UNIMORE
UNIVERSITÀ DEGLI STUDI DI MODENA E REGGIO EMILIA

Dipartimento di Economia Marco Biagi

DEMB Working Paper Series

N. 184

A comparison of machine learning methods for predicting stock returns in the US market

Silvia Muzzioli[1], Giovanni Campisi[2], Bernard De Baets[3]

January 2021

[1] University of Modena and Reggio Emilia, Department of Economics Marco Biagi
E-mail: silvia.muzzioli@unimore.it

[2] University of Modena and Reggio Emilia, Department of Economics Marco Biagi
E-mail: giovanni.campisi@unimore.it

[3] Ghent University, Department of Mathematical Modelling, Statistics and Bioinformatics
E-mail: Bernard.DeBaets@UGent.be

ISSN: 2281-440X online

A comparison of machine learning methods for predicting stock returns in the US market

Silvia Muzzioli* **Giovanni Campisi †** **Bernard De Baets ‡**

Abstract

In this paper we investigate the information content of option-based indicators to predict future stock returns. To this end, we apply different machine-learning techniques. The data set consists of stock index returns and option-based indicators of the US stock market from October 2014 to September 2019. The goal is to achieve a good prediction accuracy out of sample.

Keywords: Machine learning; Volatility indices; Market risk.

1 Introduction

Stock market prediction has always been an important issue in finance (Giot (2005), Rubbaniy *et al.* (2014), Lubnau and Todorova (2015), Gonzalez-Perez (2015), Elyasiani *et al.* (2017)). Unfortunately, although the numerous scientific attempts, none of the methodology has accurately predicted future stock returns. However, in the last decades the amount of information at disposal to researchers has increased enormously and this has had a significant impact on stock markets. Moreover, new efficient decision-making algorithms have become increasingly common in literature, this is the case of machine learning algorithms. Therefore, combining these two elements, we contribute to the ongoing literature on empirical asset pricing linking the role of machine learning methods to the information contained in risk indices.

The aim of the paper is to make extensively use of machine learning (ML) techniques to model and analyze the direction of $S\&P500$ stock returns using different risk indices. Risk indices, such as implied volatility indices, are essential for asset pricing and risk management since they contain information embedded in option prices which reflect the investors' opinion about future underlying asset evolution. The most part of the papers dealing with the use of volatility indices in predicting stock returns have

*Marco Biagi Department of Economics, University of Modena and Reggio Emilia, Viale Jacopo Berengario 51, 41121, Modena, Italy, e-mail: silvia.muzzioli@unimore.it

†Marco Biagi Department of Economics, University of Modena and Reggio Emilia, Viale Jacopo Berengario 51, 41121, Modena, Italy, e-mail: giovanni.campisi@unimore.it

‡Department of Mathematical Modelling, Statistics and Bioinformatics, Ghent University, Belgium and KERMIT, e-mail: Bernard.DeBaets@UGent.be

extensively used linear regressions (see Rubbaniy *et al.* (2014) and Elyasiani *et al.* (2017), for example) and only recently quantile regressions (Ma and Pohlman (2008)). Furthermore, while the role of implied volatility indices has been widely studied using traditional techniques, there is a lack of works combining the volatility indices and machine learning methods. At the best of our knowledge, only Rosillo *et al.* (2014) analyzed the impact of VIX on the directional weekly movement of the $S\&P500$. However, the authors consider only Support Vector Machines and include VIX and other technical indicators in their study. In our work we rely on 11 relevant risk indices in financial literature applying 10 different machine learning techniques.

The use of machine learning in finance is spreading widely lately, with few some exceptions (Hutchinson *et al.* (1994), Yao *et al.* (2000)). It has been demonstrated that machine learning algorithms are very efficient methods in predicting stock returns with respect to traditional methods, in particular classical regression methods. Indeed, these methods are more flexible than more traditional econometric prediction techniques (Gu *et al.* (2020)). Machine Learning methods are data-driven and they concern with the development of algorithms. One main difference between ML and traditional statistical methods lies in their purpose, as the former remains focused on making predictions as accurate as possible, while the latter are aimed at inferring relationships between variables (Athey and Imbens (2019)).

There is strong evidence that volatility indices provide useful information about current and future stock returns. To this end, Giot (2005) argues that high implied volatility levels indicate oversold markets and could be viewed as short- to medium-term buy signals. Zhu (2013) investigates the US stock and bond returns using a distribution-based framework and finds evidence that the VIX helps to forecast the US stock returns distribution. Gonzalez-Perez (2015) provides a comprehensive literature review of forecasting volatility models. Other examples of works highlighting the importance of volatility indices as indicators of future stock returns are Elyasiani *et al.* (2017), Lubnau and Todorova (2015), Rubbaniy *et al.* (2014), for example.

