

摘要

在一般的股票市場中，有三大層面的分析方式，基本面、技術面、籌碼面。

1. 基本面:透過影響個別公司財務與獲利之所有因素的分析、基本經濟學分析、產業分析，來預測該公司未來狀況之方法。
2. 籌碼面:這種分析方式主要就是跟著該公司的大股東走，來做出對自己有利的行為。
3. 技術面:透過研究市場過去和現在的行為，進一步去預測市場未來的變化，市場行為包括價格的高低、價格的變化、發生這些變化所伴隨的成交量，以及完成這些變化所經過的時間，在基本面會探討公司內部的運作情形，甚至產業的走向，但是在技術面是不討論這些的，因為技術面認為基本面的資訊早就反映在股價上了，所以技術面也就以這個為基礎，單看市場上的供給需求來預測未來股市走向。

技術分析其實更像是一門統計學，透過過去所擁有的資料來預測未來的資料，而這樣的分析模式也在近幾年因為人工智慧技術蓬勃發展而廣受大家所使用，所以本次專題我們就也就以技術面的方式來進行股市分析及預測。

方法及步驟

1.1 資料收集:

我們採用了台灣經濟新報 TEJ 資料庫裡面各支股票的資料，隨機收集橫跨了 30 種類，總共 50 家公司，每家公司採用了共十年的原始資料(2009-03-02~2019-02-27)以天為單位，所以總資料量為 122800 筆資料。

表一:公司種類+所有公司

水泥工業:	台泥，亞泥	食品工業:	味全，味王	塑膠工業:	南亞
紡織纖維:	新纖	電機機械:	土電，東元	電器電纜:	華電，聲寶
化學生技醫療:	上曜，中化	電腦及週邊設備業:	全友	資訊服務業:	三商電，凌群
油電燃氣業:	山隆，台塑化	金融保險:	彰銀，京城銀	鋼鐵工業:	中特鋼，中鋼
玻璃陶瓷:	台玻，冠軍	造紙工業:	台紙，土紙	汽車工業:	裕隆
橡膠工業:	南港，泰豐	其他電子業:	金寶，鴻海	電子工業:	首利,信錦
建材營造:	國產	觀光事業:	萬企，華園	化學工業:	榮化
航運業:	台船，長榮	電子通路業:	聯強，精技	光電業:	中環，光磊
貿易百貨:	益航	生技醫療業:	葡萄王	通信網路業:	台揚，東訊
其他:	中福	半導體業:	麗正，聯電	電子零組件業:	台達電

這些原始資料當中，我們只取了開盤價、收盤價、最高價、最低價，用這四種應用來做出往後的資料分析部分。

1.2 資料處理:

我們透過技術分析指標法+每天的漲跌，得到我們想要的資料特徵。

1. KD 指標:

要算出 KD 之前我們得先求未成熟隨機值(RSV)

$$RSV = \frac{\text{今日收盤價} - \text{最近九天最低價}}{\text{最近九天最高價} - \text{最近九天最低價}}$$

而從公式可以看出，把 9 天內的股價總波動當分母、當天收盤價跟 9 天內最低點的差當作分子，衡量當天收盤價在這 9 天內相對位置是強還是弱，而一旦算出了 RSV 值就可以算出當天的 K 值與 D 值了。

$$K \text{ 值} = \frac{2}{3} \times \text{昨天的 K 值} + \frac{1}{3} \times \text{今天的 RSV 值}$$

$$D \text{ 值} = \frac{2}{3} \times \text{昨天的 D 值} + \frac{1}{3} \times \text{今天的 K 值}$$

