Learning From Data Coursework

November 2022

1 Introduction

Technical analysis of financial markets means to study the relation between the price and volume of a trading stock. There are many technical indicators to measure the performance of a stock which assist traders in deciding the best trading strategy. On exploring the indexprocessed.csv data set with a different question in mind, and creating different kinds of indicators, I found there to be two types: those which oscillated around a mean — such as RSI (Relative strength index) which oscillates around 0 — and those which follow a time dependant trend — like moving averages. Regression models attempt to predict the exact value of an outcome variable by assuming that there is a linear transformation between features and output. In this report, I will train different regression models on the two types of financial indicator to predict the next day's close price of a stock exchange index to see which is more effective.

2 Method

2.1 Data Exploration

I am going to be using four indexes from the indexprocessed.csv data, NASDAQ (IXIC), NYSE(NYA), DAX(GDAXI), and HSI. These each have unique features, fundamental or technical 1. The NASDAQ represents the largest tech companies in the US; NYSE represents the largest stock exchange in the world; DAX represents top German firms, and is the most volatile index; and HSI consists of the largest banks in China.

	NASDAQ		DAX		NYSE		HSI	
	mean	std	mean	std	mean	std	mean	std
CloseUSD	2389.70	2573.01	7216.11	4452.54	5760.60	3920.55	1976.07	1056.47
Volume	$1.21x0^9$	$1.111x0^9$	$6.83x10^7$	$6.47x10^7$	$1.62x10^9$	$1.95x10^9$	$8.43x10^8$	$1.02x10^9$

Table 1: Index data

1 shows us the difference in scales between volume and price. The standard deviation is also very high, showing how much the price varies over our date range.

The columns are highly positively correlated. Fig 1 is the correlation heat map for NYSE, and the other three indexes produced the same graph.

This does not imply dependence on one another — knowing the low for a day can not let us predict the high of the day for example — but, more likely, that they are dependant on the same things.

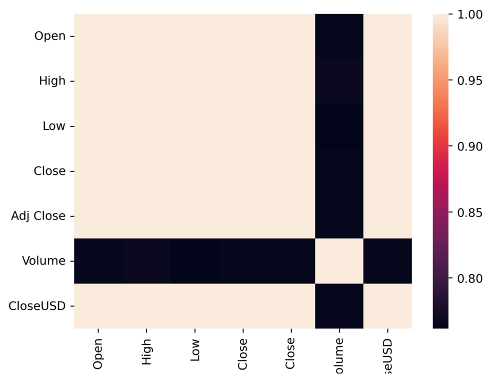


Figure 1: Correlation of NYSE data fields

2.2 Data Cleaning

There were some issues with the data. Some indexes didn't contain any volume information until a certain time. Other indexes had zero gain or loss for the start of their data (such as NYSE). Because this will skew the coefficients of our regression models, the data I'll use will start from the first row with no zero features. This still gave us between 5000 and 10000 data points depending on the index and the features. There are also around 300 missing days for each index — business days with no entry — which I will attribute this to bank holidays.

2.3 Feature engineering

The oscillators I chose were the Relative Strength Indicator (RSI), the average increase (negative if it was a decreasing period) over the previous 3,8,21,55, and 200 days, and the MACD — the difference between the 12 and 26 day moving average. I wrote custom code for each. 2 shows the histograms for these indicators on the HSI — the other indexes look similar. Each of these indicators roughly oscillates around a mean, often 0. However, the average increases are more negatively skewed the higher the time period, because in general each stock increases over our whole time period and longer average increases represent that, but we still expect it to vary around a mean 0.

For the non-oscillating indicators I'll be using the Volume, the OBV —On Balance Volume, for measuring momentum — and the moving averages, over the past 9,12,26,55, and 200 days, of the close price in USD. 3 Compared to the oscillating features, these don't share the same variation around a mean. Instead they follow the close price more closely 3h.

Table 2 shows the correlation between the oscillating features and the closing price of the data. This is expected since oscillators don't increase or decrease with respect to time like the close price does.

	MACD	3d_av_inc	8d_av_inc	21d_av_inc	55d_av_inc	200d_av_inc	RSI
DAX	0.1113	0.0166	0.0282	0.0444	0.0734	0.0696	0.0587
NASDAQ	0.2520	0.0184	0.0276	0.0413	0.0768	0.2411	-0.055
NYSE	0.1469	0.0302	0.0467	0.0786	0.1021	0.0650	0.0334
HSI	0.0715	-0.059	-0.103	-0.165	-0.230	-0.349	-0.234

Table 2: Oscillators correlation to Close price

In contrast, table 3, has much higher correlation between the features and the close price.

