OTTO VON GUERICKE UNIVERSITÄT MAGDEBURG | INF | FAKULTÄT FÜR INFORMATIK

Understanding Deep Learning on stock data

Presented by:
Saiyudh Mannan - 225982
Gracy Joseph - 225968

DKE| Fakultät für Informatik
Mathematics and Numeric of Deep Neural Networks for Physical Simulations
Otto-von-Guericke-Universität Magdeburg

Recent Works & Study

Recent works:

- Combining the desirable aspects of RNN and CNN architectures
- Convolutional LSTM[1], Quasi-RNN model[6], dilated RNN[7]
- Multiple thorough evaluations of RNN architectures on representative sequence modeling tasks[8]

Current paper:

Focussed on a comparison of generic convolutional (TCN)[2] and recurrent architectures (LSTM and GRU)[2]

Time series Sequential model

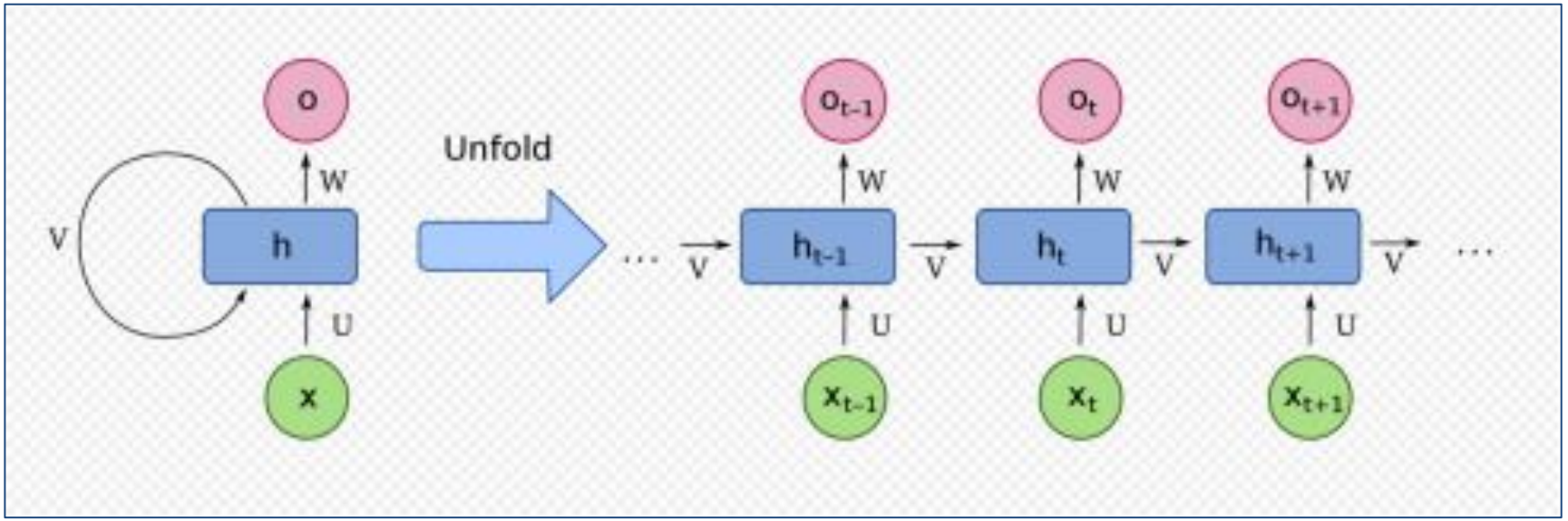
- Deep learning models that input or output sequences of data
- Time series observation from the sequence of discrete-time of successive intervals
- Seq models have to deal with input and output sequences of different lengths

Applications of Sequence Models

1. Stock Data Analysis
2. Trajectories of physical systems
3. Forecasting

Recurrent Neural Network

- variant of the conventional feedforward artificial neural networks
- work for time series data or data that involves sequences
- has theoretical ability to retain information through sequences of unlimited length



- RNNs have feedback in the hidden layer
- during the next iteration, the data from the previous layer is also included in the computation

Source: https://commons.wikimedia.org/wiki/File:Recurrent_neural_network_unfold.svg

Recurrent Neural Network on sequential data[2]

- specialized for sequential data
- used in the field of Natural Language Processing
- used in time series predictions
- takes historical information for computation
- backpropagation is done at each point in time

LSTM (Long Short-Term Memory)[9]

- to overcome vanishing gradient problem
- can capture long-range dependencies
- gating mechanism allowed the network to learn the conditions

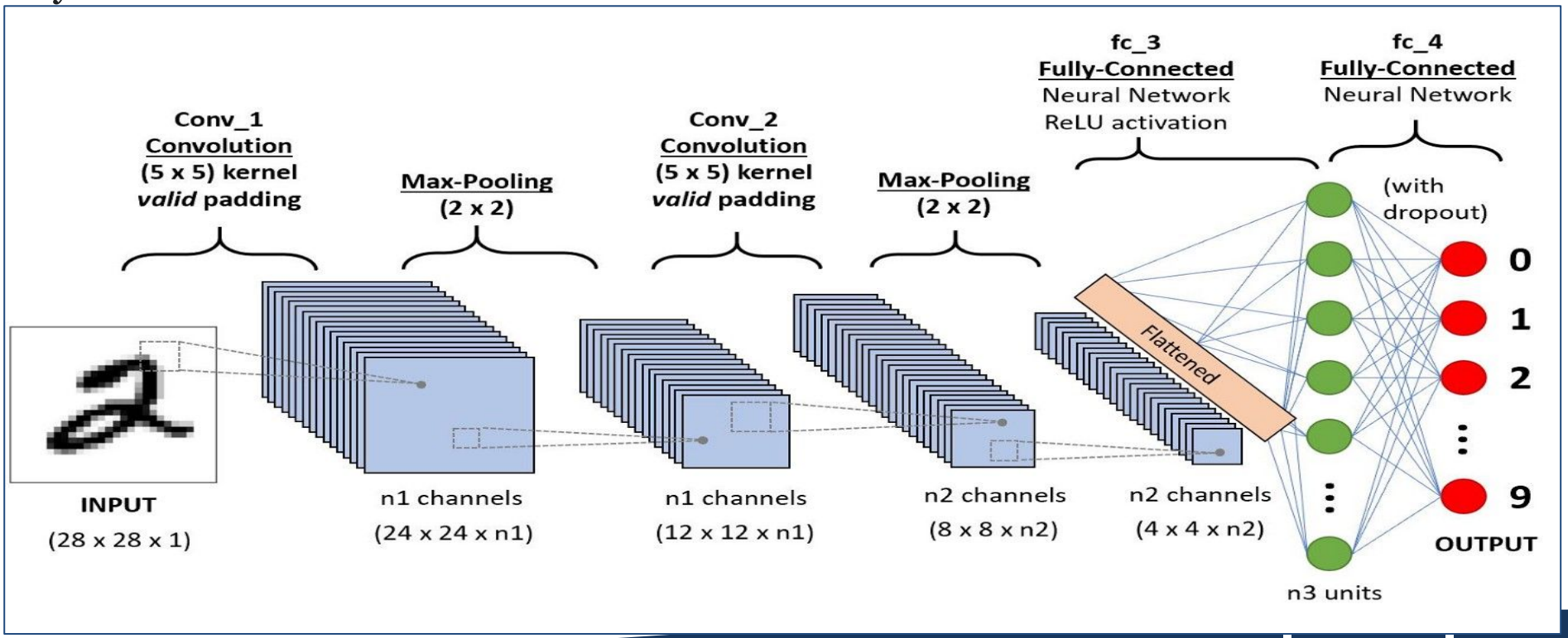
e.g.: Apple's Siri and Google's voice search

How does LSTM work:

- a cell state is passed to the next time step
- the data from the previous layer is also included in the omputation

Convolutional Neural Network

- formed by a stack of distinct layers that transform the input volume into an output volume
- applied to analyze visual imagery
- connectivity pattern between neurons resembles the organization of the animal visual cortex
- use convolution in place of general matrix multiplication in at least one of their layers



Source: https://www.analyticsvidhya.com/blog/2020/10/what-is-the-convolutional-neural-network-architecture/

Convolutional Network on sequential data[2]

- can reach state-of-the-art accuracy
- has a backpropagation path different from the temporal direction of the sequence

TCN

- memory characteristics: Longer memory than LSTM
- more accuracy
- comparatively easier to train
- the TCN is causal (no information leakage from the future to the past)
- can map any sequence to an output sequence of the same length

Background

Paper:

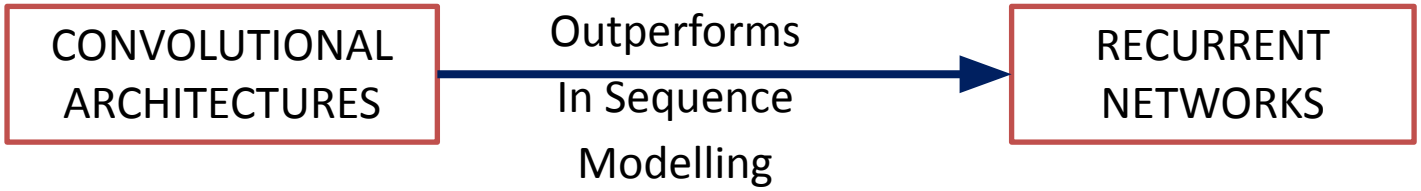
An Empirical Evaluation of Generic Convolutional and Recurrent Networks for Sequence Modeling[2]

(Authors: Shaojie Bai, J. Zico Kolter, Vladlen Koltun)

- Sequence modeling focusses on recurrent networks
- Conducted a systematic evaluation of generic convolutional and recurrent architectures for sequence modeling
- Architecture temporal convolutional network (TCN) was compared to LSTMs and GRUs
- Model evaluation across a broad range of standard tasks

Paper:

An Empirical Evaluation of Generic Convolutional and Recurrent Networks for Sequence Modeling



- gating mechanisms or skip connections were not added to either TCNs or RNNs.
- standard regularizations used

Synopsis

TCN architecture with minimal tuning outperforms canonical recurrent architectures

Sequence Modeling Task	Model Size (≈)	Models			
		LSTM	GRU	RNN	TCN
Seq. MNIST (accuracy^h)	70K	87.2	96.2	21.5	**99.0**
Permuted MNIST (accuracy)	70K	85.7	87.3	25.3	**97.2**
Adding problem T=600 (loss$^\ell$)	70K	0.164	**5.3e-5**	0.177	**5.8e-5**
Copy memory T=1000 (loss)	16K	0.0204	0.0197	0.0202	**3.5e-5**
Music JSB Chorales (loss)	300K	8.45	8.43	8.91	**8.10**
Music Nottingham (loss)	1M	3.29	3.46	4.05	**3.07**
Word-level PTB (perplexity$^\ell$)	13M	**78.93**	92.48	114.50	88.68
Word-level Wiki-103 (perplexity)	-	48.4	-	-	**45.19**
Word-level LAMBADA (perplexity)	-	4186	-	14725	**1279**
Char-level PTB (bpc$^\ell$)	3M	1.36	1.37	1.48	**1.31**
Char-level text8 (bpc)	5M	1.50	1.53	1.69	**1.45**

Which architecture should one use?

	RNNs **LSTM and GRU**	**CNNs** **TCN**
Parallelism	sequentially	long input sequence can be processed as a whole
Flexible receptive field size	not as easier as TCN to adapt to different domains	better control of the model's memory size
Stable gradients	major issue of vanishing gradients	avoids the problem of exploding/vanishing gradients
Low memory requirement for training	can use up a lot of memory to store the partial results for their multiple cell gates	less memory usage
Variable length inputs	sequential data of arbitrary length	take in inputs of arbitrary lengths

Dataset

CNNpred: CNN-based stock market prediction using a diverse set of variables Data Set[5]

Data Set Characteristics:	**Sequential, Time-Series**
Number of Instances:	**8915**
Number of Attributes:	**82**

Features:

S&P 500, NASDAQ Composite, Dow Jones Industrial Average, RUSSELL 2000, and NYSE

Preprocessing and Design choices

Missing values:

- 37.5% of instances in dataset had missing values
- forward fill
- zero fill

Feature Scaling:

- Normalized 82 features
- Standardized the features following Hoseinzade and Haratizadeh [3,4]

Feature Set:

- We considered three different feature sets: a *full* feature set with 82 total features
- a reduced *PCA* feature set comprising 30 principal components explaining 90.7% variance of the original dataset

Preprocessing and Design choices [continued]

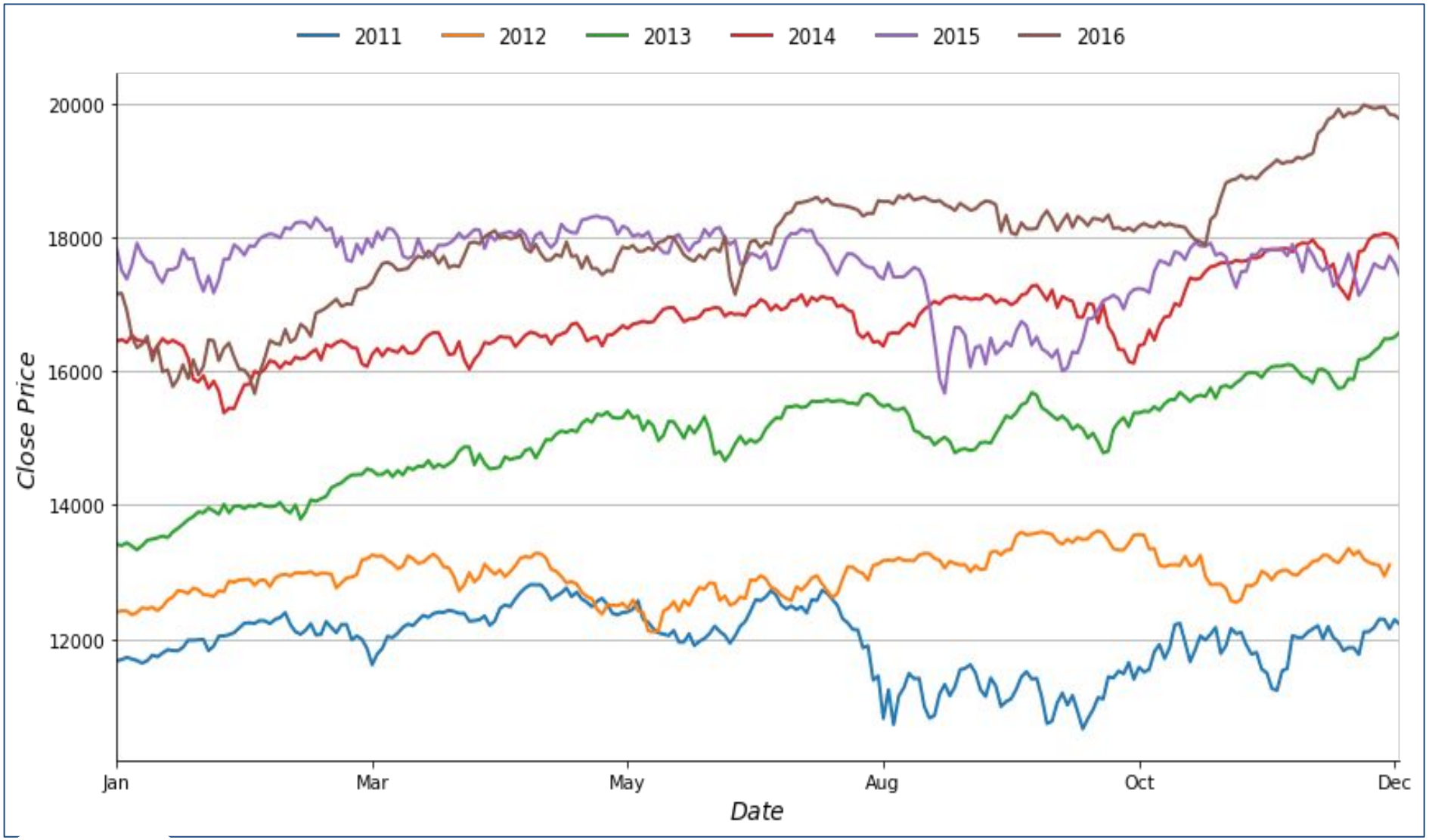
Data Split:

- Cross Validation 80:10:10 [Train: Test: Validation]
- [1395: 173 : 156 instances]
- For 2D ,3D CNN we did not shuffle but aggregated the whole date to avoid cheating(using future data or same data from different markets to estimate predictions in another dataset or market.)

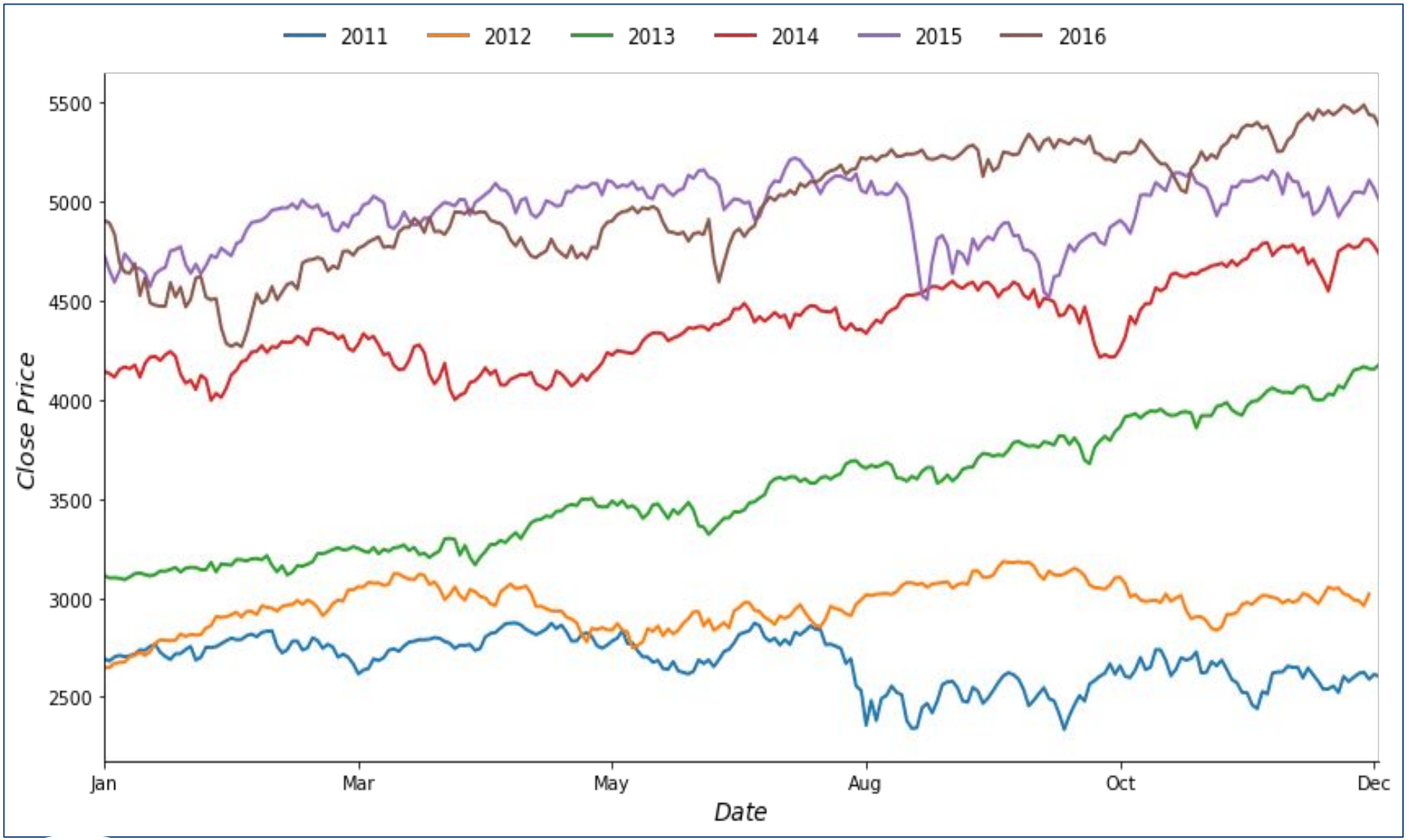
Evaluation metrics:

- Accuracy
- macro-average F1

Time series data exploration based on close price vs year[DJI]