Our work also builds on the recent strand of the literature predicting stock returns as a function of many characteristics at once. Recently, Patel *et al.* (2015) address the problem of predicting stock direction for stocks and the stock index in the Indian stock market using Artificial Neural Network, Support Vector Machine, random forest and naive-Bayes using 10 years of historical data. They found that random forest and naive-Bayes performed better in terms of accuracy. Gu *et al.* (2020) provide a comparative analysis of machine learning methods applied to the two canonical problems of empirical asset pricing, i.e. predicting returns in cross-section and time series. They find that machine learning methods enhance the performance relatively to traditional forecasting methods. Bracke *et al.* (2019) develop a systematic analytical framework for approaching explainability questions in real world financial applications. In particular they use the Quantitative Input Influence method of Datta *et al.* (2016) for predicting mortgage defaults. Ryll and Seidens (2019) presents and analyses a vast array of literature on machine learning applications for financial time series analysis. Gerlein *et al.* (2016) analyse the role of simple machine learning models to achieve profitable trading through a series of trading simulations in the FOREX market. The authors focus on the choice of the optimal combination of attributes to enhance the classification performance of the models.

Different strategies will be considered in order to learn the most important indicators among the ones

considered. In our study, seven different Machine Learning techniques have been discussed and applied to predict price movement. Starting with LDA (Linear Discriminant Analysis), we also consider Logistic Regression, Ridge, Lasso, Bagging, Random Forest and Boosting. We explore the power of Machine Learning techniques to analyze the role of different risk indices in financial markets.

As in Gu *et al.* (2020), our purpose is to describe an asset's excess return as an additive prediction error model. For this purpose we have considered two types of models: classification and regression. In the first case, the prediction focus on the direction of the S&P500 stock market returns movements (i.e. rise or fall) and the target variable is discrete. This is a two-class classification problem and we analyze it with LDA and Logistic Regression. In the second case, the response variable is continuous and the focus is on forecasting the change of the S&P500 returns with 30 days time to maturity. We exploit this model with shrinkage methods: Ridge and Lasso. Additionally, due to their higher performance in terms of accuracy, we employ Ensamble methods in both classification and regression models. In particular, we consider Bagging, Random Forest and Boosting. Finally, in evaluating the performance of all the models we transforming the results of the continuous variables into a binary variable in order to choose the models that give us the best fit to the data in terms of AUC (are under the curve) and test error rate.

The reminder of the paper is organized as follows. In Section 2, we describe our dataset. In Section 3, we outline the model specifications and investigate the forecasting power of the different methods proposed. Section 4 provides results and associated comments. The last section concludes.

2 Data

The data source of this paper is based on the Bloomberg database. We use daily data on 11 input variables: VIX, VIX9D, VIX3M, VIX6M, VVIX, SKEW, VXN, VIX3M/VIX, GVZ, OVX, PUTCALL. The data cover a period from October 2014 to September 2019 of total 1232 daily observations. Table (1) shows the summary statistics of our dataset. In the following, we provide the particular detail of the selected attributes. The *VIX* is the CBOE volatility index computed by the bid and ask prices from the cross section of $S\&P500$ options[1]. The *VIX9D*, is the CBOE $S\&P500$ 9-Day Volatility Index and estimates the expected 9-day volatility of $S\&P500$ stock returns. The *VIX3M* is the CBOE 3-Month Volatility Index that is designed to be a constant measure of 3-month implied volatility of the $S\&P500$ Index options[2]. The *VIX6M*, represents the CBOE $S\&P500$ 6-Month Volatility Index and is an estimate of the expected 6-month volatility of the $S\&P500$ Index. It is calculated using the well-known VIX methodology applied to SPX options that expire 6-to-9 months in the future. The CBOE *VVIX* Index represents a volatility of volatility in the sense that it measures the expected volatility of the 30-day forward price of VIX. This forward price is the price of a hypothetical VIX futures contract that expires in 30 days. The CBOE SKEW Index estimates the skewness of $S\&P500$ returns at the end of a 30-day horizon. Similar to VIX the price of $S\&P500$ tail risk is calculated from the prices of $S\&P500$ out-of-the-money options. *SKEW* typically ranges from 100 to 150. Values above the threshold level 100 tend to point to a negative risk-neutral skewness and a distribution skewed to the left (i.e. negative

[1]More details can be found at www.cboe.com.

[2]On September 18, 2017 the ticker symbol for the CBOE 3-Month Volatility Index was changed from "VXV" to "VIX3M"

returns are more often expected than positive returns). For values below 100 it indicates a positive risk-neutral skewness and a distribution skewed to the right (i.e. positive returns are more often expected than negative returns). The CBOE NASDAQ-100 Volatility Index (VXN) is a key measure of market expectations of near-term volatility conveyed by NASDAQ-100 Index (NDX) option prices. It measures the market's expectation of 30-day volatility implicit in the prices of near-term NASDAQ-100 options. The VXN is quoted in percentage points. The ratio *VIX3M/VIX* provides useful information on the term structure of $S\&P500$ option implied volatility (i.e. the average volatility that traders expect to prevail over non-overlapping time intervals). The CBOE Gold ETF Volatility Index ("Gold VIX", Ticker - GVZ) measures the market's expectation of 30-day volatility of gold prices by applying the VIX methodology to options on SPDR Gold Shares (Ticker - GLD). The Cboe Crude Oil ETF Volatility Index ("Oil VIX", Ticker - OVX) measures the market's expectation of 30-day volatility of crude oil prices by applying the VIX methodology to United States Oil Fund. The *PUT/CALL* is a proportion between all the put options and all the call options purchased on any given day. *Returns30* are the $S\&P500$ daily returns at time t that refer to a window of 30 days.

