BITCOIN PREDICTION USING MACHINE LEARNING

Siddharth Tiwari
SRMIST KTR
603203
sh5456@srmist.edu.in

Abstract

Bitcoins can be seen from an economic as well as computer science angle; this makes bitcoin an exciting field of study. Studying its behavior is of great importance to countries as many of them are now legalizing this form of virtual currency. Thus predicting its price and analyzing the trend in the open market will be boon not only to an economically growing country but also to every person who wants to invest in Bitcoins. This paper/project focuses on predicting the prices of Bitcoin, the most in-demand crypto-currency of today's world. We propose to predict the prices accurately by gathering data available at coinmarketcap while taking various hyper-parameters into consideration which have affected the bitcoin prices until now. We will be using various machine learning models like linear and polynomial regression, Bayesian regression and Time-Series models like AR(autoregressive), MA(moving average), ARIMA, ARCH and VAR. We will try to explain each model's advantages and disadvantages to see which one works the best. We will also extend the observations to show why Time series models are used heavily for forecasting purposes. GitHub repo for code - https://github.com/siddharthhh21/BTC-Price-Prediction-ML-Project

1. Introduction

Bitcoin was invented in 2008 by an anonymous person/group under the name Satoshi Nakamoto. It is a form of crypto-currency which revolutionized the way we think about money. It is a decentralized digital currency with no central bank, can be sent from user-to-user and verified by peers as it involves a public ledger. Bitcoins price started soaring from 2017 and have been high since then. By the emergence of BTC in the year 2009 with the initial value of around one dollar, no one predicted that in 8 years it would pass all previous records and would reach to the unbelievable value of $18000. BTC presents an impressive proof to this as it is a time series prediction problem in a market still in its transient stage. As a result, it is highly unpredictable in the market [4], and this provides an opportunity in terms of prediction for time-series prediction models. A time series is a series of data points indexed (or listed or graphed) in time order.

Time series forecasting is the use of a model to predict future values based on previously observed values. Models for time series data can have many forms and represent different stochastic processes. When modelling variations in the level of a process, three broad classes of practical importance are the autoregressive (AR) models, the integrated (I) models, and the moving average (MA) models. These three classes depend linearly on previous data points. Thus in this paper, we have focused mainly on the combinations like ARMA, ARIMA, SARIMAX (seasonal ARIMA exogenous) using RMSE as the evaluation metric for different models.

2. Literature Survey

Over the years, many algorithms have been developed for forecasting time series in stock markets. The most widely adopted are based on the analysis of past market movements[2].[3] presented a forecasting model based on chaotic mapping, firefly algorithm, and support vector regression (SVR) to predict stock market prices. Unlike other widely studied time series, still very few researches have focused on the bitcoin price prediction. One of them is the prediction of Bitcoin prices using Recurrent Neural Network (RNN)[4] and Artificial neural network(ANN) [5]. The study resulted in the best average accuracy gained by 98.76% in the training data and 97.46% in test data. Naimy & Hayek (2018) tried to forecast the volatility of the Bitcoin price using GARCH (Generalized Autoregressive Conditional Heteroscedasticity) models [6]. Some studies using the ARIMA method for predictions have also been performed, such as Stock Price Prediction of PT. BRI, TBK Using ARIMA Method" [7].[8] presented a forecasting Bitcoin exchange rate model in a high volatility environment, using autoregressive integrated moving average (ARIMA) algorithms. The preprocessing and collection of data are explained significantly in [9]. ARIMA is a predictive method that entirely ignores independent variables in forecasting, making it suitable for corresponding statistical data (dependent) and has some assumptions that need to be fulfilled like autocorrelation, trend, or seasonality [10].

3. Dataset and Pre-processing

3.1. Dataset collection and cleaning.

The dataset consists of day-wise bitcoin statistics from 29 Apr '13 to 1 Oct '20. The unit used for the price is USD. The data was taken from coinmarketcap [11]. The data was pretty consistent without null and missing fields, so there was no need for cleaning the dataset.

3.2. Feature Extraction

The dataset initially consisted of various attributes that can be listed as 'time_open', 'time_close', 'time_high', 'time_low', 'USDopen', 'USD.high', 'USD.low', 'USD.close', 'USD.volume', 'USD.market_cap' and 'USD.timestamp'. The attributes retained after Feature Extraction is Date- the time stamp of the observation, USD.low- the all-day lowest bitcoin price in USD.high - the all-day highest bitcoin price, USD.open -the opening bitcoin price in USD for the USD.close - the closing bitcoin price in USD for the USD.volume- the total volume of bitcoins traded in USD for the day. After Feature Crafting, we had a new attribute as Mean = (USD.low+USD.high) /2. This gives a more realistic bitcoin price value for the day. We have used this Mean as the forecast attribute in all the models.

3.3. Stationarity

A time series is said to be stationary if it does not show seasonal or trend effects on an aggregate level, i.e. no change in mean or variance throughout the data. This can be observed by analyzing the seasonal decomposition of the time series attribute to be forecast (i.e. Mean in our case). So to check if data is stationary or not, we used the Augmented Dickey-Fuller (ADF) test.

It is the most popular statistical method to find if the series is stationary or not. It is also called as Unit Root Test.