	Volume	OBV	9d_MA	12d_MA	26d_MA	55d_MA	200d_MA
DAX	-0.159	-0.088	0.9979	0.9973	0.9946	0.9890	0.9681
NASDAQ	0.7844	0.5639	0.9993	0.9992	0.9984	0.9969	0.9906
NYSE	0.2983	0.1593	0.9975	0.9968	0.9936	0.9872	0.9572
HSI	0.6143	0.4229	0.9959	0.9949	0.9897	0.9787	0.9224

Table 3: time dependant features correlation to close price

2.4 Modelling

I will train each model with Linear Regression and Polynomial regression, as well as Polynomial regression with LASSO(Least Absolute Shrinkage Selection Operator) and Ridge regularisation to predict the exact closing price in USD. In the Hyper-parameters section, I vary the order of the polynomial regression as well as the shrinkage parameter for both Lasso and Ridge. Since I'm using regression, I'll also standardise the features using the min-max scalar.

$$x_s = \frac{x - min}{max - min}$$

This should also help LASSO and Ridge converge since some indicators are orders of magnitude higher than others. I used a 80:20 train test split as this article interestingly describes.

2.5 Hyper-parameters

I calculated the test score and train score of Lasso and Ridge regression for different values of alpha on the processed data before running each experiment. Here is a sample of the results from the processed NASDAQ data with the oscillating features.

alpha	1	2	3	4	5	10	20	50	100
Ridge train score	0.4224	0.3901	0.3707	0.3573	0.3469	0.3146	0.2784	0.2201	0.1720
Ridge test score	-2.093	-2.324	-2.477	-2.593	-2.687	-3.002	-3.356	-3.849	-4.170
Lasso train score	0.3699	0.3284	0.2970	0.2753	0.2584	0.1687	0.0487	0.0	0.0
Lasso test score	-2.594	-2.844	-3.068	-3.239	-3.388	-4.148	-4.688	-4.598	-4.598
% params reduced	0.6388	0.6666	0.7222	0.7777	0.8055	0.8888	0.9166	1.0	1.0

Table 4: Testing regularisation parameter

I also tested different degrees of polynomial regression.

degree	1	2	3	4	5
training score	0.18900	0.48341	0.63552	0.72821	0.80654
test score	-4.4016	-1.6205	-227.78	-2788.4	-187409

Table 5: Testing degree of polynomial regression

From this, I chose alpha = 1 and the degree of polynomial regression to be 2.

2.6 Analysis

To test the performance of my models, I'll be using the coefficient of determination of the prediction (R^2). Where,

$$R^2 = \frac{\Sigma_i (T_i - P_i)^2}{\Sigma_i (T_i - \bar{T})^2}$$

where T are the true values and P are the values predicted by the models. Close to 1 is good. So that I can keep track of over-fitting models, I will calculate the score for the training data and the testing data and compare the two.

3 Results

	NYSE		NASDAQ		DAX		HSI	
	test	train	test	train	test	train	test	train
Lin Reg	0.0842	-15.390	0.188	-4.400	0.0931	-25.753	0.193	-13.818
Poly Reg	0.191	-14.943	0.483	-1.620	0.384	-26.027	0.339	-11.628
Poly Lasso	0.171	-14.351	0.360	-2.594	0.322	-26.245	0.293	-12.425
Poly Ridge	0.176	-14.021	0.422	-2.093	0.322	-25.010	0.321	-11.926

Table 6: Oscillating features

	NYSE		NASDAQ		DAX		HSI	
	test	train	test	train	test	train	test	train
Lin Reg	0.99400	0.95838	0.99730	0.99604	0.99488	0.93696	0.99188	0.92297
Poly Reg	0.99435	0.94254	0.99752	0.98974	0.99507	0.93404	0.99230	0.92146
Poly Lasso	0.99336	0.94932	0.99689	0.99506	0.99424	0.92579	0.99041	0.91427
Poly Ridge	0.99332	0.94155	0.99692	0.99014	0.99393	0.92485	0.99075	0.91139

Table 7: non-oscillating features

Table 6 shows the scores of the models with oscillating features and 6,7,5,4, show the performance of the models graphically. Table 7 shows the scores of the models with non-oscillating features and 10,11,9,8, show the performance of the models graphically.