Figure 1 demonstrates the time series of the *Returns30* variable, which represents the response variable of all the regression models we use in our analysis. In particular, the returns fluctuate around the mean value of zero and display the phenomenon of volatility clustering (i.e. consecutive large volatility periods alternating with several consecutive periods of limited volatility).
The summary statistics of the variables have been reported in Table 1. As can be seen, of the 12 variables considered, *SKEW*, *VVIX* and *OVX* posses the highest mean value. We observe the lowest mean value for *Returns30*, *PUTCALL* and $VIX3M\_VIX$ variables. The test for skewness and kurtosis reveals that *Returns30* and $VIX3M\_VIX$ are skewed to the left (or negatively skewed), which means that the tail of the left side of the probability density function is longer than the right side and the majority of the values are situated to the right of the mean. All other indices are positively skewed. Moreover, the *VIX9D* and the *VVIX* are leptokurtosis (i.e. their distribution displays fat tails compared to the normal distribution), whereas all other variables are platykurtosis (i.e. they have fatter middles or fewer extreme values). The reported autocorrelations show that all variables are highly persistent. We also investigate if all the series are stationary by employing the augmented Dickey-Fuller (ADF) unit-root test. We conclude that all the series are stationary, indeed the series display the rejection of unit roots at the 1% level. Finally, the Jarque-Bera test rejects the null hypothesis at the 1% significance level for each of the twelve variables, since the test statistic, J-B statistic, is greater than the critical value, which is 5.8461.

From Table 1 it is evident that *SKEW*, *VVIX* and *OVX* exhibit higher mean values than all other variables. In this respect, we have proceeded to standardize the dataset in order to maintain all variables on the same scale. Moreover, this is necessary also because we are using methods involving distances in the loss function (in particular, lasso and ridge) resulting in estimations that are dependent on the scale of the predictors. Therefore, before applying the learning algorithms standardization is needed (see James *et al.* (2013)). After standardization all variables have a mean of 0 and standard deviation of 1, whereas all other statistics reported in Table 1 remain the same.

In Table 2 we provide the correlations between all predictors. We can observe that the most part of variables is moderately correlated. However, as can be expected, the correlation between volatility

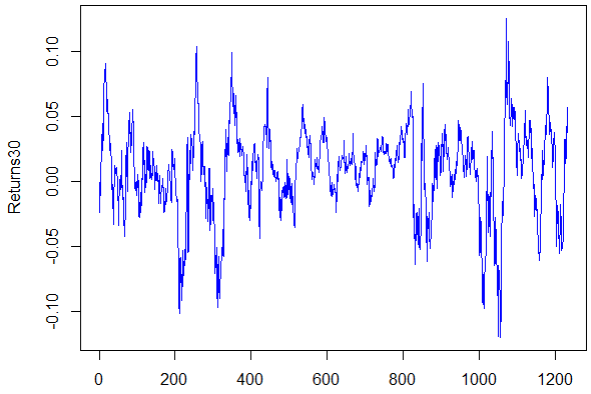

Figure 1: Time series of $S\&P500$ 30-days log-returns

indices at different time intervals (*VIX, VIX9D, VIX3M, VIX6M, VIX3M_VIX*) is very high. Although correlation does not imply causation, we need to exclude the problem of multicollinearity in our analysis. In order to overcome the presence of multicollinearity and to select representative features for prediction, we use Lasso regression which performs both parameter shrinkage and variable selection, generating more stable and interpretable estimates in models with a large number of regressors (Rapach *et al.* (2013)).

Table 1: Summary statistics

Statistic	N	Mean	St. Dev.	Skewness	Kurtosis	ρ_1	ρ_2	ρ_3	ADF	JB statistic
VIX	1,233	15.176	4.289	1.461	3.108	0.933***	0.872***	0.820***	−14.5632***	939.0446
VIX9D	1,233	14.622	5.645	2.003	7.270	0.891***	0.806***	0.738***	−14.3858***	3554.5864
VIX3M	1,233	16.975	3.257	0.988	1.066	0.956***	0.917***	0.886***	−17.6622***	260.1662
VIX6M	1,233	18.259	2.626	0.805	0.599	0.966***	0.935***	0.909***	−17.3656***	152.127
VVIX	1,233	94.128	12.723	1.800	6.250	0.897***	0.798***	0.715***	−14.1254***	2684.1922
SKEW	1,233	129.214	8.437	0.404	-0.240	0.877***	0.826***	0.780***	−14.5625***	36.4302
VXN	1,233	18.104	4.534	1.333	1.995	0.945***	0.895***	0.859***	−15.3199***	572.4133
VIX3M_VIX	1,233	1.143	0.099	-0.538	0.137	0.911***	0.830***	0.756***	−14.4138***	60.6049
GVZ	1,233	14.624	3.494	0.576	-0.305	0.971***	0.950***	0.932***	−12.9092***	72.9266
OVX	1,233	36.681	10.554	0.933	0.378	0.982***	0.965***	0.949***	−10.7339***	186.6909
PUTCALL	1,233	0.962	0.088	0.643	0.456	0.969***	0.916***	0.848***	−11.1408***	96.0166
Returns30	1,233	0.007	0.033	-0.570	1.012	0.936***	0.873***	0.813***	−12.3499***	120.2973

