

통계데이터마이닝 - 박창이 교수님

_KOSPI DATA 에 대한 PCA 및 DNN 학습

1. 요약

- A. 프로젝트를 위한 데이터로 KOSPI 주식 데이터를 선택했다. 주식 데이터를 이용해 적절한 회귀모형을 만들기 위해, $a - 14$ 일 ~ $a - 1$ 일 데이터에 대한 Volume 과 전날 대비 주가 데이터를 이용해 a 일의 주가를 predictor 로 이용하는 데이터 구조를 만들고 그 주성분을 추출해 차원을 축소시키기 위해 PCA 를 진행했다. 그후 DNN 회귀 모델을 학습시켜보았다.

2. 데이터 전처리

- A. Python 의 pandas API 를 이용해 yahoo finance server 에서 KOSPI 전체에 대한 일별 데이터를 받았다. 다음 코드는 그 과정의 일부이다.

```
for li in KOSPIList[1:]:
    print(li)
    try:
        tmpDf = listModule.data.DataReader(
            li,
            'yahoo',
            listModule.START
        )
    except listModule._utils.RemoteDataError as RDE:
        print("ERROR with " + li + ": " + RDE)
        continue
    else:
        tmpDf.to_csv(listModule.dataPath + li + '.csv')
        gotList.append(li + '.csv')
```

이에 대한 전체 코드는 [project gitub \[1\]/Data preprocess/getData.py](#) 에서 참조할 수 있다.

- B. 이 데이터를 14 일을 기준으로 Shingling 하고, 각 날짜의 거래 Volume 과 전날대비 주가수치, 그리고 14 일의 다음날의 전날대비 주가수치를 계산했다.

```
def getClosePerOpenAndVol(code, cmpBase):
    tmpKOSPIData = LM.csv.reader(open(LM.dataPath + code))
    tmpValueList = []
    KOSPIDataList = list(tmpKOSPIData)[1:]
    for idx, val in enumerate(KOSPIDataList):
        try:
            if val[1] != '':
                tmpPer = ((float(val[4]) - float(val[1]))/float(val[1]))
                tmp = list()
```

```

tmpI = 0
if (idx - LM.N >= 0):
    tmpI = idx - LM.N
    for i in range(tmpI, idx):
        if KOSPIDataList[i][1] == '':
            break
        # if len(KOSPIDataList[i]) == 7:
        #     tmpWeek =
datetime.datetime.strptime(str(KOSPIDataList[i][0]), "%Y-%m-%d").weekday()
        #     KOSPIDataList[i].insert(1, DAY[tmpWeek])
        tmp.append([100*((float(KOSPIDataList[i][4])
float(KOSPIDataList[i][1]))/float(KOSPIDataList[i][1])), KOSPIDataList[i][6]])
        if len(tmp) == 14:
            tmp.append(100*tmpPer)
            tmp.append(datetime.datetime.strptime(str(val[0]),
"%Y-%m-%d").weekday())
            tmpValueList.append(tmp)
    except ValueError as e:
        print(e)
return tmpValueList

```

이에 대한 전체 코드는 project gitub [1]/Data preprocess/classifyAndSaveModule.py 에서 참조할 수 있다.

- C. R 환경에서 구동가능한 Machine learning library 인 H2O 를 이용해, PCA 를 진행하여 다음과 같이 5 개의 주성분을 얻었다.

MODEL PARAMETERS

OUTPUT

OUTPUT - IMPORTANCE OF COMPONENTS

OUTPUT - SCORING HISTORY FOR GRAMSVD

OUTPUT - TRAINING METRICS

OUTPUT - ROTATION

	pc1	pc2	pc3	pc4	pc5
Day.0	0.0	-0.0	0.0	-0.0	0.0
Day.1	0.0	-0.0	0.0	0.0	0.0
Day.2	0.0	-0.0	0.0	0.0	0.0
Day.3	0.0	0.0	0.0	-0.0	0.0
Day.4	0.0	0.0	0.0	-0.0	0.0
Per1	0.0	-0.0	0.0	0.0	0.0
Vol1	0.2494	-0.2844	0.3389	0.3665	0.3398
Per2	0.0	-0.0	-0.0	-0.0	-0.0
Vol2	0.2583	-0.3287	0.3077	0.2675	0.1189
Per3	0.0	-0.0	-0.0	-0.0	-0.0

Ready

Connections: 0 H2O

Help

Using Flow for the first time?

Quickstart Videos

Or, view example Flows to explore and learn H2O.

STAR H2O ON GITHUB!