When changing the hyper parameters, as exemplified in the precious section, I found that for both types of features increasing the order of the polynomial regression over-fit the model to the training data and the model was less able to correctly predict the test data (5 has a score of -187409 for a 5^{th} degree polynomial). Increasing the shrinkage parameter for Lasso and Ridge regression didn't reduce over fitting for either set of features. They helped reduce the training score, but the testing score fell too. Lasso and ridge didn't perform any better than polynomial or linear regression, across all features, failing to reduce over fitting in a significant way. Lasso is useful for when we have a few important features amongst many unimportant ones; its low performance could tell us that all of our features are equally important or unimportant.

This should have benefited ridge regression, which doesn't shrink features to zero, but because we used min-max standardisation the relation between the range of each feature was still preserved and features like volume or MACD were still larger relatively. Another option would have been to normalise the data which might have improved this. The NASDAQ was significantly less over-fit than the others when trained with both types of features (which you can see most clearly from 6). This can be attributed to it's mean being lower than it's median, meaning that it was negatively skewed. Since our training set is the *first* 80% of the data, the training set for the NASDAQ is more representative of the test data than the others. The HSI also shared the negative skew but it was a more volatile market leading to great error. Opposite to the NASDAQ, the DAX was the most over-fit across both sets of features. It covers the widest range of values (min: 1141, max: 18934) and is the most positively skewed. We can tentatively say, therefore, that regression models are more accurate when trained on smoother data. All regression models struggled to fit to the oscillating

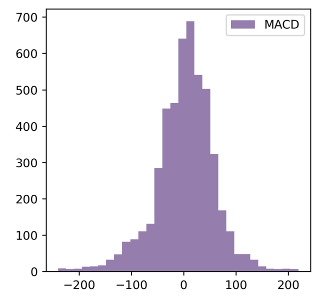
features, with polynomial regression faring only marginally better due to its increased complexity. In contrast, all regression models performed well with the non-oscillating features, with R^2 scores very close to 1. This is due to the correlation between the features and the output variable and the nature of regression models. Regression models are linear transformations which preserve the relationship between the features — if the features are oscillating, so will be the output. Polynomial regression performs slightly better with oscillating features, because the complexity means that it can vary in a wider range and get closer to the correct values (Figure 4 shows how polynomial regression oscillates over a larger range), not because it follows the trends in the data. Regression fails to detect the underlying trend, when trained on these oscillators, despite the oscillators being dependant on the non-oscillating data, which *is* highly correlated with the output data. Regression cannot find the underlying pattern because it assumes there is a direct linear transformation between input and output; the output needs to be able to increase/decrease along an underlying trend, not only oscillate. What the oscillators are very good at is predicting changes and the magnitudes of them; figure 6a is a good example of this. Oscillators are better suited to predicting the gain or loss for a particular day rather than the exact closing price. They would be better for a Multilayer perceptron or Convolutional Neural Network where the output would be binary and the models can recognise these underlying trends.

4 Conclusion

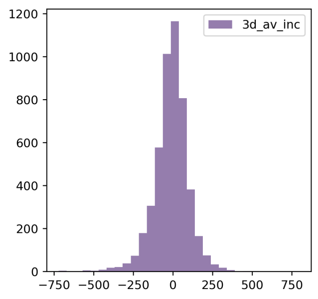
In conclusion, training regression models to predict the closing price of a stock exchange index on highly correlated, non-oscillating features produces more accurate results than training the same models on oscillating features calculated using the same data set, which do not correlate to the closing price. Regression models are limited by their inability to predict outcomes which are not directly correlated to the features. In further experiments, I would use MLPs or CNNs to predict increase or decrease using oscillators, and investigate how the weights change depending on the rate of oscillation or correlation to the output variable.

