Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 00 (2023) 000–000

Procedia Computer Science

www.elsevier.com/locate/procedia

International Neural Network Society Workshop on Deep Learning Innovations and Applications (INNS DLIA 2023)

Application Of Machine Learning Algorithms to Free Cash Flows Growth Rate Estimation

Ivan Evdokimov[a,*], Michael Kampouridis[a], Tasos Papastylianou[b]

[a]School of Computer Science and Electronic Engineering, University of Essex, Wivenhoe Park, Colchester CO4 3SQ, UK
[b]Institute of Public Health and Wellbeing, University of Essex, Wivenhoe Park, Colchester CO4 3SQ, UK

Abstract

Machine learning (ML) demonstrates superior accuracy in financial time-series forecasting compared to traditional statistical models. While most studies focus on applying ML algorithms to high-frequency pricing data, the availability of fundamental financial data is limited as it is generated quarterly. This paper investigates the performance of nine ML algorithms in small sample data sets, against an ARIMA model — frequently used for financial time-series forecasting — serving as a benchmark. Results obtained from 100 US companies indicate that the majority of ML algorithms exhibit low error rates on the test set, outperforming benchmark results. Notably, the k-nearest neighbor algorithm achieves the highest prediction accuracy among the algorithms considered, even with only 33 data observations, while avoiding overfitting.

Keywords: Machine Learning; Forecasting; Financial Fundamentals; Time-Series Modeling; Free Cash Flows Forecasting

1. Introduction

Much of the academic literature on the applications of ML algorithms to finance tends to focus on pricing data [15], i.e. the market price of a financial security (e.g. common stocks, bonds, etc.). This is typically the case due to the open-source availability and abundance of such data: each year a total of 252 data points are created, assuming daily frequency. Fundamental data, on the other hand, such as revenues and free cash flows, usually used by creditors, auditors and investors to perform their analyses, is in short supply as companies only disclose such figures on a quarterly basis, thus generating four data points per annum.

Free Cash Flows (FCF) is a measure of cash available for shareholders after all the necessary expenses and investments were met. Fundamental investors aim to predict the growth rate of FCFs in order to estimate a business'

* Corresponding author.
E-mail address: ie20391@essex.ac.uk

intrinsic value. Compared to the price of financial securities, intrinsic value is a measure of how much a business is worth [11] given its future FCFs growth, discounted at an appropriate rate. The model that performs this computation is known as Discounted Free Cash Flows (DCF). Specifically, the DCF model takes the FCF growth and discount rates as inputs, and outputs an estimate of the intrinsic value of a stock. While there exists a consensus among investment practitioners and researchers on how to derive the discount rate [23], the best way to derive the growth rate is still an open question. This is partly due to the fact that the growth rate at a particular point in time tends to be highly dependent on a company's financial and operating conditions, which are generally not constant.

This paper investigates the effectiveness of a representative sample of regression algorithms taken from the Machine Learning (ML) literature, benchmarked against an Auto Regressive Integrated Moving Average (ARIMA) model, which is traditionally treated as the gold-standard approach for growth rate prediction in the finance literature [6]. In our regression setting, we treat the Free Cash Flows growth rate as our target variable, and a set of past lags, the mean and standard deviation of the target variable, and financial ratios as our feature matrix.

The motivation for this research stems from the fact that while traditional quantitative models require that the data conforms to a strict set of assumptions[1], the models we consider here do not require such strict assumptions on the data. Hence, when these assumptions are violated, the ML models are more likely to demonstrate higher predictive power.

The present work makes two important contributions to the state of the art. Firstly, the focus of this paper is on finding a more accurate way of predicting the FCF growth rate than the established ones, which, when applied to the DCF model would result in a more precise estimate of the 'true' worth of a business. This, in turn, would result in better investment decisions undertaken by the investment community. Secondly, the data sets presented in the paper are characterized by data that is sparse in terms of available number of observations. Therefore, we present an overview of the performance of a representative set of machine learning algorithms in this setting, and in addition, we outline a method for time-series forecasting using an expanding window, which allows to preserve as many training observations as possible.

The rest of this paper is structured as follows: Section 2 provides the background information; Section 3 details the literature review conducted for this study; Section 4 outlines the methodology used in our approach; Section 5 details the experimental setup in which we consider the properties of data and parameter tuning for our algorithms; Section 6 describes the results obtained from our experiments and the computational complexity involved; finally Section 7 provides conclusions and implications from the results.