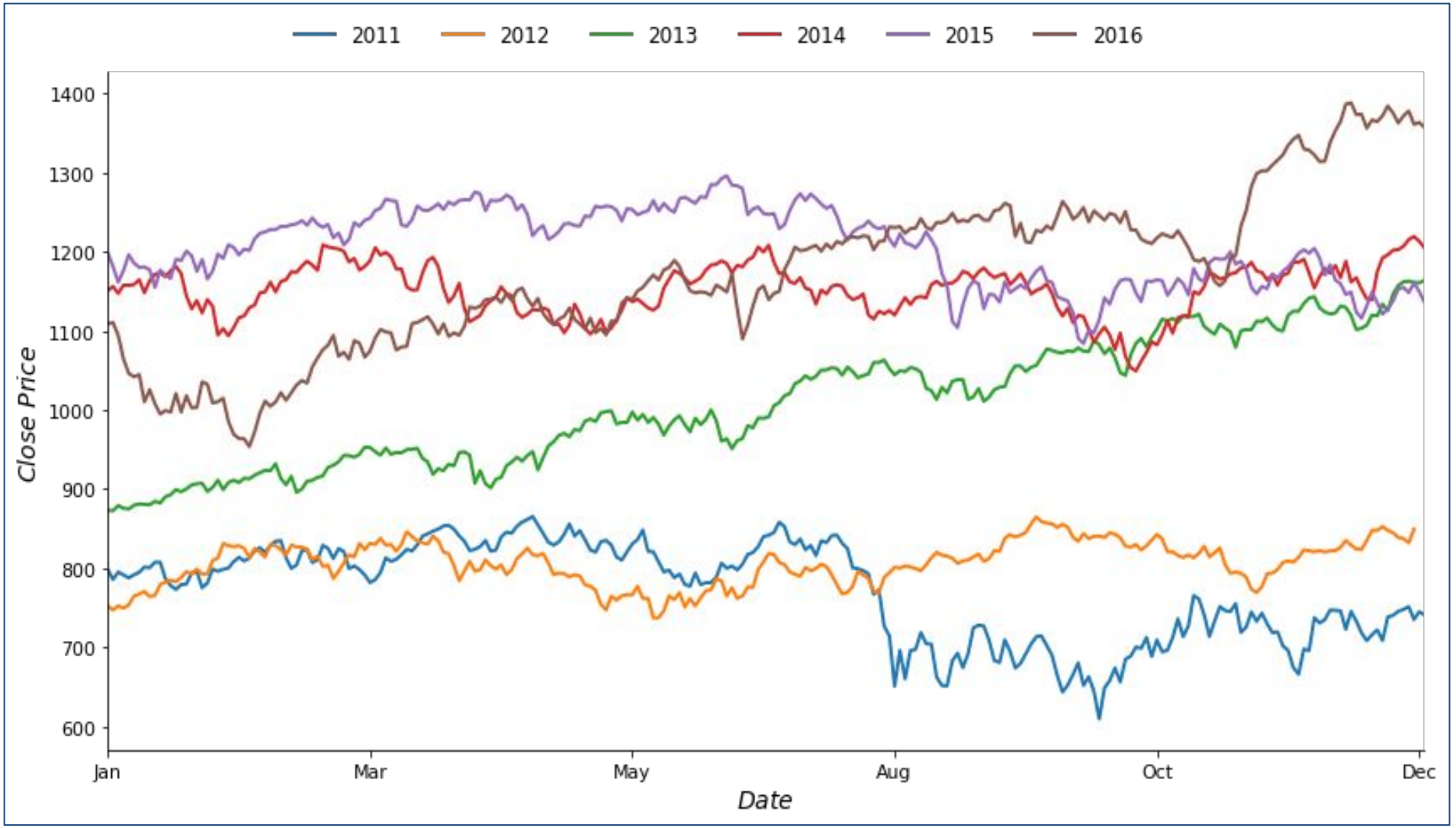


Time series data exploration based on close price vs year[NASDAQ]

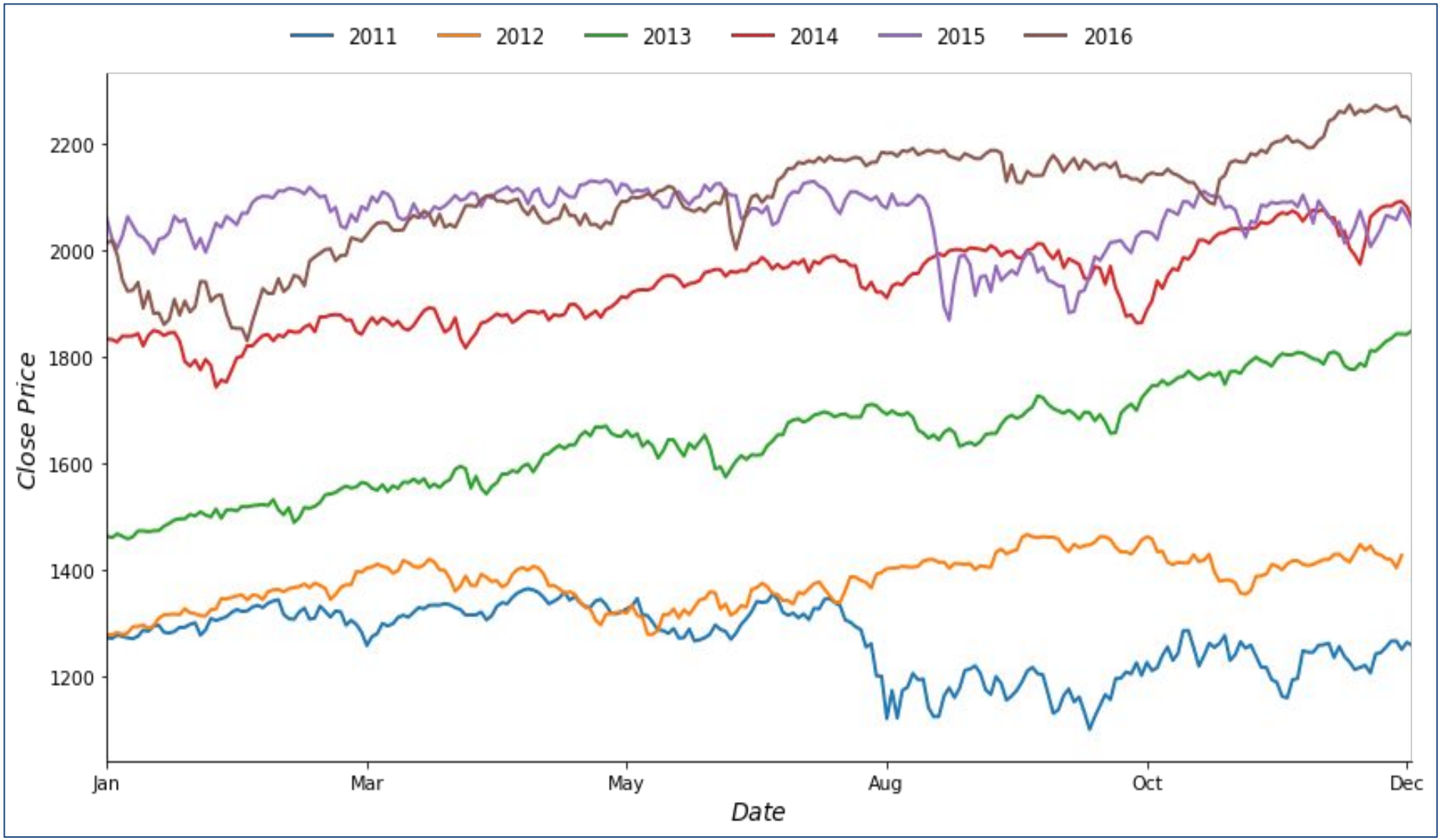


Time series data exploration based on close price vs year[NYSE]

Time series data exploration based on close price vs year[Russel]

