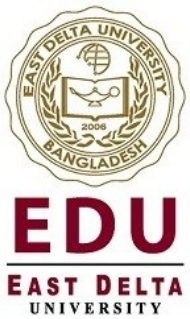
Bachelor of Science in Electrical & Electronics Engineering

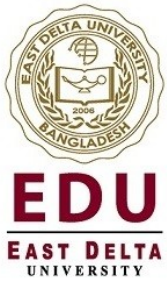


Forecasting Stock Price Using Machine Learning Technique

By

Md. Tanvir Rahman (ID: 143000410)

This thesis is submitted in partial fulfillment of the requirement for the degree of Bachelor of Science in Electrical & Electronics Engineering.



By

Md. Tanvir Rahman (ID: 143000410)

Supervised by

Mohammed Nazim Uddin, PhD

Associate Professor & Associate Dean

School of Science, Engineering and Technology

East Delta University (EDU)

Department of Electrical & Electronics Engineering

School of Science, Engineering & Technology

East Delta University

EDU Permanent Campus, Noman Society, East Nasirabad, Chittagong

ABSTRACT

Stock market is an emerging sector in any country of the world . Many people directly related to this sector . Stock market prediction is the act of trying to determine the future value of a company stock or other financial instrument .When publicly traded companies issue shares of stock to investors, each of those shares is assigned a monetary value, or price. Stock prices can go up or down depending on different factors. Stock prices can be affected by a number of things including volatility in the market, current economic conditions, and popularity of the company. The successful prediction of a stock's future price could yield significant profit .Along with the development with the stock market ,forecasting become an important topic .Since finance market has become more and more competitive, stock price prediction has been a hot research topic in the past few decades .predicting stock price is regarded a challenging task because stock market is essentially non linear ,non-parametric,noisy,and a chaotic system .Trend of a market depends on many things like liquid money human behavior, news related to stock market etc. All this together controls the behavior of trends in a stock market with the advancement of the computing technology we use machine learning technique,like Support Vector Regression,K-nearest-

neighbor,liner Regression, Random forest Regression , for analyzing time series data to predict stock price. In this paper we try to develop a forecasting model with stacking multiple method to find the best forecast of the stock price.

ACKNOLEDGEMENTS

First and foremost, I would like to thank the Almighty for providing me with His blessings. Without His kindness, it would not be possible to continue my journey towards the completion of study and thesis.

Then, I would like to thank my supervisor, Mohammed Nazim Uddin, PhD, for his amazing and endless support throughout the thesis. I am indebted to him for his expert suggestions which always motivated me to become the best version of myself. I also take this opportunity to thank my dear faculties at East Delta University for their constant support and encouragement.

Finally and most importantly, I would like to convey my greatest gratitude to my parents for being so considerate, caring and understanding all the time.

Table of Contents

Table of Contents

List of Tables

List of Figures

Chapter 1
Introduction

1.1 Background of the study

In the business and economic environment. its very important to predict various kinds of financial variables to develop proper strategies and avoid the risk of potential large losses .the forecast of a variety of economic indices has profound impact on the development of the economy. Especially in the case of stocks markets,the task become more important because the dynamic change of the market of the market behavior and immeasurable economic benefits According to the prediction of stock market indices,risk manager and practitioners can realize whether their portfolio will decline in the future and they may want to sell it before it becomes depreciated.Therefore, the research of predicting the future trends of financial indices is significant and necessary for people who are interested in the stock markets .However, the behavior of stock markets depends on many factors such as political, economic, natural factors and many others.The stock markets are dynamic and exhibit wide variation, and the prediction of stock market is a highly challenging task due to the highly nonlinear nature and complex dimensionality.Time series forecasting is the basic study to analysis data process over period of time. This is a series of statistical observations recorded over time series. It can be used to realize past behavior of the series and based on past behavior it can forecast future behavior of the series. The target of sales forecasting is to help the organization to determine demands of products and improve their strategy for the future.

1.2 Problem Statement

The purpose of this thesis to create a stock price prediction model for the various international companies .The resulting model is intended to be used as a decision support tool or as an autonomous tools that predict the future value of the stock prices by analyzing the previous stock price data.

1.3 Objective of the study

This study seeks the goal is to take time series data, find the equation that best fits the data, and be able forecast out a specific value. Time series data is a continuous data statistical observations recorded over a specific period of time. This model will try to understand the pattern of the continuous data[1] by combining different method and produce a best fit line that fits the data. The target is to determine the future stock price and improve their strategy for future. regression[2] models are among the most known regression models used in the machine learning community and recently many researchers have examined their sufficiency in ensembles[3][4]. Although many methods of ensemble design have been proposed, there is as yet no obvious picture of which method is best. One notable successful adoption of ensemble learning is the distributed scenario. In this work, we propose an efficient distributed method that uses the same training set with the parallel usage of an averaging methodology that combines linear regression[5] and KNN regression models[6],Support Vector Regression[7],random Forest

Regresssion[8][9]. We performed a comparison of the presented ensemble with other ensembles that use either the linear regression as base learner and the performance of the proposed method was better in most cases. Using averaging methodology, we expect to obtain better results because both theory and experiments show that averaging helps most if the errors in the individual regression models are not positively correlated. linear regression is a linear approach to modelling the relationship between a scalar response dependent variable and one or more explanatory variables (or independent variables). The case of one explanatory variable is called simple linear regression. For more than one explanatory variable, the process is called multiple linear regression support vector machines are supervised learning models algorithms that analyze data used for classification and regression analysis. Given a set of training examples, each marked as belonging to one or the other of two categories, an SVM training algorithm builds a

model by creating a hyper plane[9] that assigns new examples to one category or the other . in support vector regression (SVR). The model produced by support vector classification (as described above) depends only on a subset of the training[10] data, because the cost function for building the model does not care about training points that lie beyond the margin. In

pattern recognition, the k-nearest neighbors algorithm (k-NN) is a non-parametric method used for classification and regression. In both cases, the input consists of the k closest training examples in the feature space. The output depends on whether KNN is used for classification or regression.In case of the knn regression the output is the property value for the object. This value is the average of the values of its k nearest neighbors.a useful technique can be used to assign weight to the contributions of the neighbors, so that the nearer neighbors contribute more to the average than the more distant ones A model that combines KNN regression,Linear regression,Support Vector Regression, Random Forest regression model used for predicting stock prices can forecast better price accuracy

1.4 Outline of the study

This study can be narrowed down to five parts. Section 2 describes related work on the Stock price prediction. Properties and aims of the time series data will be discussed in section 3. Section 4 presents the model for building ensembles .Section IV contains Proposed methodology. The experimental design is elaborated in section V ,Section VI contains Implementation and Evaluation . Section VII contains conclusion

Chapter 2
Related Work

2 Related Work

A lot of research has been done to predict the stock price. Many algorithms of data mining have been proposed topredict stock price. Neural Network, Genetic Algorithm, Decision Tree and Fuzzy systems are widely used. Pattern discovery is beneficial for stock market prediction and public sentiment is also related to predicting stock price. There is a certain correlation among them. previous studies on stock price forecasting shows the prevalent use of technical indicators with artificial neural networks (ANN) for stock market prediction One of the well-researched and most important algorithm in the field of Data mining is Association Rule Mining (ARM), Decision trees are excellent for making financial decisions where a lots of complex data needs to be taken into account. They provide an effective framework in which alternative decisions and the implications of taking those decisions can be laid down and evaluated. They also form an accurate, an algorithm called AIS was proposed for mining association rules [11]. For last fifteen years many algorithms for rule mining have been proposed. Wanzhong Yang [12] also proposed one innovative technique to process the stock data named Granule mining technique, which reduces the width of the transaction data and generates the association rules. R.V.Argiddi [13] has proposed fragment based mining which deals mainly with reducing the time and space complexity involved in processing the data in association rule mining technique. As in granule mining, fragment based approach fragments the data sets into fragments for processing thereby reducing the input size of data sets fed to the algorithm. In contrast to granule mining, in fragment based mining the condition and decision attributes are summed for obtaining generalized association rules. Kannika Nirai Vaani M,E Ramaraj [14] has now proposed new

approach to generate association rules Providing faster generation of frequent item sets to offer interesting and useful rules in an effective and optimized way with the help of Genetic Algorithm approach. From the above literature review, technical indicators with different data mining techniques had been widely used, while there are only fewstudies of the use of fundamental indicators. The impact of fundamental analysis variables has been largely ignored like Price earnings ratio, Moving average, rumors etc. But Prashant S. Chavan , Prof. Dr. Shrishail. T. Patil [15] has said that hybridized parameters gives better & more accurate results that applying only single type of input variables. A.A. Adebiyi , C.K. Ayo, M.O Adebiyi and S.O. Otokiti [16] has proposed predictive model has the potential to enhance the quality of decision making of investors in the stock market by offering more accurate stock prediction using hybrid parameters. They used ANN for this but their performance is not always satisfactory. Robert K. Laia , Chin-Yuan Fanb, Wei-Hsiu Huang b, Pei-Chann Chang [6,17],[9,18] has proposed forecasting model that integrates a data clustering technique, fuzzy decision tree (FDT) and genetic algorithm (GA) to construct a decision-making system based on historical data and technical indices.Public information such as news, blogs, twitter mood, social networking sites and stock articles can also affect stock market trend. Web has been treated as a great source of financial information; many papers proposed stock price predicting approaches based on analyzing web sentiments using text mining.Schumaker and Chen examined different textual representations of news articles to predict future stock price, which was compared to linear regression with SVM.

.

Chapter 3
Time series Data

3.Introduction to Time Series Data

Time series data are taken by a variable over time (such as daily sales revenue, weekly orders, monthly overheads, yearly income,daily stock prices) and tabulated or plotted as chronologically ordered numbers or data points. there are two fundamental ways, how time series data are recorded. The first way, values are measured just for the specific time stamps, what may occur periodically, or occasionally according to concrete conditions, but anyway, result will be a discrete set of values, formally called discrete time series. This is very common case and frequently observed in practice. In economy sector, most of the indicators are measured periodically with the specific periods, therefore economic indicators represent an appropriate example of discrete time series. The second option is, that data are measured and recorded continuously along the time intervals. Electrical signals from sensors, various indicators from medicine, like ECG, or many other scientific sensors, they all represent a continuous measurement of corresponding physical quantity. This kind of processes produces a continuous time series.To yield valid statistical inferences, these values must be repeatedly measured, often over a four to five year period. Time series consist of four components: (1) Seasonal variations that repeat over a specific period such as a day, week, month, season, etc., (2) Trend variations that move up or down in a reasonably predictable pattern, (3) Cyclical variations that correspond with business or economic 'boom-bust' cycles or follow their own peculiar cycles, and (4) Random variations that do not fall under any of the above three classifications .

3.1 Aims of Time Series Analysis

Time series modeling is a dynamic research area which has attracted attentions of researchers community over last few decades. The main aim of time series modeling is to carefully collect and rigorously study the past observations of a time series to develop an appropriate model

which describes the inherent structure of the series. There are two main requirement of time series analysis:

1. Identification of the important parameters and characteristics, which adequately describe the time series behavior.

2 identification of the best time series model.

This model is then used to generate future values for the series, i.e. to make forecasts. Time series forecasting thus can be termed as the act of predicting the future by understanding the past . Due to the indispensable importance of time series forecasting in numerous practical fields such as business, economics, finance, science and engineering, etc. proper care should be taken to fit an adequate model to the underlying time series. It is obvious that a successful time series forecasting depends on an appropriate model fitting. A lot of efforts have been done by researchers over many years for the development of efficient models to improve the forecasting accuracy. As a result, various important time series forecasting models have been evolved in literature.

Chapter 4
ENSEMBLES OF REGRESSION MODELS

Bagging is a "bootstrap" ensemble method that creates indivrated by randomly drawing, with replacement, N examples - where N is the size of the original set; many of the original examples may be repeated in the resulting training set while others may be left out. After construction of several regression models, averaging the predictions of each regression model performs the final prediction . Instability (responsiveness to changes in the training data) is a prerequisite for bagging to be effective Another approach for building ensembles of regression models is to use a variety of learning algorithms on all of the training data and combine their predictions. When multiple regression models are combined using averaging methodology, we expect to obtain good results based on the belief that the majority of experts are more likely to be correct in their decision when they are close in their opinions Stacked generalization or Stacking, is a more sophisticated approach for combining predictions of different learning algorithms. Stacking combines multiple regression models to induce a higher-level regression model with improved performance. In detail, the original data set constitutes the level zero data and all the base regression models run at this level. The level one data are the outputs of the base regression models. A learning algorithm is then used to determine how the iduals for its ensemble by training each regression model on a random redistribution of the training set. Each regression model's training set is generated by randomly drawing, with replacement, N examples - where N is the size of the original set; many of the accurately learned. After several cycles, the prediction is performed by taking a weighted average of the predictions of each regression model, with the weights being proportional to each regression model's performance on its training set.

Chapter 5

Forecasting Method

for stock price prediction we use four technique K – nearest Neighbor,Multiple Linear regression,Support Vector Regression,Random Forest Regression

5.1 K – Nearest Neighbor

One of the oldest, accurate and simplest method for pattern classification and regression is K-Nearest-Neighbor (kNN) . kNN algorithms have been identified as one of the top ten most influential data mining algorithms for their ability of producing simple but powerful classifiers. It has been studied at length over the past few decades and is widely applied in many fields. The kNN rule classifies each unlabeled example by the majority label of its k-nearest neighbors in the training dataset. KNN is a non-parametric lazy learning algorithm. That is a pretty concise statement. When we say a technique is non parametric, it means that it does not make any assumptions on the underlying data distribution. This is pretty useful, as in the real world, most of the practical data does not obey the typical theoretical assumptions made. Non parametric algorithms like KNN come to the rescue here.It is a lazy algorithm. What this means is that it does not use the training data points to do any generalization. In other words, there is no explicit training phase or it is very minimal. This means the training phase is pretty fast. Lack of generalization means that KNN keeps all the training data. More exactly, all the training data is needed during the testing phase. .Most of the lazy algorithms – especially KNN – makes decision based on the entire training data set. The dichotomy is pretty obvious here – There is a non-existent or minimal training phase but a costly testing phase. The cost is in terms of both time and memory. More time might be needed as in the worst case, all data points might take point in

decision. More memory is needed as we need to store all training data.Despite its simplicity, the kNN rule often yields competitive results. A recent work on prototype reduction, called Weighted Distance Nearest Neighbor is based on retaining the informative instances and learning their weights for classification. The algorithm assigns a non negative weight to each training instance tuple at the training phase. Only the training instances with positive weight are retained (as the prototypes) in the test phase. Although the WDNN algorithm is well formulated and shows encouraging performance, in practice it can only work with K = 1. A more recent approach tries to reduce the time complexity of WDNN and extend it to work for values of K greater than 1.Chawla and Liu in one of their recent work presented a novel K-Nearest Neighbors weighting strategy for handling the problem of class imbalance. They proposed CCW (class confidence weights) that uses the probability of attribute values given class labels to weight prototypes in kNN. While the regular kNN directly uses the probabilities of class labels in the neighborhood of the query instance, they used conditional probabilities of classes. They have also shown how to calculate CCW weights using mixture modeling and Bayesian networks. The method performed more accurately than the existing state-of-art algorithms. KaiYan Feng and others defined a new neighborhood relationship known as passive nearest neighbors. For two points A and B belonging to class L, point B is the local passive k th -order nearest neighbor of A, only and only if A is the k th nearest neighbor of B among all data of class L. For each query point, its k actual nearest neighbor and k passive nearest neighbors are first calculated and based on it, a overall score is calculated for each class. The class score determines the likelihood that the query points belong to that class. In another recent work [25], Evan and others proposes to use geometric structure of data to mitigate the effects of class imbalance. The method even works, when the level of imbalance changes in the training data, such as online streaming data. For each query point, a k dimensional vector is calculated for each of the classes present in the data. The vector consist of distances of the query point to it’s k nearest neighbors in that class.

Based on this vector probability that the query point belongs to a particular class is calculated. However the approach is not studied in depth. Yang Song and others proposes [41] two different versions of kNN based on the idea of informa-tiveness. According to them, a point is treated to be informative, if it is close to the query point and far away from the points with different class labels. One of the proposed versions LI-KNN takes two parameters k and I, It first find the k nearest neighbor of the query point and then among them it find the I most informative points. Based on the class label of the informative points, class label is assigned to the query point. They also showed that the value of k and I have very less effect on the final result. The other version GI-KNN works on the assumption that some points are more informative then others. It tries to find global informative points and then assigns a weight to each of the points in training data based on their informativeness. It then uses weighted euclidean metric to calculate distances. In another recent work a k Exemplar-based Nearest Neighbor (kENN) classifier was proposed which is more sensitive to the minority class. The main idea is to first identify the exemplar minority class instances in the training data and then generalize them to Gaussian balls as concept for the minority class. The approach is based on extending the decision boundary for the minority class.

5.1.1 Assumption of KNN

KNN assumes that the data is in a feature space. More exactly, the data points are in a metric space. The data can be scalars or possibly even multidimensional vectors. Since the points are in feature space, they have a notion of distance – This need not necessarily be Euclidean distance although it is the one commonly used.

Each of the training data consists of a set of vectors and class label associated with each vector. In the simplest case, it will be either + or – (for positive or negative classes). But KNN, can work equally well with arbitrary number of classes.

We are also given a single number "k". This number decides how many neighbors (where neighbors is defined based on the distance metric) influence the classification. This is usually an odd number if the number of classes is 2. If k=1, then the algorithm is simply called the nearest neighbor algorithm. We estimate the distance between the data points by euclidian distance.

$$\mathbf{dist\ (\ A\ ,\ B)} = \sqrt{\sum(xi - yi)/2m} \qquad (1)$$

5.2 Support Vector Regression

In machine learning, support vector machines (SVMs, also support vector networks) are supervised learning models with associated learning algorithms that analyze data used for classification and regression analysis. Given a set of training examples, each marked as belonging to one or the other of two categories, an SVM training algorithm builds a model that assigns new examples to one category or the other, making it a non-probabilistic binary linear classifier (although methods such as Platt scaling exist to use SVM in a probabilistic classification setting). An SVM model is a representation of the examples as points in space, mapped so that the examples of the separate categories are divided by a clear gap that is as wide as possible. New examples are then mapped into that same space and predicted to belong to a category based on which side of the gap they fall. In addition to performing linear classification, SVMs can efficiently perform a non-linear classification using what is called the kernel trick, implicitly mapping their inputs into high-dimensional feature spaces. When data is unlabelled, supervised learning is not possible, and an unsupervised learning approach is required, which

attempts to find natural clustering of the data to groups, and then map new data to these formed groups. The support vector clustering algorithm, created by Hava Siegelmann and Vladimir Vapnik, applies the statistics of support vectors, developed in the support vector machines algorithm, to categorize unlabeled data, and is one of the most widely used clustering algorithms in industrial applications.support vector machine constructs a hyperplane or set of hyperplanes in a high- or infinite-dimensional space, which can be used for classification, regression, or other tasks like outliers detection. Intuitively, a good separation is achieved by the hyperplane that has the largest distance to the nearest training-data point of any class (so-called functional margin), since in general the larger the margin the lower the generalization error of the classifier. The original problem may be stated in a finite dimensional space, it often happens that the sets to discriminate are not linearly separable in that space. For this reason, it was proposed that the original finite-dimensional space be mapped into a much higher-dimensional space, presumably making the separation easier in that space. To keep the computational load reasonable, the mappings used by SVM schemes are designed to ensure that dot products of pairs input data vectors may be computed easily in terms of the variables in the original space, by defining them in terms of a kernel function selected to suit the problem. The hyperplanes in the higher-dimensional space are defined as the set of points whose dot product with a vector in that space is constant, where such a set of vector is an orthogonal (and thus minimal) set of vectors that defines a hyperplane. The vectors defining the hyperplanes can be chosen to be linear combinations with parameters of images of feature vectors that occur in the data base.With this choice of a hyperplane, the points in the feature space that are mapped into the hyperplane are defined by the relation: Note that if becomes small as grows further away from , each term in the sum measures the degree of closeness of the test point to the corresponding data base point . In this way, the sum of kernels above can be used to measure the relative nearness of each test point to the data points originating in one or the other of the sets to be discriminated. Note the

fact that the set of points mapped into any hyperplane can be quite convoluted as a result, allowing much more complex discrimination between sets which are not convex at all in the original space. Then generate SVM regression, the input (X)tis first mapped onto a m-dimensional feature space using some fixed (nonlinear) mapping, and then a linear model is constructed in this feature space. Using f(x,w) mathematical notation, the linear model (in the feature space) is given by where g(x) denotes a set of nonlinear transformations, and b is the "bias"term.

$$f(\mathbf{x}, \omega) = \sum_{j=1}^{m} \omega_j g_j(\mathbf{x}) + b \qquad (2)$$

loss function is estimated by this formula

$$L_\varepsilon(y, f(\mathbf{x}, \omega)) = \begin{cases} 0 & \text{if } |y - f(\mathbf{x}, \omega)| \le \varepsilon \\ |y - f(\mathbf{x}, \omega)| - \varepsilon & \text{otherwise} \end{cases}$$

5.3 Multiple Linear Regression

In statistical modeling, **regression analysis** is a set of statistical processes for estimating the relationships among variables. It includes many techniques for modeling and analyzing several variables, when the focus is on the relationship between a dependent variable and one or more independent variables.More specifically, regression analysis helps one understand how the typical value of the dependent variable changes when any one of the independent variables is varied, while the other independent variables are held fixed. Most commonly, regression analysis estimates the conditional expectation of the dependent variable given the independent variables –

that is, the average value of the dependent variable when the independent variables are fixed. Less commonly, the focus is on a quantile, or other location parameter of the conditional distribution of the dependent variable given the independent variables. In all cases, a function of the independent variables called the regression function is to be estimated. In regression analysis, it is also of interest to characterize the variation of the dependent variable around the prediction of the regression function using a probability distribution.Multiple linear regression attempts to model the relationship between two or more explanatory variables and a response variable by fitting a linear equation to observed data. Every value of the independent variable *x* is associated with a value of the dependent variable *y*. The population regression line for *p* explanatory variables x_1, x_2, ... , x_p is defined to be $\mu_y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + ... + \beta_p x_p$. This line describes how the mean response μ_y changes with the explanatory variables. The observed values for *y* vary about their means μ_y and are assumed to have the same standard deviation σ. The fitted values b_0, b_1, ..., b_p estimate the parameters β_0, β_1, ..., β_p of the population regression line. Since the observed values for *y* vary about their means μ_y, the multiple regression model includes a term for this variation. In words, the model is expressed as DATA = FIT + RESIDUAL, where the "FIT" term represents the expression

$$\beta_0 + \beta_1 x_1 + \beta_2 x_2 + ... \beta_p x_p.$$

The "RESIDUAL" term represents the deviations of the observed values *y* from their means μ_y, which are normally distributed with mean 0 and variance σ. The notation for the model deviations is ε.

Formally, the model for multiple linear regression, given *n* observations, is

$$y_i = \beta_0 + \beta_1 x_{i1} + \beta_2 x_{i2} + \ldots \beta_p x_{ip} + \varepsilon_i \text{ for } i = 1,2, \ldots n. \qquad (4)$$

In the least-squares model, the best-fitting line for the observed data is calculated by minimizing the sum of the squares of the vertical deviations from each data point to the line (if a point lies on the fitted line exactly, then its vertical deviation is 0). Because the deviations are first squared, then summed, there are no cancellations between positive and negative values. The least-squares estimates b_0, b_1, ... b_p are usually computed by statistical software.