2. Background Information

The essence of fundamental stock valuation is researching a company's financial documentation. Specifically, investors, suppliers, and creditors focus on two key documents, namely the '10-Q', which discloses data from the first three quarters of a year, and the '10-K', which summarizes the same information for a whole year. Specifically, financial analysts will pay particular attention to the key figures from the 'Balance Sheet' — which is a summary of what a company owns and owes; the 'Income/Loss section' — a summary of how much money from sales a company generated; and the 'Cash Flow section' — a summary of how the capital was allocated within a company. The aim of this due diligence is to estimate the fair value of a business, given the growth of sales, earnings, credit and capital relationship changes and available equity.

In economic theory, the value of an asset is determined by the future cash flows it generates, discounted at a suitable rate that reflects the associated risk for the investor [6]. The concept of "intrinsic value" is emphasized in financial literature [11], and refers to an estimated reasonable price for a stock based on its future cash flows, discounted at an appropriate rate. Various methods are available for estimating this intrinsic value, with the discounted cash flow (DCF) model being the most popular approach [23]. The DCF model can be formally represented as follows:

[1] e.g. the absence of a deterministic trend and time anomalies, constancy of parameter coefficients, and homoscedasticity of the error process, all of which are frequently violated by fundamental financial data [24]

$$V = \sum_{t=1}^{T} \frac{\mathrm{FCF}_{t+1}}{(1 + \mathrm{WACC})^t}, \tag{1}$$

where $t = 1, ..., T$ is the time index, with $t = 1$ corresponding to the first quarter; T denotes a sufficiently distant point of interest in the future; V denotes the intrinsic value of a stock; FCF_{t+1} denotes the Free Cash Flows at quarter $t + 1$; and WACC is the weighted average cost of capital (explained below).

In other words, in order to arrive at the intrinsic value with respect to a future point in time, researchers predict the future Free Cash Flows $T+1$ periods into the future, apply a discounting rate by scaling each future FCF by $(1 + \mathrm{WACC})^{-t}$, and then sum up the resulting values. We calculate the Free Cash Flows as:

$$\mathrm{FCF}_t = \mathrm{CFO}_t - \mathrm{CUI}_t, \tag{2}$$

where CFO_t denotes the 'Cash From Operations', and CUI_t denotes the 'Cash Used Investing'. These two variables are reported quarterly by a company and can be found in the 'Cash Flow Statement'. The one quarter ahead value of FCF is estimated as:

$$\mathrm{FCF}_{t+1} = \mathrm{FCF}_t(1 + g_{t+1}), \tag{3}$$

where g_{t+1} — the target variable we aim to estimate in this research — is the projected growth rate to the next quarter, i.e. $t+1$. The final component of the DCF model is the weighted average cost of capital (WACC), which is widely used to estimate the level of risk associated with common stock [23]. The WACC considers both the default risk associated with a firm's leverage and the risk related to stock market shocks, taking into account the amount of equity capital the firm possesses [14], formally:

$$\mathrm{WACC} = r_D \frac{D}{D+E} + r_E \frac{E}{D+E}, \tag{4}$$

where D denotes the total debt a firm possesses, E denotes the total equity, r_D denotes the required return on debt, and r_E the required return on equity. The values for D and E are displayed on the Balance Sheet of a firm, usually labeled as 'Total Liabilities' and 'Total Shareholders' Equity', respectively. The cost of debt, r_D is usually assigned by the credit rating agency, such as Moody's, Standards and Poors, and Fetch, and is disclosed in 10K/Q financial reports. To estimate r_E, researchers usually apply the Capital Asset Pricing Model (CAPM):

$$r_E = R_f + \beta(R_m - R_f), \tag{5}$$

where R_f denotes the risk free rate, which is typically estimated via the 3-Month US Treasury bills rate; R_m denotes the expected (average) return on a broader market index, such as the S&P500; and β denotes the slope-coefficient from an Ordinary Least Squares regression, performed on 3 years' worth of monthly returns on stock, against the financial index's price returns. More specifically, this is derived from:

$$R_s = \beta R_{idx} + \epsilon, \tag{6}$$

where R_s is the vector of stock returns; R_{idx} is the vector of market returns; the slope-coefficient β is the quantity required for Equation 5; and lastly, ϵ denotes the regression error [21].