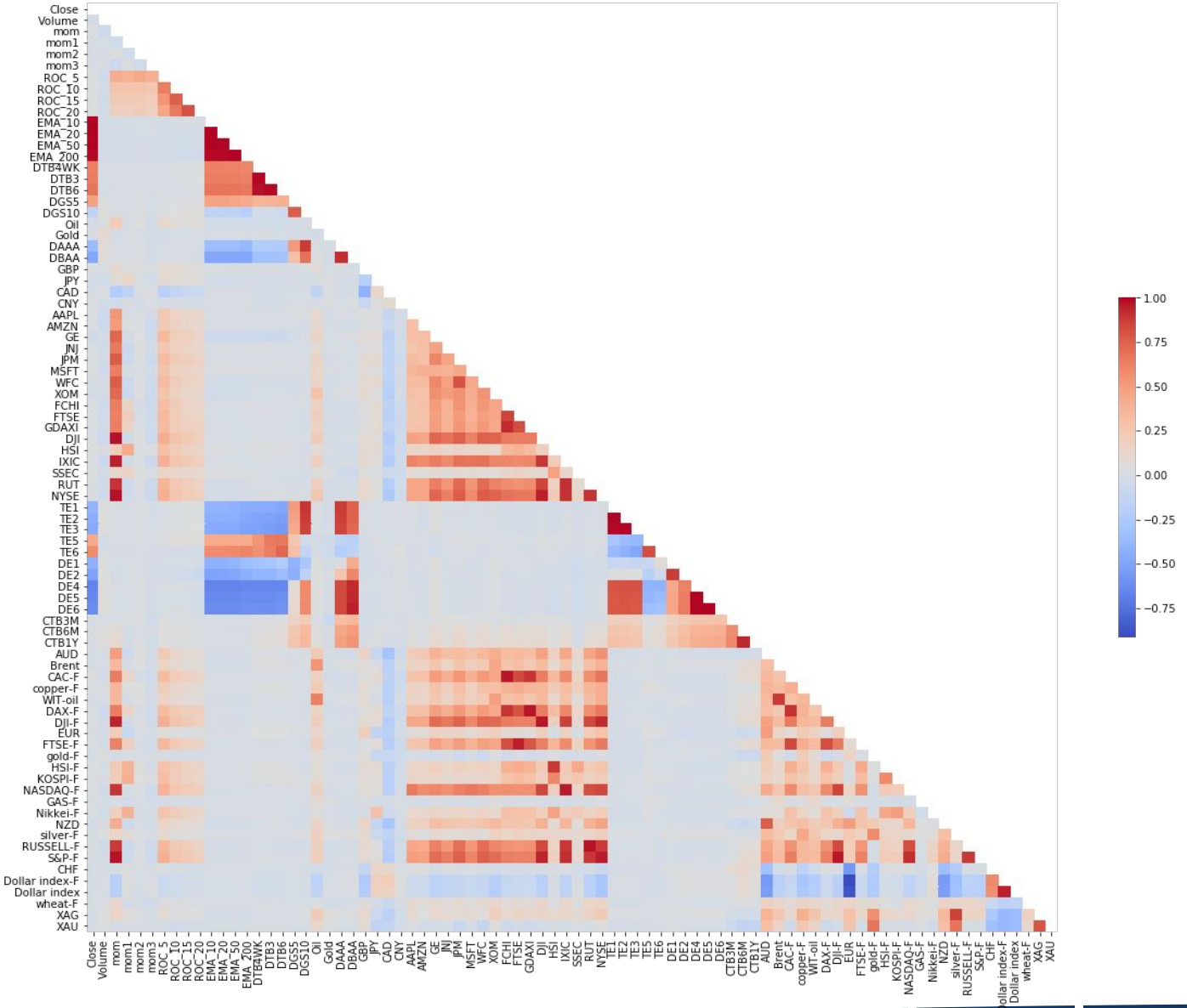


Time series data exploration based on close price vs year[S&P 500]

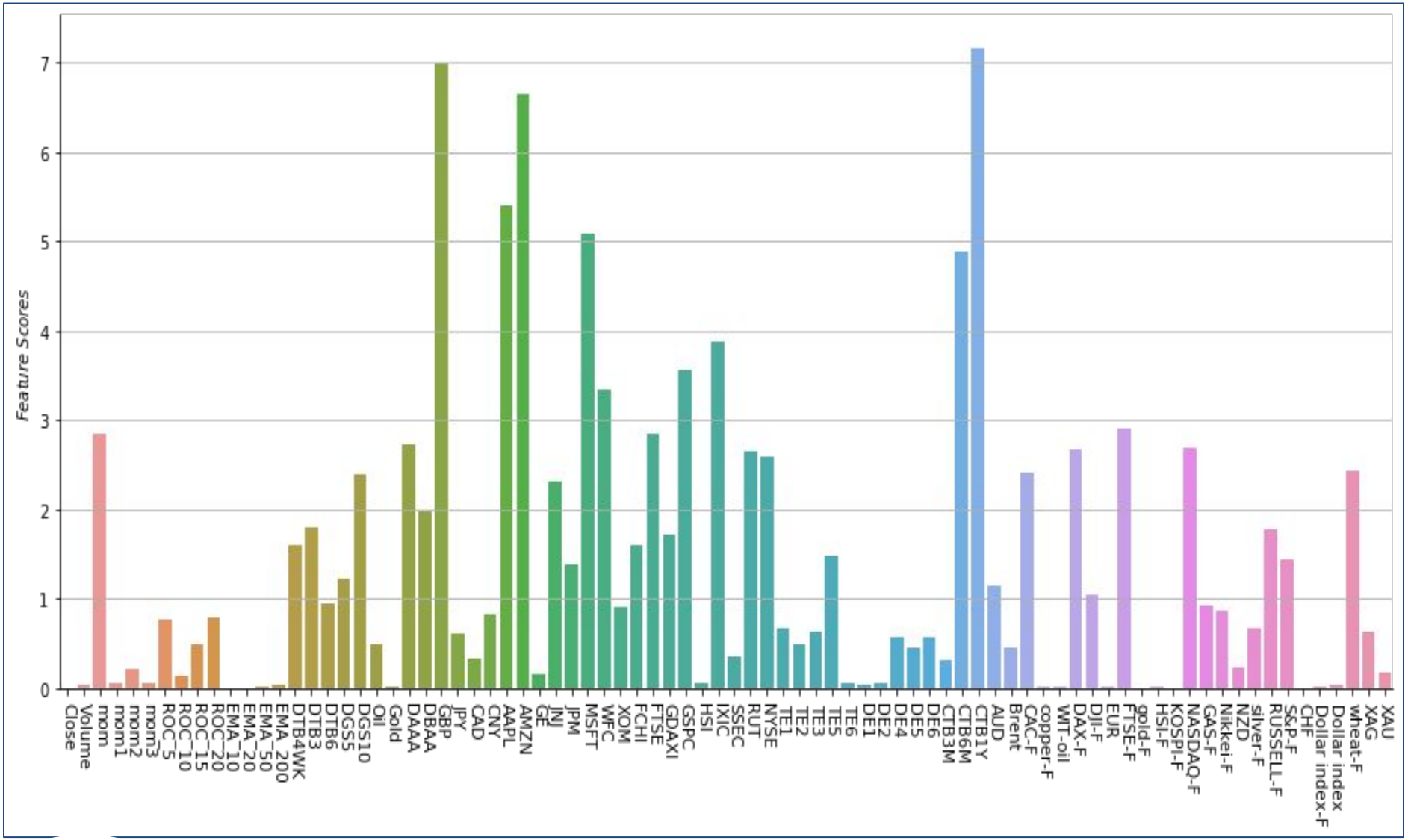


Correlation matrix of features

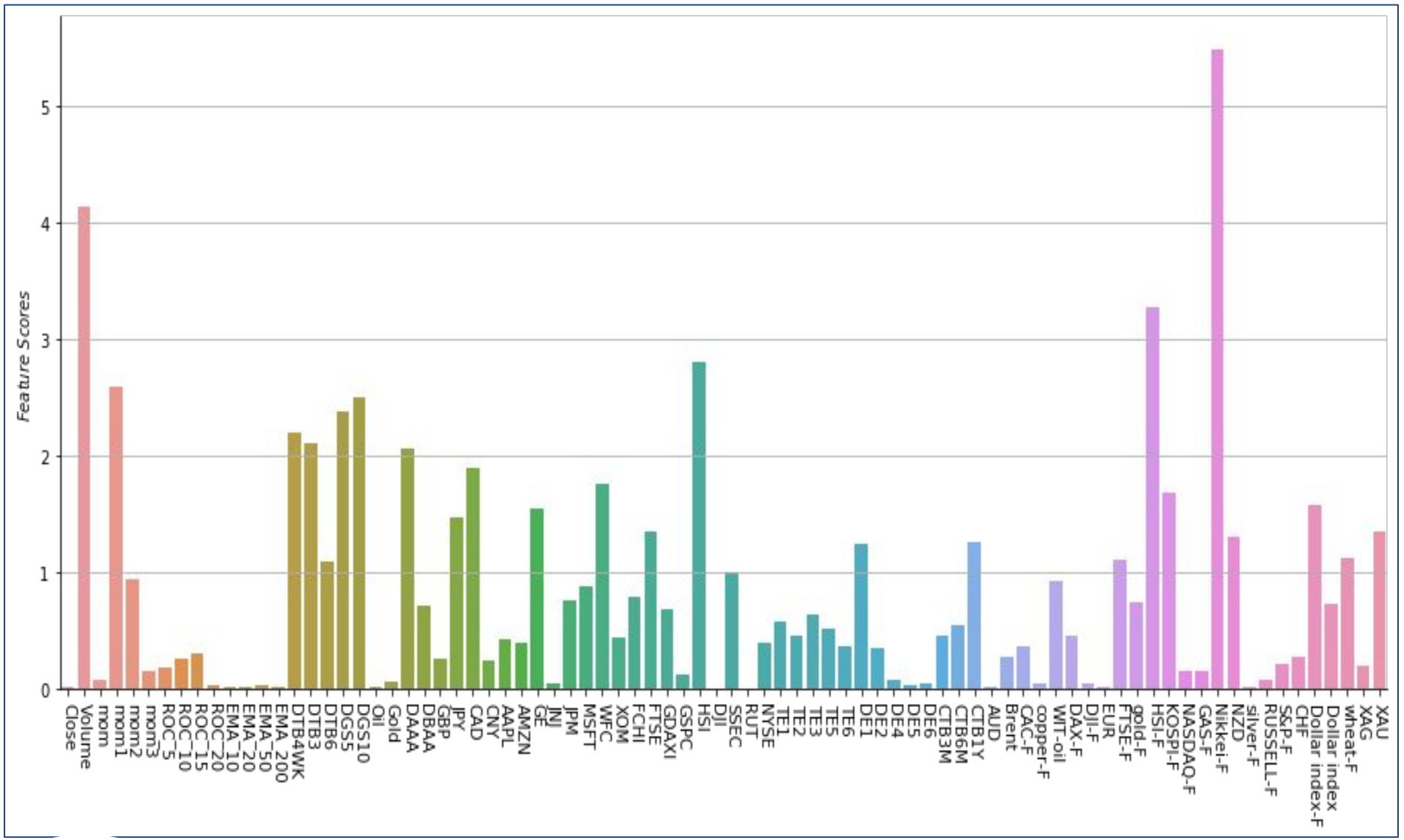
1. DJI,
2. NASDAQ
3. RUSSEL
4. S&P 500
5. NYSE