其實 K 值也就是將 RSV 值做加權平均而 D 值就是做 K 值的加權平均，而 K 值這樣做是為了減少 RSV 的反應過於靈敏而造成許多雜訊的問題，那 D 值也就是更鈍化的 K 值了。總結來說其實做 KD 值是希望能靈敏地反映出價格在市場上的變化大小。

```
In [ ]: def get_KD(stock_data, company_keys):
        KD = OrderedDict({})
        for key in company_keys:
            KD[key] = {}
            RSV = get_RSV(key, stock_data)
            K = get_K(key, stock_data, RSV)
            D = get_D(key, stock_data, K)
            KD[key]['RSV'] = RSV
            KD[key]['K'] = K
            KD[key]['D'] = D
        return KD

    def get_RSV(key, stock_data):
        nineDayLowest = np.array([False] * 8)
        nineDayHighest = np.array([False] * 8)
        RSV = np.array([False] * 8)
        for i in range(8, len(stock_data[key]['年月日'])):
            nineDayLowest = np.append(nineDayLowest, min(stock_data[key]['最低價'][i - 8:i + 1]))
            nineDayHighest = np.append(nineDayHighest, max(stock_data[key]['最高價'][i - 8:i + 1]))
        RSV = np.append(RSV, ((stock_data[key]['收盤價'][8:] - nineDayLowest[8:]) / (nineDayHighest[8:] - nineDayLowest[8:])) * 100)
        return RSV

    def get_K(key, stock_data, RSV):
        K = np.array([False] * 7 + [50])
        for i in range(8, len(stock_data[key]['年月日'])):
            todayK = (2/3.) * K[i-1] + (1/3.) * RSV[i]
            K = np.append(K, todayK)
        return K

    def get_D(key, stock_data, K):
        D = np.array([False] * 7 + [50])
        for i in range(8, len(stock_data[key]['年月日'])):
            todayD = (2/3.) * D[i-1] + (1/3.) * K[i]
            D = np.append(D, todayD)
        return D
```

圖一:KD 指標程式碼

2. SMA 指標:

SMA 為簡單移動平均

$$SMA = \frac{\text{最近 } n \text{ 天收盤價總和}}{n}$$

這邊我們讓 $n = 10$ 。SMA 是一種簡單的方式反映出每天收盤價的趨勢。

```
In [ ]: def get_SMA(stock_data, company_keys, t):
        SMAx = OrderedDict({})
        for key in company_keys:
            SMAx[key] = np.array([False] * (t - 1))
            for j in range((t - 1), len(stock_data[key]['*年月日'])):
                SMAx[key] = np.append(SMAx[key], stock_data[key]['收盤價'][j - (t - 1):j + 1].sum() / float(t))
        return SMAx
```

圖二:SMA 程式碼

3. RSI 指標:

RSI 為相對強弱指標，是以某段時間，股價總漲幅跟股價總跌幅的比例來預測未來股價的漲跌機率。

$$RSI = \frac{\text{最近 9 天的總漲幅}}{\text{9 天的總跌幅} + \text{9 天的總漲幅}}$$

```
In [ ]: def get_RSI(stock_data, company_keys):
    RSI = OrderedDict({})
    for key in company_keys:
        RSI[key] = np.array([False] * 9)
        for j in range(9, len(stock_data[key]['*年月日'])):
            rs = get_RS(stock_data[key], j)
            if(rs == 'infinity'):
                rsi = 100
                RSI[key] = np.append(RSI[key], rsi)
            else:
                rsi = 100 * (rs / float((1 + rs)))
                RSI[key] = np.append(RSI[key], rsi)
    return RSI

def get_RS(oneData, day):
    rise = float(0)
    fall = float(0)
    for i in range(9):
        if(oneData['收盤價'][i + (day - 8)] - oneData['收盤價'][i + (day - 9)] > 0):
            rise += (oneData['收盤價'][i + (day - 8)] - oneData['收盤價'][i + (day - 9)])
        elif(oneData['收盤價'][i + (day - 8)] - oneData['收盤價'][i + (day - 9)] < 0):
            fall += abs(oneData['收盤價'][i + (day - 8)] - oneData['收盤價'][i + (day - 9)])
    if((fall/9.) == 0):
        return 'infinity'
    else:
        return ((rise/9.) / (fall/9.))
```

圖三:RSI 程式碼

4. MACD 指標:

MACD 是利用兩條不同天數的指數平均線算出差異值，再將差值做一次指數平線。

DIF = EMA(收盤價,12) – EMA(收盤價,26)

MACD = EMA(DIF,9)