5.4 Random forest Regression

Random Forest is a flexible, easy to use machine learning algorithm that produces, even without hyper-parameter tuning, a great result most of the time. It is also one of the most used algorithms, because it's simplicity and the fact that it can be used for both classification and regression tasks.The general method of random decision forests was first proposed by Ho in 1995. Ho established that forests of trees splitting with oblique hyperplanes can gain accuracy as they grow without suffering from overtraining, as long as the forests are randomly restricted to be sensitive to only selected feature dimensions. A subsequent work along the same lines concluded that other splitting methods, as long as they are randomly forced to be insensitive to some feature dimensions, behave similarly. Note that this observation of a more complex classifier (a larger forest) getting more accurate nearly monotonically is in sharp contrast to the common belief that the complexity of a classifier can only grow to a certain level of accuracy before being hurt by overfitting.Random forests or random decision forests are an ensemble learning method for classification, regression and other tasks that operates by constructing a multitude of decision trees at training time and outputting the class that is the mode of the classes (classification) or mean prediction (regression) of the individual trees.Random decision forests

correct for decision trees habit of overfitting to their training set. Random Forest has nearly the same hyperparameters as a decision tree or a bagging classifier. Fortunately, we don’t have to combine a decision tree with a bagging classifier and can just easily use the classifier-class of Random Forest. Random Forest adds additional randomness to the model, while growing the trees. Instead of searching for the most important feature while splitting a node, it searches for the best feature among a random subset of features. This results in a wide diversity that generally results in a better model.

Therefore, in Random Forest, only a random subset of the features is taken into consideration by the algorithm for splitting a node. we can even make trees more random, by additionally using random thresholds for each feature rather than searching for the best possible thresholds

5.5 Forecasting stock prices method architecture

The prediction system has a two tier architecture top tier is dedicated to preparing the data sets from multiple information sources to make them ready for the predication tasks in the next tier. It is composed of two major parts. The first part is data prepossessing . In this process we process the data by adding more feature and removing unnecessary feature and removing the bad data and also the absence of the data . The second part is the data alignment. The second tier is dedicated to the market volatility analysis and prediction through the model integration and training, which uses multiple kernel learning methodology to train the model It consists of three tasks: First, we build one regression model per source. Second, we train the model with the same data sets ,then we create a stacked algorithm using these algorithm. In this paper, we use the multi-kernel learning.

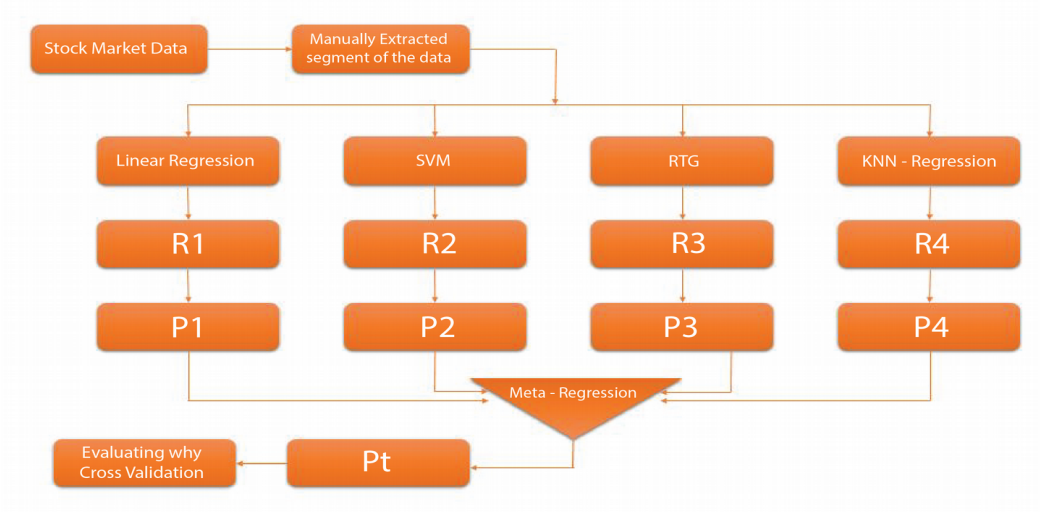


Figure 1: Forecasting of stock prices method architecture

Stacking is concerned with combining multiple classifiers generated by different learning algorithms **L1,....Ln** on a single data set S, which is composed by a feature vector Si = (Xi,Yi).

• The stacking process can be broken into two phases:

1. Generate a set of base-level classifiers C1,...,Cn. Where Cl=Ll (S)

2. Train a meta-level classifier to combine base level classifier

3.The training set for the meta-level classifier is generated through a leave-one-out cross validation process.

$$\forall i = 1, \ldots n \text{ and } \forall k = 1, \ldots, N$$

$$Cik = Lk(S - si) \qquad (5)$$

• The learned classifiers are then used to generate predictions

for **Yki=C ik (xi)**

The meta-level data sets consists of examples of the form where the features are the predictions of the base-level classifiers and the class is the correct class of the example in hand.

Chapter 6
Experimental Evaluation

6.1 Data Collection

Dataset is taken for **Quandl** is a platform for financial, economic, and alternative data that serves investment professionals. Quandl sources data from over 500 publishers .All Quandl's data are accessible via an API.API access is possible through packages for multiple programming languages including R, Python, Matlab, Maple and Stata.Quandl's sources include open data from providers such as the UN, Worldbank and central banks; core financial data from providers such as CLS Group, Zacks, and ICE; and alternative data from Dun & Bradstreet, along with numerous confidential sources. Some examples of the data set is given in the figure bellow

Date	Open	High	Low	Close	Volume	Dividend	Split	Adj_Open	Adj_High	Adj_Low	Adj_Close	Adj_Volume
1962-01-02	578.5	578.5	572.0	572.00	19360.0	0.0	1.0	14.792167	14.792167	14.625963	14.625963	387200.0
1962-01-03	572.0	577.0	572.0	577.00	14400.0	0.0	1.0	14.625963	14.753813	14.625963	14.753813	288000.0
1962-01-04	577.0	577.0	571.0	571.25	12800.0	0.0	1.0	14.753813	14.753813	14.600393	14.606786	256000.0
1962-01-05	570.5	570.5	559.0	560.00	18160.0	0.0	1.0	14.587608	14.587608	14.293555	14.319125	363200.0
1962-01-08	559.5	559.5	545.0	549.50	27200.0	0.0	1.0	14.306340	14.306340	13.935577	14.050641	544000.0

Table 6.1 Training Datasets example of the NIKE corporation

This is an example of the intel corporation stock price datasets

Date	Open	High	Low	Close	Volume	Dividend	Split	Adj_Open	Adj_High	Adj_Low	Adj_Close	Adj_Volume
1980-03-17	62.5	63.50	62.50	62.50	56900.0	0.0	1.0	0.212218	0.215614	0.212218	0.212218	10924800.0
1980-03-18	62.5	63.00	62.00	62.00	88900.0	0.0	1.0	0.212218	0.213916	0.210520	0.210520	17068800.0
1980-03-19	63.5	64.50	63.50	63.50	96400.0	0.0	1.0	0.215614	0.219009	0.215614	0.215614	18508800.0
1980-03-20	63.5	64.25	63.25	63.25	58200.0	0.0	1.0	0.215614	0.218160	0.214765	0.214765	11174400.0
1980-03-21	62.0	62.00	61.00	61.00	63400.0	0.0	1.0	0.210520	0.210520	0.207125	0.207125	12172800.0

Table 6.2 Traning Datasets example for intel datasets

This is an example of the Microsoft stock prices dataset

Date	Open	High	Low	Close	Volume	Dividend	Split	Adj_Open	Adj_High	Adj_Low	Adj_Close	Adj_Volume
1986-03-13	25.50	29.25	25.5	28.00	3582600.0	0.0	1.0	0.058190	0.066748	0.058190	0.063895	1.031789e+09
1986-03-14	28.00	29.50	28.0	29.00	1070000.0	0.0	1.0	0.063895	0.067318	0.063895	0.066177	3.081600e+08
1986-03-17	29.00	29.75	29.0	29.50	462400.0	0.0	1.0	0.066177	0.067889	0.066177	0.067318	1.331712e+08
1986-03-18	29.50	29.75	28.5	28.75	235300.0	0.0	1.0	0.067318	0.067889	0.065036	0.065607	6.776640e+07
1986-03-19	28.75	29.00	28.0	28.25	166300.0	0.0	1.0	0.065607	0.066177	0.063895	0.064466	4.789440e+07

Table 6.3 Traning Datasets example for Microsoft datasets

6.2 Data Pre-processing

In the real world, many data sets are very messy.Most stock price/volume data is pretty clean, rarely with missing data, but many data sets will have a lot of missing data. filter out other unimportant feature from the feature because not all the feature will be included into the final feature list .The reason behind it is the unnecessary feature and those value which has no relation with the stock market prediction will reduce the accuracy of the prediction. in our study, we used the following attributes Adjusted Close price, Volatility, Percentage change, Adjusted open price,Adjusted Volume

6.3 Tools and Language

For data analysis and evaluation python is used as programming language. Python is an interpreted and objected oriented programming language.

Anaconda : Anaconda is a free and open-source distribution of the Python and R programming languages for scientific computing (data science, machine learning applications, large-scale data processing, predictive analytics, etc.), that aims to simplify package management and deployment. Package versions are managed by the package management system *conda*. The Anaconda distribution is used by over 6 million users and includes more than 1400 popular data-science packages

Pandas: It is a Python package providing fast, flexible, and expressive data structures designed to make working with "relational" or "labeled" data both easy and intuitive. It aims to be the fundamental high-level building block for doing practical, real world data analysis in Python. Additionally, it has the broader goal of becoming the most powerful and flexible open source data analysis / manipulation tool available in any language. It is already well on its way toward this goal.

Numpy: It’s provided given below.

- An array object of arbitrary homogeneous items
- Fast mathematical operations over arrays
- Linear Algebra, Fourier Transforms, Random Number Generation

Skikit-learn:**Scikit-learn** (formerly **scikits.learn**) is a free software machine learning library for the Python programming language.It features various classification, regression and clustering algorithms including support vector machines, random forests, gradient boosting, *k*-means and

DBSCAN, and is designed to interoperate with the Python numerical and scientific libraries NumPy and SciPy.

Matplotlib.pyplot: It is a state-based interface to matplotlib. It provides a MATLAB-like way of plotting.pyplot is mainly intended for interactive plots.

Chapter 7
Result And Conclusion

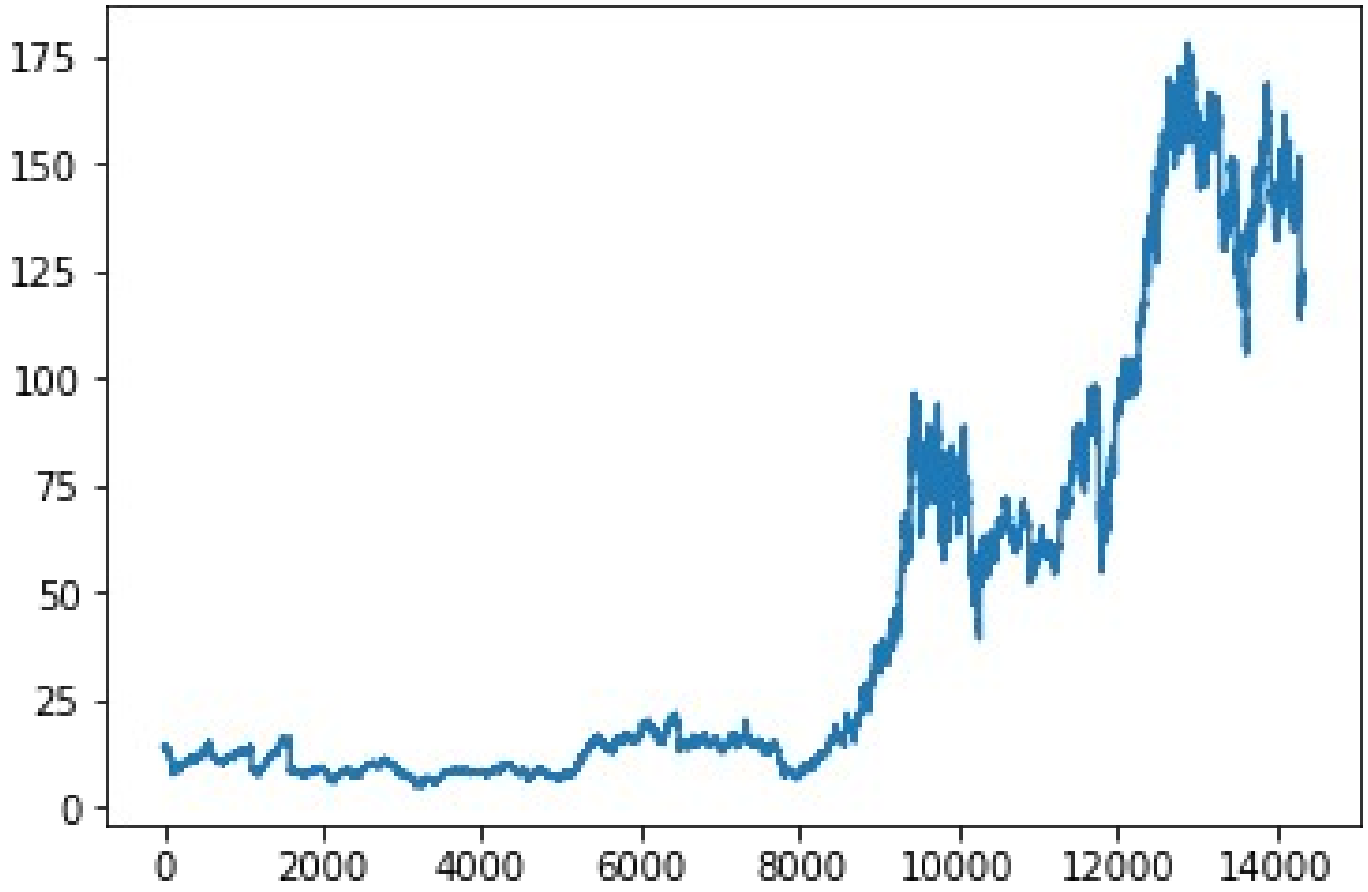
7.1 Final Forecasting result

In the given table final forecasting result calculation is shown.

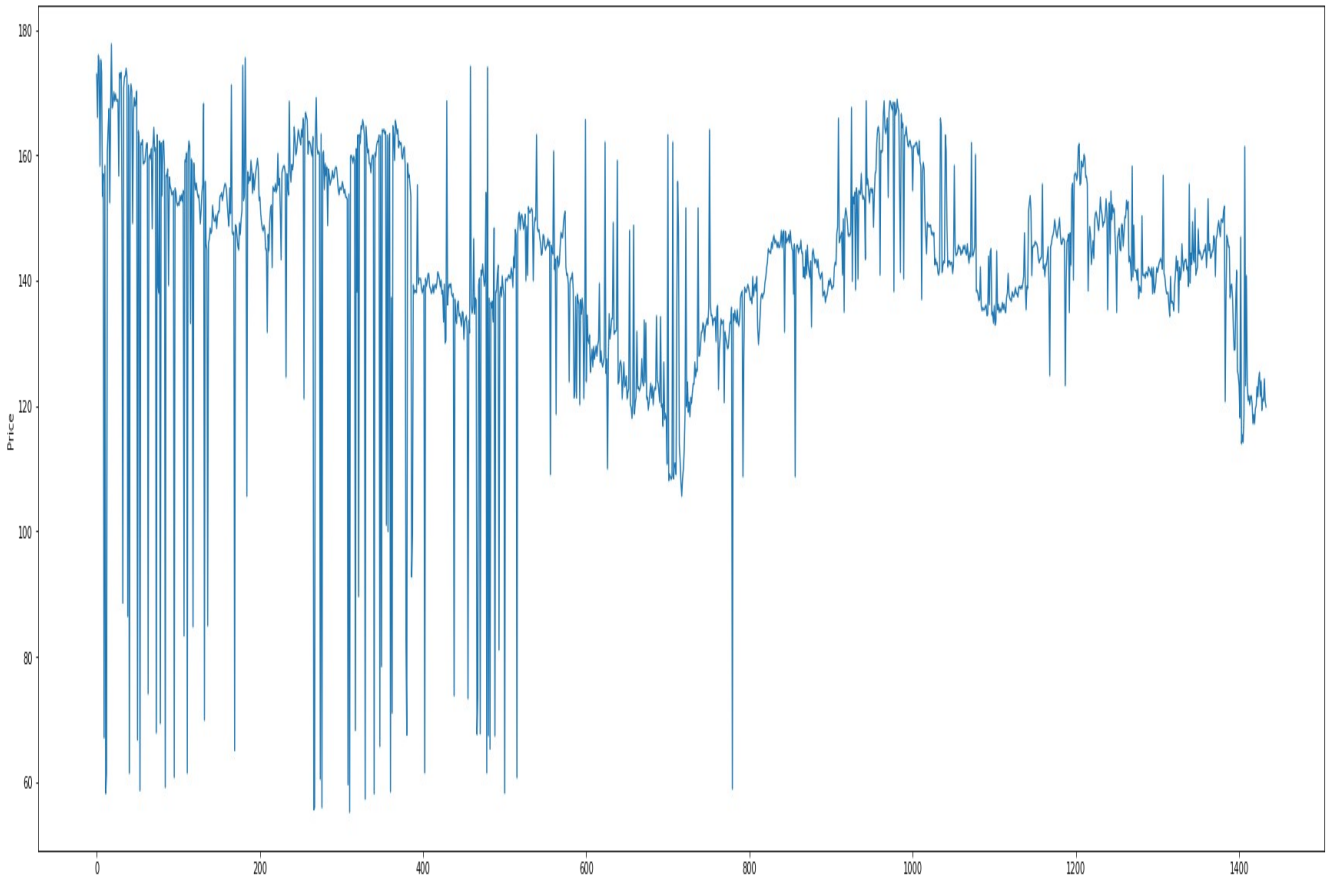
Table 1: Final forecasting result

Accuracy Based on Cross Validation					
Company	Linear Regression	KNN Regression	SVR	Random Forest Regression	**Proposed Method**
Google	**87.9%**	89.18%	74.32%	88.6%	**90.72%**
Apple	**92.2%**	91.4%	91.0%	93.4%	**94.8%**
Microsoft	**91.4%**	91.4%	91.4%	91.4%	**91.4%**
IBM	**73.6%**	71.3%	77.8%	78.4%	**80.8%**
NIKE	**94.0%**	94.9%	95.05%	92.9%	**96.8%**
MACDONALD	**93.77%**	91.78%	91.09%	92.70%	**94.7%**
Walt Disney	**74.36%**	69.66%	77.31%	75.05%	**81.43%**
Intel	**52.66%**	**61.30%**	65.72%	67.36%	**72.77%**

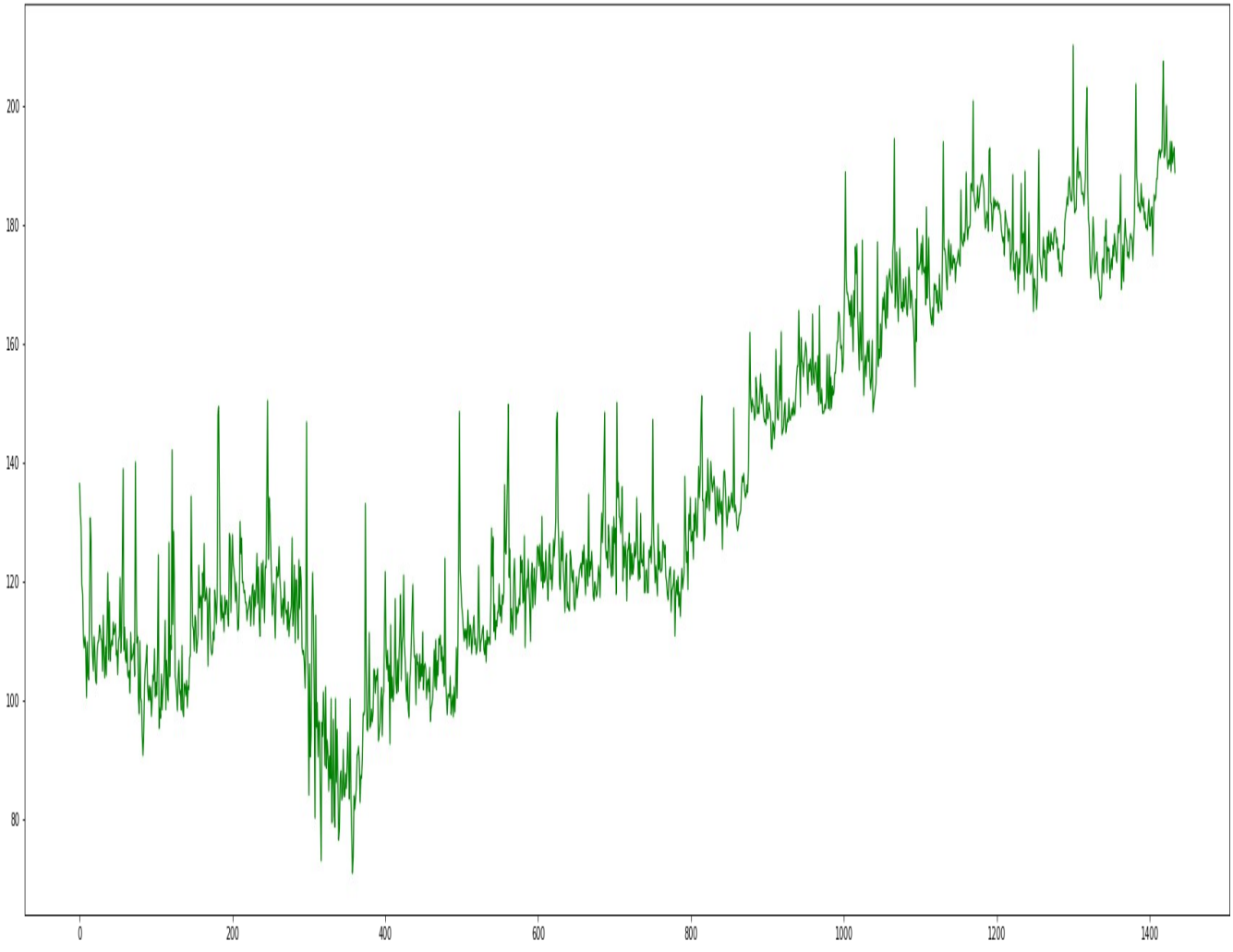
Following curve is shown the previous stock price of IBM corporation. where X axis denotes days and Y axis denotes closing share price



Following curve is shown comparison between actual sales and predicted sales in IBM where X axis denotes days and Y axis denotes closing share price in KNN Regression

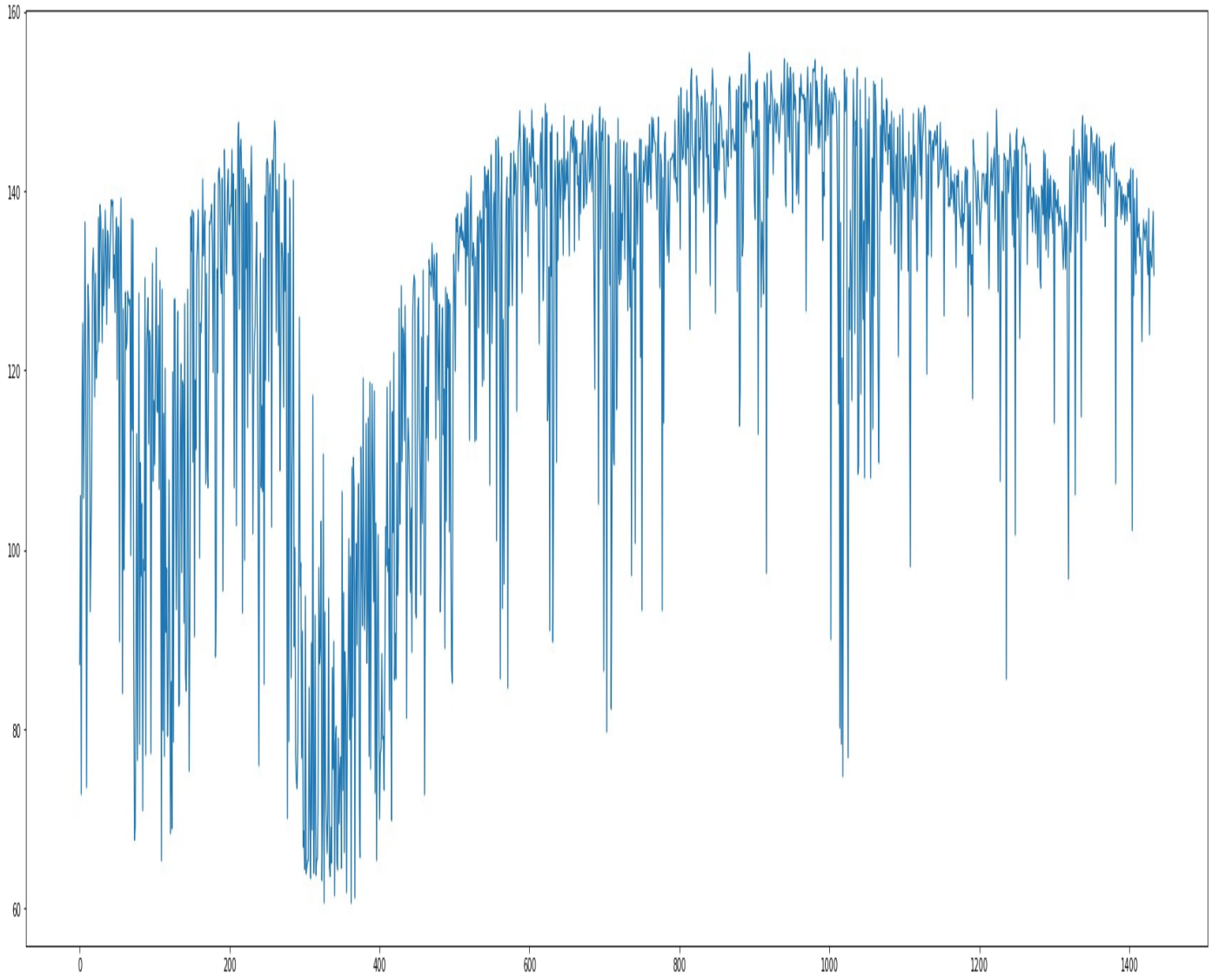


Following curve is shown comparison between actual sales and predicted sales of IBM where X axis denotes days and Y axis denotes closing share price in Linear Regression