5 Graphs

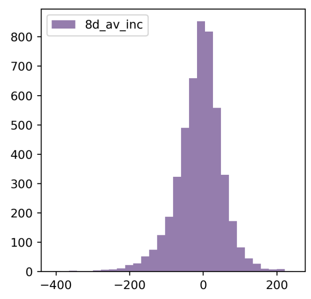
5.1 Feature Histograms



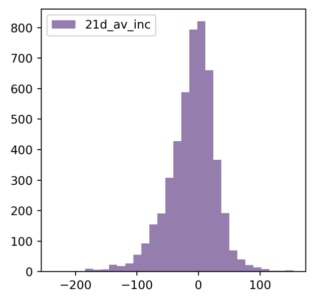
(a) MACD



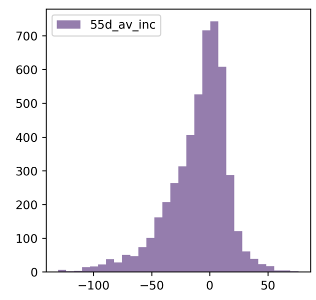
(b) 3 day average inc



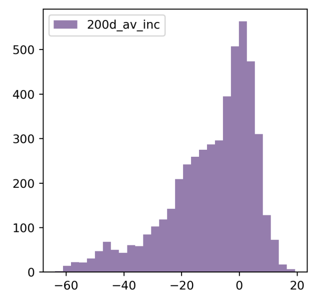
(c) 8 day average inc



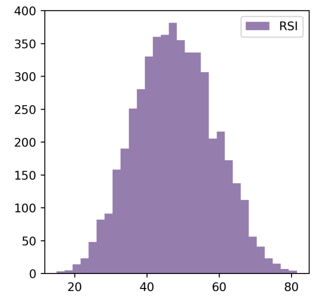
(d) 21 day average inc



(e) 55 day average inc

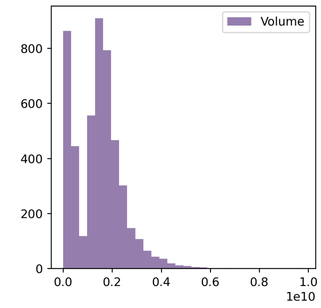


(f) 200 day average increase

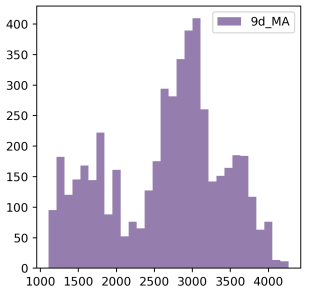


(g) RSI

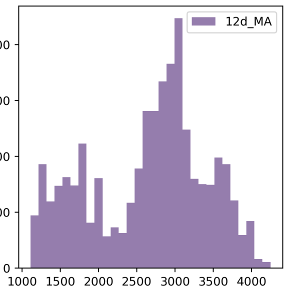
Figure 2: Histograms showing distribution of oscillating indicators