Table 2: Correlation matrix

	VIX	VIX9D	VIX3M	VIX6M	VVIX	SKEW	VXN	VIX3M_VIX	GVZ	OVX	PUTCALL	Returns30
VIX	1	0.968	0.967	0.907	0.667	-0.249	0.823	-0.867	0.322	0.536	0.607	-0.644
VIX9D	0.968	1	0.905	0.833	0.705	-0.236	0.766	-0.867	0.271	0.451	0.556	-0.647
VIX3M	0.967	0.905	1	0.978	0.579	-0.253	0.810	-0.744	0.448	0.662	0.620	-0.586
VIX6M	0.907	0.833	0.978	1	0.499	-0.252	0.747	-0.632	0.555	0.734	0.604	-0.530
VVIX	0.667	0.705	0.579	0.499	1	0.116	0.480	-0.649	0.117	0.094	0.340	-0.515
SKEW	-0.249	-0.236	-0.253	-0.252	0.116	1	-0.300	0.241	-0.143	-0.278	-0.190	0.172
VXN	0.823	0.766	0.810	0.747	0.480	-0.300	1	-0.723	0.128	0.423	0.578	-0.540
VIX3M_VIX	-0.867	-0.867	-0.744	-0.632	-0.649	0.241	-0.723	1	-0.055	-0.248	-0.495	0.597
GVZ	0.322	0.271	0.448	0.555	0.117	-0.143	0.128	-0.055	1	0.672	0.262	-0.049
OVX	0.536	0.451	0.662	0.734	0.094	-0.278	0.423	-0.248	0.672	1	0.400	-0.284
PUTCALL	0.607	0.556	0.620	0.604	0.340	-0.190	0.578	-0.495	0.262	0.400	1	-0.632
Returns30	-0.644	-0.647	-0.586	-0.530	-0.515	0.172	-0.540	0.597	-0.049	-0.284	-0.632	1

3 Empirical model

In this section we explore the power of ML techniques to analyze the role of different risk and sentiment indices in financial markets. In our analysis we consider the problem of forecasting the stock index returns by using different indicators in two settings: classification and regression. In the classification types of problem we consider: logistic regression, linear discriminant analysis, random forests, bagging and gradient boosting. In the regression problem we use: lasso, ridge, random forests, bagging and gradient boosting.

To maintain our approach as general as possible, we first describe the model in its general form, then we illustrate the characteristic of each method outlined in detail.

Following Gu *et al.* (2020) we aim to describe an asset's excess return as an additive prediction error model:

$$r_{t+1} = \mathbb{E}_t(r_{t+1}) + \epsilon_{t+1} \tag{1}$$

where

$$\mathbb{E}_t(r_{t+1}) = g^*(z_t) \tag{2}$$

r_{t+1} is the S&P500 stock return at time $t+1$ with $t = 1, ..., T$. Our objective is to apply a statistical learning method in order to find a function $g^*(z_t)$ expressed in terms of the predictor variables, z_t, that maximizes the out-of-sample explanatory power for the realized return at time $t+1$, r_{t+1}. The function $g^*(z_t)$ represents the conditional expectation of r_{t+1} and it is a flexible function of the predictors z_t.

The output variable differs with respect to the method used. Indeed, we admit that our variable can be binary or continuous. In the former case, we face a classification problem and our goal is to predict if the market is bearish (i.e. returns go down) or bullish (i.e. returns go up). When our response variable is continuous we consider our problem as a regression set up in order to forecast the S&P500 returns with 30 days time to maturity. However, the two problems can be considered as a whole, transforming the results of the continuous variables into a binary variable in order to choose the model that give us the best fit to the data.

To avoid overfitting, we rely on the choice of a set of hyper-parameters (also called tuning parameters), i.e. those parameters that maximize the model performance allowing to control model complexity. Following the most common approach in the literature we select tuning parameters from the data using k-fold cross validation. In particular, we randomly divide our dataset into k folds of approximately equal size. First, we train the model on $k-1$ subsets and then we test it on the remaining one subset. Finally, we calculate the test error (i.e. the mean squared error). This procedure is repeated k times obtaining k estimates of the test error. The k-fold cross validation estimate is computed by averaging these values (see James *et al.* (2013) for futher details).

We use the open source software package R to run our machine-learning models. In the following, we give an overview of the two problems.

3.1 Classification models

In our classification models we consider the response variable as binary: it represents positive and negative S&P500 stock returns. We consider a two-class classification problem where the training algorithm

consists of pairs $(\mathbf{x}, y)$ with $\mathbf{x} \in X$ is the feature vector, and $y \in (0, 1)$ is the response variable and it represents $Pr(y = 1|X)$, that is the probability that the response variable belong to the category 1 given X.

Logistic regression models the probability that y belongs to a particular category. In our case, we face a binary classification problem and the response variable falls into one of two categories: bearish or bullish. Consequently, if we label the class 1 as bullish and the class 0 as bearish, then $Pr(y = 1|X)$ represents the probability that the market is bullish given X. The log-likelihood function is expressed as follows:

$$\ell(\beta) = \sum_i [y_i \log p(\mathbf{x_i}) + (1 - y_i) \log(1 - p(\mathbf{x_i}))] \tag{3}$$

where $p(\mathbf{x_i}) = \frac{e^{\mathbf{x_i}\beta}}{1+e^{\mathbf{x_i}\beta}}$.