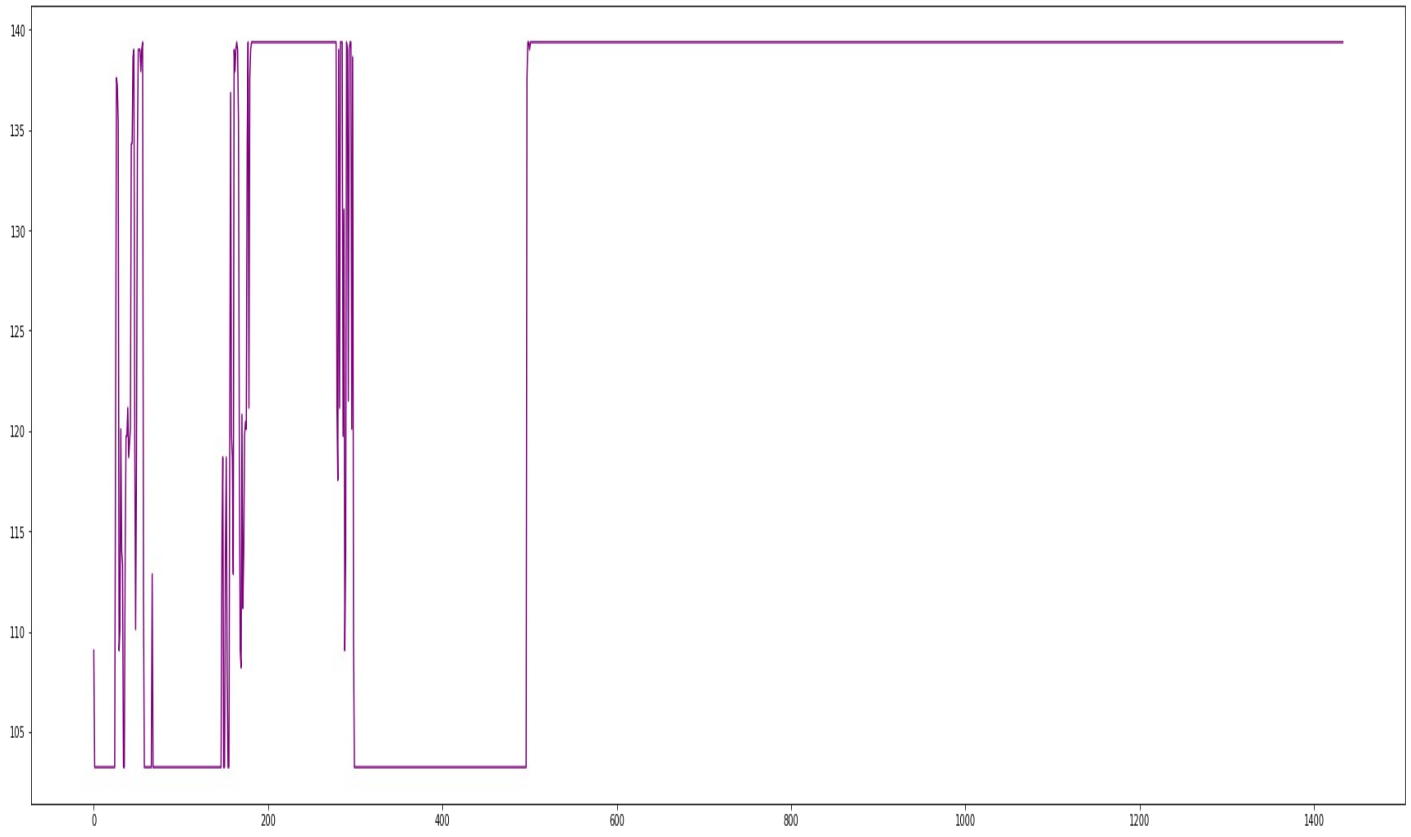


svr

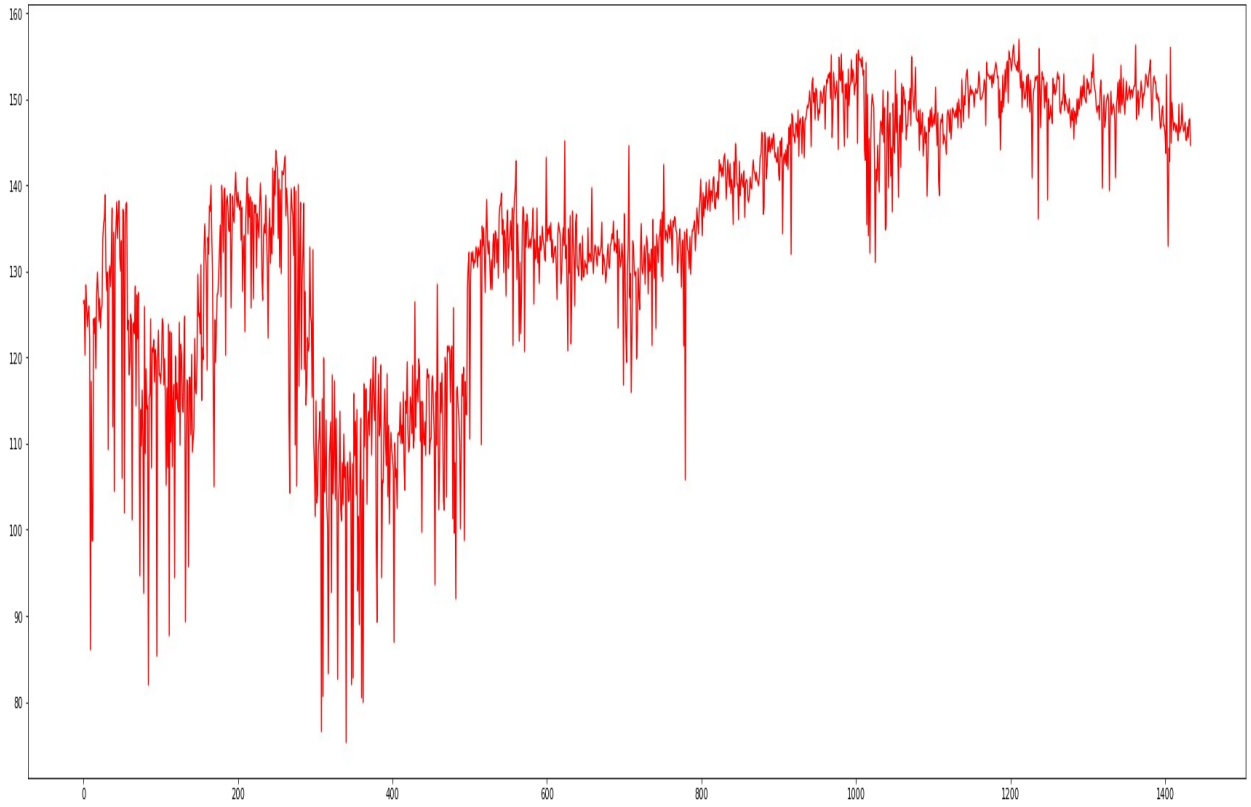
Following curve is shown comparison between actual sales and predicted sales of IBM where X axis denotes days and Y axis denotes closing share price in SVR



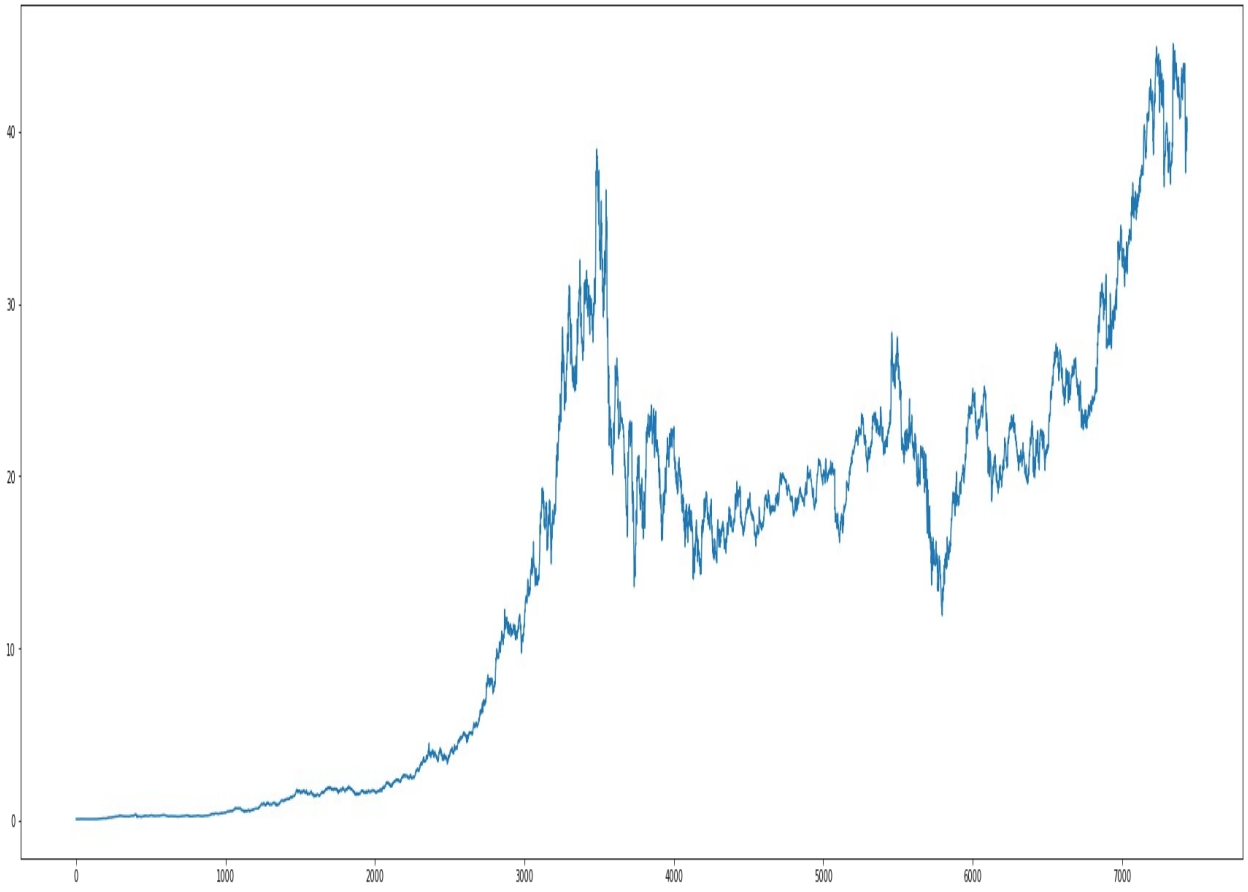
Following curve is shown comparison between actual sales and predicted salesof IBM where X axis denotes days and Y axis denotes closing share price in random forest Regression



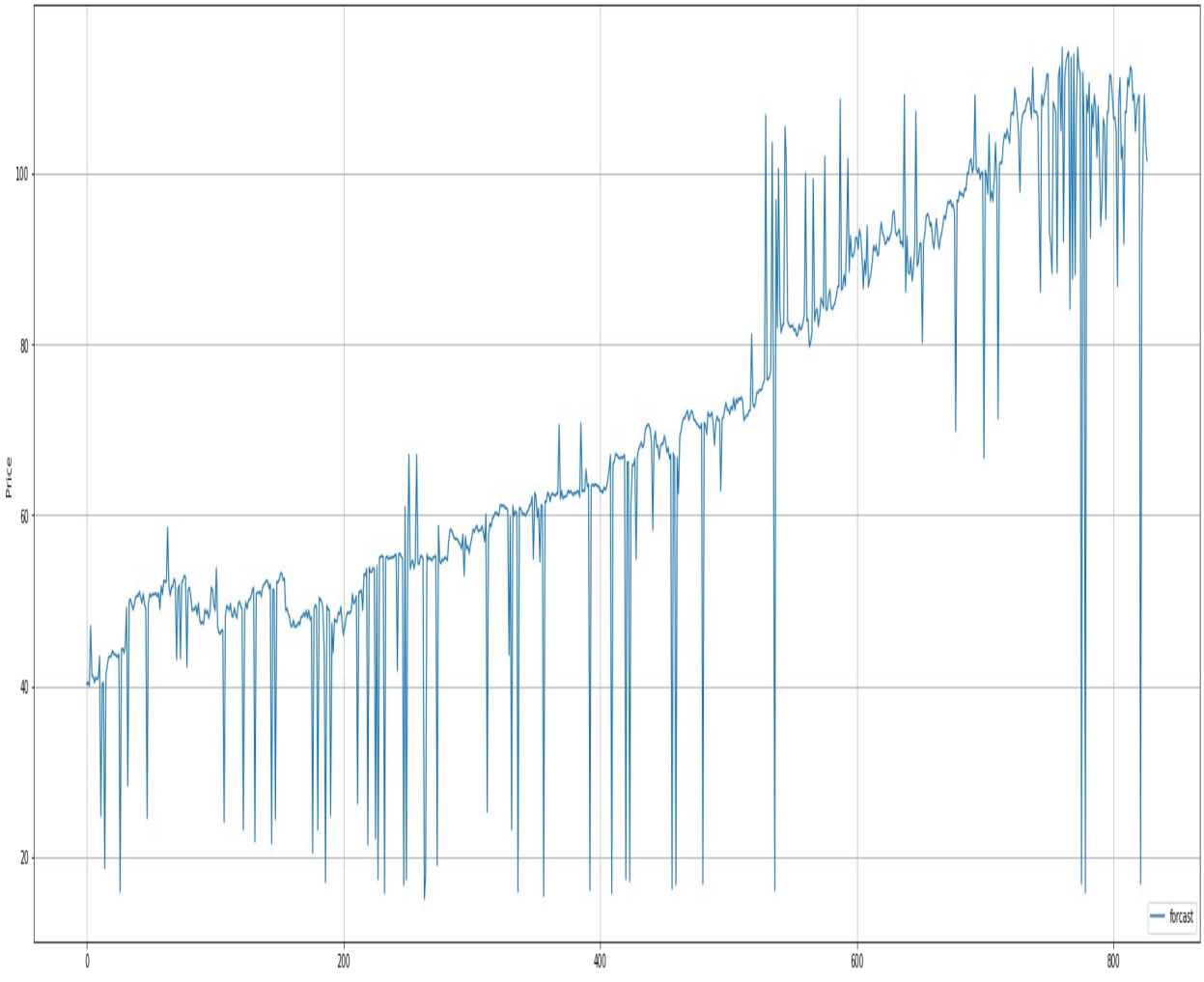
Following curve is shown comparison between actual sales and predicted sales where X axis denotes days and Y axis denotes closing share price in Proposed Regression model



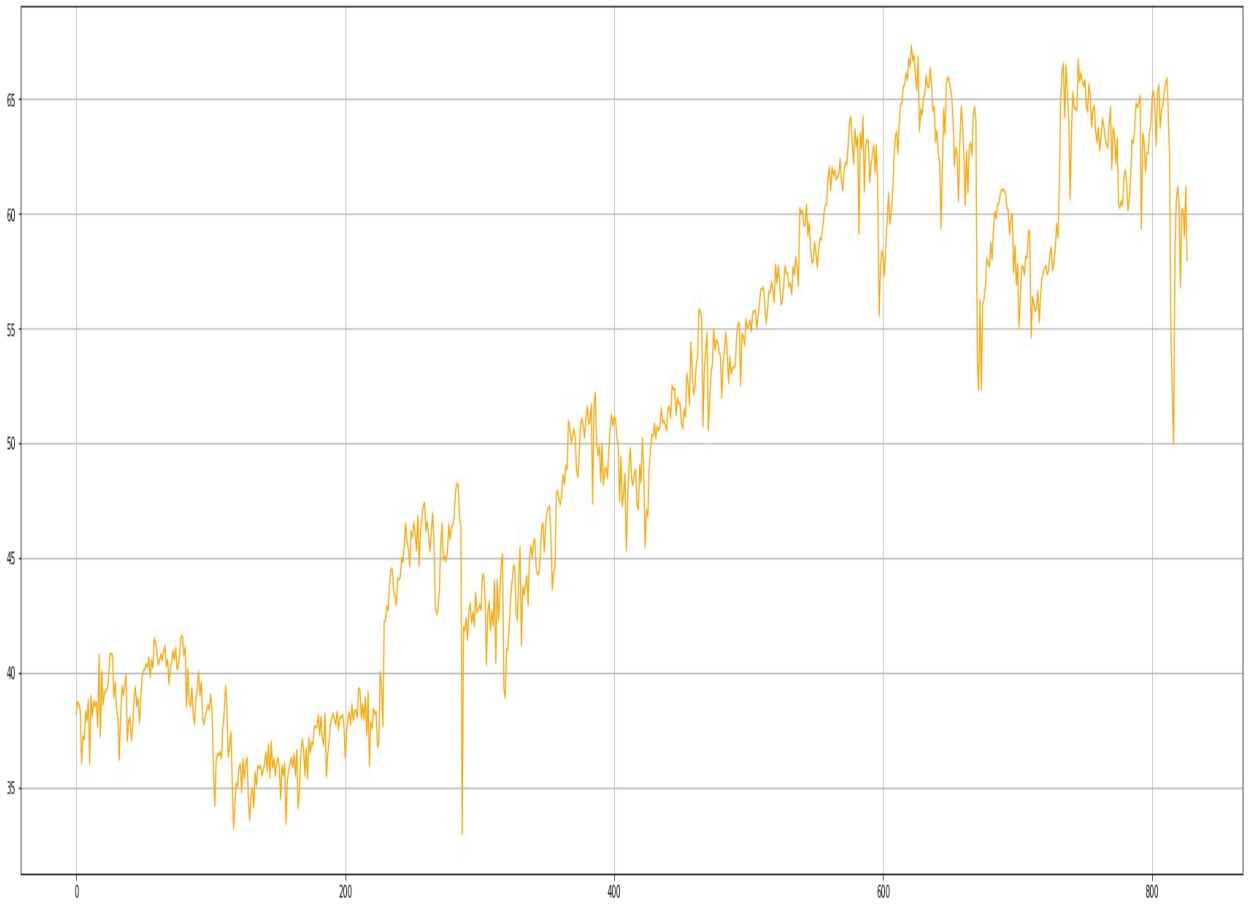
Following curve is shown the previous stock price of Microsoft corporation. where X axis denotes days and Y axis denotes closing share price



Following curve is shown comparison between actual sales and predicted sales of Microsoft where X axis denotes days and Y axis denotes closing share price in KNN Regression

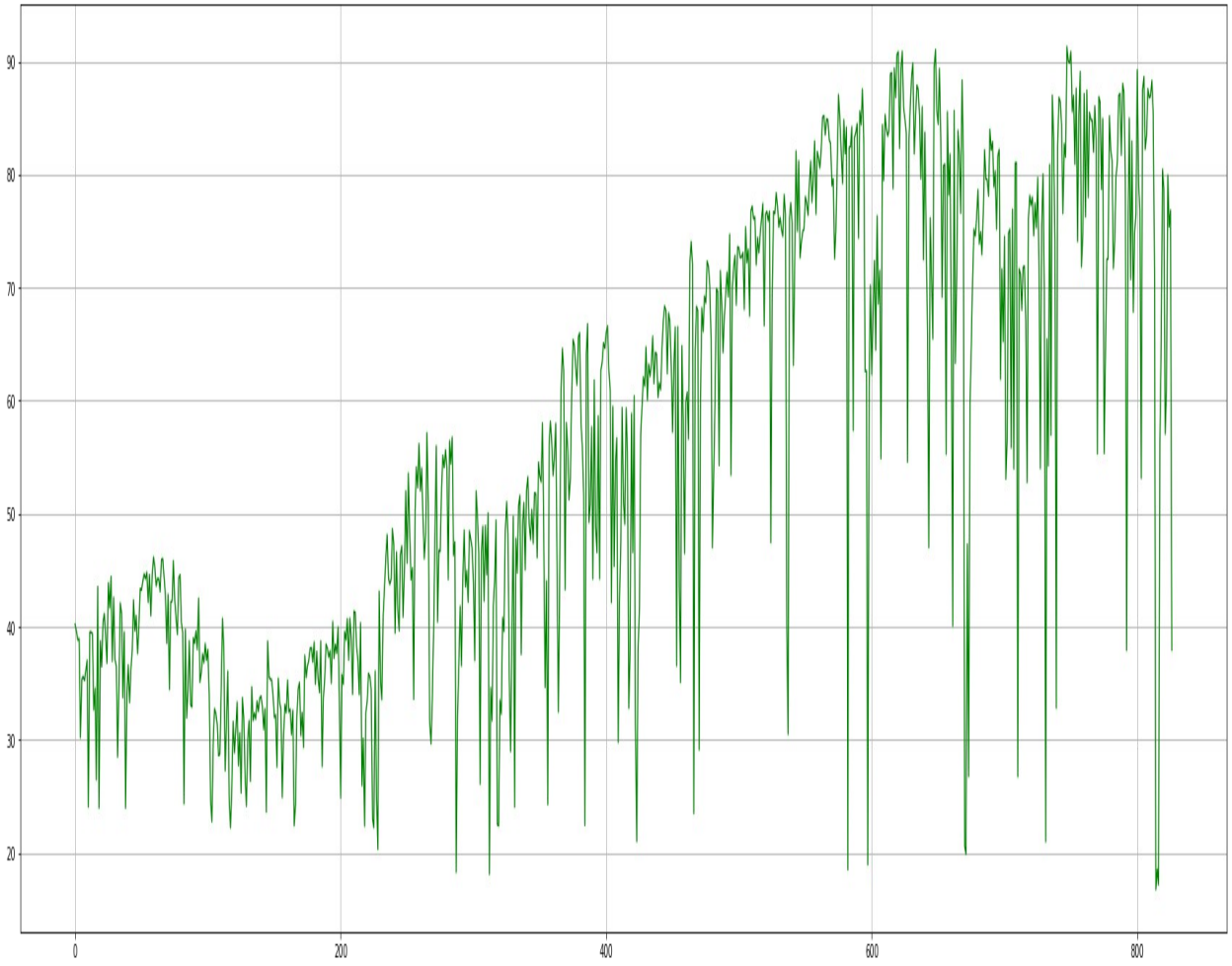


Following curve is shown comparison between actual sales and predicted sales of microsoft where X axis denotes days and Y axis denotes closing share price in Linear Regression