Star 2,674

GENERAL

- Flow Web UI ...
- ... Importing Data
- ... Building Models

D. H2O 에서 지원하는 DNN 을 이용해 다음과 같이 학습하였다.

nd_biases":false,"mini_batch_size":1,"elastic_averaging":false}

Started at 4:04:22 pm

Job

Run Time 00:00:09.322

Remaining Time 389:58:36.384

Type Model

Key Q deeplearning-6a77406d-1dd1-4b4c-b8c3-42c98042f04e

Description DeepLearning

Status RUNNING

Progress 1%

Iterations: 2, Epochs: 0.00641438, Speed: 1,520 samples/sec, Estimated time left: 646:15:14.937

Actions View Cancel Job

Ready

Connections: 0 H2O

이때 사용한 네트워크는 maxout with dropout / size: 256*512*128 의 일반적인 DNN 이다.

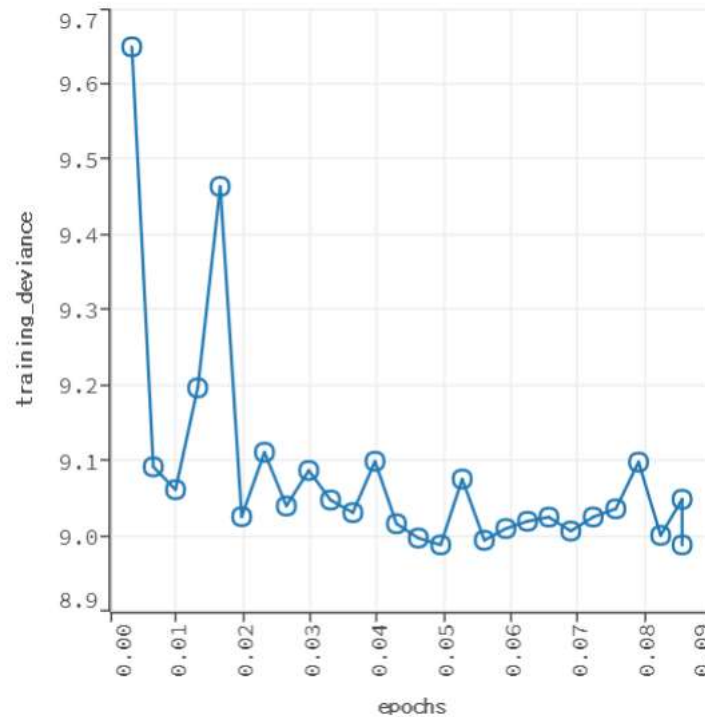
E. 다음은 학습결과 Iteration 에 대한 사진이다.

Date	Time	Duration	Obs/sec	Loss	Metric 1	Metric 2	Metric 3	Metric 4
2017-12-17	16:05:28	1 min 7.634 sec	1615 obs/sec	0.0623	19	53889.0	3.0033	9.0199
2017-12-17	16:05:31	1 min 10.877 sec	1614 obs/sec	0.0655	20	56643.0	3.0042	9.0254
2017-12-17	16:05:34	1 min 14.028 sec	1622 obs/sec	0.0688	21	59476.0	3.0012	9.0070
2017-12-17	16:05:38	1 min 17.152 sec	1629 obs/sec	0.0721	22	62369.0	3.0043	9.0256
2017-12-17	16:05:41	1 min 20.399 sec	1633 obs/sec	0.0755	23	65296.0	3.0061	9.0365
2017-12-17	16:05:44	1 min 23.710 sec	1635 obs/sec	0.0789	24	68230.0	3.0163	9.0983
2017-12-17	16:05:47	1 min 26.884 sec	1639 obs/sec	0.0822	25	71096.0	3.0002	9.0012
2017-12-17	16:05:50	1 min 29.983 sec	1643 obs/sec	0.0854	26	73860.0	3.0082	9.0490

Ready

Connections: 0 H2O

F. 다음은 학습결과 Training deviance-epochs 의 그래프는 다음과 같다.



그래프에서 알 수 있듯이, 학습이 특별한 효과없이 종료되었다. RMSE 또한 3.1 ~ 2.9 를 왔다갔다 하는 형상을 보이며 그 이상의 학습은 진행하지 못했다.

3. 결론

14 일간의 거래량과 전날 대비 주가 데이터를 이용해 다음 날의 주가를 예측하는 DNN 모델을 만들어보았다. KOSPI 데이터를 얻어 원하는 형태로 전처리를 진행한 후, PCA 를 이용해 데이터의 차원을 축소시켰다. 그 결과를 DNN 을 이용해 학습시켰으나 특별한 진전없이 학습이 종료되었다. 이는 애초에 데이터의 x-y 관계가 규칙적이지 않기 때문으로 보인다. 즉 14 일간의 거래량과 전날대비 주가 데이터로는 다음 날의 주가를 예측하기 어려운 것으로 결론지을 수 있다.

4. 참고

[1] Project GITHUB: <https://github.com/johnnyapu15/DataMiningGrad2017Proj>