Unlike logistic regression, linear discriminant analysis (LDA) uses a different estimation procedure to estimate parameters. In particular, logistic regression relies on maximum likelihood whereas LDA fits all parameters using the first two moments extracted from a multivariate Normal distribution. With LDA it is assumed that each predictor, $\mathbf{x_i}$ is drawn from a multivariate normal distribution. Formally, consider a $n-$dimensional random variable X then we assume that $X \sim N(\mu, \Sigma)$ where $E(X) = \mu$ is the mean of X, and $Cov(X) = \Sigma$ is the nxn variance-covariance matrix of X. In both methods, the model to analyse takes the following form:

$$p(\mathbf{x_i}) = \frac{e^{\mathbf{x_i}\beta}}{1 + e^{\mathbf{x_i}\beta}} = \log\left(\frac{p(\mathbf{x_i})}{1 - p(\mathbf{x_i})}\right) = \mathbf{x_i}\beta \tag{4}$$

3.2 Regression models

The first two regression models we consider, i.e. ridge and lasso, fall into the category of skrinkage methods. Indeed, both methods allow us to fit a model involving all the predictor provided that the estimated coefficient are constrained or shrunken towards zero. These methods, also called penalized methods, append a penalty to the original loss function. There are several choices for the penalty function, we rely on the ridge and lasso penalty. These regularization approaches have the advantage of reducing variance. Moreover, depending on what type of shrinkage is performed, some of the coefficients may be estimated to be exactly zero (in particular, in the case of lasso).
Following Gu *et al.* (2020), in order to provide a description of the ridge and the lasso methods, we start from the baseline model of the linear regression. We consider to approximate the conditional expectation that have a linear form:

$$g(z_t; \theta) = z_t^{'}\theta \tag{5}$$

The traditional method for estimating the regression function is least squares, resulting in the following objective function:

$$\mathcal{L}(\theta) = \frac{1}{T}\sum_{t=1}^{T}(r_{t+1} - g(z_t; \theta))^2 \tag{6}$$

Minimizing $\mathcal{L}(\theta)$ yields the OLS estimator.
Instead of directly optimizing (6), a term (i.e. a penalty) is added to the objective function to penalize

the complexity of the model. One common form of regularization is to add a penalty term to the original loss function:

$$\mathcal{L}(\theta;\rho) = \mathcal{L}(\theta) + \phi(\theta;\rho) \tag{7}$$

According to the functional form given to the penalty function we can distinguish between the ridge and the lasso. In particular, we consider the following functional form of $\phi(\theta;\rho)$:

$$\phi(\theta;\rho) = \rho\|\theta\|_k \tag{8}$$

where $\rho > 0$ is the tuning parameter. This parameter serves to control the relative impact of the regression coefficient estimates. When $\rho = 0$, the penalty function 8 has no effect in the objective function 7 and we find the same results of the least squares estimates. When $\rho \to \infty$, the coefficient estimates approach to zero. The notation $\|\theta\|_k$ denotes the l_k norm of a vector, and is defined as $\|\theta\|_k = \sum_{i=1}^{N} |\theta_i|^k$. The $k = 1$ corresponds to the lasso. For $k = 2$, this corresponds to the ridge regression. Both methods differ to the choice of the penalty function. However, there is an important difference among these two models. While lasso lead to solutions with a number of the regressor coefficients exactly equal to zero, in ridge regression all of the estimated regression coefficients generally differ from zero. In this respect, we say that the lasso yields sparse models referring to models that involve only a subset of the variables while ridge regression is a shrinkage method that helps prevents coefficients from becoming excessively large in magnitude. We adaptively optimize the tuning parameter using k-fold cross validation.

3.3 Ensemble methods

Ensemble methods are aimed at combining multiple weak learning algorithm to produce a strong learning algorithm. They belong to the class of *decision tree* methods because the predictor space is segmented into a number of simple regions that can be summarize in a tree. These methods can be applied to both classification and regression models. In our work we consider three types of ensemble learning approaches: bagging (or bootstrap aggregation), random forests and boosting.
Ensemble methods aim at generating multiple version of a predictor and using these to get an aggregated predictor. This procedure leads to get better prediction accuracy.

Bagging (Breiman (1996)) aggregates many predictions from different training set of the population. Given that, in reality, only one training set is available we fit classification or regression models to bootstrap samples from the unique training data and combine by majority vote (in classification), i.e. the commonly occurring class among the B predictions, or averaging over individual tree predictions (in regression). Unlike previous models analyzed, with bagging there is no need to perform k-fold cross validation to obtain the test error. This error is obtained from the bootstrap procedure taking into consideration the observations that are not considered in the fit of the model, i.e. the out-of-bag (OOB) observations. For each observation we get a single prediction by averaging the predicted responses (in regression) or by taking the majority vote (in classification). Finally we obtain the overall OOB MSE or classification error depending on the model used, i.e. regression or classification respectively.

Random forests (Breiman (2001)) is a variant of bagging. In particular, it involves the same bootstrap procedure to generate samples from the original dataset. However, at each step of splitting, in random

forests we use only a random subsample (usually $M = \sqrt{P}$) of all features, P. For instance, if a random forest is built using $M = P$, then we lead to bagging. The final random forest output is given by the average of the outputs of all B trees:

$$\hat{g}(z_t; d, B) = \frac{1}{B} \sum_{b=1}^{B} \hat{g}_b(z_t; \hat{\theta}_b, d) \tag{9}$$

where B is the number of trees and d is the number of splits in each tree which allows to control for the complexity of the model.