程式碼:

```
In [ ]: def get_MACD(stock_data, company_keys):
    MACD = OrderedDict({})
    EMA12 = OrderedDict({})
    EMA26 = OrderedDict({})
    DIFF = OrderedDict({})
    for key in company_keys:
        EMA12[key] = get_EMAx(stock_data[key]['收盤價'], 12)
        EMA26[key] = get_EMAx(stock_data[key]['收盤價'], 26)
        DIFF[key] = np.array([False] * 25)
        DIFF[key] = np.append(DIFF[key], EMA12[key][25:] - EMA26[key][25:])
        MACD[key] = get_DIFFEMA(DIFF[key], 9)
    return MACD

def get_EMAx(x, t):
    oneEMA = np.array([False] * (t - 1))
    oneEMA = np.append(oneEMA, sum(x[:t])/float(t))
    for i in range(t, len(x)):
        value = oneEMA[i - 1] + (2/float(t+1)) * (x[i] - oneEMA[i - 1])
        oneEMA = np.append(oneEMA, value)
    return oneEMA

def get_DIFFEMA(x,t):
    oneEMA = np.array([False] * 33)
    oneEMA = np.append(oneEMA, sum(x[25:34])/9.)
    for i in range((t+25), len(x)):
        value = oneEMA[i - 1] + (2/float(t+1)) * (x[i] - oneEMA[i - 1])
        oneEMA = np.append(oneEMA, value)
    return oneEMA
```

圖四:MACD 程式碼

5. 每天漲跌:

最後我們加上每天的(收盤價)-(開盤價)。

程式碼:

```
In [ ]: def get_rise_fall(stock_data, company_keys, each_year_data):
    rise_fall = OrderedDict({})
    for key in company_keys:
        rise_fall[key] = (stock_data[key]['收盤價'] - stock_data[key]['開盤價'])
    return rise_fall
```

圖五:每天漲跌程式碼

表二:各支股票 precision,recall

	class1 precision	class1 recall	class2 precision	class2 recall	class3 precision	class3 recall	class4 precision	class4 recall
1101 台泥	0.745	0.677	0.509	0.539	0.525	0.579	0.731	0.672
1102 亞泥	0.763	0.735	0.563	0.603	0.538	0.562	0.735	0.671
1201 味全	0.742	0.693	0.623	0.777	0.333	0.05	0.741	0.786
1203 味王	0.76	0.694	0.539	0.6	0.592	0.722	0.732	0.524
1303 南亞	0.782	0.77	0.507	0.431	0.551	0.683	0.784	0.669
1409 新纖	0.771	0.626	0.688	0.855	0.525	0.323	0.705	0.732
1503 土電	0.734	0.81		0	0.647	0.839	0.772	0.557
1504 東元	0.699	0.603	0.543	0.688	0.485	0.448	0.701	0.597
1603 華電	0.736	0.665	0.607	0.746	0.563	0.55	0.711	0.601
1604 聲寶	0.696	0.701	0.638	0.638	0.632	0.726	0.758	0.614
1316 上曜	0.734	0.657	0.627	0.8	0.478	0.369	0.712	0.614
1701 中化	0.776	0.642	0.645	0.858		0	0.776	0.74
1802 台玻	0.748	0.656	0.575	0.746	0.534	0.428	0.751	0.641
1806 冠軍	0.766	0.731	0.685	0.85	0.562	0.301	0.704	0.637
1902 台紙	0.72	0.676	0.58	0.691	0.603	0.664	0.723	0.509
1903 士紙	0.726	0.743	0.582	0.649	0.543	0.571	0.736	0.591
2002A 中鋼特	0.718	0.736	0.514	0.456	0.622	0.828	0.865	0.431
2002 中鋼	0.738	0.774	0.567	0.538	0.575	0.632	0.742	0.657
2101 南港	0.755	0.621	0.622	0.858	0.5	0.01	0.742	0.712
2102 泰豐	0.805	0.696	0.652	0.873	0.462	0.167	0.7	0.704
2201 裕隆	0.749	0.724	0.664	0.83		0	0.746	0.701
1471 首利	0.753	0.711	0.692	0.873		0	0.76	0.706
1582 信錦	0.817	0.662	0.611	0.833		0	0.741	0.75
2504 國產	0.811	0.687	0.72	0.848	0.617	0.404	0.665	0.698
2208 台船	0.799	0.651	0.661	0.884		0	0.775	0.754
2603 長榮	0.764	0.705	0.649	0.846	0.389	0.059	0.725	0.761
2701 萬企	0.7	0.859	0.4	0.123	0.679	0.765	0.722	0.673
2702 華園	0.686	0.743	0.631	0.67	0.562	0.595	0.745	0.583
2801 彰銀	0.682	0.723	0.55	0.518	0.636	0.771	0.735	0.5
2809 京城銀	0.749	0.648	0.6	0.775	0.538	0.326	0.739	0.753
2601 益航	0.743	0.626	0.685	0.868	0.233	0.04	0.694	0.705
1435 中福	0.753	0.778	0.625	0.662	0.629	0.696	0.758	0.601
1704 榮化	0.752	0.63	0.597	0.847	0.167	0.004	0.72	0.721