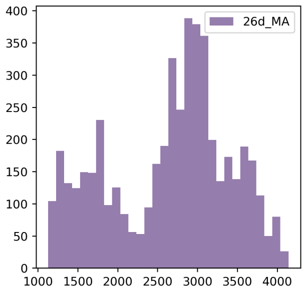
(a)
Volume



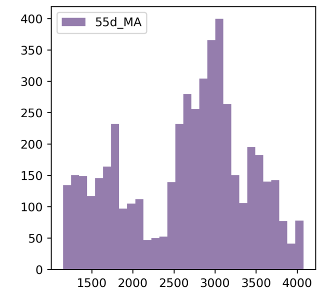
(b) 9 day
moving average



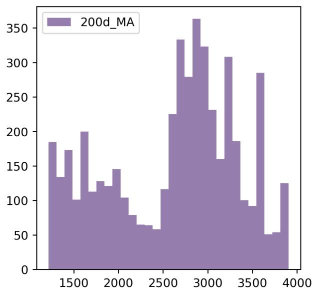
(c) 12 day
moving average



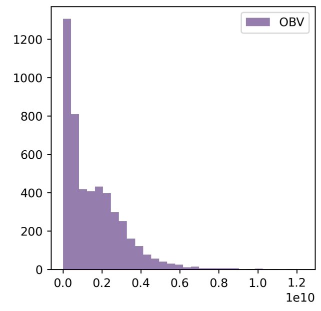
(d) 26 day
moving average



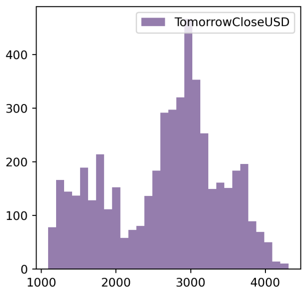
(e) 55 day
moving average



(f) 200 day
moving average



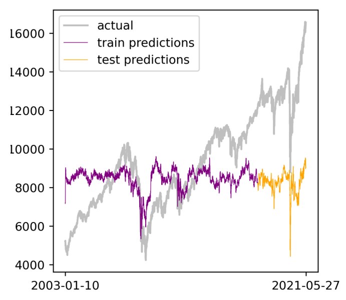
(g) OBV



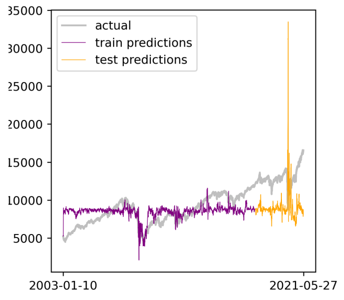
(h) Close price USD

Figure 3: Histograms showing distribution of non-oscillating indicators

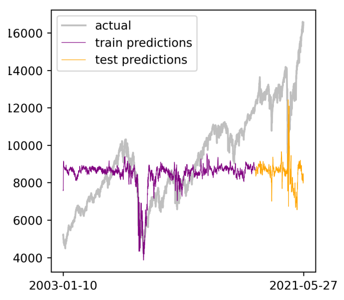
5.2 Models



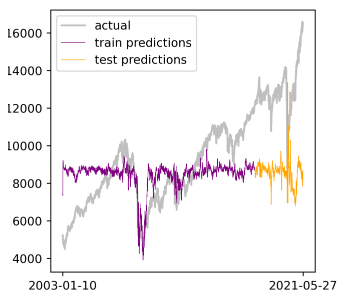
(a)
Linear Regression



(b) 2^{nd} order
Polynomial

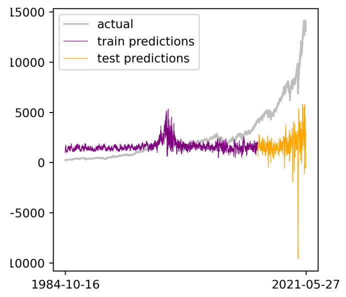


(c)
Lasso Regression

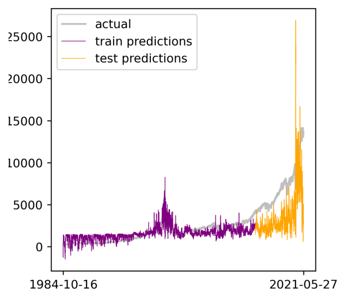


(d)
Ridge Regression

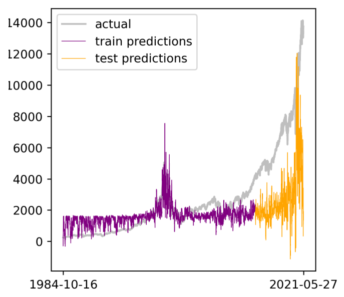
Figure 4: Oscillating features modelling NYSE



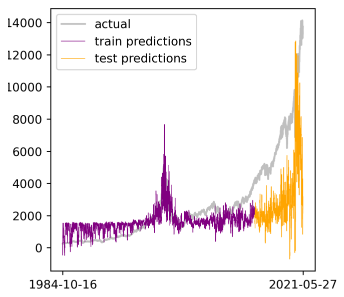
(a)
Linear Regression



(b) 2nd order
Polynomial

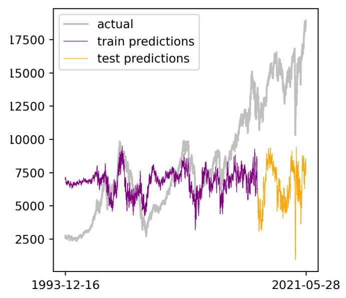


(c)
Lasso Regression

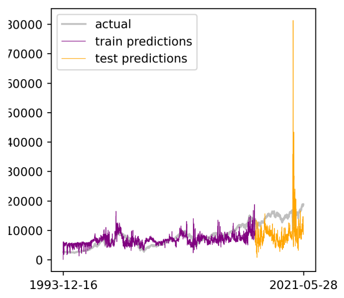


(d)
Ridge Regression

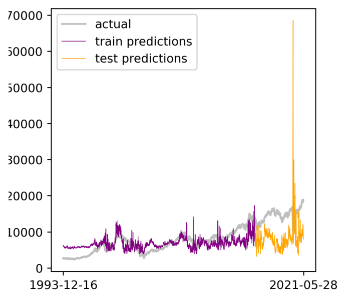
Figure 5: Oscillating features modelling NAS



