range(0,len(index)):
        #print(typ[index[i]])
        if(typ[index[i]]=="goup"):
            up=up+1
        if(typ[index[i]]=="godown"):
            down=down+1
        if(typ[index[i]]=="common"):
            common=common+1
    if(up>=2):
        F7.write("goup\n")
        result_arr.append("goup")
    if(down>=2):
        F7.write("godown\n")
        result_arr.append("godown")
    if(common>=2):
        F7.write("common\n")
        result_arr.append("common")
    if(up==1 and down==1 and common==1):
        F7.write(typ[index_pre])
        F7.write("\n")
        result_arr.append(typ[index_pre])
    #print(len(index))
acc=0
for i in range(0,len(result_arr)):
    if(result_arr[i]==typ[i]):
        acc=acc+1

print(acc/len(result_arr))
```

準確率