1707 葡萄王	0.737	0.694	0.584	0.811		0	0.724	0.674
2616 山隆	0.727	0.578	0.57	0.674	0.621	0.731	0.752	0.597
6505 台塑化	0.747	0.667	0.511	0.608	0.517	0.501	0.71	0.673
2302 麗正	0.79	0.671	0.705	0.876	0.456	0.23	0.674	0.676
2303 聯電	0.733	0.637	0.593	0.774	0.374	0.163	0.69	0.728
2305 全友	0.756	0.742	0.679	0.762	0.642	0.605	0.705	0.636
2323 中環	0.782	0.601	0.742	0.901			0.932	0.75
2340 光磊	0.784	0.653	0.645	0.837		0	0.719	0.733
2314 台揚	0.782	0.643	0.674	0.841		0	0.75	0.776
2321 東訊	0.687	0.885	0.318	0.042	0.73	0.862	0.76	0.562
2308 台達電	0.757	0.538	0.532	0.772	0.394	0.211	0.681	0.709
2347 聯強	0.762	0.671	0.576	0.794	0.25	0.004	0.726	0.779
2414 精技	0.77	0.684	0.662	0.829	0.504	0.291	0.73	0.69
2427 三商電	0.787	0.68	0.684	0.866	0.477	0.129	0.725	0.724
2453 凌群	0.788	0.639	0.71	0.888		0	0.89	0.73
2312 金寶	0.799	0.718	0.671	0.81	0.35	0.042	0.704	0.838
2317 鴻海	0.785	0.646	0.587	0.826	0	0	0.723	0.752

表三:各支股票 K,D,SMA,RSI 統計資料

	K 平均值	K 標準差	D 平均值	D 標準差	SMA 平均值	SMA 標準差	RSI 平均值	RSI 標準差
1101 台泥	51.67	23.615	51.643	20.731	37.057	5.1	50.769	20.466
1102 亞泥	53.677	24.872	53.655	22.125	33.828	4.697	51.738	21.235
1201 味全	45.27	22.95	45.23	20.302	30.584	10.871	48.189	22.187
1203 味王	46.077	19.572	46.084	17.075	23.224	1.901	50.458	19.322
1303 南亞	54.78	25.365	54.783	22.415	67.456	8.755	52.125	21.189
1409 新纖	46.502	21.931	46.503	19.352	10.392	1.667	50.478	19.483
1503 土電	54.862	21.923	54.841	19.298	38.131	2.889	50.427	19.057
1504 東元	51.417	22.857	51.406	19.86	24.58	6.002	51.064	18.921
1603 華電	46.373	21.397	46.41	18.865	14.833	6.16	49.917	20.255
1604 聲寶	50.599	23.369	50.584	20.906	11.647	3.402	52.488	23.045
1316 上曜	44.08	21.881	44.09	19.246	15.617	5.819	48.048	22.844
1701 中化	44.652	21.325	44.616	18.681	20.624	2.808	49.623	20.015
1802 台玻	46.616	23.898	46.588	21.127	24.234	9.008	49.138	20.763
1806 冠軍	44.556	21.084	44.551	18.525	11.984	5.779	48.52	21.021
1902 台紙	46.149	21.877	46.12	19.303	13.075	3.389	49.343	21.007
1903 土紙	45.341	21.697	45.311	19.129	43.511	13.242	47.694	19.552
2002A 中鋼特	56.382	24.922	56.32	22.06	40.646	1.519	54.551	21.075