Boosting (Schapire (1990) and Freund (1995)) is a variant of both bagging and random forests. Unlike these methods, boosting does not involve bootstrap sampling. Instead, it allows each tree to grow using information from previously grown trees. In this respect, the trees grow sequentially, indeed boosting is a method that learns slowly (James *et al.* (2013). It consists of fitting a decision tree using the residuals as response variable. Then, this new decision tree is added into the fitted function in order to update the residuals. In the update process the residual forecast is added to the total with a shrinkage weight of λ. This parameter controls the rate at which boosting learns. This procedure is iterated B times, which corresponds to the number of trees. At each new step b, the tree is fitted to the residuals from the model with $b-1$ trees and then used to update the model ($\hat{g}_b$). The final model output is

$$\hat{g}_B(z_t; B, \lambda, d) = \sum_{b=1}^{B} \lambda \hat{f}_b(\cdot) \tag{10}$$

where (B, λ, d) are the tuning parameters which we adaptively choose in the k-fold cross validation procedure. In particular, B is the number of trees, λ is the shrinkage parameter and d is the number of splits in each tree which allows to control for the complexity of the boosted ensemble.

3.4 Performance measures

To asses the predictive performance of our models, we rely on two metrics: the classification error rate and the area under the receiver operating characteristic curve (AUC). Prediction accuracy is based on a confusion matrix shown in Table 3. The classification error rate is defined in terms of accuracy. The accuracy of a model is calculated by the diagonal elements of the confusion matrix and represents the correct classification by the classifier,i.e.:

$$\text{Accuracy} = \frac{TP + TN}{TP + FP + TN + FN} \tag{11}$$

Then, the classification error rate is given by:

$$\text{Classification error} = 1 - \text{Accuracy} = \frac{FP + FN}{TP + FP + TN + FN} \tag{12}$$

The second measure we use is the AUC which is a ranking-based measure of classification performance. Its value can be interpreted as the probability that a classifier is able to distinguish a randomly chosen positive example from a randomly chosen negative example. In contrast to many alternative performance

Table 3: confusion matrix

		Predicted	
		Positive	Negative
Actual	Positive	True positive (TP)	False Negative (FN)
	Negative	False positive (FP)	True Negative (TN)

measures, AUC is invariant to relative class distributions and class-specific error costs (see Airola *et al.* (2009)). AUC values range from 0.5, for a classifier with no predictive value, to 1, for a perfect classifier.

In line with Airola *et al.* (2009), we define the AUC using the following formula:

$$A(S, f_Z) = \frac{1}{|S_+||S_-|} \sum_{x_i \in S_+} \sum_{x_j \in S_-} H(f_Z(x_i) - f_Z(x_j)) \tag{13}$$

where H is the Heaviside step function, for which $H(a) = 1$ if $a > 0$, $H(a) = 1/2$ if $a = 0$ and $H(a) = 0$ if $a < 0$. f_Z is a prediction function returned by a learning algorithm based on a fixed training set Z. S is a sequence of examples, with $S_+ \subset S$ and $S_- \subset S$ denote the positive and negative examples in S, respectively.

4 Discussion of the results

In our work we compare ten models in total, including Logistic regression, Linear Discriminant Analysis (LDA), Ridge, Lasso, Random Forest, Bagging and Boosting. The ensemble methods have been employed in both classification and regression models.

Regarding the choice of the variables, we used all market risk indicators that are extensively used in financial literature proving that these indices has a significant role in predicting stock market returns. Indeed, there is strong evidence that volatility indices provide useful information about current and future stock returns. In this respect, Rubbaniy *et al.* (2014), Giot (2005) examine the predictive power of VIX and VXN on the underlying index returns. Mora-Valencia *et al.* (2021) find that the Skew Index reveal salient information for expected financial downturns. Regarding the role of OVX, Kang *et al.* (2015) find that OVX impacts stock returns in the US market. Moreover, Dutta *et al.* (2020) underline the crucial role of OVX in asset pricing and risk analysis and its capability in forecasting crude oil returns. Gokmenoglu and Fazlollahi (2015) find evidence that volatility in one market affect the price index of the other market using GVX and OVX. Finally, Simon and Wiggins III (2001) investigates the predictive power of VIX, PUTCALL and tradign index on the $S\&P500$ future contract obtaining statistically and economically significant forecasting power. However, none of the works consider all our indices in their analysis neither in a machine learning set-up. Instead, we use these risk indices predictors to forecast the direction of the $S\&P500$ stock returns combining 10 machine learning algorithms.