Earlier we noted that intrinsic value estimation typically relies on the DCF model and calculating the WACC using the CAPM and financial statements. However, determining the future FCF growth rate (g_{t+1}) lacks a universally accepted approach. This research addresses this gap by formulating the derivation as a regression problem. The aim is to predict the future FCF growth rate (g) using financial ratios. As previously mentioned, a more accurate FCF growth rate prediction would result in a more precise estimate of the 'true' worth of a company, and subsequently in better investment decisions undertaken by the investment community. The next section explores existing methods in the literature for approximating the growth rate.

3. Literature Review

The finance literature highlights the usefulness of ML algorithms for classification and regression problems [15]. Typical regression problems revolve around predicting revenues and earnings, where these variables are indicative of future stock price performance. Typical classification problems focus on determining probability of important events occurring, e.g. the probability of bankruptcy of enterprises.

It is worth noting that when assessing the value of a company, investors often rely on projections provided by financial institutions and investment banks. However, these projections can be akin to a black box and may exhibit biases [9]. Another commonly used approach involves utilizing financial variables or historical average FCF growth rates [6]. However, this method overlooks external factors such as economic recessions or seasonal fluctuations in consumer demand. Additionally, as companies evolve over time, their growth rate tends to decrease as they mature [6], rendering past growth rates a less reliable predictor of future performance.

Traditionally used quantitative forecasting models like ARIMA have shown robust performance [3]. However, these models rely on assumptions that financial data often violates over longer periods [12]. While ML algorithms have gained interest for their ability to bypass stationarity and linearity assumptions [10], their application has mainly focused on forecasting earnings per share, revenues, and cash flows [7, 16, 25]. Recent research evaluated various ML algorithms against analysts' expectations and found that ML models provided comparable or superior results [16].

The forecasting of business fundamentals and earnings per share has been approached as a classification problem. For instance, [5] used Random Forests and Stochastic Gradient Descent Boosting to predict a year ahead directional changes in corporate earnings. Ensemble learning models show robust performance on out-of-sample data, primarily due to the capability of such models to automatically choose significant variables from high-dimensional data. [13] developed a tree-based technique, named TreeNet, to estimate the probability of bankruptcy in Chinese public companies using financial ratios, valuation multiples, macroeconomic and other variables. The model showed improved accuracy of more than 90% in predicting bankruptcy.

In [23], ML algorithms were applied for fundamental company valuation, specifically focusing on the WACC and FCF growth rate. A linear regression model was used, incorporating detailed data from balance sheets and income/loss statements, polynomials representing past FCF lags, and selected macroeconomic variables. The study demonstrated that ML-enhanced valuation models provide less biased estimates of intrinsic value compared to market analysts. Similarly, in [1], the directional changes of enterprise profitability measures, including the FCF growth rate, were investigated. Multiple Random Forest Classification algorithms were trained, adding one feature per training iteration. Results revealed that ML algorithms achieved 55-67% accuracy in predicting directional changes in future FCF growth rates, surpassing the 50% accuracy of the Random Walk model when tested on out-of-sample data.

Based on the literature reviewed, it is evident that ML algorithms are gaining popularity in the field of financial forecasting, particularly in predicting per share earnings and revenues. While there is ongoing exploration of the forecasting ability of FCFs, most studies focus on predicting directional changes, cash flows from projects, and employ simple algorithms such as Linear Regression. Nonetheless, the overall trend indicates a growing interest in applying ML techniques to such tasks. The next section considers the methodology pursued over the course of this study.

4. Methodology

This research investigates the effectiveness of a number of ML algorithms, benchmarked against ARIMA, commonly applied to financial time-series data. The target variable, i.e. the 'Free Cash Flows growth rate', is a frequently used measure for calculating the intrinsic value of a business, which is used in finance to make buy/sell decisions related to common stock investing. At its core, we have a regression problem, where we regress the target variable — the FCF growth rate — against past lags of the target variable, its empirical mean and standard deviation, and a set of financial ratios, assuming that the next quarter's FCF growth rate, depends on the previous quarters' growth, their moments, and financial ratios.

Financial ratios serve as performance indicators for specific aspects of a company's performance, while Free Cash Flows (FCFs) represent the amount of money available for distribution to shareholders after meeting all obligations. The growth rate of FCFs measures the percentage change in FCFs over time. To ensure reproducibility, data for our research is obtained from the Securities Exchange Commission's EDGAR archive, which covers all securities instruments in US Exchanges. The remaining sections are organized as follows: Section 4.1 explains the data preprocessing methodology, and Section 4.2 provides an overview of the regression problem tackled in this research.