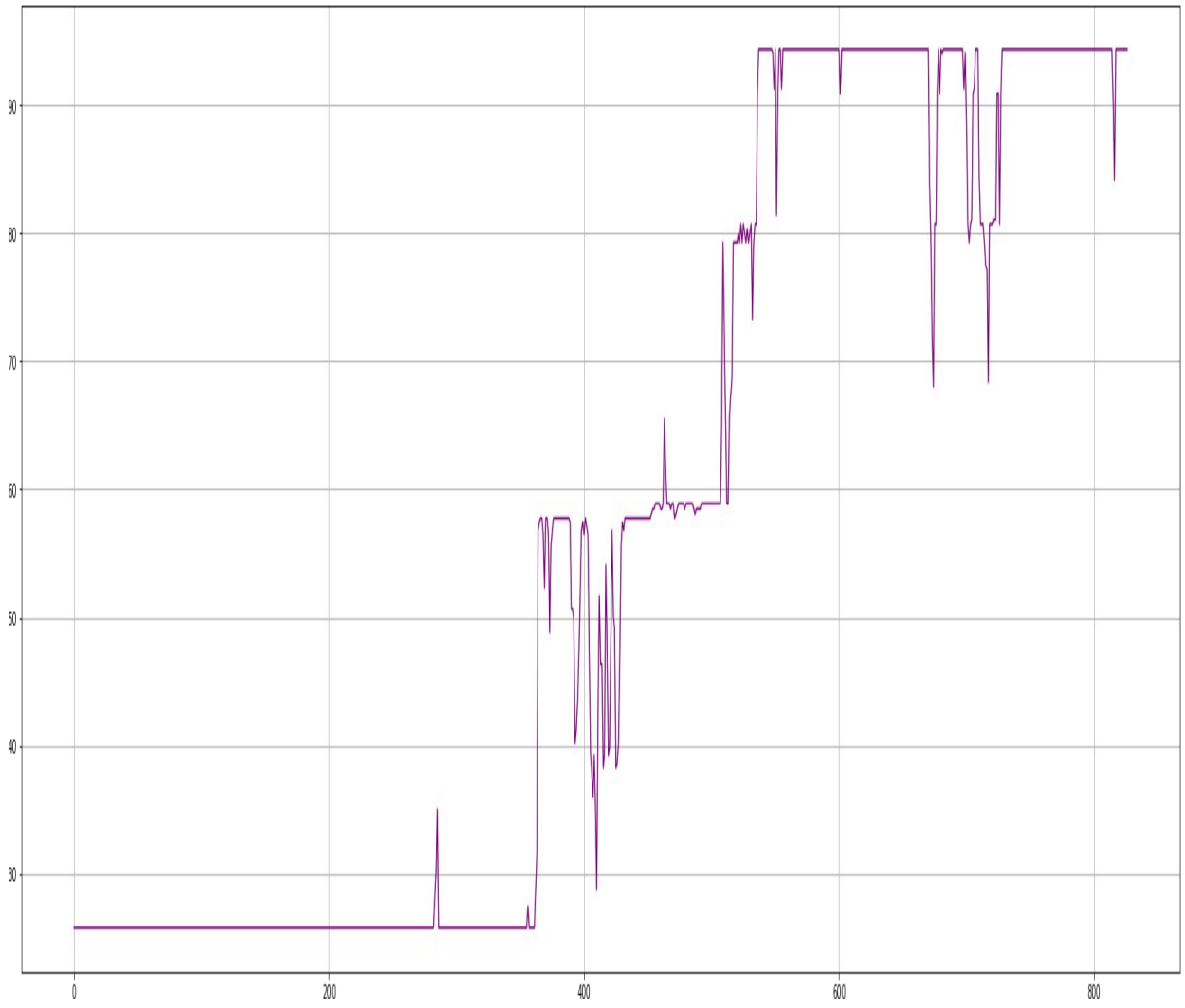


svr

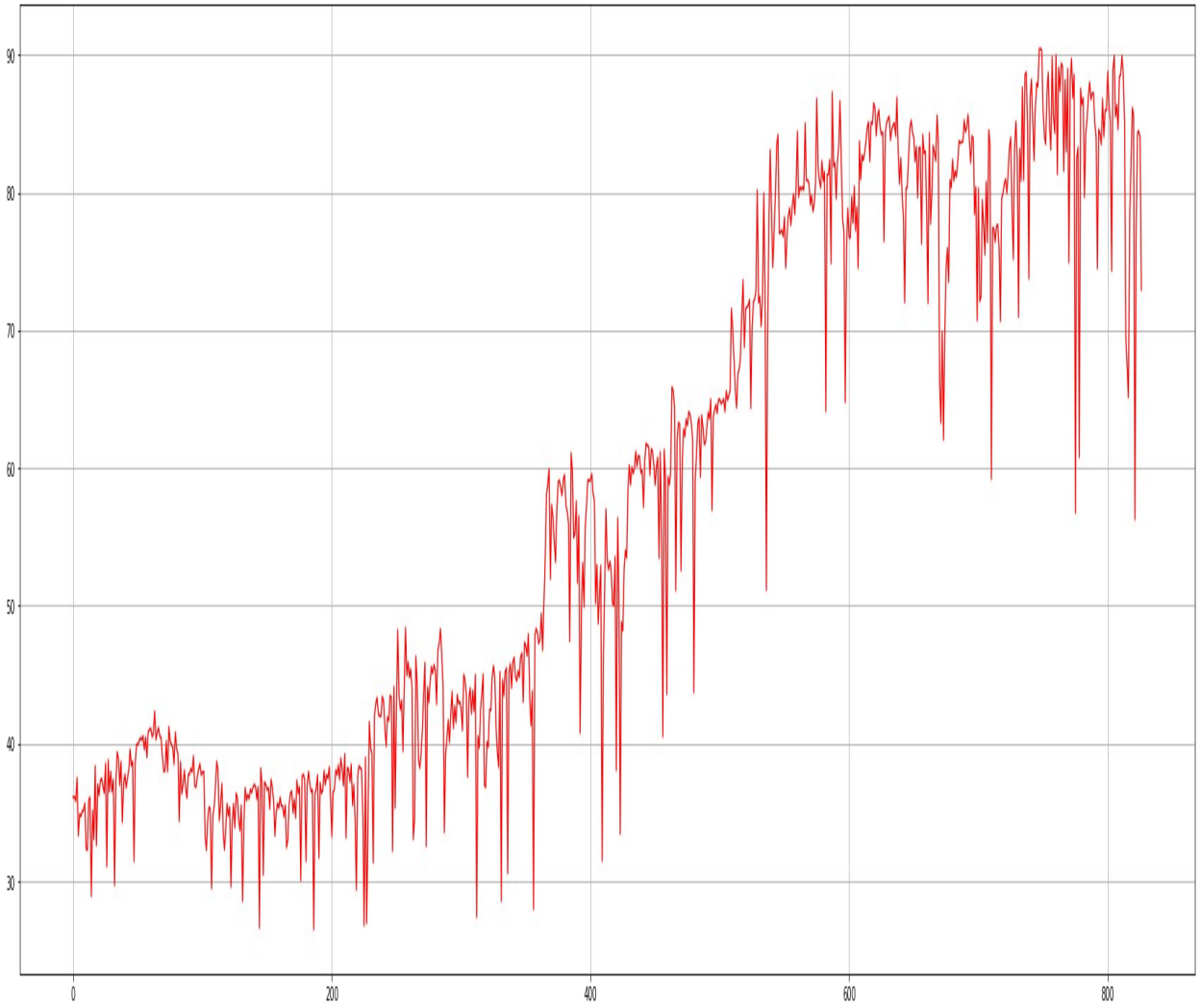
Following curve is shown comparison between actual sales and predicted sales of microsoft where X axis denotes days and Y axis denotes closing share price in SVR



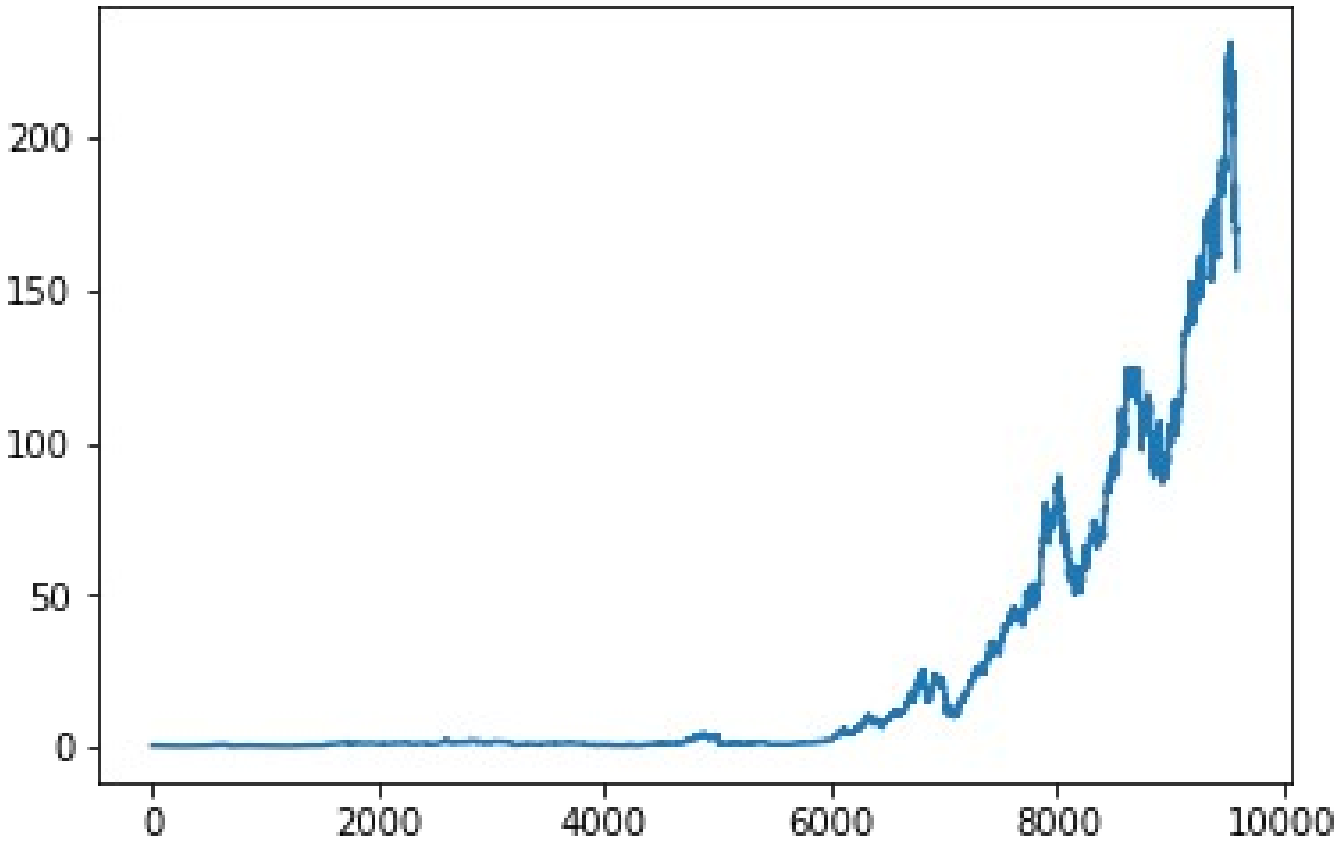
Following curve is shown comparison between actual sales and predicted salesof microsoft where X axis denotes days and Y axis denotes closing share price in Random forestRegression



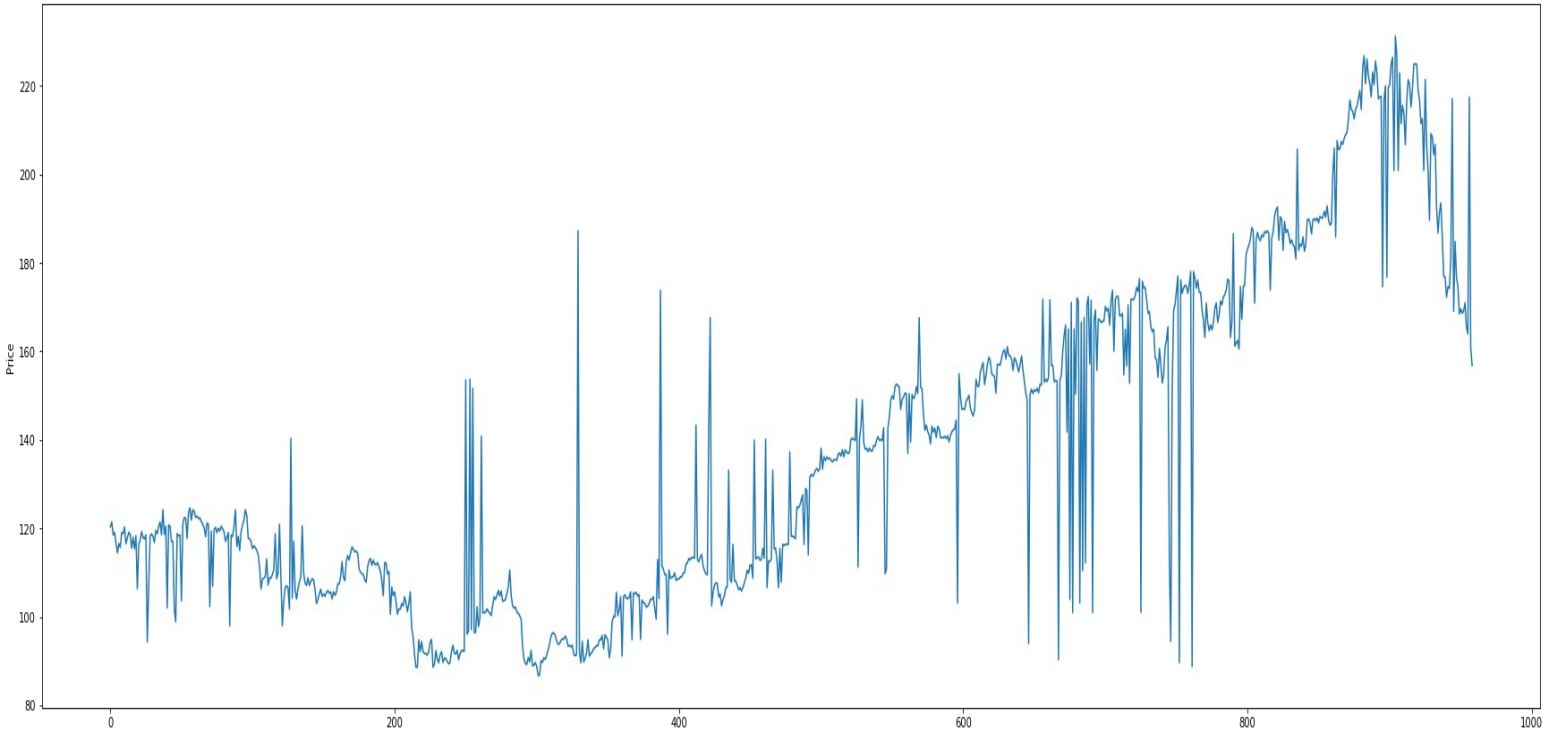
Following curve is shown comparison between actual sales and predicted sales where X axis denotes days and Y axis denotes closing share price in proposed Regression Model



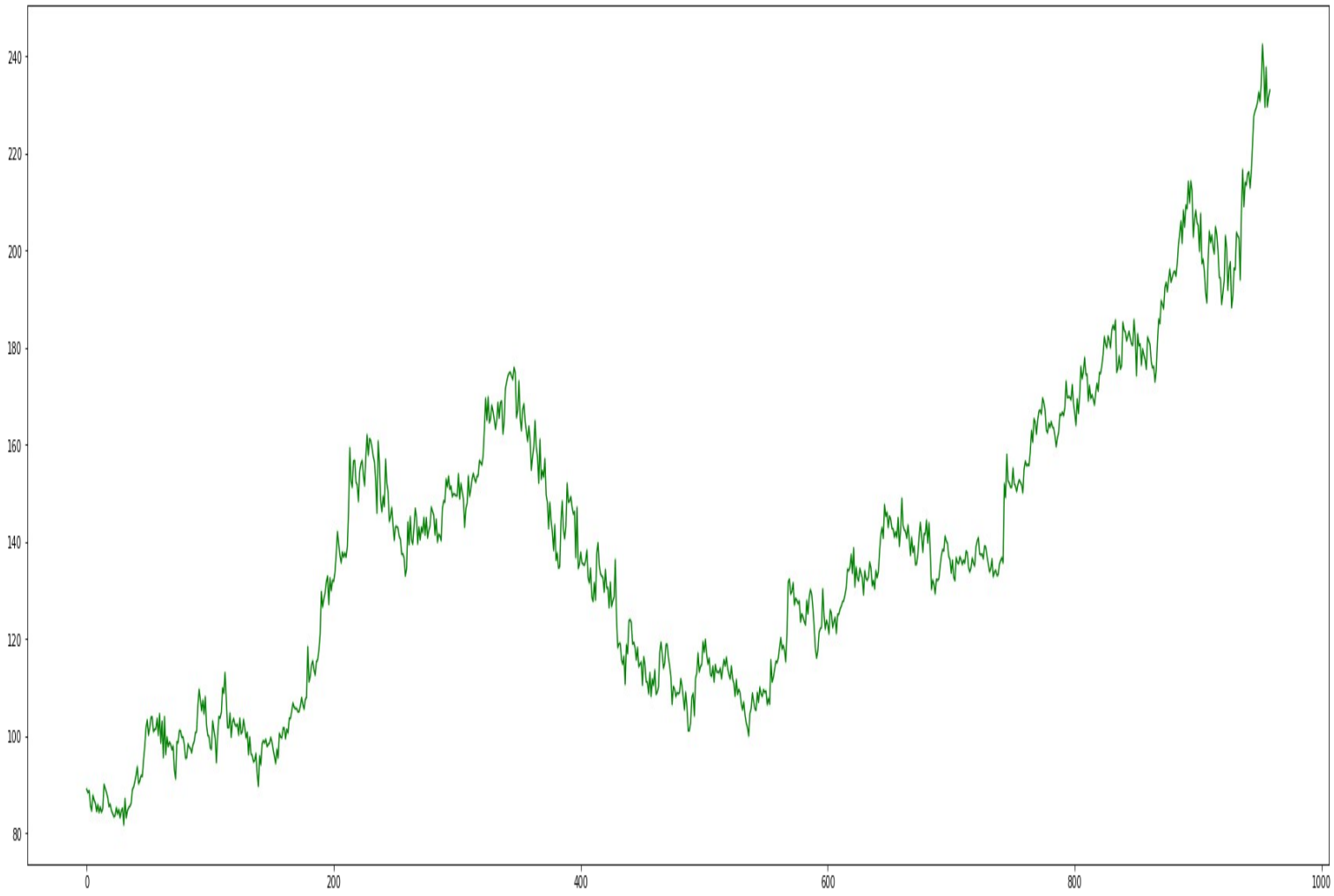
Following curve is shown the previous stock price of Apple corporation. where X axis denotes days and Y axis denotes closing share price



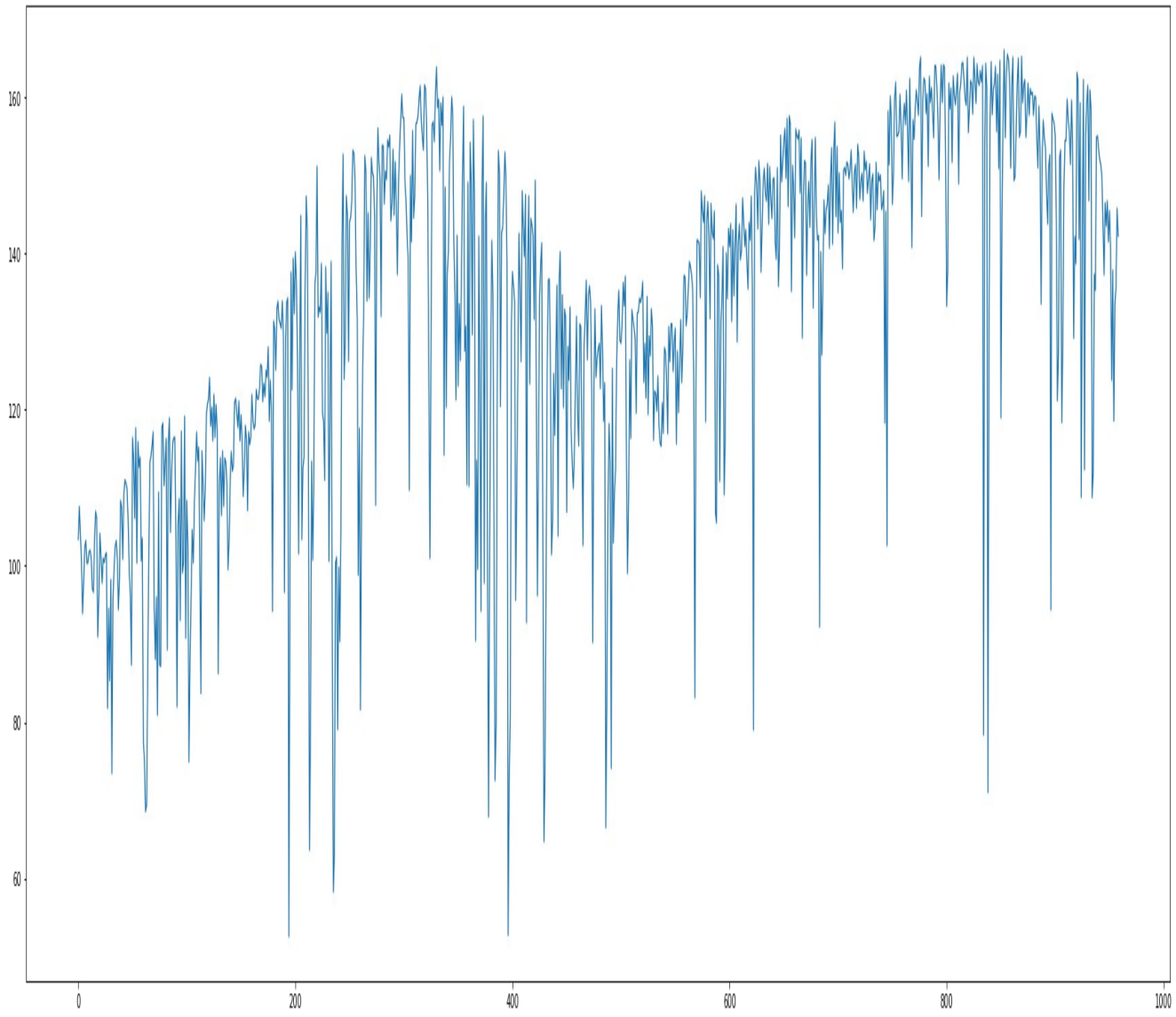
Following curve is shown comparison between actual sales and predicted sales of apple where X axis denotes days and Y axis denotes closing share price in KNN Regression