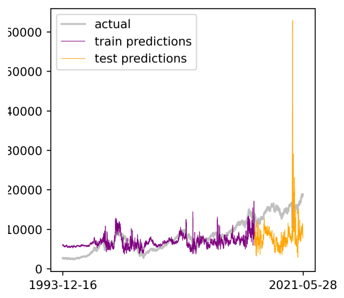
(a)
Linear Regression



(b) 2nd order
Polynomial

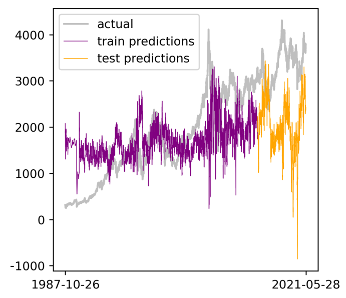


(c)
Lasso Regression

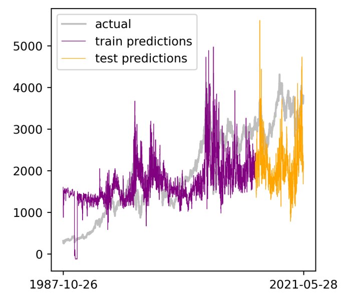


(d)
Ridge Regression

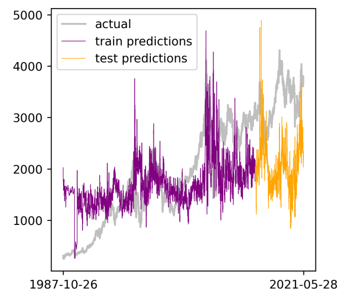
Figure 6: Oscillating features modelling DAX



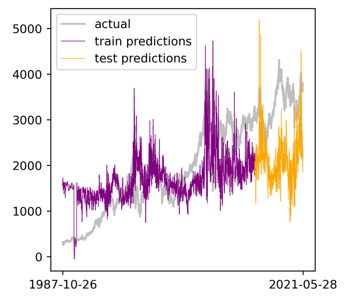
(a)
Linear Regression



(b) 2nd order
Polynomial

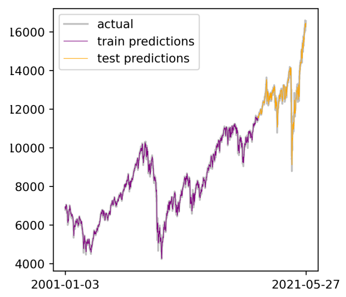


(c)
Lasso Regression

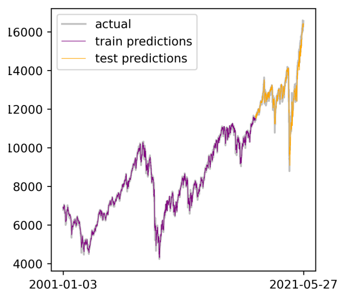


(d)
Ridge Regression

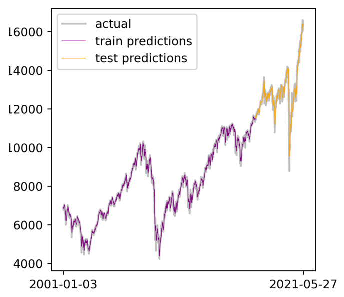
Figure 7: Oscillating features modelling HSI



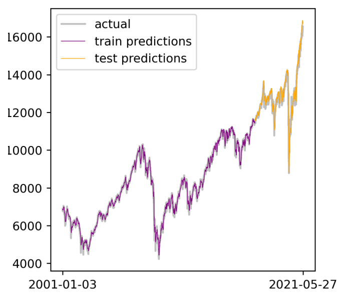
(a)
Linear Regression



(b) 2^{nd} order
Polynomial

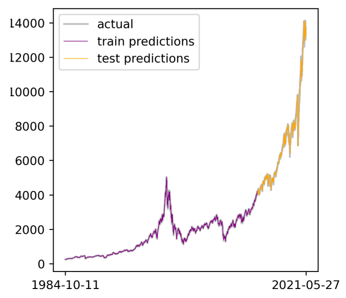


(c)
Lasso Regression

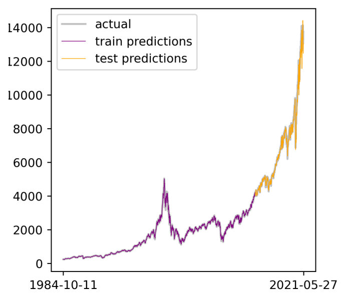


(d)
Ridge Regression

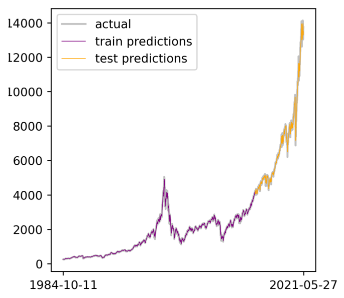
Figure 8: Non-oscillating features modelling NYSE