2002 中鋼	51.742	24.607	51.741	21.533	26.333	3.803	49.814	20.287
2101 南港	42.962	22.278	42.938	19.499	35.151	7.747	47.541	20.74
2102 泰豐	43.549	22.332	43.526	19.725	17.336	3.877	47.233	20.666
2201 裕隆	44.674	24.187	44.632	21.619	41	14.308	47.192	21.746
1471 首利	41.273	21.505	41.282	18.75	10.699	7.163	46.722	21.23
1582 信錦	48.957	22.44	48.923	19.887	56.316	10.259	50.508	21.775
2504 國產	44.604	22.881	44.586	20.23	10.862	2.226	48.259	22.321
2208 台船	43.901	22.182	43.848	19.416	19.966	5.951	47.679	21.249
2603 長榮	46.78	23.599	46.772	20.88	17.155	3.967	48.845	20.696
2701 萬企	48.057	20.766	48.064	18.113	14.503	1.234	49.762	18.316
2702 華園	44.508	21.068	44.539	18.648	19.726	4.841	48.21	19.774
2801 彰銀	53.769	24.513	53.73	21.661	17.595	2.213	52.392	20.988
2809 京城銀	50.373	22.402	50.351	19.54	24.655	7.307	52.044	19.667
2601 益航	42.404	22.871	42.369	20.143	25.824	19.711	47.153	21.733
1435 中福	46.082	21.798	46.07	19.56	5.599	1.259	46.243	20.793
1704 榮化	47.84	23.778	47.804	21.091	40.926	14.503	50.678	21.769
1707 葡萄王	48.309	21.174	48.331	18.703	132.736	69.287	51.908	19.295
2616 山隆	54.912	22.188	54.885	19.623	24.023	5.214	54.744	21.625
6505 台塑化	52.736	22.713	52.726	19.638	90.17	15.509	51.002	18.624
2302 麗正	42.098	21.914	42.136	19.137	6.239	1.473	47.955	21.399
2303 聯電	47.775	23.096	47.761	20.274	13.487	1.793	49.608	19.897
2305 全友	43.171	21.785	43.156	19.397	5.92	2.214	48.108	20.682
2323 中環	40.398	21.96	40.417	19.233	5.297	1.642	47.896	20.929
2340 光磊	44.926	21.737	44.888	18.938	15.938	4.926	49.643	20.935
2314 台揚	42.574	21.406	42.578	18.452	17.758	7.529	47.59	20.909
2321 東訊	44.515	23.006	44.524	20.029	3.74	2.776	45.419	20.969
2308 台達電	52.136	22.709	52.16	19.679	137.513	34.3	51.145	18.956
2347 聯強	48.395	22.932	48.417	20.249	50.418	15.203	49.375	20.181
2414 精技	48.028	21.888	48.023	19.151	16.935	2.215	51.923	20.989
2427 三商電	43.505	22.01	43.463	19.409	10.063	3.213	48.834	20.865
2453 凌群	40.379	20.586	40.363	18.004	12.915	2.15	48.42	20.572
2312 金寶	46.844	21.813	46.863	19.16	10.175	2.177	50.713	20.551
2317 鴻海	49.915	24.112	49.903	21.18	93.245	17.039	50.531	20.49