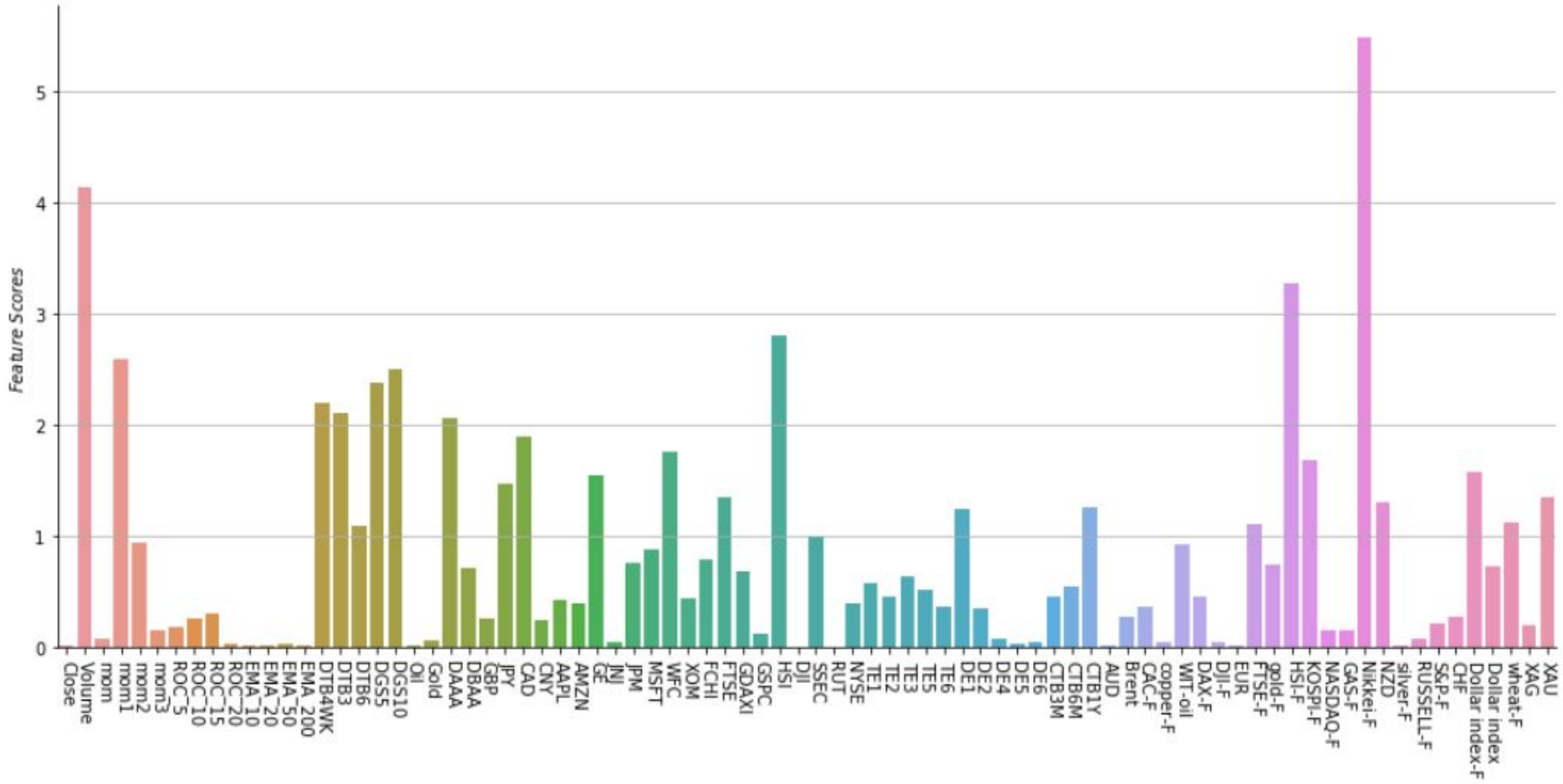
Feature Selection w.r.t to labels[DJI]



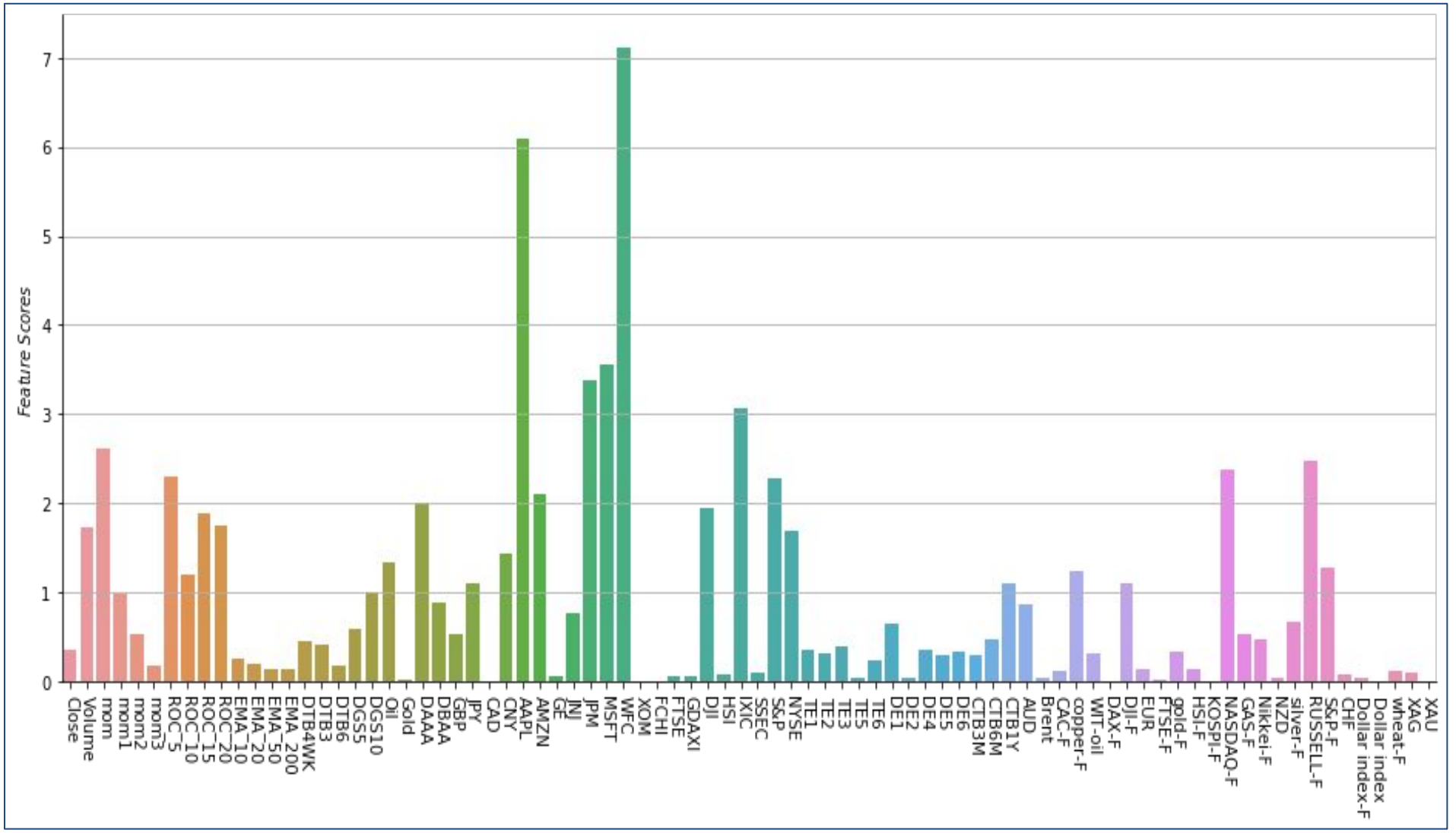
Feature Selection w.r.t to labels[NASDAQ]