4.1. Data Preprocessing

Financial ratios are a frequently used measure of how company is doing at a point in time and how it compares to similar companies in a given industry. In our research, we utilise financial ratios that measure the performance of a company in areas such as liquidity (Current Ratio), product profitability (Gross Profit Margin), financial health (Debt–Equity Ratio), assets utilisation (Return On Assets) and investors' capital efficiency (Return On Equity) [17]

FCFs represent the funds available to shareholders after all expenses and investments are accounted for [14]. The computation formula for FCFs can be found in Section 2 (Eq. 2). Regulatory requirements mandate that companies report their financial figures on a 90/180/270/360-day basis. This presents a challenge for forecasting models since reported figures are inclusive of values reported previously. To address this issue, we standardize all reported figures to a 90-day basis. This ensures that each new quarter reflects the amount of money generated by the company over a fixed 90-day period.

In order to derive the growth rate and make sure that our data set is in the same units, we convert all values to a percentage change between two periods t and $t-1$, as follows:

$$v_t = \frac{v_t - v_{t-1}}{v_t}, \tag{7}$$

where v_t denotes a value (either a target or a feature vector) at time $t = 1, .., T$.

In order to account for sudden changes and fluctuations in important aspects of companies over time, it is essential to avoid using raw values that can result in large deviations. Instead, we incorporate the mean and standard deviation of the first lag of our target variables into the feature set. By including these statistics, we provide the forecasting models with information on how the distribution's moments evolve over time. To calculate the moving average and standard deviation components, we use an expanding window method. This means that the aggregate statistics are computed based on the available data at each time point. As the window expands, the moving average and standard deviation converge to the overall average and standard deviation of the distribution. This approach helps to mitigate the potential misleading effects of large shifts in the empirical distribution and provides algorithms with a better understanding of how the moments of the distribution evolve over time.

To address high kurtosis (in excess of 3) and outliers in FCFs, we apply MinMax Scaling, while using again an expanding window method. This normalization ensures unit-variance distributions, enhancing ML algorithm effectiveness.

4.2. Regression Problem Formulation

To solve our regression problem, we apply supervised learning algorithms, namely: Automatic Relevance Determination (ARD), Bayesian Ridge (BR), Decision Tree (DT), Random Forest (RF), XGBoost (XGR), Lasso linear regression, K-Nearest Neighbor (KNN), Multilayer Perceptron (MLP), and Support Vector Regressor (SVR), with the purpose of minimizing the loss function:

$$\sqrt{\epsilon} = \sqrt{\frac{1}{T}\sum_{t=1}^{T}(g_t - \widehat{g_t})^2}, \tag{8}$$

which is also known as Root Mean Squared Error (RMSE), denoted as $\sqrt{\epsilon}$. In Equation 8 g_t is the actual FCF growth rate at time t and $\widehat{g_t}$ is the model estimation of the growth rate at time t. Next section outlines the experimental setup for this regression problem.

5. Experimental Setup

In this section we present the details of our experimental setup. We first present the data sets used in our experiments, forecasting procedure, followed by discussion on hyperparameter tuning.

5.1. Data and Forecasting Method

For reasons explained in Section 2, public companies generate 4 data points annually (once every quarter). Further, popular paid financial data sources, such as Bloomberg, Compustat and Refinitiv Eikon began to store their data digitally in 1980s [15]. For most publicly traded companies there are up to 120 data points available (past 30 years). Thus, ML algorithms will not have enough data to adequately train, which results in model overfitting [2]. To overcome this problem, we use an expanding window approach, where the number of instances allocated to training expands at every iteration by adding a point from the test set. This allows us to preserve as much data as possible, while at the same time leaving enough room for performance evaluation. Implicitly, this process guarantees better fitness of the model parameters even in the presence of past outliers, as these will be present in the training data at some point.

