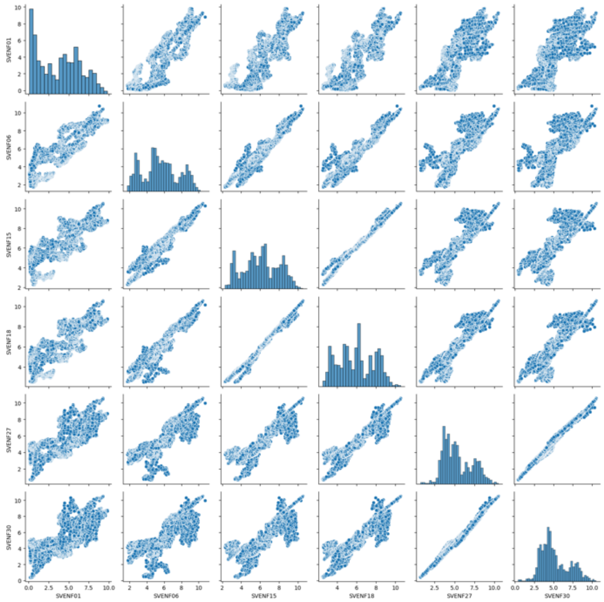
Machine Learning in Finance Lab – Week 05

Liao, Yu-Ching

Part 1. EDA:

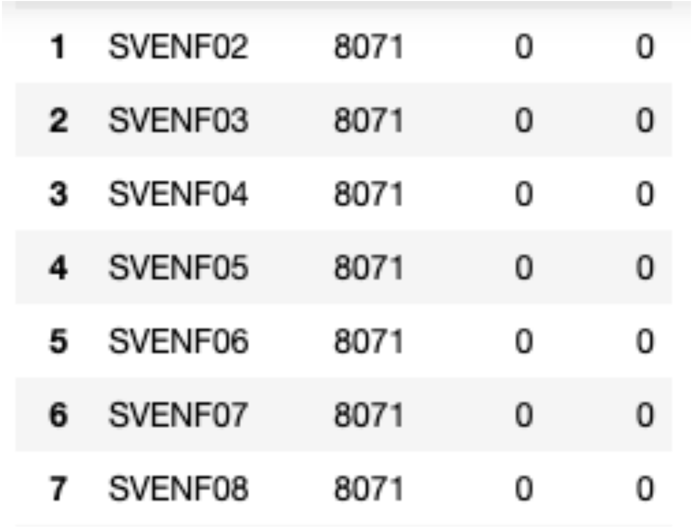
- **Scatter Plot Metrics:**



- **Shape of Data:**

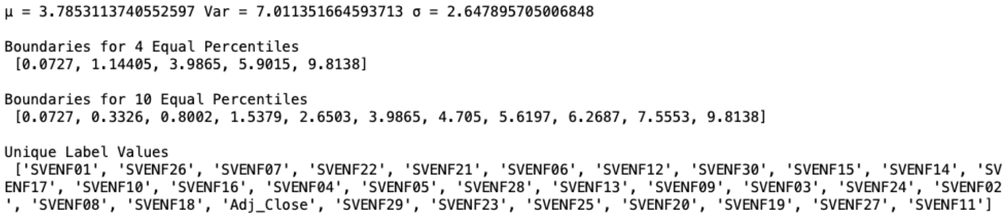
```
The number of Columns is 31 .
The number of Rows is 8071 .
```