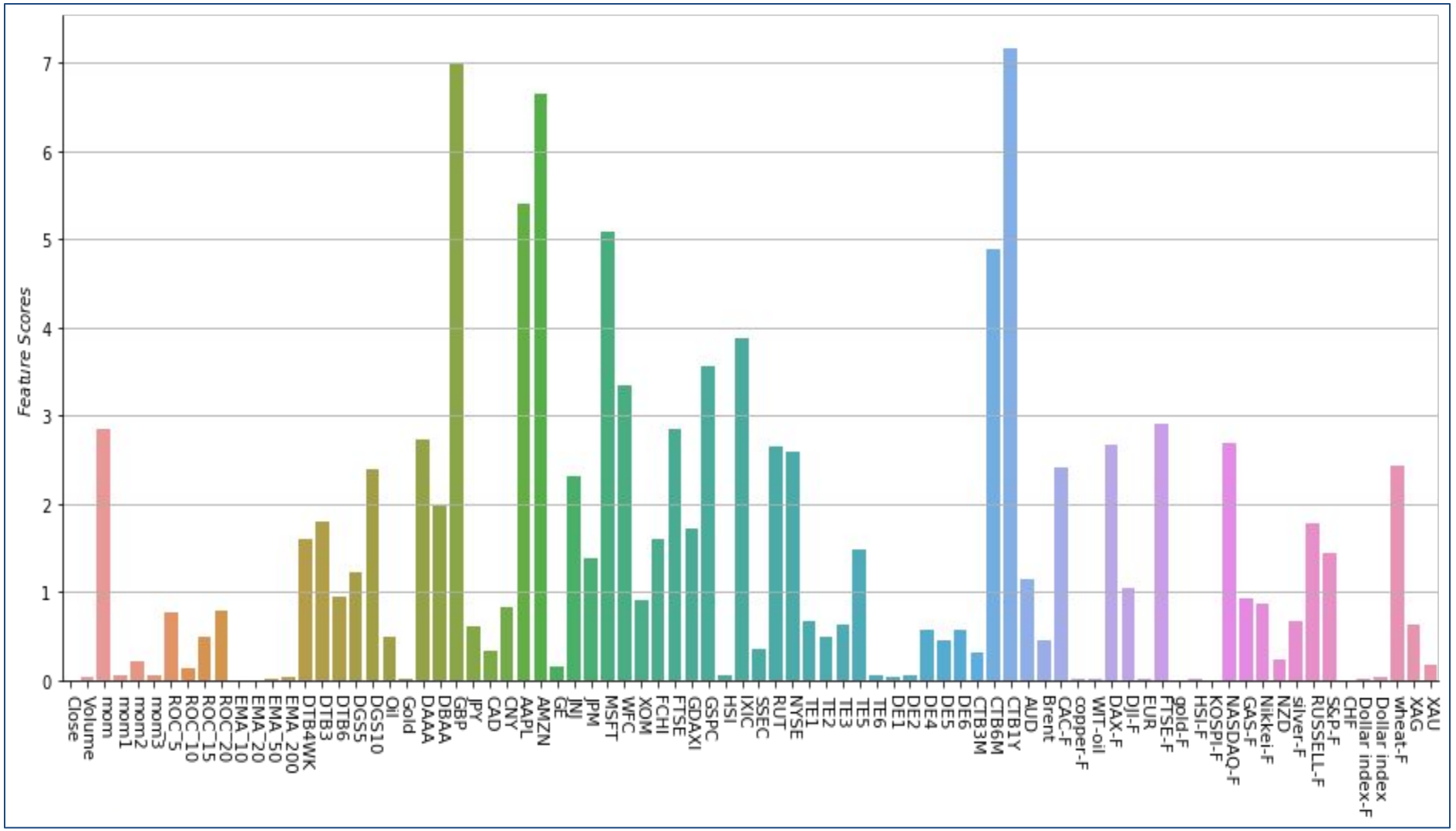
Feature Selection w.r.t to labels[NYSE]



Feature Selection w.r.t to labels[Russel]



Feature Selection w.r.t to labels[S&P 500]



NN Descriptions in brief

1) **Feedforward NN**
   - two hidden layers 64 and 32 respectively with RELU
   - ending with sigmoid layer for prediction

2) 2D- CNN
   - Conv Layer with filter size x number of features and RELU
   - 2 Conv Layer with RELU
   - max pooling of 3x1 and 2x1 used respectively
   - ending with sigmoid layer for prediction

3) 3D- CNN
   - Conv Layer with 1x1 filter and RELU
   - Conv Layer x number of markets and RELU
   - max pool 2x1 Conv Layer with filter 3x1, max pool 2x1
   - ending with a sigmoid layer for prediction

NN Descriptions in brief [continued]

4) LSTM

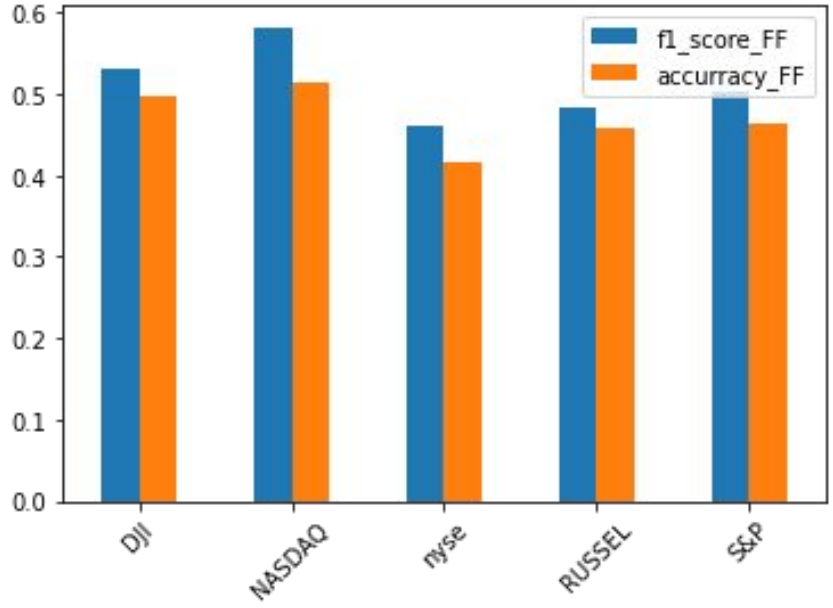
- two LSTM layers with 128 and 64 neurons with leaky RELU
- ending with sigmoid for prediction