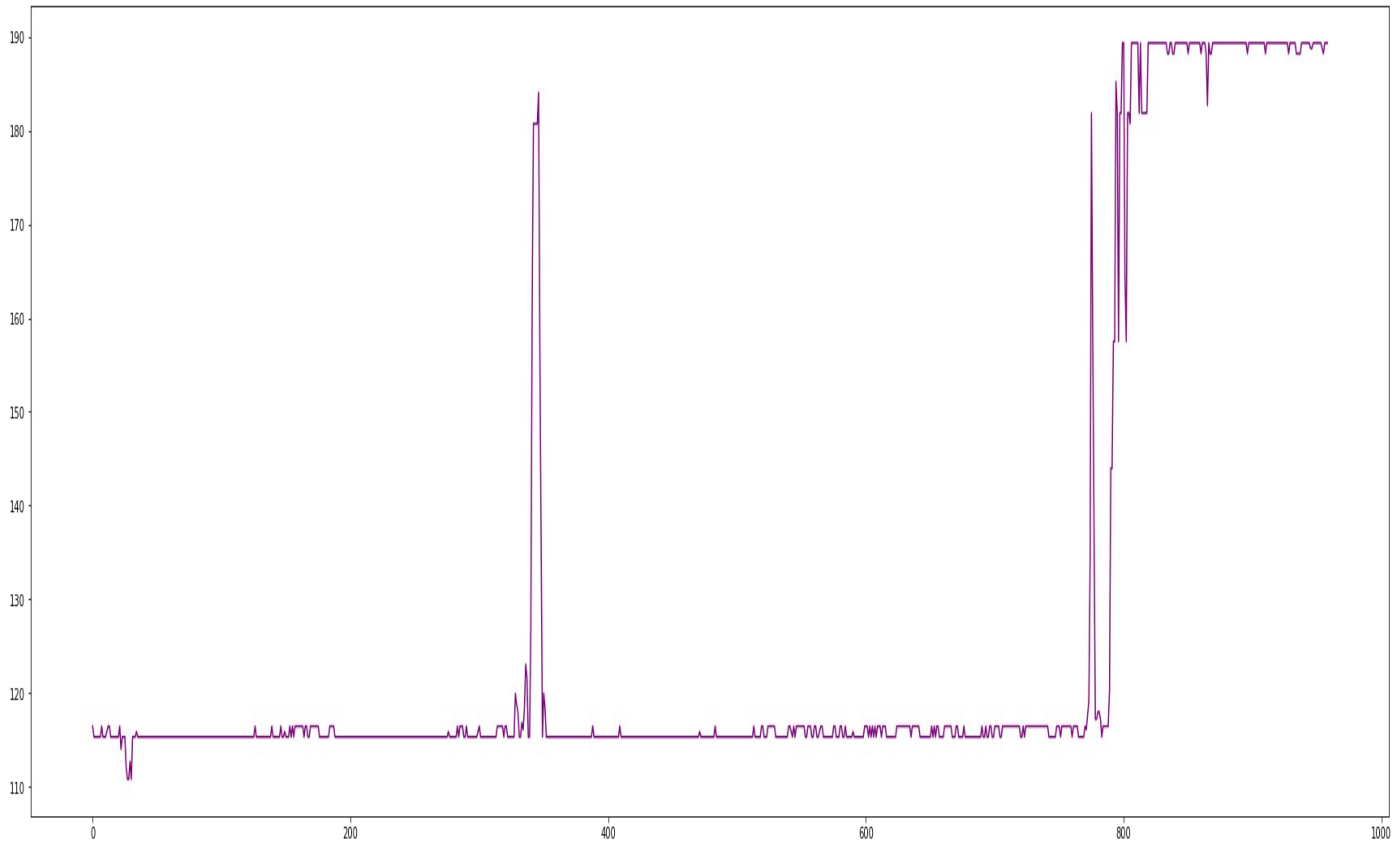
Following curve is shown comparison between actual sales and predicted sales of apple where X axis denotes days and Y axis denotes closing share price in Linear Regression



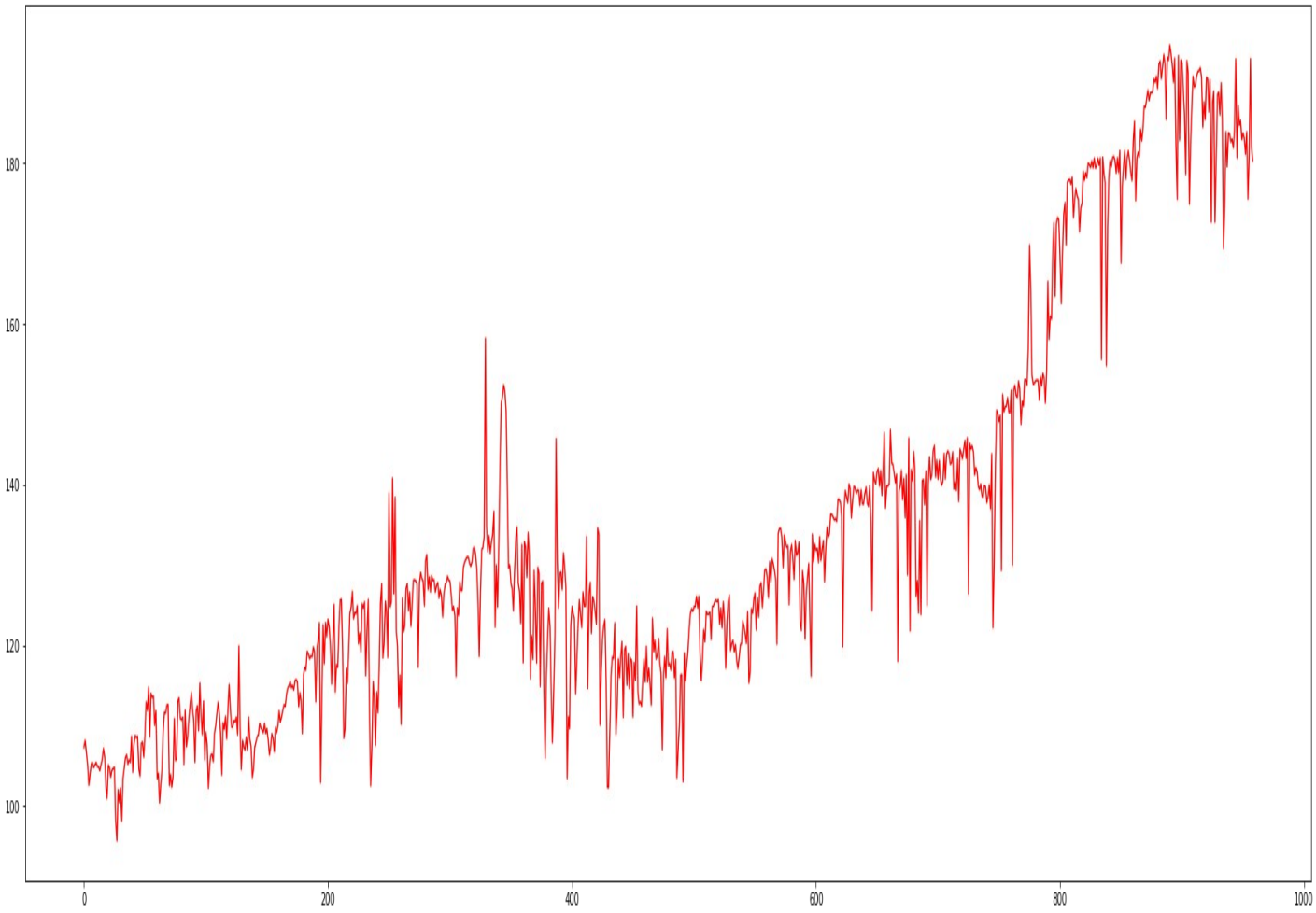
Following curve is shown comparison between actual sales and predicted sales of apple where X axis denotes days and Y axis denotes closing share price in SVR



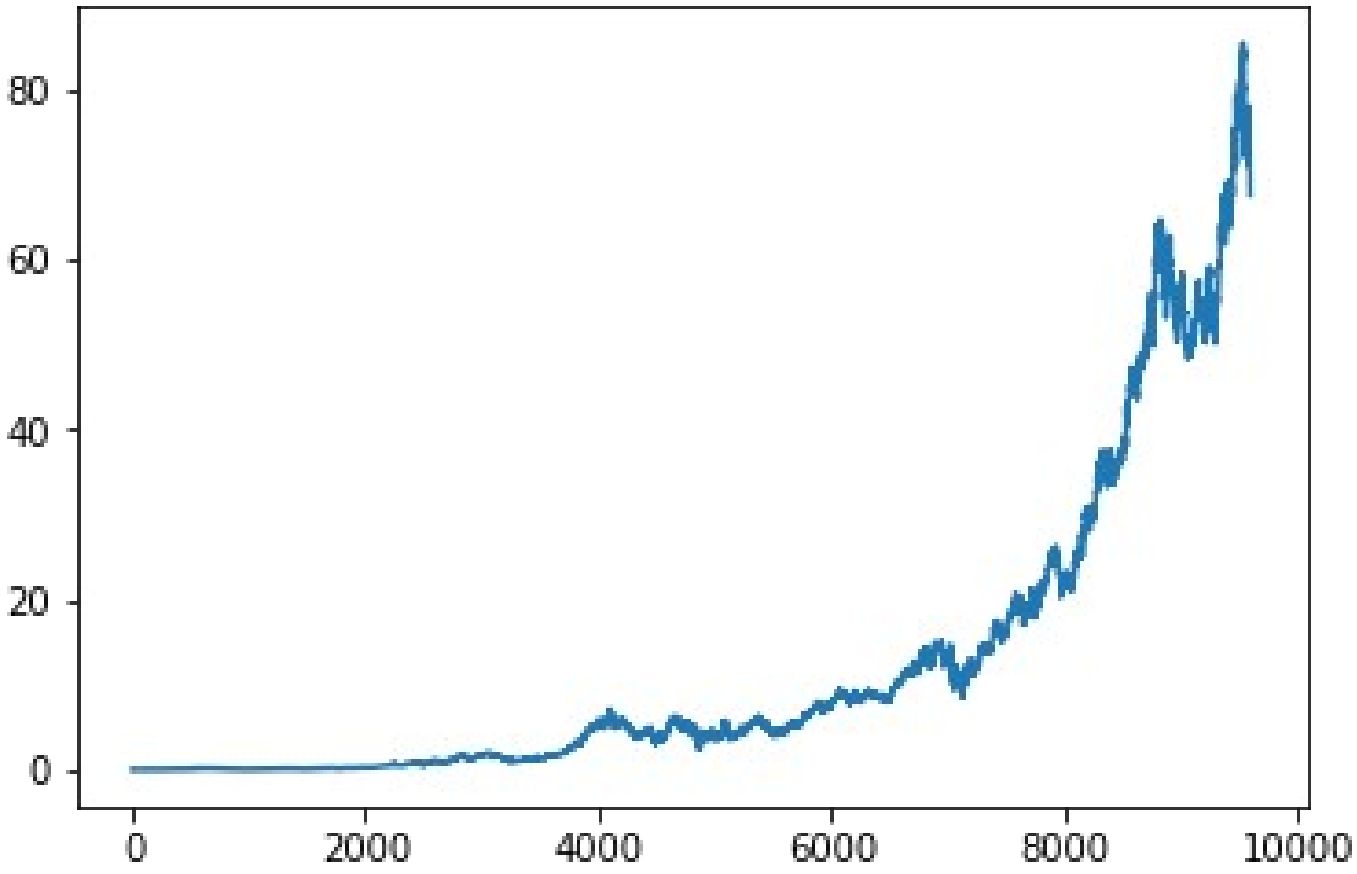
Following curve is shown comparison between actual sales and predicted sales of apple where X axis denotes days and Y axis denotes closing share price in random forest Regression



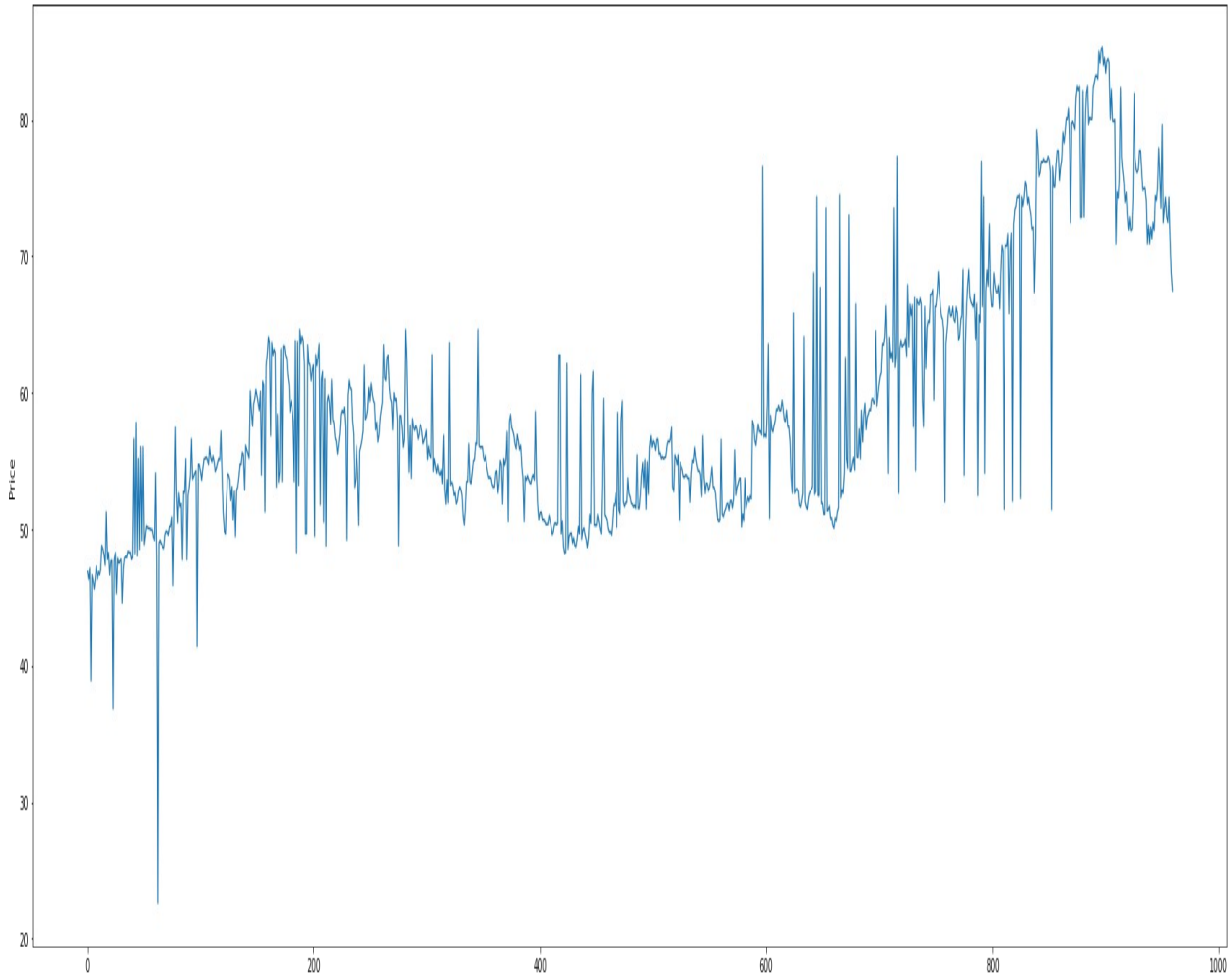
Following curve is shown comparison between actual sales and predicted sales where X axis denotes days and Y axis denotes closing share price in Proposed Regression Model



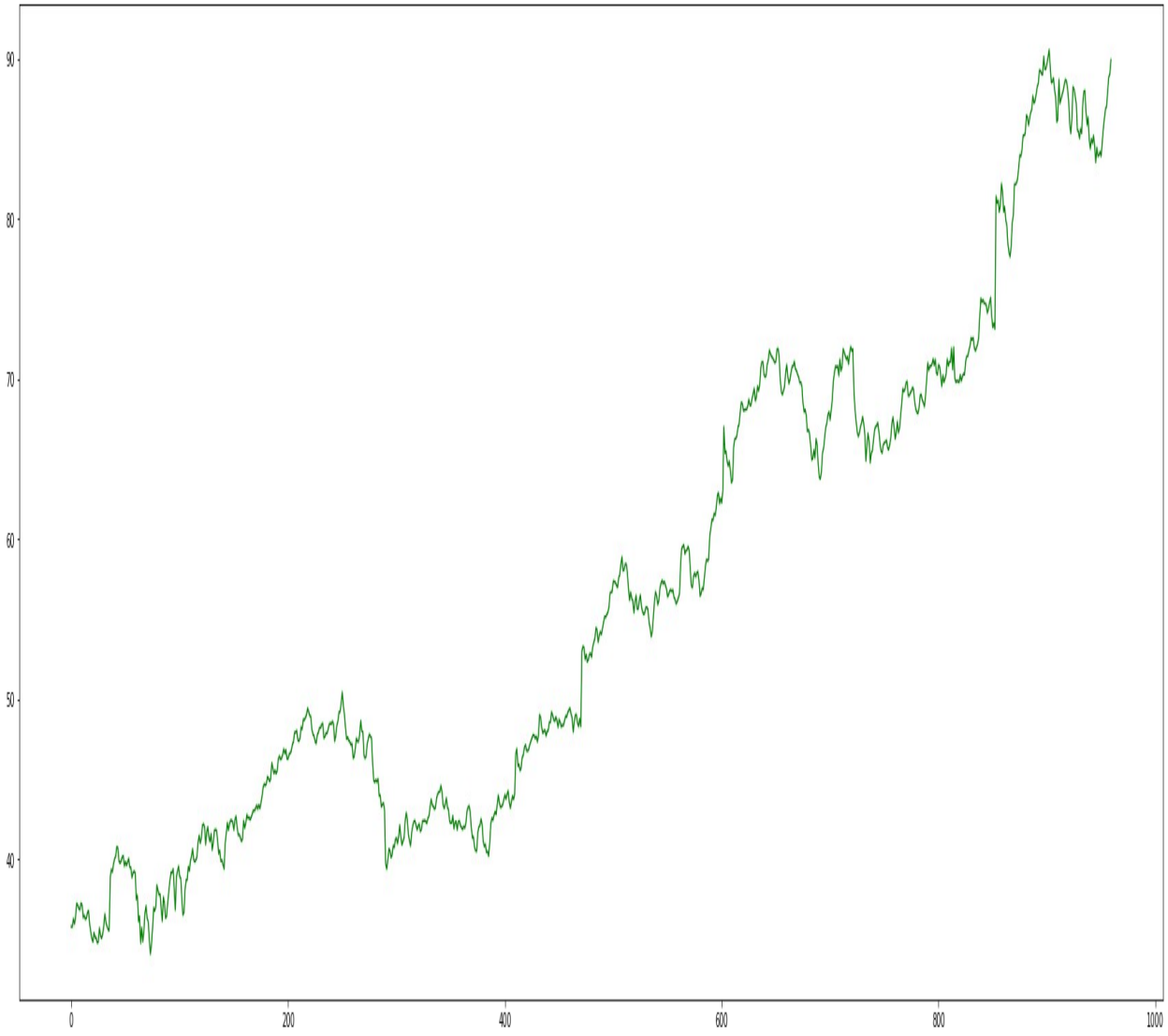
Following curve is shown the previous stock price of Nike corporation. where X axis denotes days and Y axis denotes closing share price



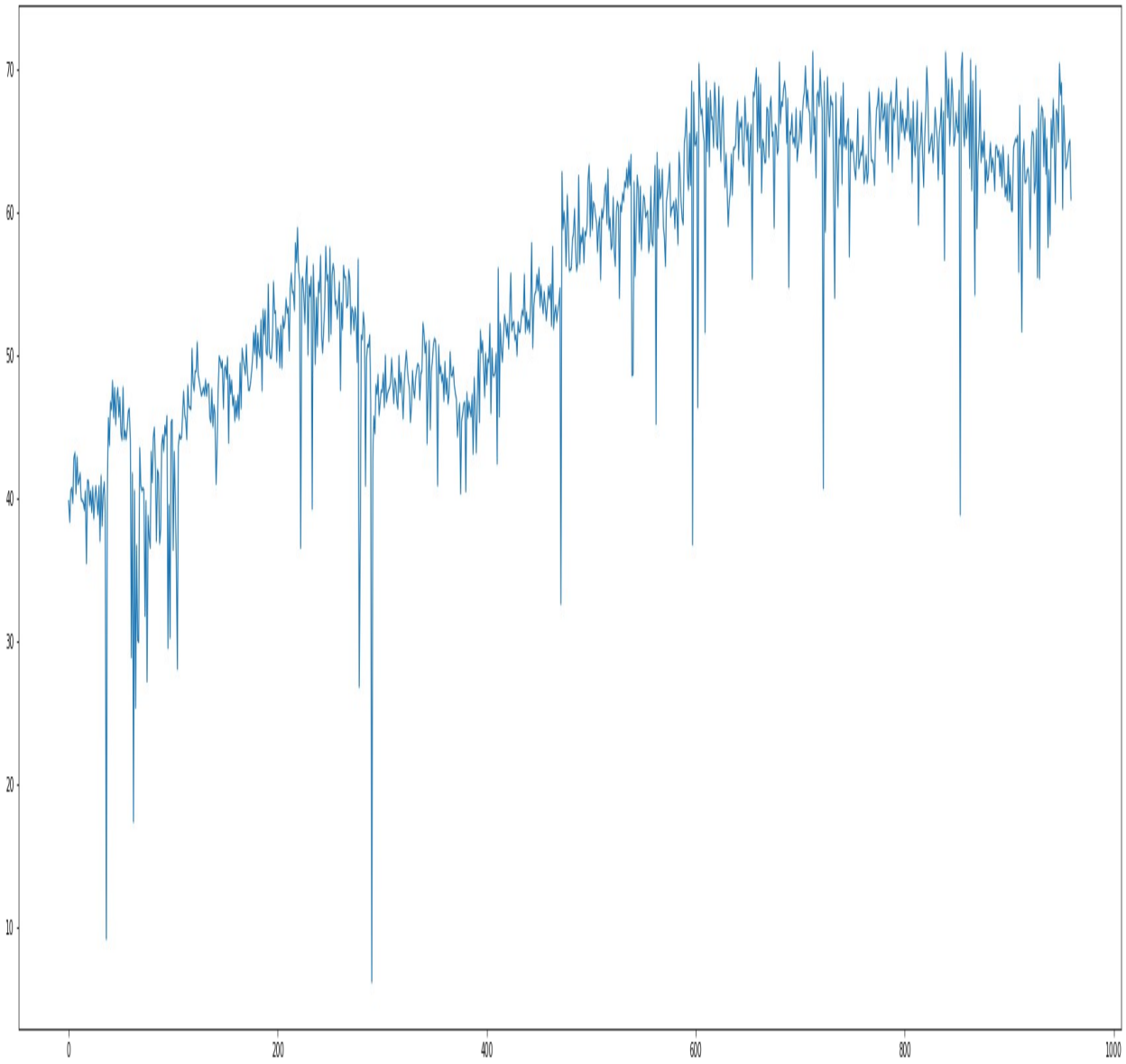
Following curve is shown comparison between actual sales and predicted sales where X axis denotes days and Y axis denotes closing share price in KNN Regression