- **Nature (truncated):**



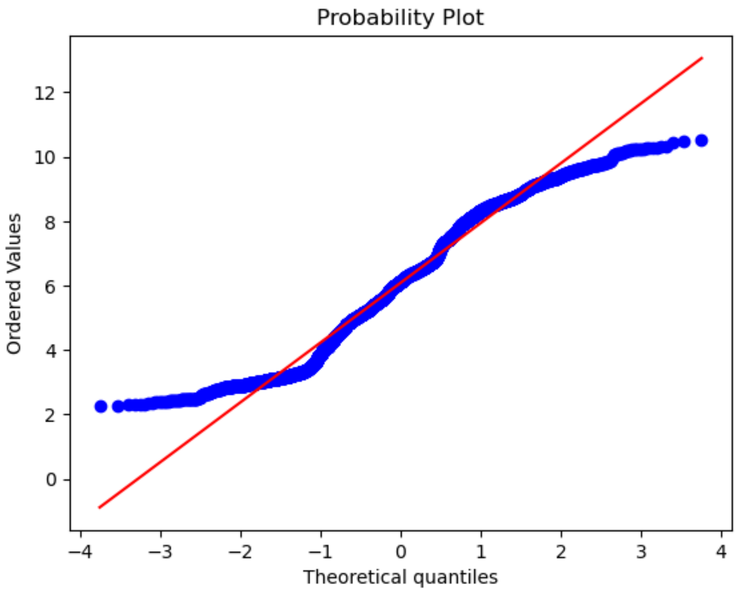
1	SVENF02	8071	0	0
2	SVENF03	8071	0	0
3	SVENF04	8071	0	0
4	SVENF05	8071	0	0
5	SVENF06	8071	0	0
6	SVENF07	8071	0	0
7	SVENF08	8071	0	0

- **Summary of Statistics:**

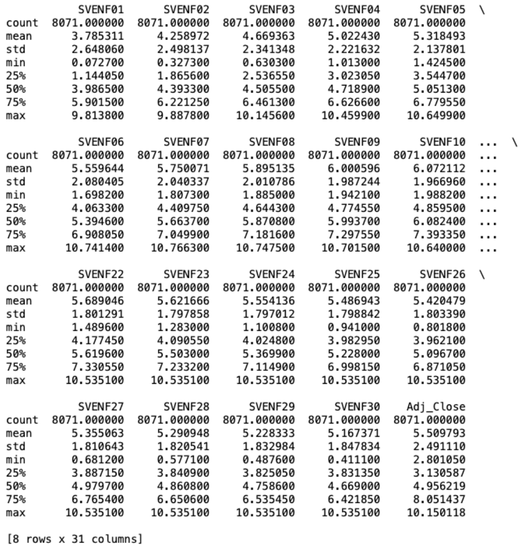


```
μ = 3.7853113740552597 Var = 7.011351664593713 σ = 2.647895705006848

Boundaries for 4 Equal Percentiles
 [0.0727, 1.14405, 3.9865, 5.9015, 9.8138]

Boundaries for 10 Equal Percentiles
 [0.0727, 0.3326, 0.8002, 1.5379, 2.6503, 3.9865, 4.705, 5.6197, 6.2687, 7.5553, 9.8138]

Unique Label Values
 ['SVENF01', 'SVENF26', 'SVENF07', 'SVENF22', 'SVENF21', 'SVENF06', 'SVENF12', 'SVENF30', 'SVENF15', 'SVENF14', 'SV
ENF17', 'SVENF10', 'SVENF16', 'SVENF04', 'SVENF05', 'SVENF28', 'SVENF13', 'SVENF09', 'SVENF03', 'SVENF24', 'SVENF02
', 'SVENF08', 'SVENF18', 'Adj_Close', 'SVENF29', 'SVENF23', 'SVENF25', 'SVENF20', 'SVENF19', 'SVENF27', 'SVENF11']
```