5) TCN

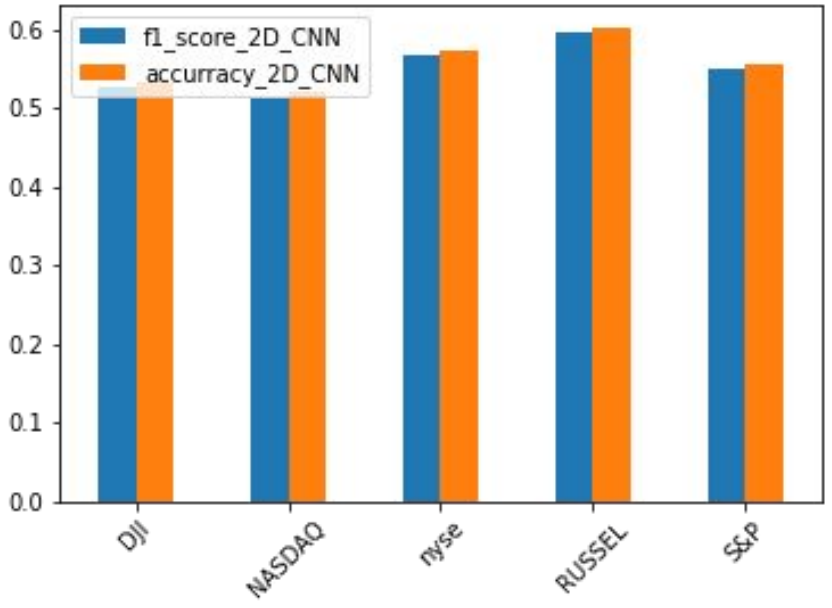
- Input followed by custom TCN from TCN library
- ending with sigmoid for prediction

F1 Measure and Accuracy [Feedforward NN,2D CNN,LSTM,TCN]

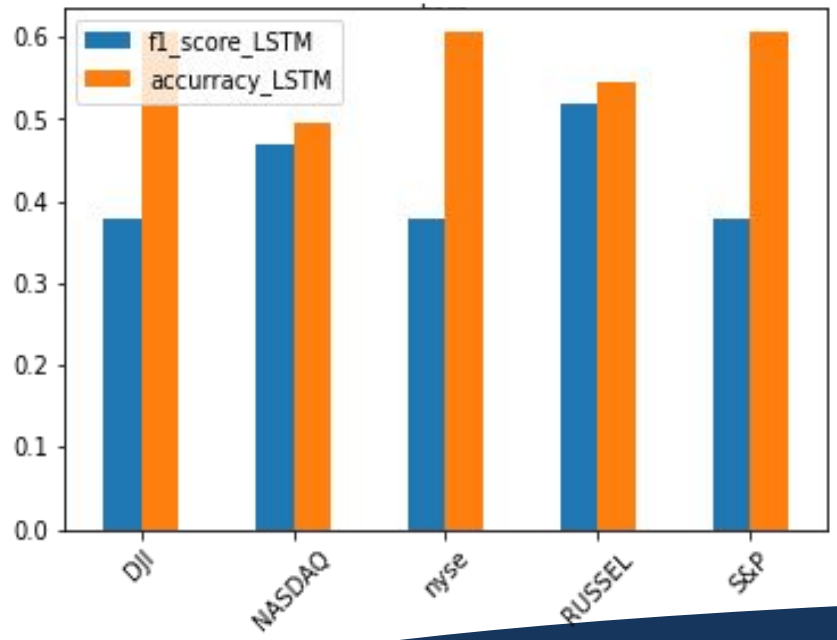
Feedforward NN



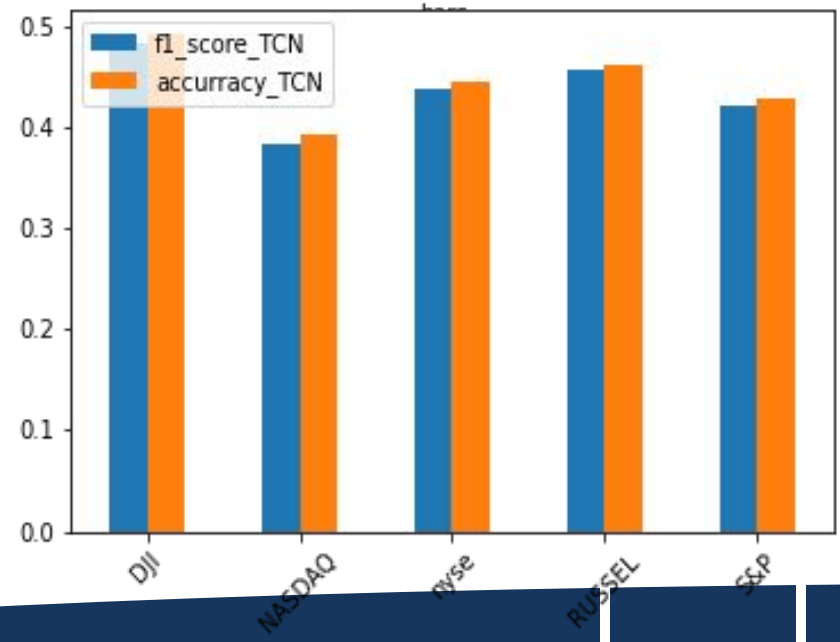
2D CNN



LSTM



TCN



Evaluation over F1 measure and Accuracy

We observe that TCN performed better than Feedforward and LSTM but got outperformed by 2D_CNN.

some possible reasons could be,

1) lack of randomization
2) low number of epochs
3) model not complex enough as we are using the most basic of models

Conclusion

The experimental results indicated that TCN models substantially was outperformed by 2D_CNN but performed better than Feedforward and LSTM.

A simple convolutional architecture is more effective across sequence modeling on stock data than recurrent architectures.

References

1. Shi, Xingjian, Chen, Zhourong, Wang, Hao, Yeung, Dit-Yan, Wong, Wai-Kin, and Woo, Wang-chun. Convolutional LSTM network: A machine learning approach for precipitation nowcasting. In NIPS, 2015
2. S. Bai, J. Z. Kolter, and V. Koltun. An empirical evaluation of generic convolutional and recurrent networks for sequence modeling. arXiv preprint arXiv:1803.01271, 2018
3. Ehsan Hoseinzade and Saman Haratizadeh. Cnnpred: Cnn-based stock market prediction using several data sources. *arXiv preprint arXiv:1810.08923*, 2018.
4. Ehsan Hoseinzade and Saman Haratizadeh. Cnnpred: Cnn-based stock market prediction using a diverse set of variables. *Expert Systems with Applications*, 129:273–285, 2019.
5. https://archive.ics.uci.edu/ml/datasets/CNNpred%3A+CNN-based+stock+market+prediction+using+a+diverse+set+of+variables

References

6. Bradbury, James, Merity, Stephen, Xiong, Caiming, and Socher, Richard. Quasi-recurrent neural networks. In ICLR, 2017.

7. Chang, Shiyu, Zhang, Yang, Han, Wei, Yu, Mo, Guo, Xiaoxiao, Tan, Wei, Cui, Xiaodong, Witbrock, Michael J., HasegawaJohnson, Mark A., and Huang, Thomas S. Dilated recurrent neural networks. In NIPS, 2017.

8. Yin, Wenpeng, Kann, Katharina, Yu, Mo, and Schutze, Hinrich. ¨ Comparative study of CNN and RNN for natural language processing. arXiv:1702.01923, 2017.

9. LSTM: https://towardsdatascience.com/sequence-models-and-recurrent-neural-networks-rnns-62cadeb4f1e1