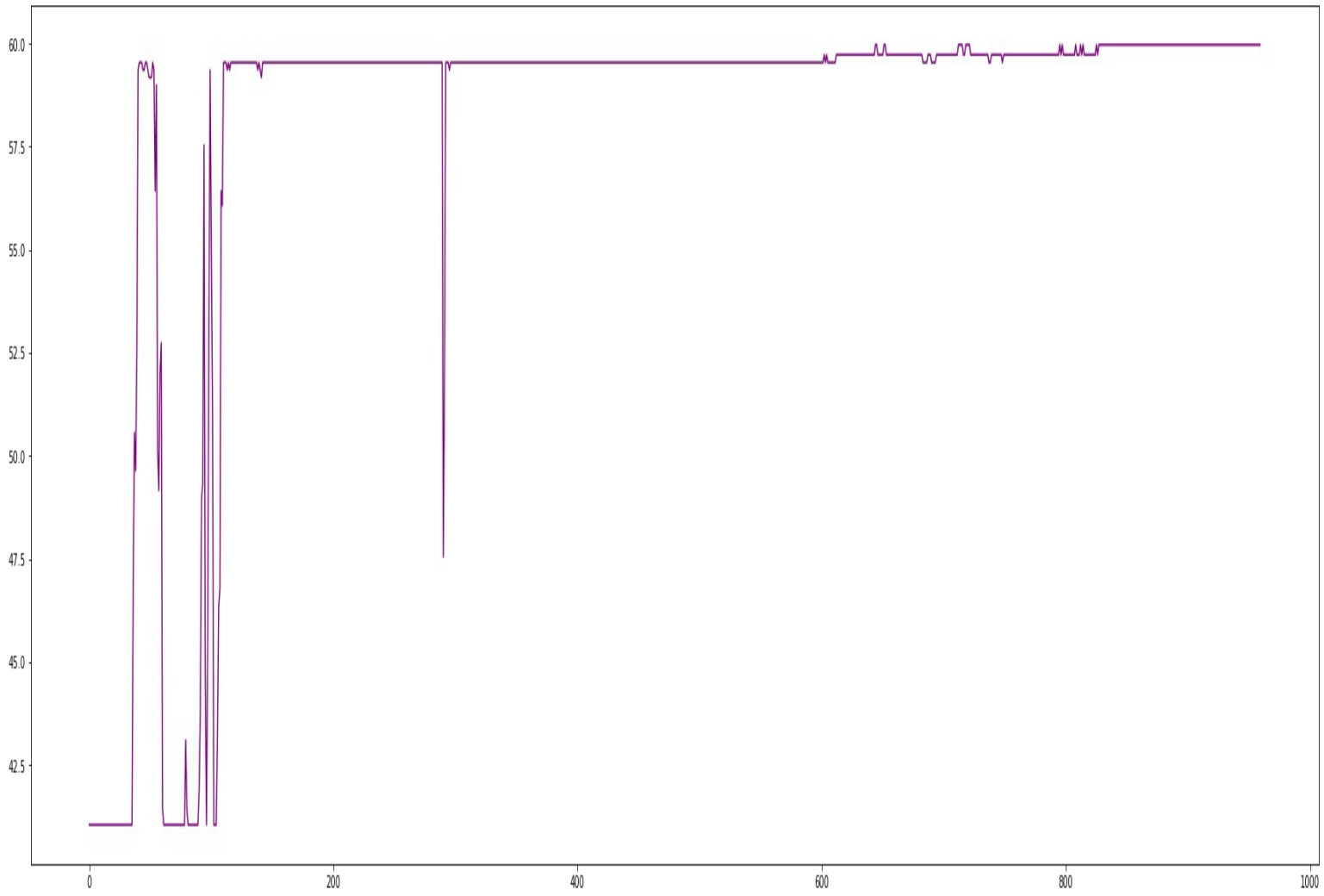
Following curve is shown comparison between actual sales and predicted sales where X axis denotes days and Y axis denotes closing share price in Linear Regression



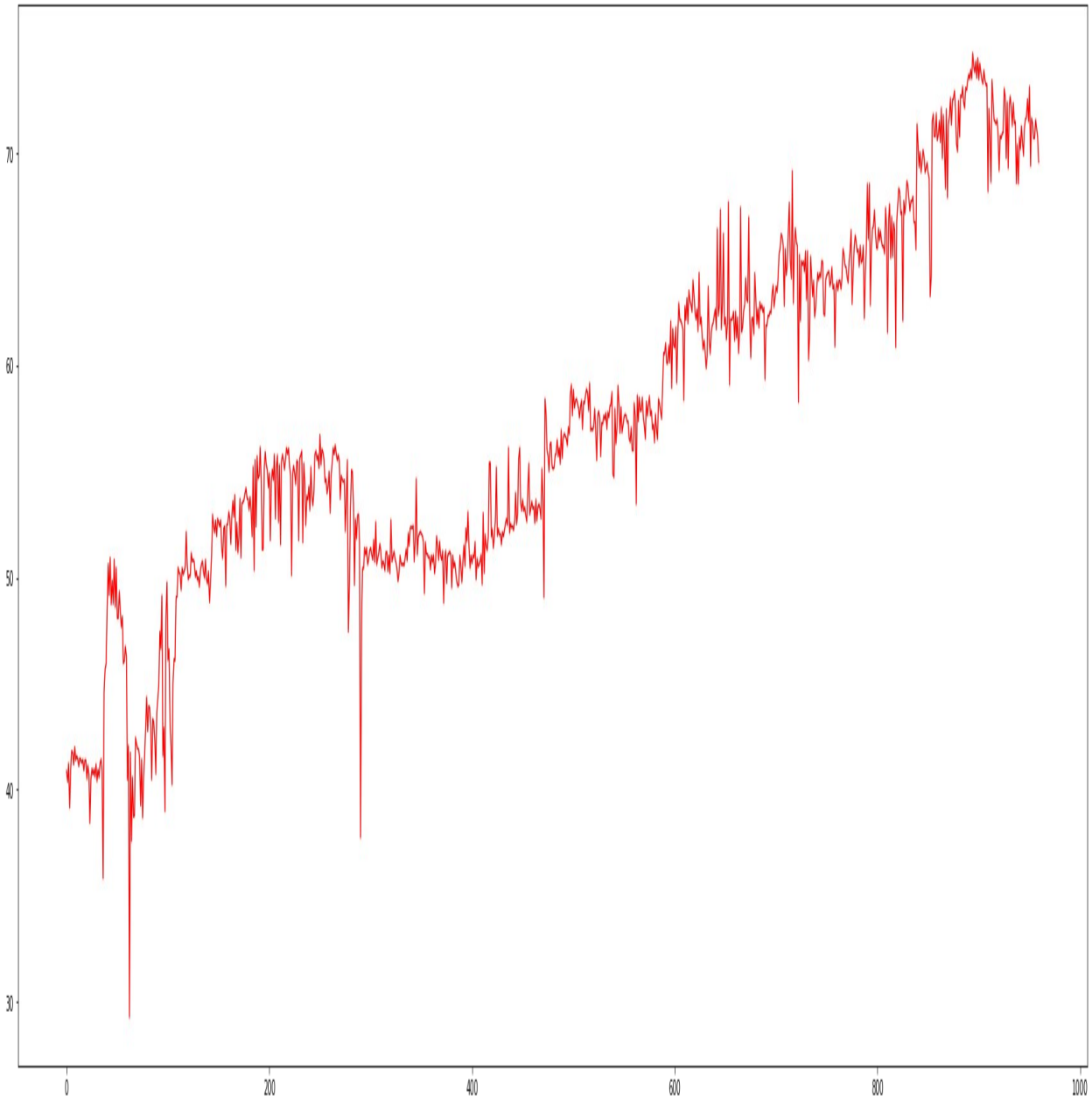
Following curve is shown comparison between actual sales and predicted sales where X axis denotes days and Y axis denotes closing share price in SVR



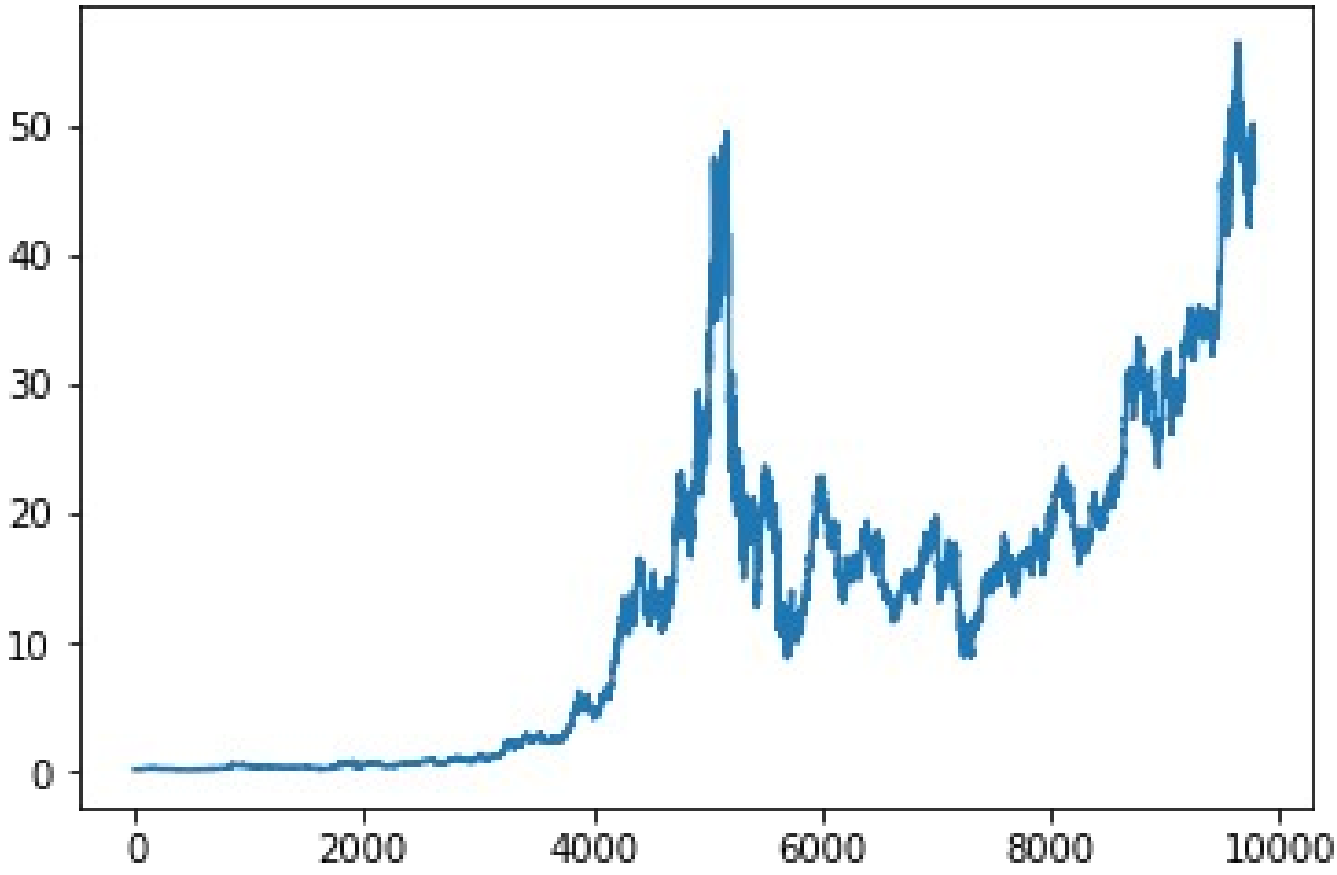
Following curve is shown comparison between actual sales and predicted sales where X axis denotes days and Y axis denotes closing share price in Random Forest Regression



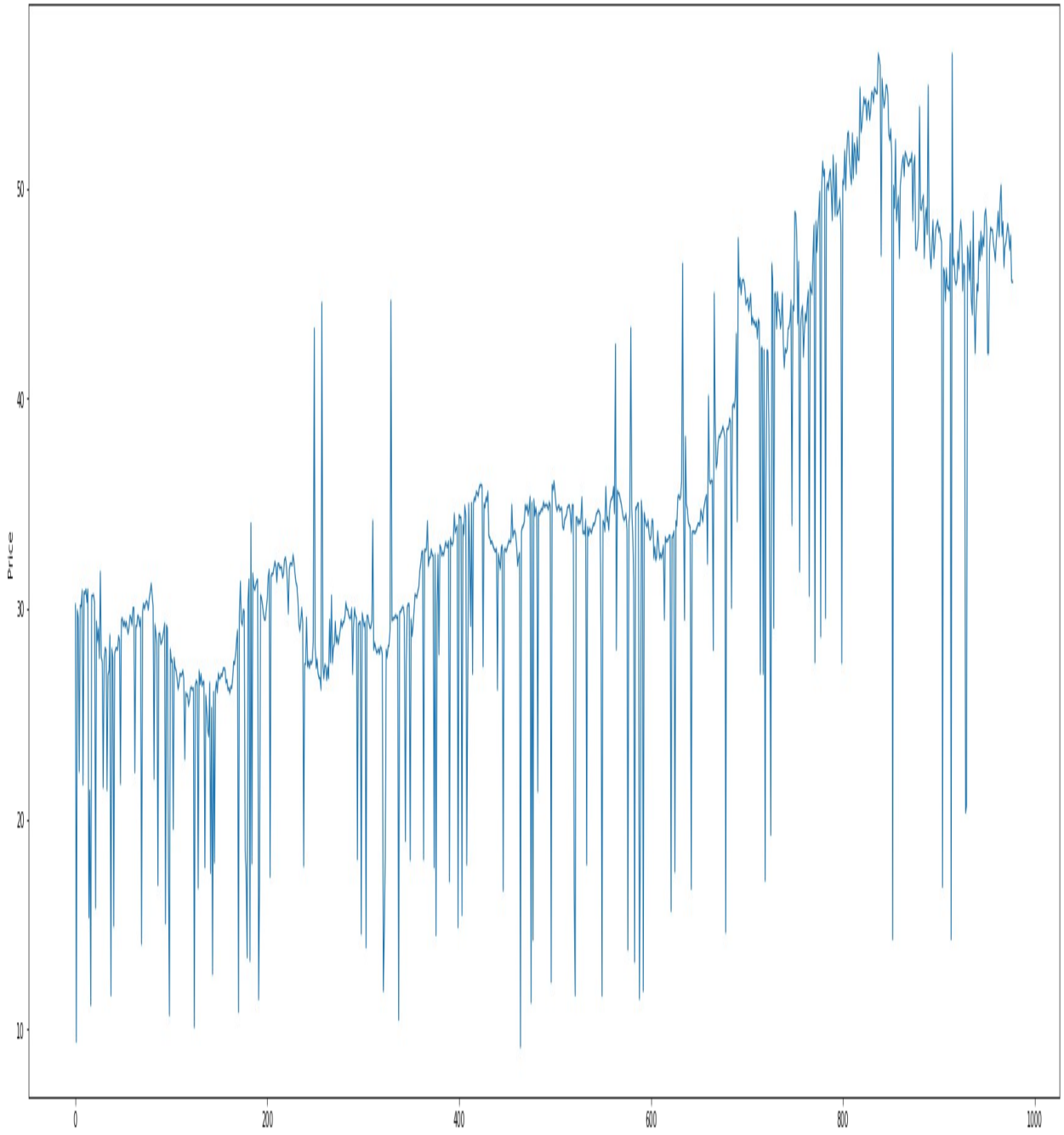
Following curve is shown comparison between actual sales and predicted sales where X axis denotes days and Y axis denotes closing share price in Proposed Regression Model



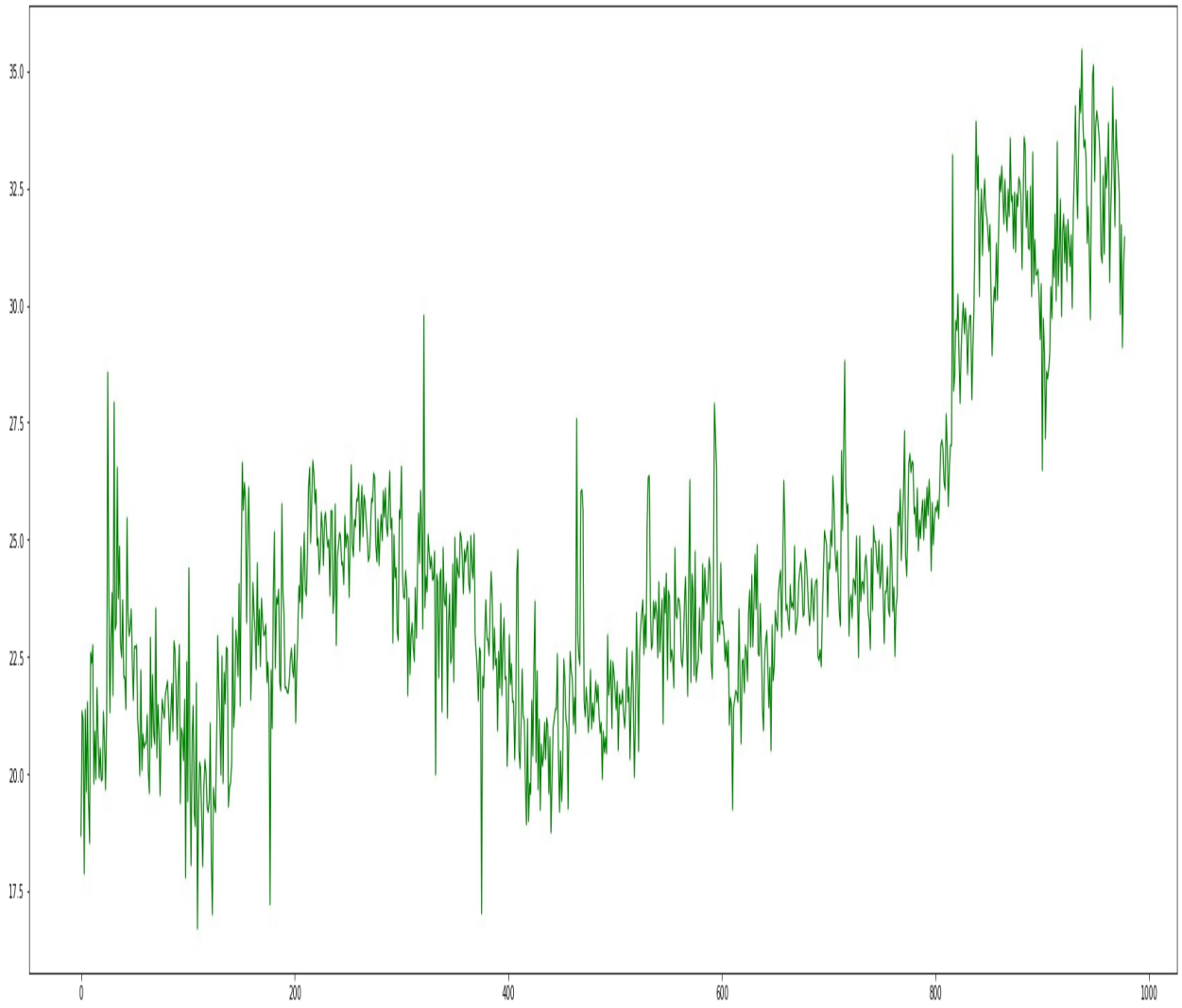
Following curve is shown the previous stock price of INtel corporation. where X axis denotes days and Y axis denotes closing share price



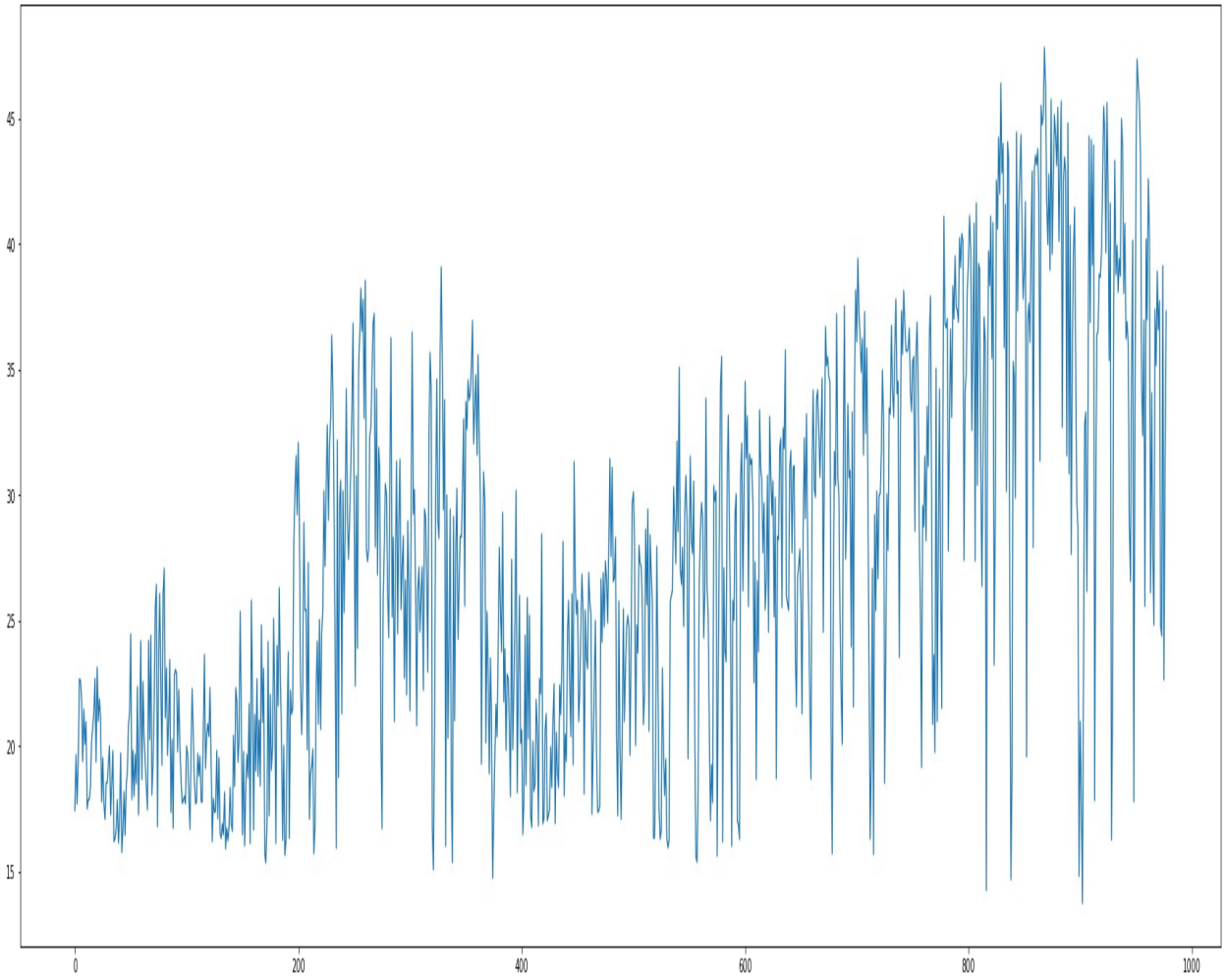
Following curve is shown comparison between actual sales and predicted sales where X axis denotes days and Y axis denotes closing share price in KNN Regression



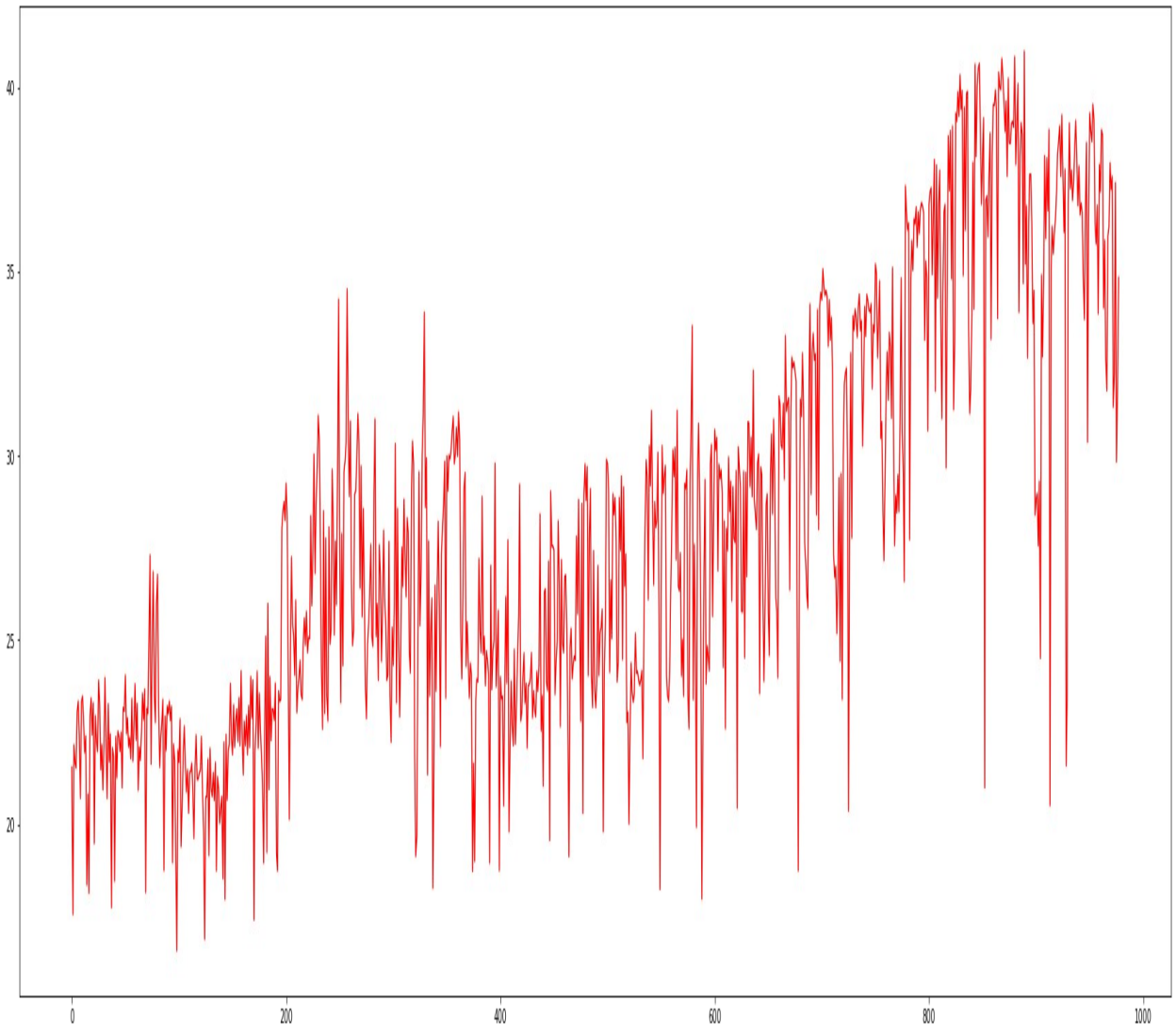
Following curve is shown comparison between actual sales and predicted sales where X axis denotes days and Y axis denotes closing share price in Linear Regression