- **QQ Plot:**

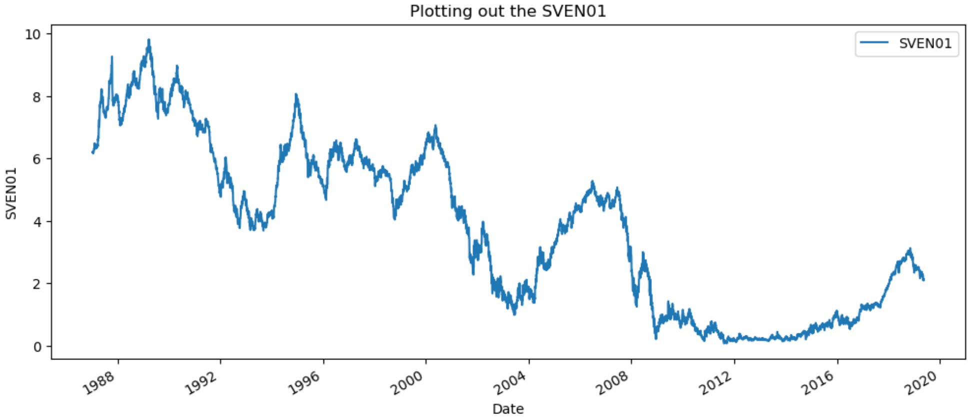


- **Summary of data:**

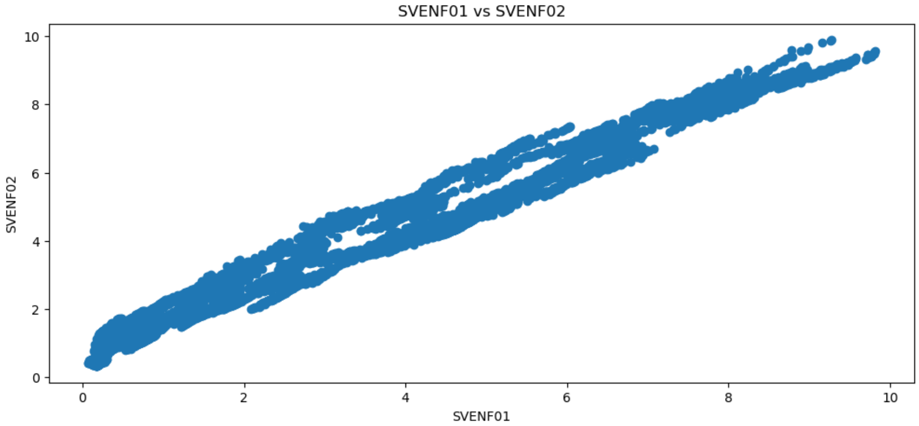


```
           SVENF01      SVENF02      SVENF03      SVENF04      SVENF05  \
count  8071.000000  8071.000000  8071.000000  8071.000000  8071.000000
mean      3.785311     4.258972     4.669363     5.022430     5.318493
std       2.648060     2.498137     2.341348     2.221632     2.137801
min       0.072700     0.327300     0.630300     1.013000     1.424500
25%       1.144050     1.865600     2.536550     3.023050     3.544700
50%       3.986500     4.393300     4.505500     4.718900     5.051300
75%       5.901500     6.221250     6.461300     6.626600     6.779550
max       9.813800     9.887800    10.145600    10.459900    10.649900

           SVENF06      SVENF07      SVENF08      SVENF09      SVENF10  ...  \
count  8071.000000  8071.000000  8071.000000  8071.000000  8071.000000  ...
mean      5.559644     5.750071     5.895135     6.000596     6.072112  ...
std       2.080405     2.040337     2.010786     1.987244     1.966960  ...
min       1.698200     1.807300     1.885000     1.942100     1.988200  ...
25%       4.063300     4.409750     4.644300     4.774550     4.859500  ...
50%       5.394600     5.663700     5.870800     5.993700     6.082400  ...
75%       6.908050     7.049900     7.181600     7.297550     7.393350  ...
max      10.741400    10.766300    10.747500    10.701500    10.640000  ...

           SVENF22      SVENF23      SVENF24      SVENF25      SVENF26  \
count  8071.000000  8071.000000  8071.000000  8071.000000  8071.000000
mean      5.689046     5.621666     5.554136     5.486943     5.420479
std       1.801291     1.797858     1.797012     1.798842     1.803390
min       1.489600     1.283000     1.100800     0.941000     0.801800
25%       4.177450     4.090550     4.024800     3.982950     3.962100
50%       5.619600     5.503000     5.369900     5.228000     5.096700
75%       7.330550     7.233200     7.114900     6.998150     6.871050
max      10.535100    10.535100    10.535100    10.535100    10.535100

           SVENF27      SVENF28      SVENF29      SVENF30    Adj_Close
count  8071.000000  8071.000000  8071.000000  8071.000000  8071.000000
mean      5.355063     5.290948     5.228333     5.167371     5.509793
std       1.810643     1.820541     1.832984     1.847834     2.491110
min       0.681200     0.577100     0.487600     0.411100     2.801050
25%       3.887150     3.840900     3.825050     3.831350     3.130587
50%       4.979700     4.860800     4.758600     4.669000     4.956219
75%       6.765400     6.650600     6.535450     6.421850     8.051437
max      10.535100    10.535100    10.535100    10.535100    10.150118

[8 rows x 31 columns]
```