Based on the results of the Lasso, after features selection we excluded three variables: VIX3M, VIX6M and VIX3M_VIX. Tables 4 and 5 presents the comparison of machine learning techniques in terms of their test error rate and AUC before and after features selection with Lasso was performed. In line

with Smialowski *et al.* (2010) and Zhang *et al.* (2014), we employ a supervised preprocessing of the data considering the general characteristic of the training set to select the key regressors independently: as a results, the test set was not used in this procedure. After features selection the results of our models do not change significantly. First of all, considering the test error as performance measure, looking at Table 5, Random Forest significantly outperforms all the other classifiers in both classification and regression models. Gradient Boosting in the regression model performs the worst, whereas LDA attains the highest test error in classification model. According to the AUC measure, Ridge reaches the lowest value of 0.8939 followed by Lasso with AUC equals to 0.8951. In the classification model, the lowest values are those of Logistic regression and LDA with AUC equal to 0.90 and 0.9047, respectively. In both models, Bagging possesses the highest AUC apart from Random Forest. In general, all the classifiers applied in the classification model outperform those applied in the regression model.

Predicting the direction of the stock market returns is of interest for most investors and for this purpose many studies have faced this problem. It is interesting to compare our results with other related studies that use machine learning algorithms for the same purpose. In particular, we can observe that our machine learning algorithms allow us to obtain results in term of accuracy higher than other study using machine learning to predict the direction of the stock market. For example, Khan *et al.* (2020) use algorithm on social media and financial news data to discover their impact on stock market prediction accuracy. In their experiments, they found that Random forest classifier reaches the highest accuracy of 0.8322 which is lower than the accuracy we found in our work (i.e. $1 - 0.1396 = 0.8604$). Hu *et al.* (2018) use an improved back propagation neural network for predicting the directions of the opening stock prices for the $S\&P500$ and they obtain an hit ration of 0.8681. Schumaker and Chen (2009) estimate a discrete stock price twenty minutes after a news article was release, they use support vector machine obtaining an accuracy of 0.571. Moreover, our results in term of accuracy are in line with the studies in other markets reaching an accuracy level above 0.8, such as Patel *et al.* (2015) in the Indian stock market and Malagrino *et al.* (2018) in the Brazilian stock market.

Table 4: Test error and AUC before feature selection

ML method	test error	AUC
logistic regression	0.1494	0.9043
lda	0.1753	0.9062
Random Forest classification	0.1169	0.9432
Bagging classification	0.1234	0.9394
Gradient Boosting classification	0.1331	0.9261
Random Forest regression	0.1333	0.9393
Bagging regression	0.1266	0.9468
Gradient Boosting regression	0.1721	0.9212
Ridge regression	0.1818	0.8958
Lasso regression	0.1753	0.8951

Table 5: Test error and AUC after feature selection

ML method	test error	AUC
logistic regression	0.1461	0.90
lda	0.1623	0.9047
Random Forest classification	0.1136	0.9392
Bagging classification	0.1201	0.9334
Gradient Boosting classification	0.1364	0.9154
Random Forest regression	0.1396	0.9347
Bagging regression	0.1429	0.937
Gradient Boosting regression	0.1883	0.9114
Ridge regression	0.1688	0.8939
Lasso regression	0.1753	0.8951

5 Conclusions

The purpose of our study is to examine the forecasting power of market risk indices on future stock returns using machine learning algorithms. In particular, we have considered two basic models: regression and classification. The response variable in the regression model is continuous and we forecast the $S\&P500$ returns with 30 days time to maturity. In the classification model, the response variable is binary and we predict if the market is bearish (i.e. returns go down) or bullish (returns go up). Moreover, in order to compare our results in both models, we have transformed the results of continuous variables into a binary variable. We have compared our results in terms of two well known measures: classification error (called also test error) and AUC (area under the curve).

We have underlined the role of each method used in our work describing its characteristic. Moreover, we have motivated the use of our predictors observing that they are all extensively used in all relevant empirical finance analyses.

In order to solve possible problem of multicollinearity we have performed a feature selection using Lasso regression. As a consequence, three input variables were eliminated from our analysis: VIX3M, VIX6M and VIX3M_VIX. The feature selection procedure are conducted following the recent literature that suggests to exclude the test set in this stage.

The results show that Random Forest attains the highest performance followed by Bagging and in general, all the classifiers applied in the classification model outperform those applied in the regression model. Moreover, when compare our results with other related studies, we find that our performance is higher. In addition, we have highlighted that our results are also in line with other analysis conducted in other relevant financial markets, such as the Indian and the Brazilian markets.

Acknowledgements

The authors gratefully acknowledge financial support from University of Modena and Reggio Emilia for the FAR2019 project. We also wish to extend our thanks to William Bromwich for his painstaking attention to the copy-editing of this paper.

References

Airola, A., Pahikkala, T., Waegeman, W., De Baets, B., and Salakoski, T. (2009). A comparison of auc estimators in small-sample studies. In *Machine learning in systems biology*, pages 3–13.

Athey, S. and Imbens, G. W. (2019). Machine learning methods that economists should know about. *Annual Review of Economics*, **11**, 685–725.

Bracke, P., Datta, A., Jung, C., and Sen, S. (2019). Machine learning explainability in finance: an application to default risk analysis.

Breiman, L. (1996). Bagging predictors. *Machine learning*, **24**(2), 123–140.

Breiman, L. (2001). Random forests. *Machine learning*, **45**(1), 5–32.

Datta, A., Sen, S., and Zick, Y. (2016). Algorithmic transparency via quantitative input influence: Theory and experiments with learning systems. In *2016 IEEE symposium on security and privacy (SP)*, pages 598–617. IEEE.