表四:各支股票 MACD,一天漲跌統計資料

	MACD 平 均值	MACD 標 準差	(一天漲 跌)平均值	(一天漲 跌)標準差
1101 台泥	0.006	0.573	-0.019	0.529

1102 亞泥	0.004	0.532	-0.007	0.43
1201 味全	-0.073	0.674	-0.048	0.545
1203 味王	0.003	0.233	0	0.256
1303 南亞	0.06	1.009	0.016	0.851
1409 新纖	0.009	0.194	-0.017	0.171
1503 士電	0.001	0.281	0.045	0.352
1504 東元	0.015	0.355	-0.022	0.373
1603 華電	0.036	0.301	-0.007	0.257
1604 聲寶	0.027	0.186	-0.002	0.139
1316 上曜	0.013	0.585	-0.032	0.385
1701 中化	-0.016	0.354	-0.042	0.31
1802 台玻	-0.039	0.567	-0.025	0.467
1806 冠軍	-0.036	0.268	-0.027	0.255
1902 台紙	0.028	0.249	-0.005	0.221
1903 士紙	-0.131	0.784	-0.026	0.738
2002A 中鋼特	0.024	0.142	0.001	0.204
2002 中鋼	-0.017	0.277	-0.002	0.222
2101 南港	-0.026	0.555	-0.056	0.58
2102 泰豐	-0.043	0.307	-0.026	0.264
2201 裕隆	-0.065	0.883	-0.093	0.722
1471 首利	-0.084	0.354	-0.042	0.297
1582 信錦	-0.016	1.028	-0.108	0.777
2504 國產	-0.02	0.173	-0.023	0.164
2208 台船	-0.014	0.564	-0.033	0.342
2603 長榮	-0.015	0.399	-0.032	0.325
2701 萬企	-0.004	0.156	0.007	0.163
2702 華園	-0.025	0.494	-0.015	0.352
2801 彰銀	0.009	0.242	-0.002	0.197
2809 京城銀	0.065	0.388	-0.022	0.325
2601 益航	-0.108	0.835	-0.057	0.709
1435 中福	0.003	0.154	-0.005	0.125
1704 榮化	0.045	1.114	-0.075	0.857
1707 葡萄王	0.496	3.182	-0.229	2.829
2616 山隆	0.043	0.218	0.004	0.167
6505 台塑化	0.087	1.145	0.034	1.286
2302 麗正	0.012	0.174	-0.012	0.155
2303 聯電	-0.014	0.248	-0.016	0.186

2305 全友	0.004	0.189	-0.01	0.155
2323 中環	-0.002	0.158	-0.022	0.119
2340 光磊	-0.028	0.385	-0.045	0.346
2314 台揚	0.017	0.553	-0.057	0.441
2321 東訊	-0.03	0.149	0.005	0.121
2308 台達電	0.169	2.271	-0.122	2.168
2347 聯強	-0.091	0.76	-0.085	0.767
2414 精技	0.006	0.223	-0.02	0.21
2427 三商電	0.004	0.332	-0.028	0.254
2453 凌群	-0.004	0.265	-0.046	0.27
2312 金寶	0.006	0.185	-0.017	0.171
2317 鴻海	-0.203	1.599	-0.131	1.173

1.3 資料分析:

本次專題我們拿 python SkLearn 套件的 SVM model 演算法來分析歸類我們的資料。

```
In [ ]: from sklearn.svm import SVC
svm = SVC(kernel = 'rbf', probability = True, gamma = 'auto')
```

圖六:SVM 模型參數程式碼

SVM-X 軸資料:

除了每天的 KD 值、SMA 值、RSI 值、MACD 值、當天漲跌五個 feature 之外，為了讓這個 model 有學習的效果，我們還加了前四天的 KD 值、SMA 值、RSI 值、MACD 值、當天漲跌，總共 30 個 feature 當作 x 軸，然後將每個 feature 做 Z-score 的標準化。

```
In [ ]: def x_axis(feature_pair):
    x = []
    feature_pair = np.array(feature_pair)
    for i in range(6):
        feature_pair[:,i] = preprocessing.scale(feature_pair[:,i])
    reserve_K = feature_pair[:,0]
    reserve_D = feature_pair[:,1]
    reserve_SMA10 = feature_pair[:,2]
    reserve_RSI = feature_pair[:,3]
    reserve_MACD = feature_pair[:,4]
    reserve_rise_fall = feature_pair[:,5]
    for i in range(5):
        x.append(reserve_rise_fall[4 - i:len(feature_pair) - i].reshape(-1, 1))
        x.append(reserve_K[4 - i:len(feature_pair) - i].reshape(-1, 1))
        x.append(reserve_D[4 - i:len(feature_pair) - i].reshape(-1, 1))
        x.append(reserve_SMA10[4 - i:len(feature_pair) - i].reshape(-1, 1))
        x.append(reserve_RSI[4 - i:len(feature_pair) - i].reshape(-1, 1))
        x.append(reserve_MACD[4 - i:len(feature_pair) - i].reshape(-1, 1))
    x = np.concatenate(x, axis = 1)
    return np.delete(x, len(x)-1, axis = 0)
```

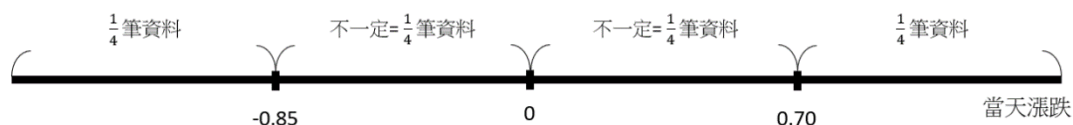
圖七: SVM-X 軸資料程式碼

SVM-Y 軸資料:

因為這份專題主要是要預測未來的漲跌價，所以 y 軸就是隔天的漲跌量，但因為是以 svm 模型來預測，所以必須將漲跌的量去做分類，每家公司的分類方式如下:

例如:1101 台泥

所有y軸資料:



圖八:分類方式

將 1、4 類的資料各占總資料的四分之一，再以數值 0(當天漲跌為 0)做為切割點，大於零分類為 3，小於 0 分類為 2，所以如果以這支股票分類就是狀況如下:

y-data<0.85 分類為 1

y-data >= 0.85 and y-data < 0 分類為 2

y-data >= 0 and y-data < 0.7 分類為 3

y-data>=0.7 分類為 4

```
In [ ]: def y_axis(rise_fall):
    total_y = np.array([])
    for i in range(len(rise_fall) - 5): total_y = np.append(total_y, sum(rise_fall[i + 1:i + 6]))
    return total_y
def y_clf(total_y):
    y = total_y.copy()
    mini = min(y)
    divide = []
    for i in range(2000):
        if(sum(y <= (mini + i*0.05)) >= ((1/4)*len(y))):
            divide.append(mini + i*0.05)
            break
    divide.append(0)
    for i in range(2000):
        if(sum(y <= (mini + i*0.05)) >= ((3/4)*len(y))):
            divide.append(mini + i*0.05)
            break
    y[(total_y >= mini) & (total_y < divide[0])] = 1
    y[(total_y >= divide[0]) & (total_y < divide[1])] = 2
    y[(total_y >= divide[1]) & (total_y < divide[2])] = 3
    y[(total_y >= divide[2]) & (total_y <= max(y))] = 4
    y = y.astype(int)
    print(divide)
    return y
```

圖九: SVM-Y 軸資料程式碼

資料調整:

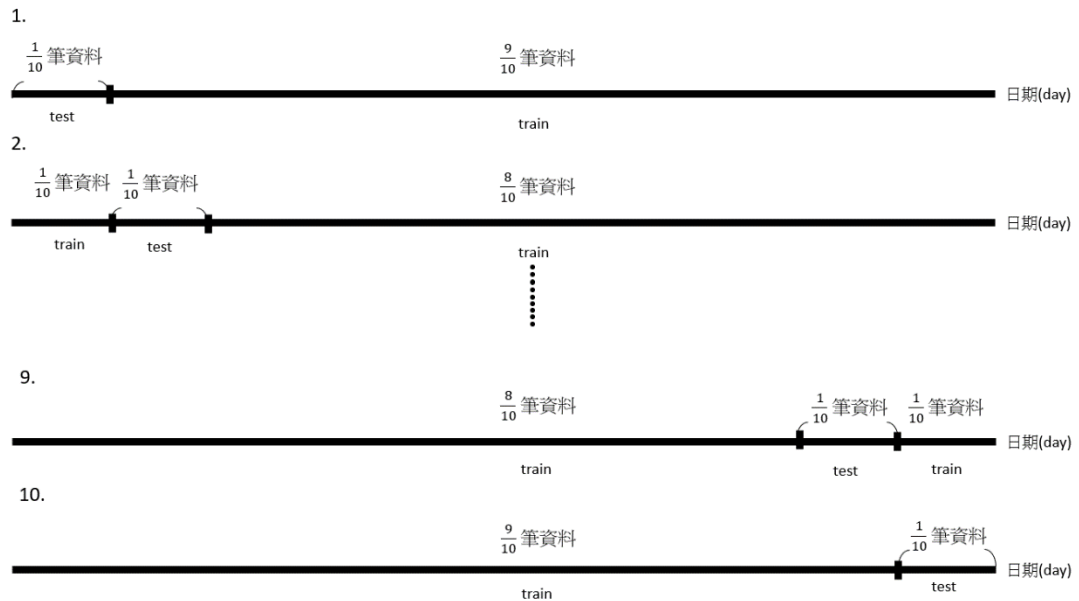
我們將 y 值資料絕對值小於 0.1 的刪除，這個目的是因為變化量小於 0.1 有點過小我們將它視為雜訊濾掉。

```
In [ ]: mask = (total_y >= 0.1) | (total_y <= -0.1)
total_y = total_y[mask]
total_x = total_x[mask]
```