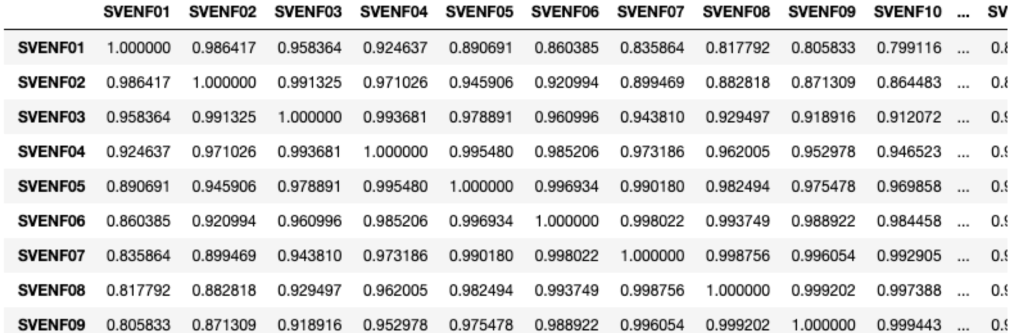
- **Plot out data:**



- **Cross Plotting Pairs:**

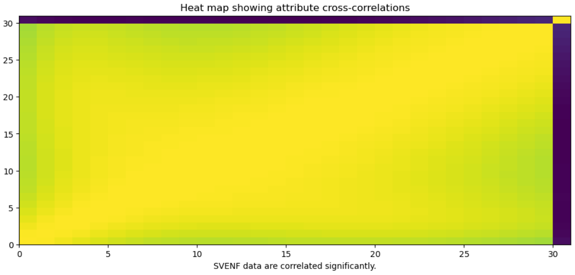


- **Correlations (truncated):**

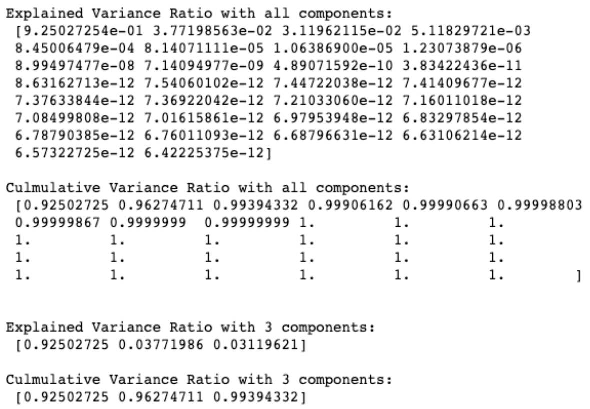


	SVENF01	SVENF02	SVENF03	SVENF04	SVENF05	SVENF06	SVENF07	SVENF08	SVENF09	SVENF10	...	SV
SVENF01	1.000000	0.986417	0.958364	0.924637	0.890691	0.860385	0.835864	0.817792	0.805833	0.799116	...	0.8
SVENF02	0.986417	1.000000	0.991325	0.971026	0.945906	0.920994	0.899469	0.882818	0.871309	0.864483	...	0.8
SVENF03	0.958364	0.991325	1.000000	0.993681	0.978891	0.960996	0.943810	0.929497	0.918916	0.912072	...	0.9
SVENF04	0.924637	0.971026	0.993681	1.000000	0.995480	0.985206	0.973186	0.962005	0.952978	0.946523	...	0.9
SVENF05	0.890691	0.945906	0.978891	0.995480	1.000000	0.996934	0.990180	0.982494	0.975478	0.969858	...	0.9
SVENF06	0.860385	0.920994	0.960996	0.985206	0.996934	1.000000	0.998022	0.993749	0.988922	0.984458	...	0.9
SVENF07	0.835864	0.899469	0.943810	0.973186	0.990180	0.998022	1.000000	0.998756	0.996054	0.992905	...	0.9
SVENF08	0.817792	0.882818	0.929497	0.962005	0.982494	0.993749	0.998756	1.000000	0.999202	0.997388	...	0.9
SVENF09	0.805833	0.871309	0.918916	0.952978	0.975478	0.988922	0.996054	0.999202	1.000000	0.999443	...	0.9