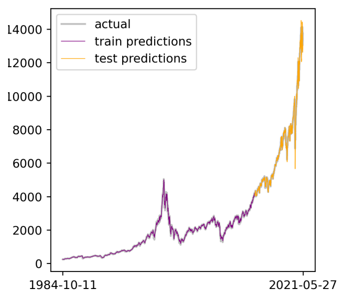
(a)
Linear Regression



(b) 2^{nd} order
Polynomial

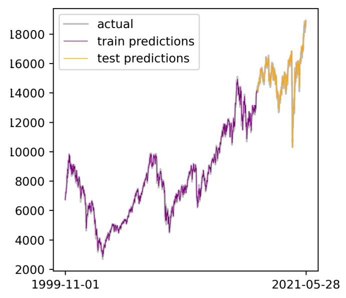


(c)
Lasso Regression

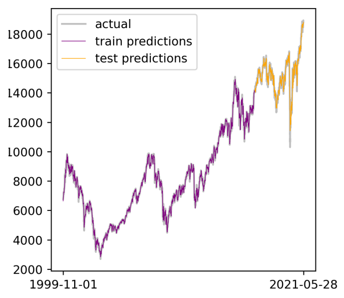


(d)
Ridge Regression

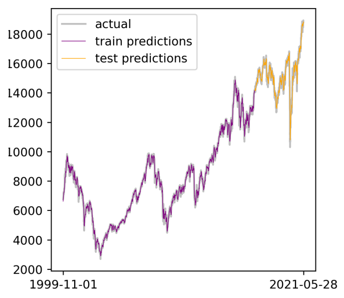
Figure 9: Non-oscillating features modelling NAS



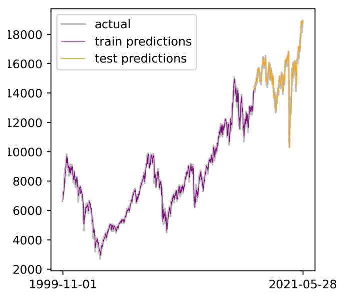
(a)
Linear Regression



(b) 2^{nd} order
Polynomial

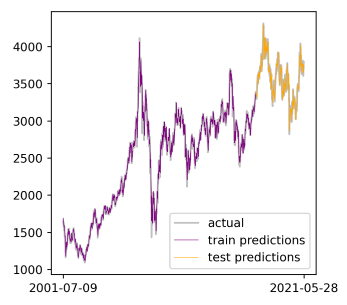


(c)
Lasso Regression

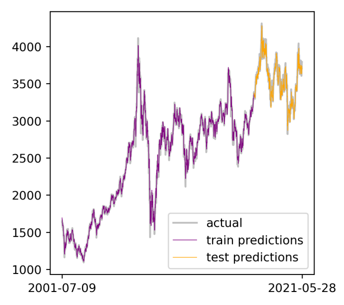


(d)
Ridge Regression

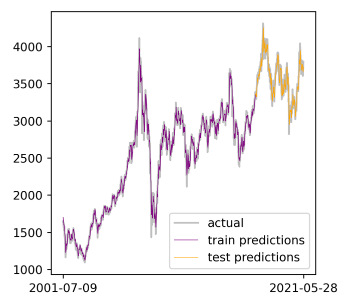
Figure 10: Non-oscillating features modelling DAX



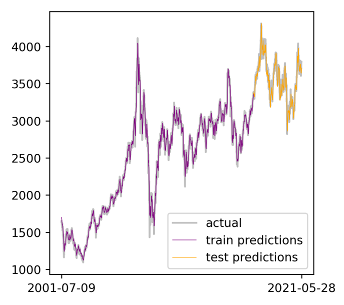
(a)
Linear Regression



(b) 2nd order
Polynomial



(c)
Lasso Regression



(d)
Ridge Regression

Figure 11: Non-oscillating features modelling HSI