圖九: 過濾資料程式碼

訓練及預測資料:

我們把訓練跟預測的資料分為 9:1(train:test)，以這種方式做每個區間的訓練及預測，方法如下圖:



圖十:資料訓練及預測方式

用這個方式就可以確保每個區間的資料都有被拿來訓練及預測。

```
In [ ]: def predict(total_x, total_y):
    svm = SVC(kernel = 'rbf', probability = True, gamma = 'auto')
    predict_y = np.array([])
    s = 10 #train:test = s-1:1
    testAmount = round(len(total_x)/s)
    for i in range(s):
        if(i == s-1):
            train_x = total_x[: (i*testAmount)]
            train_y = total_y[: (i*testAmount)]
            test_x = total_x[(i*testAmount):]
            svm.fit(train_x, train_y)
            predict_y = np.append(predict_y, svm.predict(test_x))
        else:
            train_x = np.delete(total_x, range((i*testAmount), (i*testAmount) + testAmount), axis = 0)
            train_y = np.delete(total_y, range((i*testAmount), (i*testAmount) + testAmount), axis = 0)
            test_x = total_x[(i*testAmount): (i*testAmount) + testAmount]
            svm.fit(train_x, train_y)
            predict_y = np.append(predict_y, svm.predict(test_x))
    return predict_y.astype(np.int16)
```

圖十一:資料訓練、預測程式碼

表五: 個支股票預測結果(猜對分類正確率):

台泥:61.46%	亞泥:64.11%	味全:68.74%	味王:64.64%	南亞:65.31%
上曜:65.58%	新疆:70.02%	中福:70.85%	首利:73.41%	土電:65.84%
東元:60.01%	信錦:68.47%	華電:65.60%	聲寶:66.82%	中化:70.38%
榮化:66.23%	葡萄王:65.35%	台玻: 64.29%	冠軍: 71.07%	台紙:64.74%
土紙:63.70%	中特鋼:67.55%	中鋼:65.20%	南港:68.07%	泰豐:69.41%
裕隆:70.63%	台船:73.24%	麗正:71.24%	聯電:63.29%	全友:70.63%
台達電:59.54%	金寶:71.29%	台揚:70.05%	鴻海:66.18%	東訊:71.83%
中環:77.93%	光磊:70.47%	聯強:66.04%	精技:69.39%	三商電:71.47%
凌群:76.10%	國產:71.67%	益航:69.82%	長榮:69.52%	山隆:65.34%
萬企:68.83%	華園:66.20%	彰銀:64.20%	京城銀:66.00%	台塑化:62.03%

總準確率: 67.53%

Class 1 precision: 74.93%, recall: 69.62%

Class 2 precision: 63.23%, recall: 76.55%

Class 3 precision: 58.72%, recall: 47.11%

Class 4 precision: 72.47%, recall: 68.31%

這個結果可以發現 class 3 的 precision 跟 recall 比較低，我們認為是 class 3 的資料量相較於其他的 class 少了很多，所以產生了比較低的原因。

結論

依照本次的結果來說，67.53%其實並不是非常準，可能是因為使用的資料量不夠多、原始資料的種類選擇也不夠精確，在做模型訓練之前，其實應該對原始資料的選擇做更深入的研究跟討論，因為能影響股票市場浮動的因素真的太多了，而且由於台股受限於漲跌限制，每天最多變化幅度為正負 10%，這樣的制度對於技術分析來說會更加難以分析，因為資料波動的幅度不是特別的大，所以更難用統計方式做預測。

關於股票分析其實我們認為，單純的使用技術分析方式是很難做出令人滿意的結果的，股票的波動其實不會只有單純的技術統計分析面向，例如一開始我們也有提到基本分析也是非常重要的，但是基本面分析的某些資料是我們一般人比較難掌握跟取得的。日常生活的重要大事件也是以影響整個股票市場的走動，例如國際之間的經濟政策等等，這些面向都應該被納入股票分析的範疇裡，所以未來若要分析股票市場，應該要有更大的團隊，而且整個團隊裡不能只有統計分析的專家，還要廣納各個領域的人才，才可以研究出更透徹精確的股市。