The fact that data sources such as the above are not free, makes it hard for independent users to reproduce study results. For this reason, we obtained our data from the Securities Exchange Commission's EDGAR repository — an open source provider of fundamental financial data. The full process of obtaining data from EDGAR is available as an open-source project at [8]. At EDGAR, there are 11662 US tickers[2] for which filings are available. Some of these tickers are denoted as 'secondary' issues, i.e. 'BRK-B' is a class B stock for Berkshire Hathaway. For the purposes of this study, we work with class A stocks, as these are issued by the parent company. After exclusion of secondary issues, we are left with 10813 companies. For our study we exclude all financial, blank-check, conglomerates and other non-manufacturing enterprises, because these companies usually have different financial ratios for the evaluation of the company's financial standing, due to the different nature of business activities conducted by such entities. This leaves 4266 companies to choose from. Correspondingly, we exclude all companies that went public after 2009 as this will guarantee that we get at least 45 observations per company, leaving 1669 companies available. Further, we exclude all non-US companies and those with market capitalisation of less than 1 billion (USD) as of the end of November 2022, as low capitalization companies have different regulation requirements and allowances, and non-US companies are reporting under different tax treatment, hence might require other forms of adjustment than those we describe in the methodology section. Correspondingly, we are left with 857 companies from which we randomly pick 100 to conduct the experiments on.

[2] A ticker is an abbreviation used to uniquely identify publicly traded shares of a particular stock on a particular stock exchange.

Our forecasting program automatically picks the number of lags for the target variable by choosing the number which minimizes the Akaike Information Criterion [4]. Furthermore, since the target variable displays a varying mean and variance over time, we incorporate the mean and standard deviation of the first lag as part of the feature set. The first lag is taken as a proxy for avoiding data leakage — a problem in predictive modeling when the algorithm learns from data outside the training set (the target variable, for instance). Furthermore, in order to pick up appropriate lags of features, we apply a Granger-Causality test, in which the null hypothesis is that the lagged x_t are said not to Grange-Cause variation in y_t. Hence, if the p-value is less than our significance level $\alpha = 0.05$, lags are incorporated into the feature set and are not otherwise. In order to preserve as many observations for training, hold-out validation and testing as possible, we allow for up to 5 lags of both target variables and features. Further, to train, fine tune and test our prediction algorithms, we split the data into 60/20/20 subsets.

5.2. Parameter Tuning

We use the `sklearn` [18] open-source library for the `python` programming language to train the aforementioned machine learning algorithms. Our research considers a diverse set of algorithms with different underlying optimization functions and regularization properties. In order to pick the best set of hyperparameters for each model, we split the data into train/validation/test sets and use the `GridSearch` method from the `hypopt` package on a hold-out validation set, which iterates through all the parameter values to find the minimum RMSE. The process is repeated for each company individually. Because we are dealing with a limited number of observations, we specifically focus on the regularization properties of each algorithm, to avoid overfitting. For instance, we restrict the number of estimators in a Random Forest to up to 5 trees, or the size of the MLP neural network to up to 2 hidden layers with up to 4 nodes in each. This way, the estimated parameters of the regression models are restricted to fit the majority of data observations. We run our tests on a machine with an Intel Core i5-6600, 3.30GHz processor and 8GB RAM.

6. Results

In this section, we describe the results obtained on the set of 100 companies, starting with summary statistics, followed by discussion on computational complexity.

6.1. Summary Statistics

Table 1 summarizes the ML algorithms' RMSE results in the training and test sets, sorted by each algorithm's Friedman ranking. The non-parametric Friedman statistical test with the post-hoc Bonferroni-Dunn's test, serves the purpose of identifying the best performing forecasting model. Table 1 also shows the average rank ('Ranking' column), where the lower the rank, the better the model's performance, alongside the adjusted Bonferroni p_{Bonf} value. A statistically significant difference between the model with the lowest ranking (the control '(c)' algorithm) against the rest of the models is established using p_{Bonf}: a p value less than 0.05 indicates the statistical significance at the $\alpha = 5\%$ level, meaning that the algorithm statistically outperforms the respective algorithms.

It is observed that the K-nearest neighbor yields the lowest RMSE of 0.1078 on the test set (on average). Moreover, the train set for the same algorithm yields 0.1085, indicating 0.0006 train-test sets difference. This rather small gap implies a good fitness of the KNN model to limited sample observations data. This conclusion is further evidenced by the results of the Friedman ranking test. The KNN algorithm achieves the best average rank of 4.1, followed by BR (4.24), ARD (4.725) and SVR (0.2246). Moreover, KNN's rank is statistically significant against our benchmark ARIMA model at the 5% level, as well as DT, RF, XGR, Lasso, and MLP algorithms. Overall, results are consistent with other studies examining the performance of similar algorithms in the context of problems featuring limited data observations, where KNN was also shown to be competitive against other ML algorithms, such as DT, RF, and MLP [19], [22], [20].