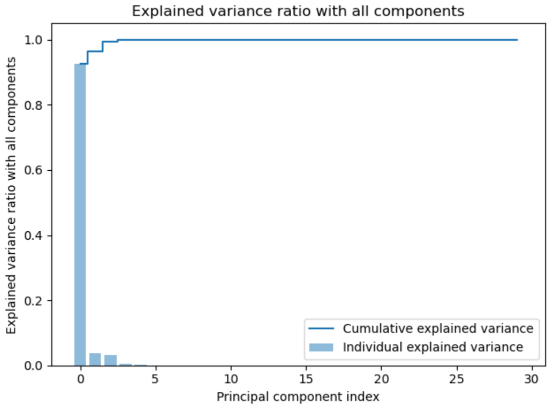
- **Correlations Visualization:**

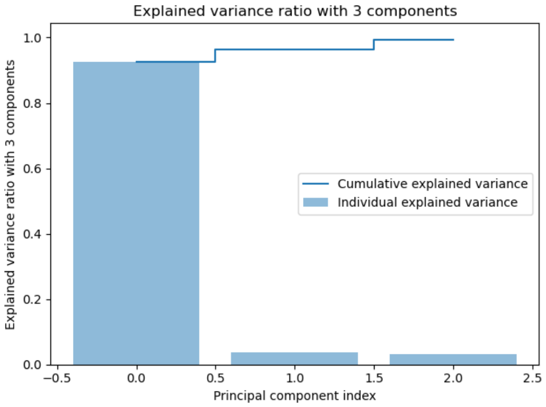


Part 2. PCA:



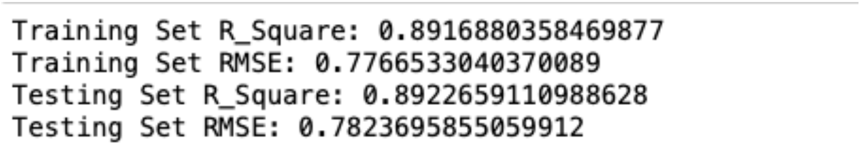
```
Explained Variance Ratio with all components:
 [9.25027254e-01 3.77198563e-02 3.11962115e-02 5.11829721e-03
 8.45006479e-04 8.14071111e-05 1.06386900e-05 1.23073879e-06
 8.99497477e-08 7.14094977e-09 4.89071592e-10 3.83422436e-11
 8.63162713e-12 7.54060102e-12 7.44722038e-12 7.41409677e-12
 7.37633844e-12 7.36922042e-12 7.21033060e-12 7.16011018e-12
 7.08499808e-12 7.01615861e-12 6.97953948e-12 6.83297854e-12
 6.78790385e-12 6.76011093e-12 6.68796631e-12 6.63106214e-12
 6.57322725e-12 6.42225375e-12]

Culmulative Variance Ratio with all components:
 [0.92502725 0.96274711 0.99394332 0.99906162 0.99990663 0.99998803
 0.99999867 0.9999999  0.99999999 1.         1.         1.
 1.         1.         1.         1.         1.         1.
 1.         1.         1.         1.         1.         1.
 1.         1.         1.         1.         1.         1.        ]


Explained Variance Ratio with 3 components:
 [0.92502725 0.03771986 0.03119621]

Culmulative Variance Ratio with 3 components:
 [0.92502725 0.96274711 0.99394332]
```