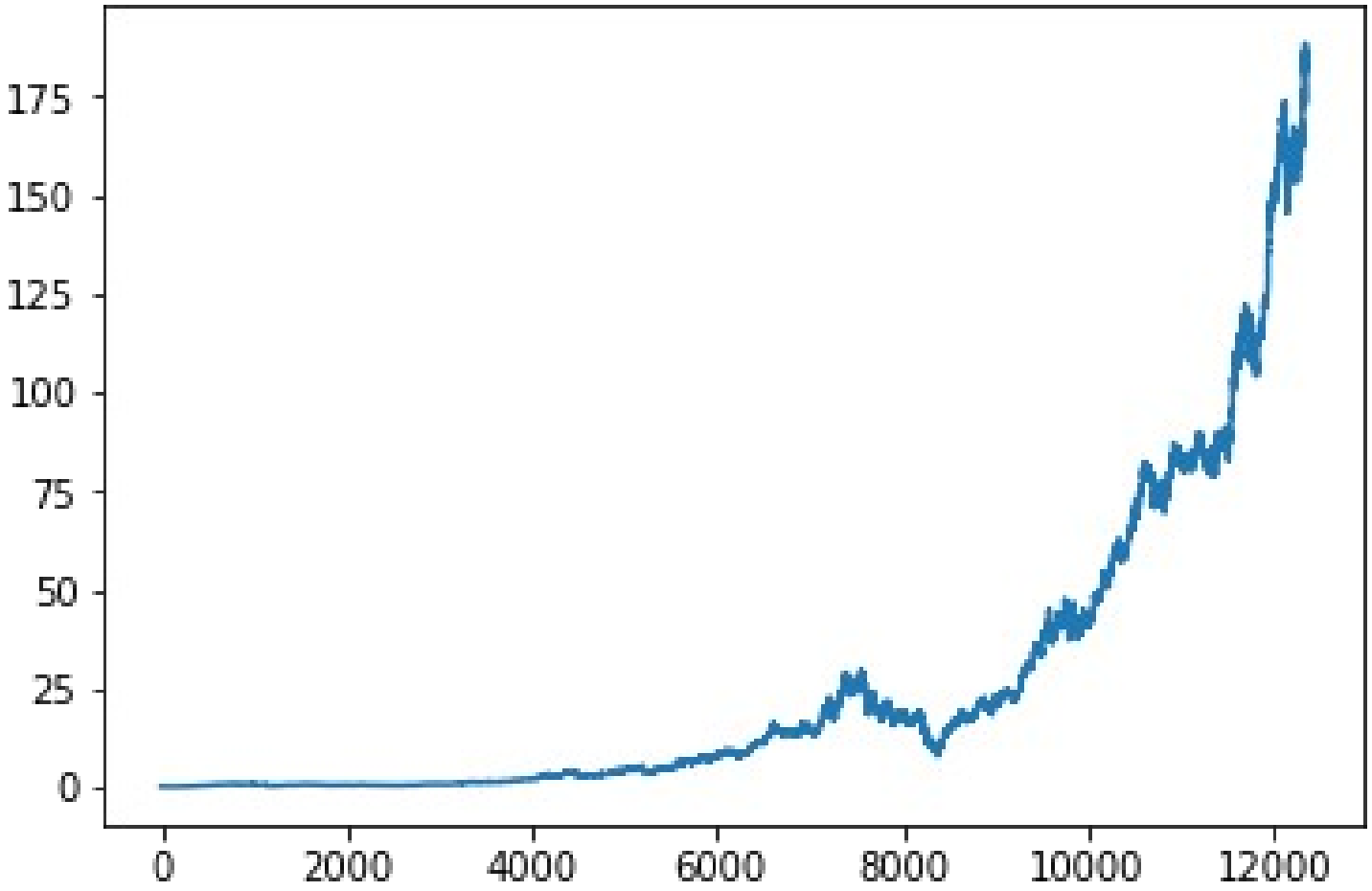
Following curve is shown comparison between actual sales and predicted sales where X axis denotes days and Y axis denotes closing share price in SVR



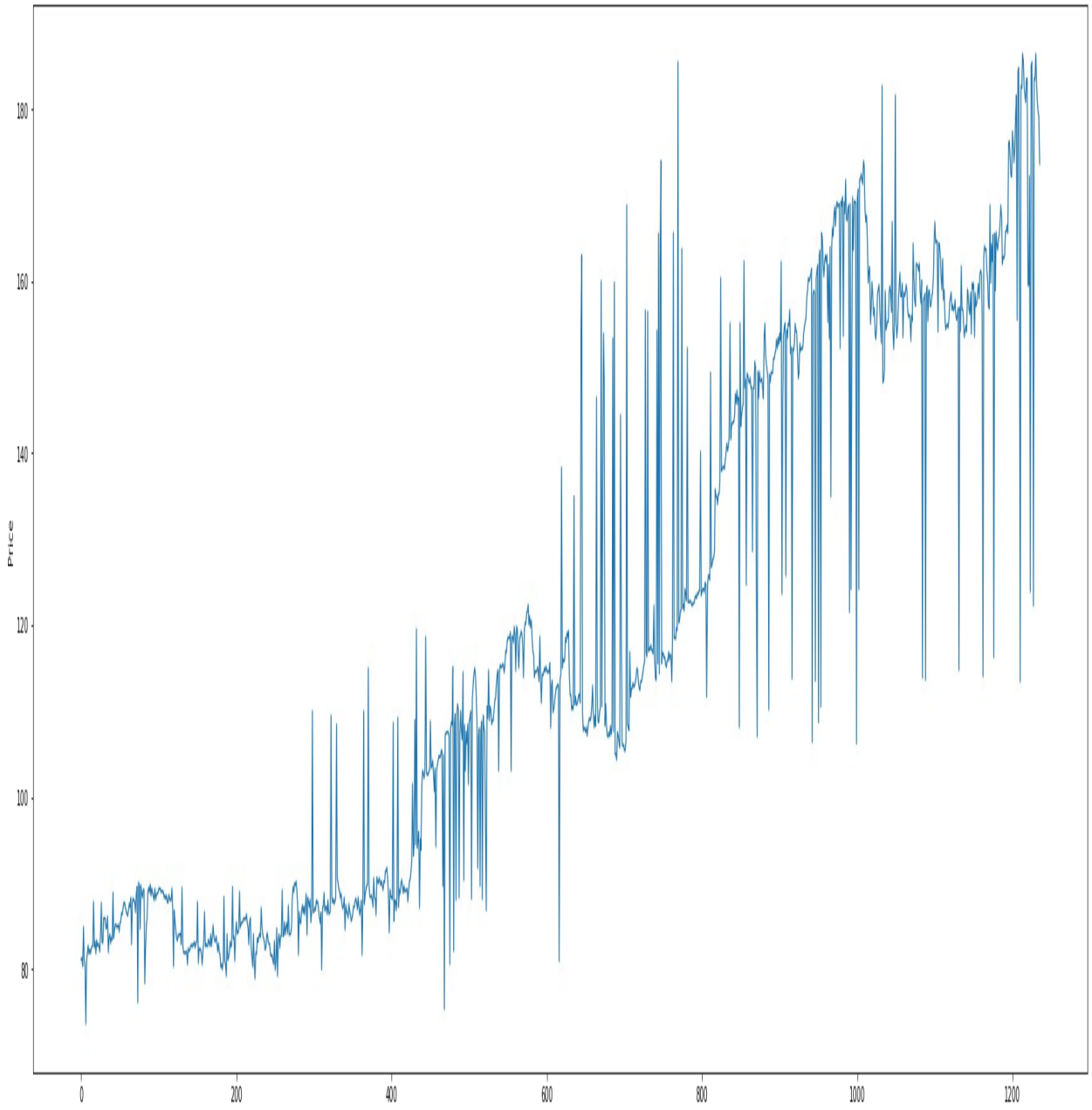
Following curve is shown comparison between actual sales and predicted sales where X axis denotes days and Y axis denotes closing share price in proposed Regression model



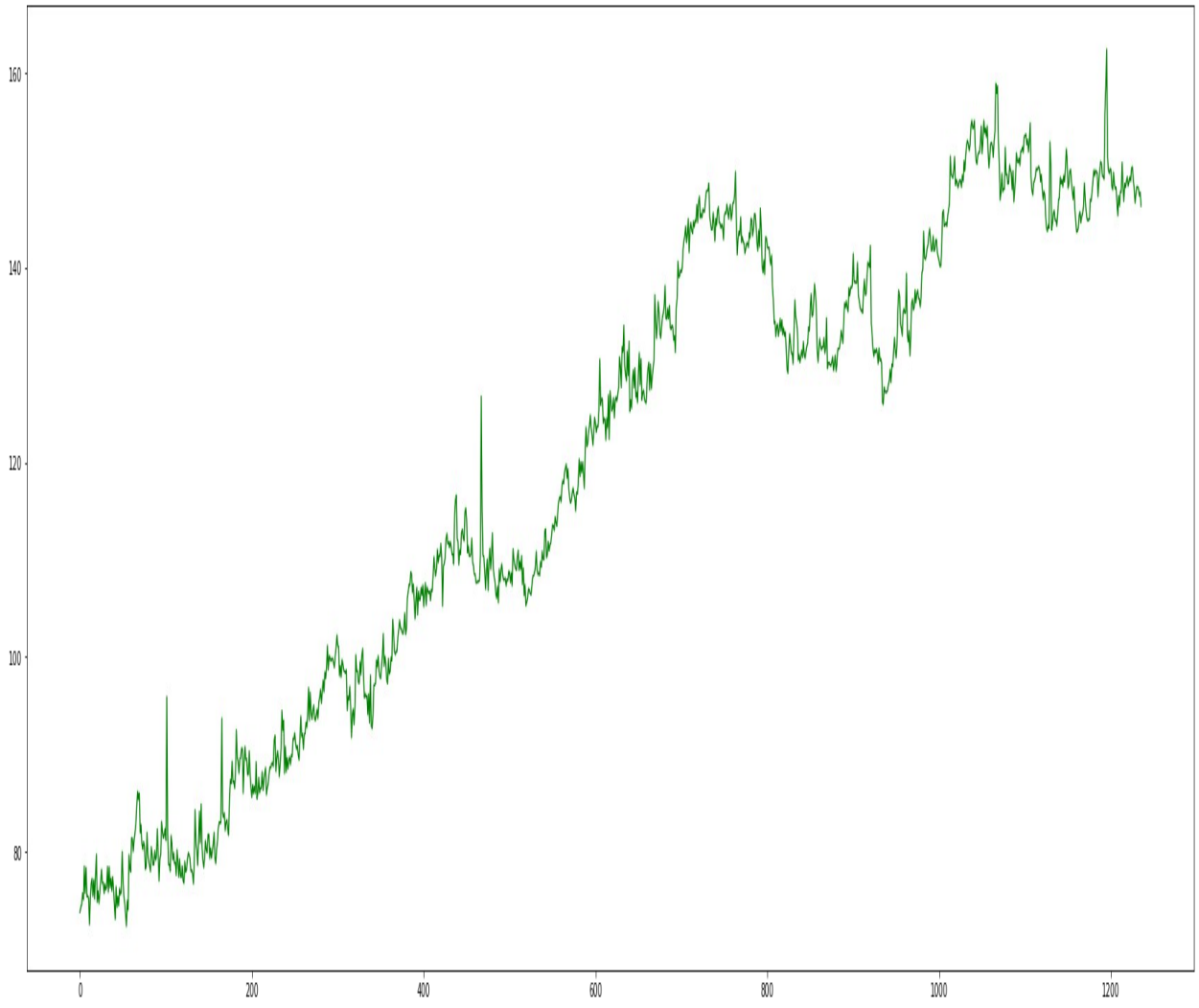
Following curve is shown the previous stock price of MACDONALD corporation. where X axis denotes days and Y axis denotes closing share price



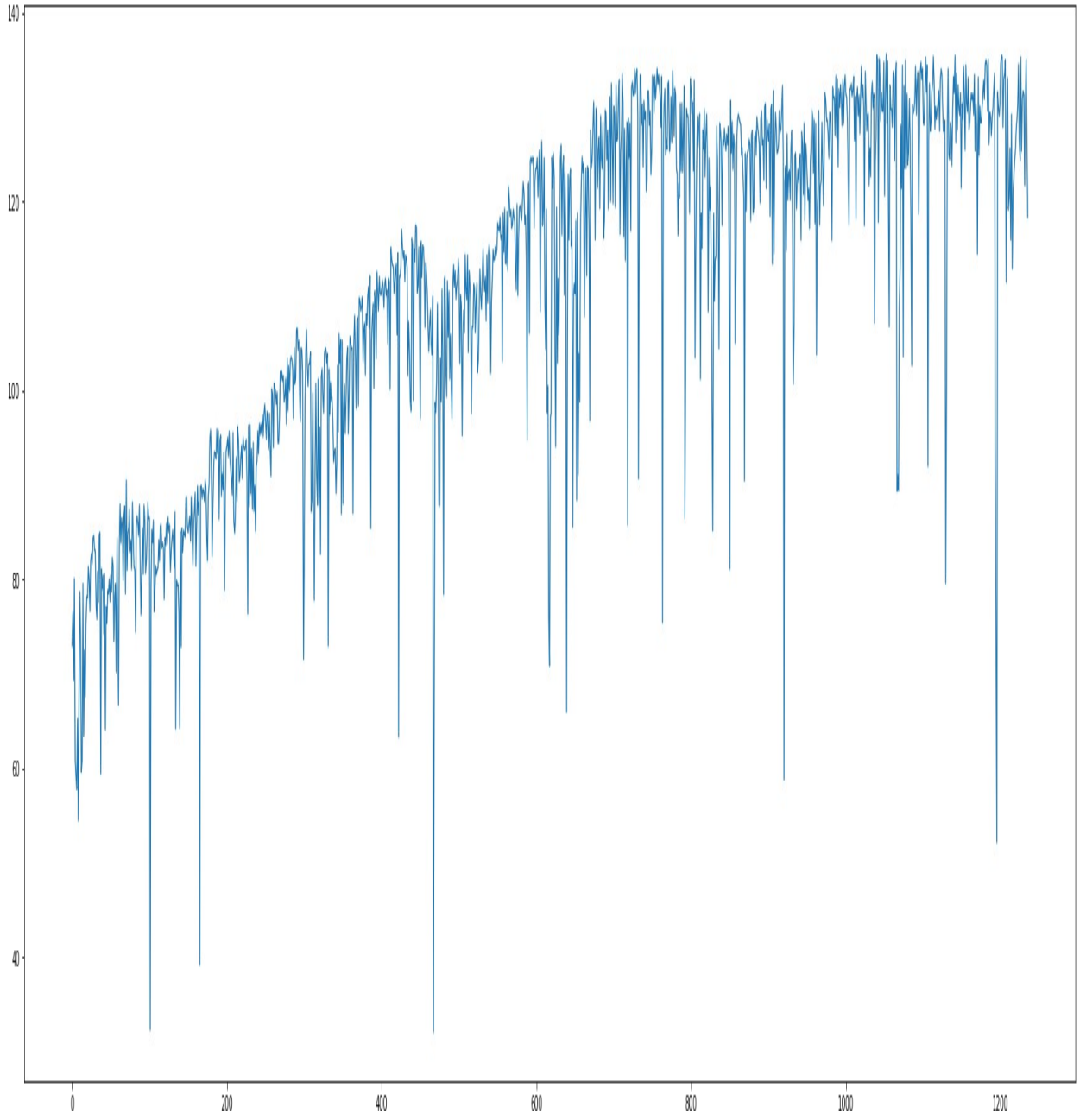
Following curve is shown comparison between actual sales and predicted sales where X axis denotes days and Y axis denotes closing share price in KNN Regression



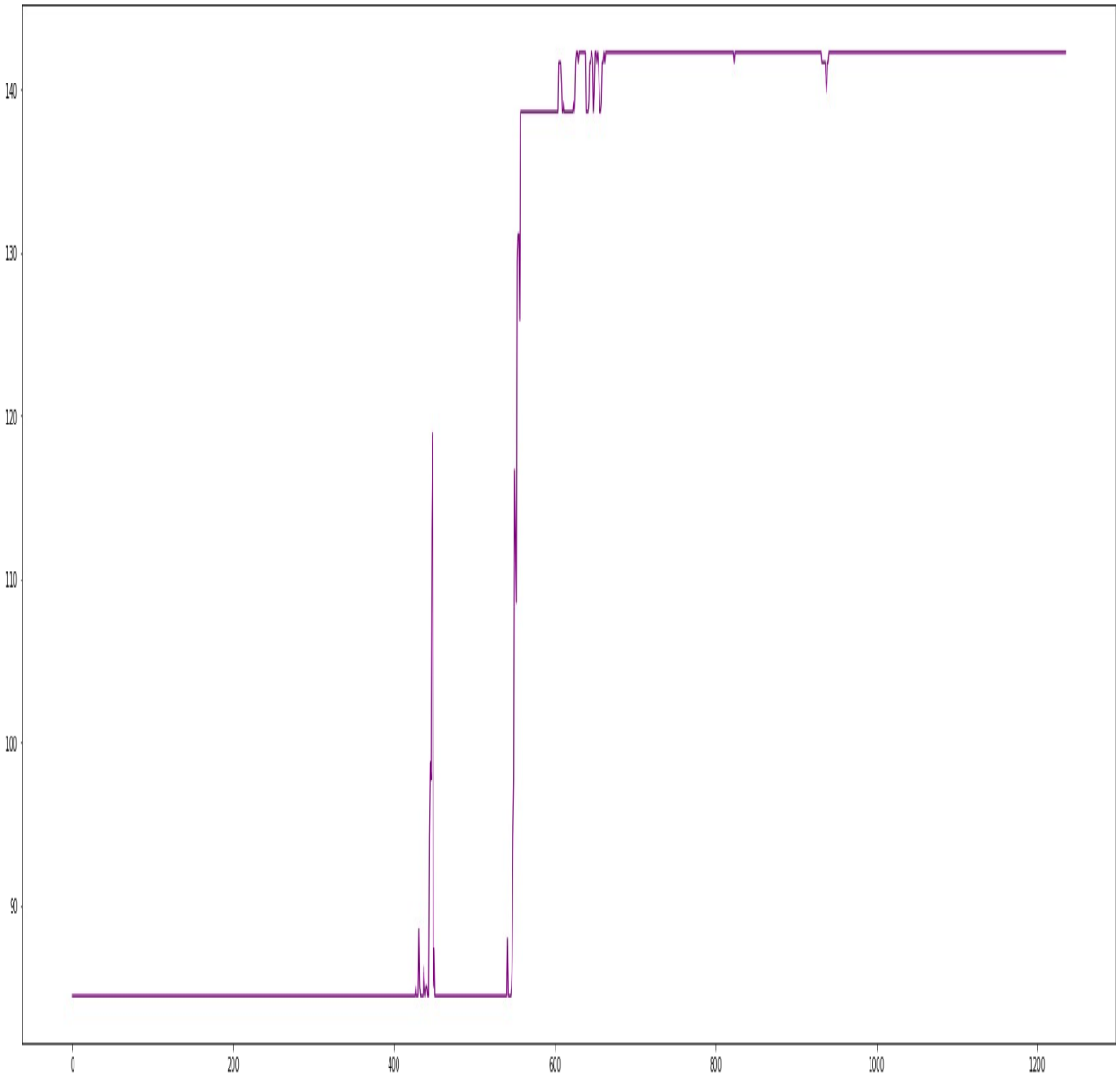
Following curve is shown comparison between actual sales and predicted sales where X axis denotes days and Y axis denotes closing share price in Linear Regression



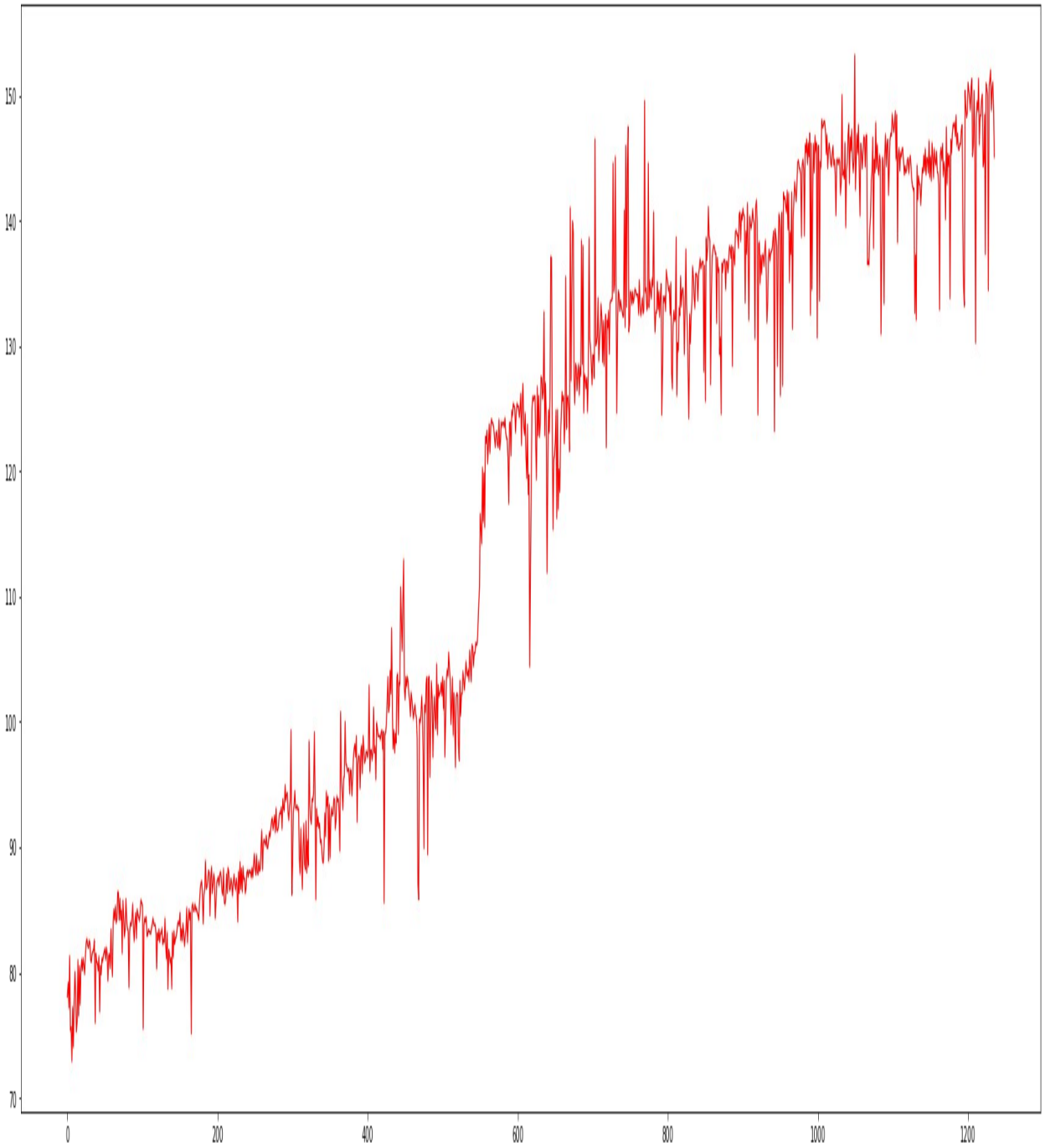
Following curve is shown comparison between actual sales and predicted sales where X axis denotes days and Y axis denotes closing share price in SVR