It is observed that, the majority of machine learning algorithms examined did not seem to overfit on our sample of 100 data sets, whereas the difference between train and test sets for ARIMA was 0.2415, meaning that the model captured patterns in the training set that could not be applied to the unseen data. Among the ML algorithms, the highest difference between train and test sets is observed with Lasso regression. Moreover, the Multilayer Perceptron

Table 1. ML Algorithms' RMSE Performance, sorted by Friedman ranking. Significant differences at the $\alpha = 5\%$ level between the control algorithm (denoted with (c)) and the other algorithms (one per row) are shown in boldface, indicating that the adjusted p value is lower than 0.05.

	TRAIN		TEST		Ranking	p_{Bonf}
Model	Mean	STD	Mean	STD		
KNN (c)	0.1085	0.1007	0.1078	0.0912	4.1	-
BR	0.1308	0.0538	0.1129	0.0782	4.24	6.6932
ARD	0.1339	0.0577	0.1213	0.0876	4.725	1.2994
SVR	0.1474	0.0707	0.1192	0.0757	5.06	0.2246
DT	0.1231	0.0677	0.1321	0.1011	**5.375**	**0.0261**
RF	0.1173	0.0519	0.1313	0.0931	**5.39**	**0.0233**
XGR	0.1467	5.38E-02	0.1336	0.0758	**5.62**	**0.0035**
Lasso	0.2253	1.03E-01	0.138	0.0895	**6.04**	**5.29E-05**
ARIMA	0.3852	4.65E-01	0.1436	0.099	**6.39**	**7.99E-07**
MLP	0.198	0.1479	0.282	0.2661	**8.06**	**2.05E-19**

Neural Network underfits the data with a difference of -0.0839. It is possible that a better result could be achieved with a larger network (increase in size of layers and/or nodes), however, such a network would require a higher number of observations, which is not possible due to the considerations outlined earlier. This model also yields the highest standard deviation among forecasting algorithms in the test set (0.2661).

As our test RMSE results contain outliers, we provide descriptive statistics for companies for which the ML algorithms demonstrate best and worst performance, along with their respective matrix dimensions. Table 2 reports this summary, denoting n as the number of instances and m as the number of feature vectors in the combined training and validation matrix. Note that the number of instances n can be different for each data set. There are two reasons for it: (i) the forecasting program automatically picks up past lags, hence the number of lags of either target or feature vectors decreases the number of observations, and (ii) different companies in our data set submitted their first financial reports to the SEC at different quarters between the years of 2008 and 2010; hence, as the year of first record varies, so does the length of the training set.

Table 2. Algorithms: best and worst forecast results

Algorithm	Best	n	m	RMSE	Worst	n	m	RMSE
KNN	ACN	33	16	6.72E-06	CAH	44	12	0.3585
BR	ACN	33	16	2.20E-06	ADM	42	12	0.3102
SVR	ACN	33	16	1.31E-05	ADM	42	12	0.3405
ARD	ACN	33	16	2.08E-05	ECL	45	17	0.4325
RF	ACN	33	16	8.86E-06	MO	46	11	0.4818
DT	ACN	33	16	1.22E-05	MO	46	11	0.4818
XGR	KMB	42	27	0.0211	MO	46	11	0.3704
LASSO	ACN	33	16	2.08E-05	ADM	42	12	0.3556
MLP	ACN	33	16	0.0002	CTXS	44	17	1.2129
ARIMA	ACN	33	16	1.74E-06	CAH	44	12	0.4587

From Table 2, it is observed that almost all algorithms show best performance on the same company except for the XGR/KMB pair, whereas the list of companies with worst performance across forecasting models is slightly more diverse. The number of observations hardly affects the performance of the forecasting algorithms, as ADM, ECL, CAH and CTX have the highest number of data records. Interestingly, KMB, which has the highest number of features (27), is showing the best performance for XGR. This could be attributed to the fact that XGR is an ensemble method and can automatically pick significant features from the data. This property of XGR is observed in RF, and DT but, possibly due to the differences in underlying optimization methods, RF and DT perform less well on KMB stock.