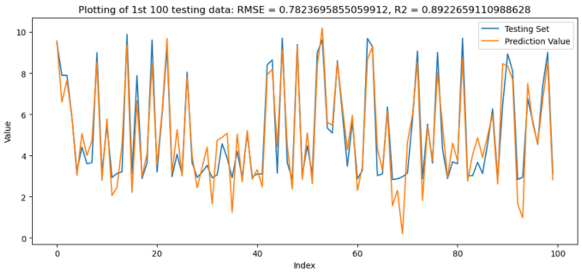


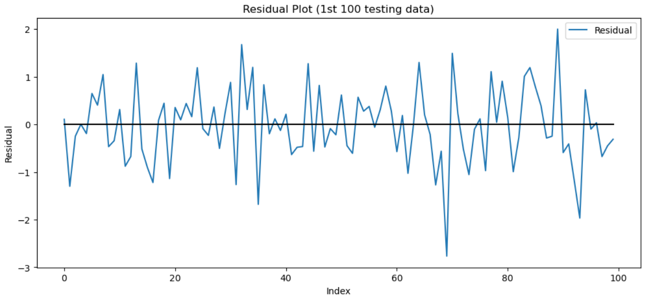
Part 3. Linear Regression VS SVM:

- **Linear Regression with all features:**

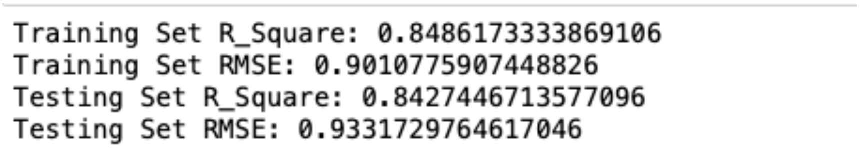


```
Training Set R_Square: 0.8916880358469877
Training Set RMSE: 0.7766533040370089
Testing Set R_Square: 0.8922659110988628
Testing Set RMSE: 0.7823695855059912
```

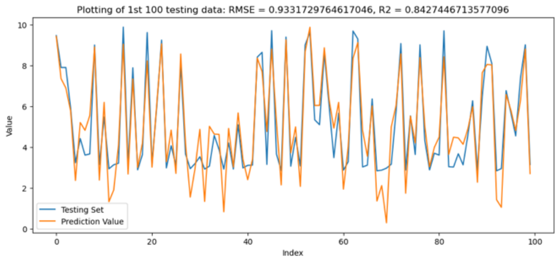


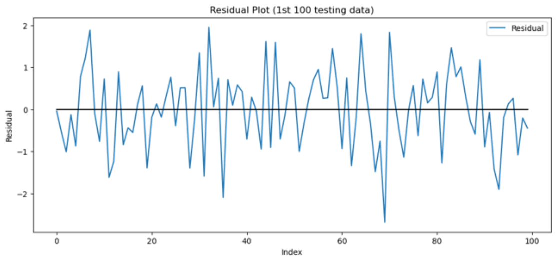


- **Linear Regression with PCA:**

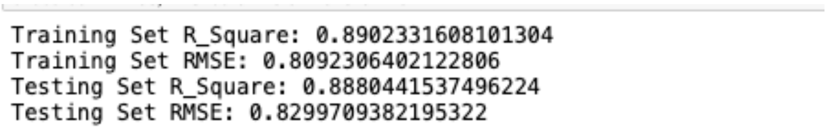


```
Training Set R_Square: 0.8486173333869106
Training Set RMSE: 0.9010775907448826
Testing Set R_Square: 0.8427446713577096
Testing Set RMSE: 0.9331729764617046
```