Dutta, A., Bouri, E., and Roubaud, D. (2020). Modelling the volatility of crude oil returns: Jumps and volatility forecasts. *International Journal of Finance & Economics*.

Elyasiani, E., Gambarelli, L., and Muzzioli, S. (2017). The information content of corridor volatility measures during calm and turmoil periods. *Quantitative Finance and Economics*, **4**(1), 454–473.

Freund, Y. (1995). Boosting a weak learning algorithm by majority. *Information and computation*, **121**(2), 256–285.

Gerlein, E. A., McGinnity, M., Belatreche, A., and Coleman, S. (2016). Evaluating machine learning classification for financial trading: An empirical approach. *Expert Systems with Applications*, **54**, 193–207.

Giot, P. (2005). Relationships between implied volatility indexes and stock index returns. *The Journal of Portfolio Management*, **31**(3), 92–100.

Gokmenoglu, K. K. and Fazlollahi, N. (2015). The interactions among gold, oil, and stock market: Evidence from s&p500. *Procedia Economics and Finance*, **25**, 478–488.

Gonzalez-Perez, M. T. (2015). Model-free volatility indexes in the financial literature: A review. *International Review of Economics & Finance*, **40**, 141–159.

Gu, S., Kelly, B., and Xiu, D. (2020). Empirical Asset Pricing via Machine Learning. *The Review of Financial Studies*, **33**(5), 2223–2273.

Hu, H., Tang, L., Zhang, S., and Wang, H. (2018). Predicting the direction of stock markets using optimized neural networks with google trends. *Neurocomputing*, **285**, 188–195.

Hutchinson, J. M., Lo, A. W., and Poggio, T. (1994). A nonparametric approach to pricing and hedging derivative securities via learning networks. *The Journal of Finance*, **49**(3), 851–889.

James, G., Witten, D., Hastie, T., and Tibshirani, R. (2013). *An introduction to statistical learning*, volume 112. Springer.

Kang, W., Ratti, R. A., and Yoon, K. H. (2015). The impact of oil price shocks on the stock market return and volatility relationship. *Journal of International Financial Markets, Institutions and Money*, **34**, 41–54.

Khan, W., Ghazanfar, M. A., Azam, M. A., Karami, A., Alyoubi, K. H., and Alfakeeh, A. S. (2020). Stock market prediction using machine learning classifiers and social media, news. *Journal of Ambient Intelligence and Humanized Computing*, pages 1–24.

Lubnau, T. M. and Todorova, N. (2015). The calm after the storm: implied volatility and future stock index returns. *The European Journal of Finance*, **21**(15), 1282–1296.

Ma, L. and Pohlman, L. (2008). Return forecasts and optimal portfolio construction: a quantile regression approach. *The European Journal of Finance*, **14**(5), 409–425.

Malagrino, L. S., Roman, N. T., and Monteiro, A. M. (2018). Forecasting stock market index daily direction: A bayesian network approach. *Expert Systems with Applications*, **105**, 11–22.

Mora-Valencia, A., Rodríguez-Raga, S., and Vanegas, E. (2021). Skew index: Descriptive analysis, predictive power, and short-term forecast. *The North American Journal of Economics and Finance*, page 101356.

Patel, J., Shah, S., Thakkar, P., and Kotecha, K. (2015). Predicting stock and stock price index movement using trend deterministic data preparation and machine learning techniques. *Expert systems with applications*, **42**(1), 259–268.

Rapach, D. E., Strauss, J. K., and Zhou, G. (2013). International stock return predictability: What is the role of the united states? *The Journal of Finance*, **68**(4), 1633–1662.

Rosillo, R., Giner, J., and de la Fuente, D. (2014). The effectiveness of the combined use of vix and support vector machines on the prediction of s&p 500. *Neural Computing and Applications*, **25**(2), 321–332.

Rubbaniy, G., Asmerom, R., Rizvi, S. K. A., and Naqvi, B. (2014). Do fear indices help predict stock returns? *Quantitative Finance*, **14**(5), 831–847.

Ryll, L. and Seidens, S. (2019). Evaluating the performance of machine learning algorithms in financial market forecasting: A comprehensive survey. *arXiv preprint arXiv:1906.07786*.

Schapire, R. E. (1990). The strength of weak learnability. *Machine learning*, **5**(2), 197–227.

Schumaker, R. P. and Chen, H. (2009). Textual analysis of stock market prediction using breaking financial news: The azfin text system. *ACM Transactions on Information Systems (TOIS)*, **27**(2), 1–19.

Simon, D. P. and Wiggins III, R. A. (2001). S&p futures returns and contrary sentiment indicators. *Journal of Futures Markets: Futures, Options, and Other Derivative Products*, **21**(5), 447–462.

Smialowski, P., Frishman, D., and Kramer, S. (2010). Pitfalls of supervised feature selection. *Bioinformatics*, **26**(3), 440–443.

Yao, J., Li, Y., and Tan, C. L. (2000). Option price forecasting using neural networks. *Omega*, **28**(4), 455–466.

Zhang, X., Hu, Y., Xie, K., Wang, S., Ngai, E., and Liu, M. (2014). A causal feature selection algorithm for stock prediction modeling. *Neurocomputing*, **142**, 48–59.

Zhu, M. (2013). Return distribution predictability and its implications for portfolio selection. *International Review of Economics & Finance*, **27**, 209–223.