Given that the number of records at each company differs, we divide the 100 companies into three groups. In order to get a meaningful number of companies in each group, we take the median number of pre-processed observations (after differencing and scaling), as these data points are used for training ML models. The median number of observations in our 100 companies sample is 44, minimum is 33 and maximum is 47. Hence, Group 1 contains all companies with number of observations below the median, Group 2 with number of observations at median point and Group 3 with number of observations above the median, resulting in 40/22/38 companies per group split. Table 3 summarizes the results in terms of the average RMSE for the test sets in each group, with lowest RMSEs shown in boldfaces.

Table 3. Algorithms: Average Test Set RMSEs By Group

Algorithm	KNN	BR	SVR	ARD	RF	DT	XGR	Lasso	MLP	ARIMA
Group 1	1.23E-01	**1.20E-01**	1.30E-01	1.28E-01	1.44E-01	1.45E-01	0.144084	1.53E-01	0.296924	1.58E-01
Group 2	0.116558	0.120529	0.12436	**0.107938**	0.132123	0.129819	0.13144	0.148773	0.337909	0.148457
Group 3	**0.087341**	0.10088	0.105107	0.121637	0.117958	0.119655	0.123939	0.115476	0.23384	0.125294

The algorithm with best RMSE per group is BR (Group 1), ARD (Group 2), and KNN (Group 3). In addition, KNN had the second best RMSE in Group 1 and 2. Thus, in two of the three groups KNN is the second best, although the difference between KNN and the best algorithm in the given group is minimal.

To summarize, we have seen that the KNN regression algorithm consistently shows low RMSE values, and is able outperform other algorithms in our study at the 5% significance level. However, while it is the best across the 100 companies, when grouped by proposed method, we get slightly different results: BR shows the lowest test set RMSE in the group with the least matrix length, ARD gives the minimum RMSE in the second group, however, the difference in both cases is close to zero.

6.2. Computational Times

In this section we consider the efficiency of each algorithm in terms of the time it takes to fine-tune the model and make predictions. For each machine learning algorithm, we summarize time estimates in Table 4.

Table 4. Computational Times

Algorithms	KNN	BR	SVR	ARD	RF	DT	XGR	Lasso	MLP	ARIMA
Hyper-Parameters space	1400	11520	25	15000	8192	12000	2400	112	2700	N/A
Fine-Tuning Time (seconds)	4.2510	34.1187	0.0841	76.0866	52.1434	31.7173	28.7062	0.3531	201.2459	N/A
Forecasting Time (seconds)	0.0359	0.0359	0.0337	0.0628	0.0806	0.0329	0.0948	0.0362	0.5489	6.3566

As observed, the least efficient algorithm is MLP both for forecasting and fine-tuning. KNN is the third best algorithm with an average training time of 4.2510 seconds, preceded by Lasso and SVR. In terms of forecasting time (after the best set of parameters has been selected), all ML outperform ARIMA. The most efficient algorithm in this aspect is Decision Trees with an average time of 0.0329 seconds. Importantly, the training/tuning and forecasting time complexity will also rely on each algorithm's individual sensitivity to dimensions of input data. We estimate the overall time complexity of the forecasting program to be $O(n^2)$, as the time it takes to produce the forecast will grow quadratically with the number of validation and test points, the number of companies to make forecasts for. It is important to mention that the forecasting approach we propose is used in an 'offline' manner and hence the runtime of each algorithm is not as important as the forecasting accuracy of the models, since it guides the accuracy of estimation of the FCF growth rate. The better the estimate of this target variable, the closer one arrives at the intrinsic value of a common stock, as computed with the DCF model.

7. Conclusions

In this paper, we investigated the performance of nine ML algorithms and ARIMA, which served as a benchmark, in the problem of forecasting the FCF growth rate. The algorithms are examined in an environment of financial data sets, where observations are limited in number (33-47 data points per data set). We show that on a set of 100 companies, ML algorithms display statistically significant superior performance of Free Cash Flows growth rate forecasting. Computational times vary with each individual model's complexity. The above result is important, for two reasons: First, our proposed methodology results in a more precise estimate of the 'true' worth of a business. This is particularly relevant to investors, as it allows to have a better understanding of a company's value. Second, given the fact that in this problem limited data points are available, we have shown that accurate predictions are still possible by using machine learning algorithms and an expanded window approach. This opens the door to applying ML forecasting to fundamental valuation techniques, with the FCF growth rate being of principal importance for this task. Further research involves use of more algorithms and modified versions of the algorithms examined in this paper.