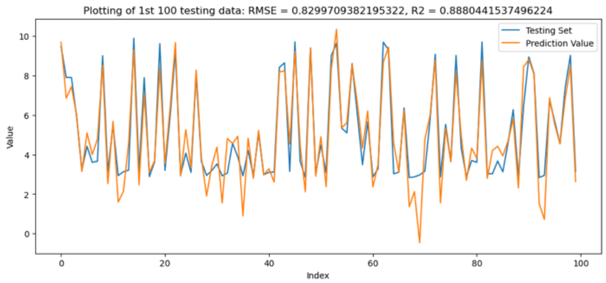


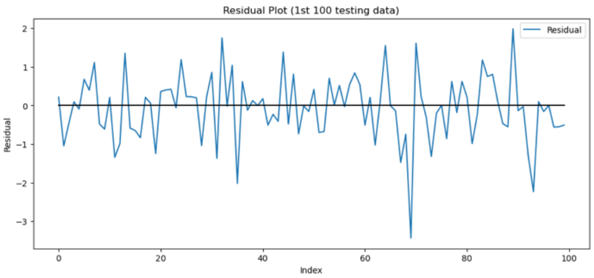


- **SVM with all features:**

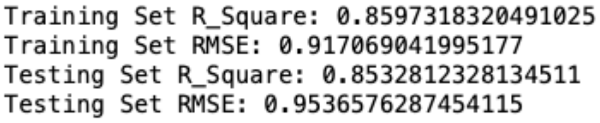


```
Training Set R_Square: 0.8902331608101304
Training Set RMSE: 0.8092306402122806
Testing Set R_Square: 0.8880441537496224
Testing Set RMSE: 0.8299709382195322
```

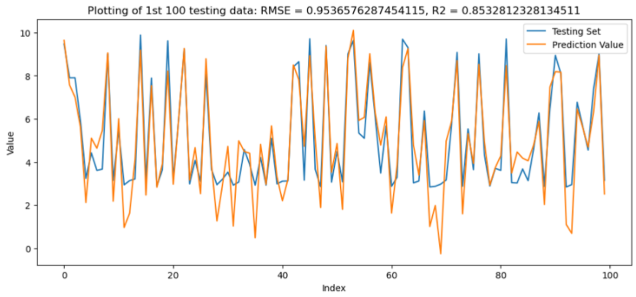


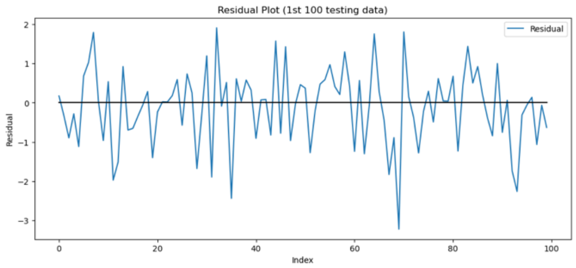


- **SVM with PCA:**



```
Training Set R_Square: 0.8597318320491025
Training Set RMSE: 0.917069041995177
Testing Set R_Square: 0.8532812328134511
Testing Set RMSE: 0.9536576287454115
```





Part 4. Conclusion:

From the R_square and RMSE, we can notice that, with PCA, the **performance of the model might decrease**. In this case, the accuracy does not decrease significantly, means that most of the signal has been retained. However, **there is an improvement in their learning time**. This might not be significant in our case, however, can be significant if we have large number of features.

And to compare Linear Regression and SVM, we can notice that **none of their performance is significantly better**. However, there is **large gap in their learning times (as shown below)**. As a result, I would still use linear regression if there were no necessity to use SVM.

In conclusion, whether to implement PCA depends on what we are expecting for. If we are only expecting for the approximate but faster outcome, PCA can be even helpful. However, if we yearn to get the exact outcome rather than approximate, we should not implement PCA since it is highly possible that it will reduce its accuracy in substitute of minimizing the calculation time.

	Linear Reg. (all)	Linear Reg. (PCA)	SVM (all)	SVM (PCA)
R_Square (Train)	0.89	0.84	0.88	0.85
R_Square (Test)	0.89	0.84	0.89	0.85
RMSE (Train)	0.77	0.90	0.81	0.91
RMSE (Test)	0.78	0.92	0.82	0.95
Times	323ms	282ms	12.9s	10.9s

Part 5. Appendix:

Github:https://github.com/yu7yu7/IE517_Machine-Learning-in-Finance-Lab/blob/main/IE517_SP23_HW5/ML_Week05_HW.ipynb

Part 6. Sign:

My name is Yu-Ching Liao.

My NetID is ycliao3.

I hereby certify that I have read the University policy on Academic Integrity and that I am not in violation.