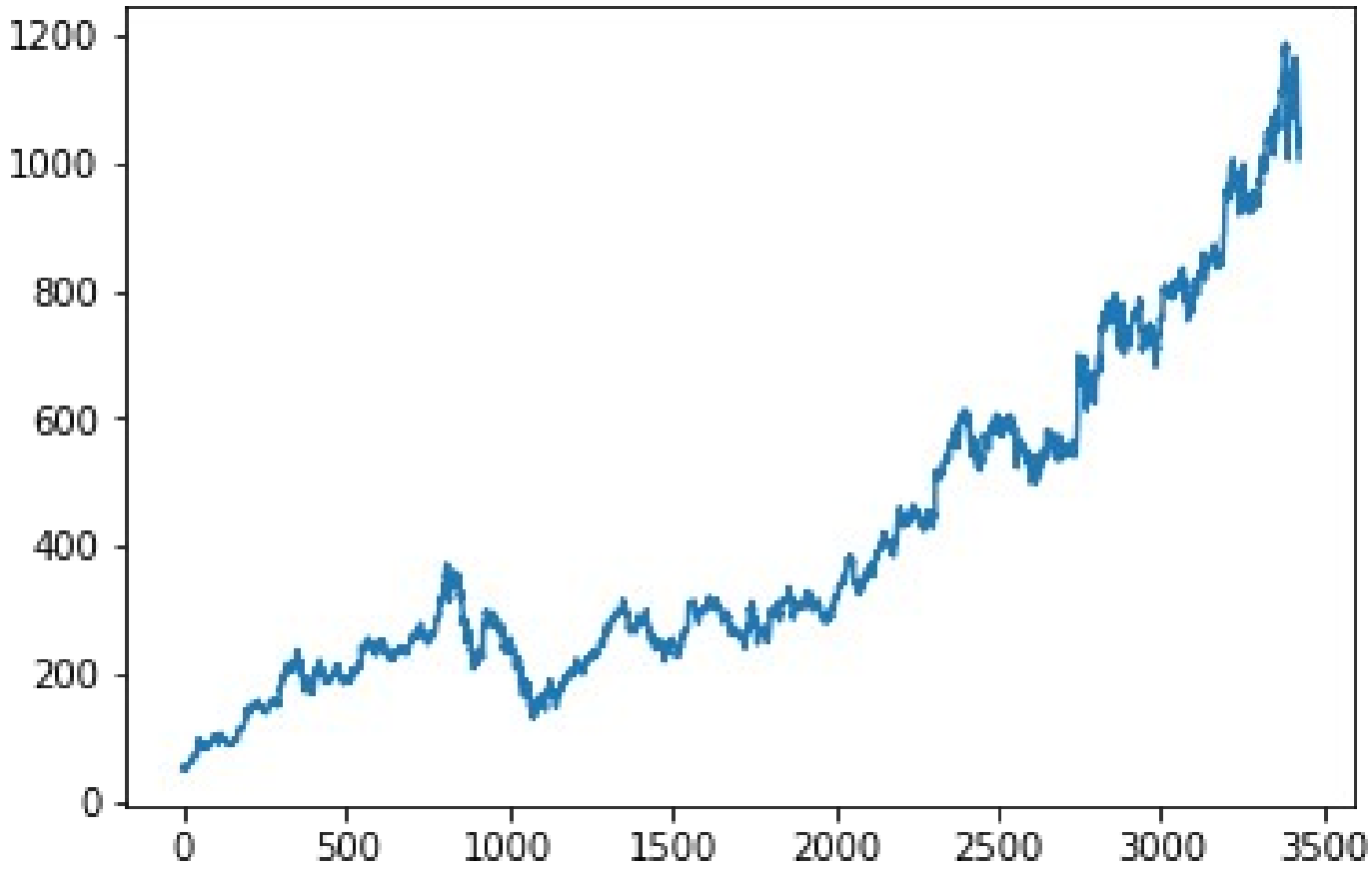
Following curve is shown comparison between actual sales and predicted sales where X axis denotes days and Y axis denotes closing share price in random forest Regression



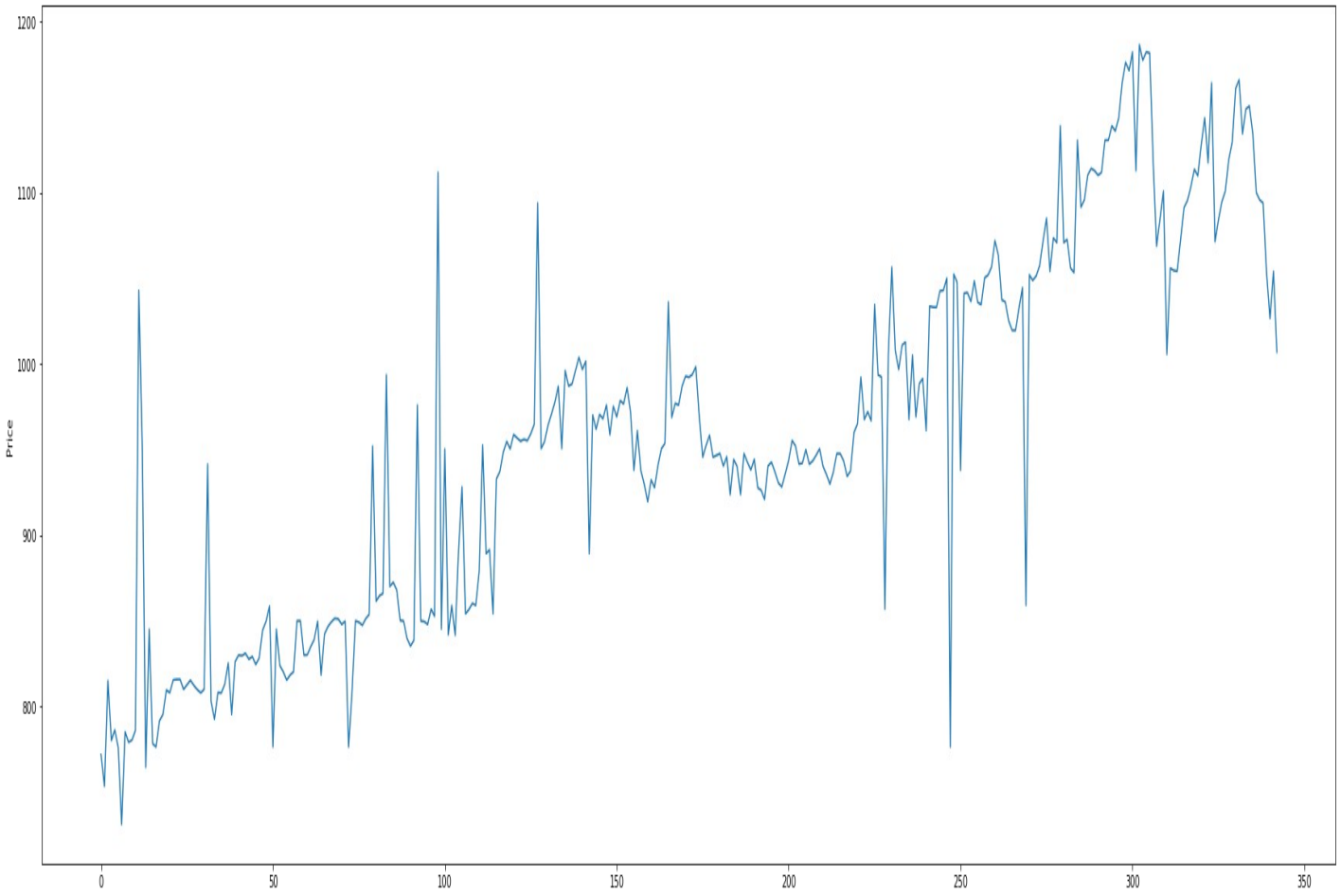
Following curve is shown comparison between actual sales and predicted sales where X axis denotes days and Y axis denotes closing share price in proposed Regression Model



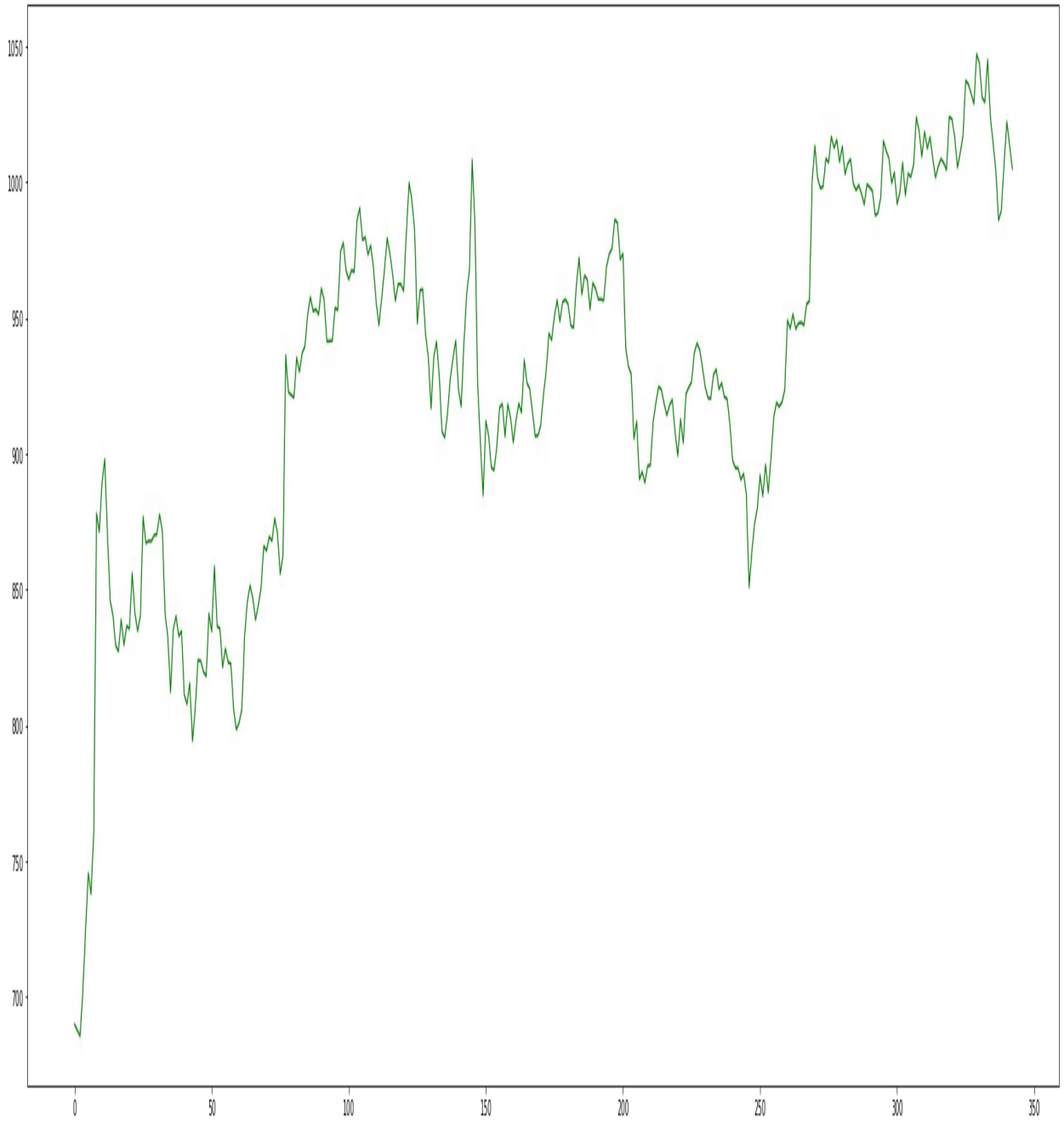
Following curve is shown the previous stock price of IBM corporation. where X axis denotes days and Y axis denotes closing share price



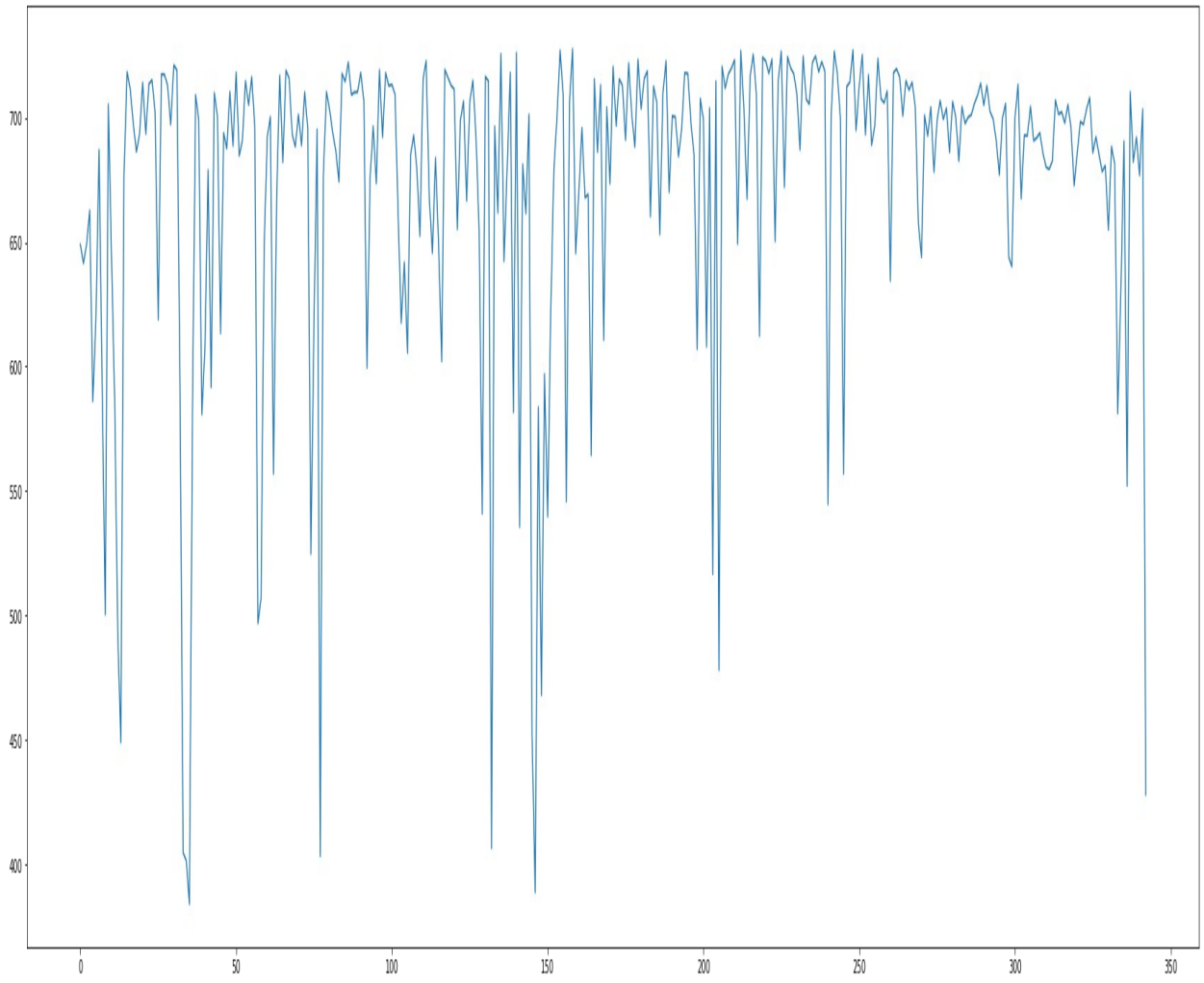
Following curve is shown comparison between actual sales and predicted sales where X axis denotes days and Y axis denotes closing share price in KNN Regression



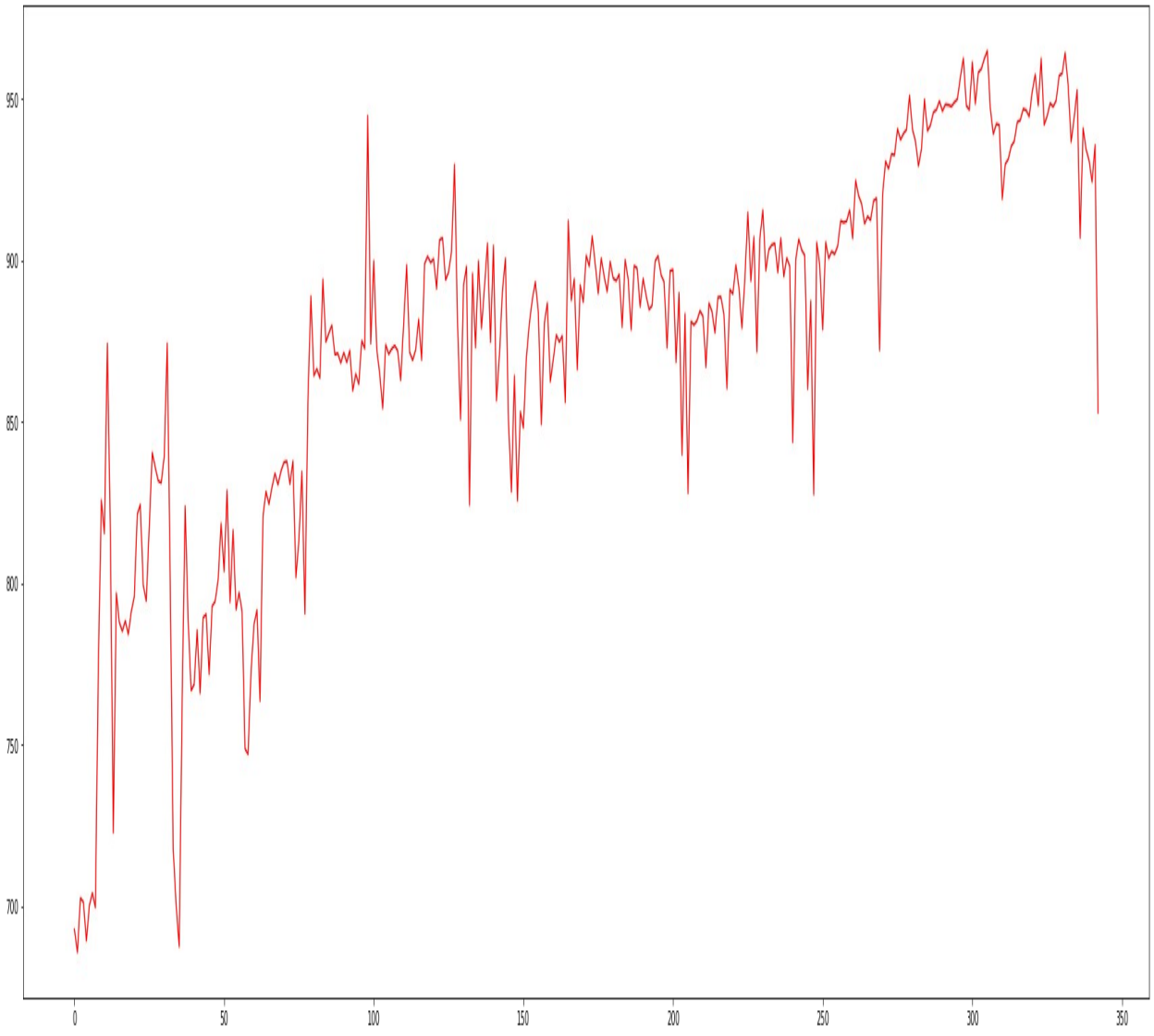
Following curve is shown comparison between actual sales and predicted sales where X axis denotes days and Y axis denotes closing share price in Linear Regression



Following curve is shown comparison between actual sales and predicted sales where X axis denotes days and Y axis denotes closing share price in SVR



Following curve is shown comparison between actual sales and predicted sales where X axis denotes days and Y axis denotes closing share price in proposed Regression



Chapter 8
Conclusion

It is known that if we are only concerned for the best possible correlation coefficient, it might be difficult or impossible to find a single regression model that performs as well as a good ensemble of regression models. In this study, we built an ensemble of regression models using four different learning methods .After determining and comparing with other models. In our proposed model we have attained the highest accuracy among all others.

REFERENCES

[1]J. Fox, Applied Regression Analysis, Linear Models, and Related Methods, ISBN: 080394540X, Sage Pubns (1997).

[2]L. Breiman, Bagging Predictors. Machine Learning, 24(3) (1996) 123-140.

[3]D. Opitz, R. Maclin, Popular Ensemble Methods: An Empirical Study, Artificial Intelligence Research, 11 (1999): 169-198, Morgan Kaufmann. 12.C. Perhch, F. J. Provost, J. S.

[4]Linear regression Analysis on Net Income of an Agronomical Company ,Supichaya Sunthornjittanon:Portland State University Efficient and Accurate

[5]knn based Classification and Regression,Harshit Dubey:International Institute of Information Technology

[6]Design and Training of Support Vector Machines,Alistair Shilton:The University of Melbourne

[7]Advances in Random Forests with Application to Classifcation,Arnu Pretorius:Stellenbosch University

[8]Cristianini and J. Shawe-Taylor.An Introduction to Support Vector Machines and other kernel-based learning method. Cambridge University Press, Cambridge, UK, 2000

[9] L. Breiman, Stacked Regression. Machine Learning, 24 (1996):49-

64.4.T.G. Dietterich, Ensemble methods in machine learning. In

Kittler, J., Roli, F., eds.:

[10] R. Agrawal, T. Imielinski, and A.Swami (1993) “Mining association

rules between sets of items in

large databases”.

(ACM SIGMOD '93).

[11] Wanzhong Yang. “Granule Based Knowledge Representation for

Intra and Inter Transaction Association Mining”, Queensland

University of Technology, July 2009.

[12] R.V.Argiddi , S.S.Apte “a study of association rule mining in

fragmented item-sets for prediction of transactions outcome in stock

trading systems” IJCET-2012.

[13] KANNIKA NIRAI VAANI M, E RAMARAJ “ AN ENHANCED

APPROACH TO DERIVE DISJUNCTIVE RULES FROM

ASSOCIATION RULE MINING WITHOUT CANDIDATE ITEM

GENERATION” IJCA-2013.

[14] Prashant S. Chavan, Prof. Dr. Shrishail. T. Patil” Parameters for Stock

Market Prediction” IJCTA | (Mar-Apr 2013).

[15] Multiple Classifier Systems. LNCS Vol. 1857, Springer (2001) 1-1 N.

Duffy, D. Helmbold, Boosting Methods for Regression, Machine

Learning, 47, (2002)

A.A. Adebiyi, 1C.K. Ayo, 1M.O Adebiyi, and 2S.O. Otokiti “An Improved Stock Price Prediction using Hybrid Market Indicators”

ICT. (2002)

[16] J. Friedman, Stochastic Gradient Boosting,Computational Statistics and Data Analysis

[17] Y. Grandvalet, Bagging Equalizes Influence, Machine Learning, Volume 55(3) (2004)

[18] N.L. Hjort, G. Claeskens, Frequentist Model Average Estimators, Journal of the American Statistical Association,

[19] Y. Morimoto, H. Ishii, S. Morishita, Efficient Construction of Regression Trees with Range and Region Splitting, Machine Learning

[20]N.L. Hjort, G. Claeskens, Frequentist Model Average Estimators,

[22] Simonoff,Tree Induction vs. Logistic Regression: A Learning-Curve Analysis. Journal of MachineLearning Research 4 (2003) 211-255

[23] Y. Wang, I. H. Witten, Induction of model trees for predicting continuous classes, In Proc.of the Poster Papers of the European Conference on ML, (1997) 128-137.