References

[1] Anand, V., Brunner, R., Ikegwu, K., Sougiannis, T., 2019. Predicting profitability using machine learning. Available at SSRN 3466478 .
[2] Bishop, C.M., Nasrabadi, N.M., 2006. Pattern recognition and machine learning. Springer.
[3] Brooks, L., Buckmaster, D., 1979. Price-change accounting models and disaggregated monetary gains and losses. Quarterly Review of Economics and Business (Spring 1979) , 115–130.
[4] Cavanaugh, J.E., Neath, A.A., 2019. The akaike information criterion: Background, derivation, properties, application, interpretation, and refinements. Wiley Interdisciplinary Reviews: Computational Statistics 11, e1460.
[5] Chen, X., Cho, Y.H., Dou, Y., Lev, B., 2022. Predicting future earnings changes using machine learning and detailed financial data. Journal of Accounting Research 60.
[6] Damodaran, A., 2012. Investment valuation: Tools and techniques for determining the value of any asset. John Wiley & Sons.
[7] Etemadi, H., Ahmadpour, A., Moshashaei, S.M., 2015. Earnings per share forecast using extracted rules from trained neural network by genetic algorithm. Computational Economics 46.
[8] Evdokimov, I., 2022. Edgar data facts to python dictionary. URL: https://github.com/ivan020/EDGAR_data, doi:10.5281/zenodo.7418203.
[9] Fairfield, P.M., Ramnath, S., Yohn, T.L., 2009. Do industry-level analyses improve forecasts of financial performance? Journal of Accounting Research 47.
[10] Goodfellow, I., Bengio, Y., Courville, A., 2016. Deep learning. MIT press.
[11] Graham, B., Dodd, D.L.F., Cottle, S., et al., 2008. Security analysis. McGraw-Hill New York.
[12] Hayashi, F., 2011. Econometrics. Princeton University Press.
[13] Jiang, Y., Jones, S., 2018. Corporate distress prediction in china: A machine learning approach. Accounting & Finance 58.
[14] Koller, T., Goedhart, M., Wessels, D., et al., 2020. Valuation: measuring and managing the value of companies. John Wiley & Sons.
[15] Kumbure, M.M., Lohrmann, C., Luukka, P., Porras, J., 2022. Machine learning techniques and data for stock market forecasting: a literature review. Expert Systems with Applications , 116659.
[16] Kureljusic, M., Reisch, L., 2022. Revenue forecasting for european capital market-oriented firms: A comparative prediction study between financial analysts and machine learning models. Corporate Ownership & Control 19.
[17] McKenzie, W., 2009. FT guide to using and interpreting company accounts. Pearson Education.
[18] Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., Duchesnay, E., 2011. Scikit-learn: Machine learning in Python. Journal of Machine Learning Research 12, 2825–2830.
[19] Pouriyeh, S., Vahid, S., Sannino, G., De Pietro, G., Arabnia, H., Gutierrez, J., 2017. A comprehensive investigation and comparison of machine learning techniques in the domain of heart disease, in: 2017 IEEE symposium on computers and communications (ISCC), IEEE. pp. 204–207.
[20] Pratama, B.Y., Sarno, R., 2015. Personality classification based on twitter text using naive bayes, knn and svm, in: 2015 international conference on data and software engineering (ICoDSE), IEEE. pp. 170–174.
[21] Sharpe, W.F., 1977. The capital asset pricing model: a "multi-beta" interpretation, in: Financial Dec Making Under Uncertainty. Elsevier, pp. 127–135.
[22] Singh, A., Halgamuge, M.N., Lakshmiganthan, R., 2017. Impact of different data types on classifier performance of random forest, naive bayes, and k-nearest neighbors algorithms. International Journal of Advanced Computer Science and Applications 8.
[23] Vayas-Ortega, G., Soguero-Ruiz, C., Rojo-Álvarez, J.L., Gimeno-Blanes, F.J., 2020. On the differential analysis of enterprise valuation methods as a guideline for unlisted companies assessment (i): Empowering discounted cash flow valuation. Applied Sciences 10.
[24] Voit, J., Lourie, R.W., 2002. The statistical mechanics of financial markets. Physics today 55, 080000–52.
[25] Zhu, L., Yan, M., Bai, L., 2022. Prediction of enterprise free cash flow based on a backpropagation neural network model of the improved genetic algorithm. Information 13.