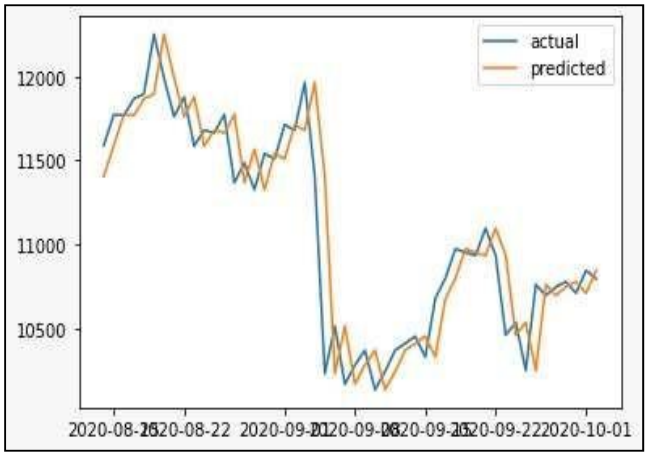

Figure 1: SARIMAX model on non- differenced data shows 1 day lag

$$y_t = c + \beta t + \alpha y_{t-1} + \phi_1 \Delta Y_{t-1} + \phi_2 \Delta Y_{t-2} \ldots + \phi_p \Delta Y_{t-p} + e_t$$

where y(t-1) = lag 1 of time series, delta Y(t-1) = first difference of the series at the time (t-1) and et is an error while alpha and c are constant and beta is coefficient of the trend.

The dataset initially was not stationary. A differencing technique was applied to make it stationary for ARIMA models to utilize their full potential. Differencing involves taking the difference of an element with another element of the same series, usually and also in our case, just the last element [8].

Time series data tend to be correlated in time and exhibit a significant autocorrelation which means that the value at time "*t* + k" is quite likely close to the index at the time "*t*", where k is an integer greater than equal to 1. As illustrated in figure 1, the SARIMAX model predicts the value at time "*t* + 1" merely by using the value at the time "*t*" as its prediction, which we see as a lag of 1 day between predicted and actual values. It is known as the persistence model. To avoid these kinds of scenario, one uses differencing on the data as a preprocessing step.

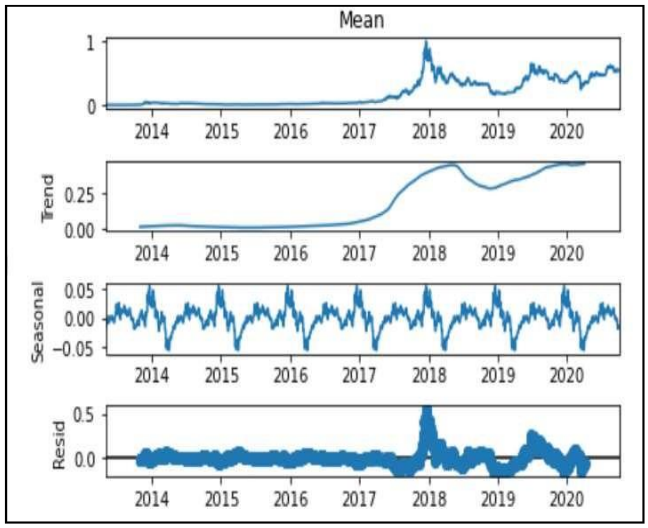

Figure 2: Seasonal decomposition of dataset with observational on top, then trend, seasonal and residual data

3.4. Seasonal Decomposition

Seasonal Decomposition separates a time series' variations into three main components

Trend: This depicts the increase or decrease in the series giving a general idea or trend about the aggregated variations.

Seasonality: This identifies the repeating short-term variation cycle in the series.

Residual: This depicts the random variation in the series sometimes also known as the noise.

Figure 2. Shows the data break down of observed data to three components. The trend shows that it's an increasing function as average values is increasing throughout the year. Seasonal data shows the data is being repeated every 365 days, thus it shows yearly seasonality. This will be used in SARIMAX as an argument with seasonal = True and m=12 .Residuals shows basically noise which adds up with seasonality and trend to give observed values.

3.5. Normalization:

We have performed min-max scaling for Normalization which scales the attributes to the range (0,1). This is done by dividing the difference between an observation and the attribute's minimum value by the range of that attribute, i.e. X_normalized = (X-Xmin) / (Xmax-Xmin)

4. Methodology

4.1. Linear Regression:

Linear Regression fits a linear model with coefficients w = (w1…wp) to minimize the residual sum of squares between the observed targets in the dataset, and the targets predicted by the linear approximation. We apply Linear regression between the mean on a given day and open value of Bitcoin. Further to establish a comparison with Time-series algorithms, we take input feature as the number of days, to predict the mean value. Mathematically, we are manipulating the Linear Regression output variable to depend upon itself.

$$y = b_0 + \vec{b}_1 \cdot \vec{x}_1 + \ldots + \vec{b}_n \cdot \vec{x}_n = b_0 + \sum_{i=1}^{n} \vec{b}_i \cdot \vec{x}_i$$

4.2. Linear Regression- Elastic Net

Linear regression with combined L1 and L2 priors as regularizes. We applied Grid search on the parameter l1_ratio, which adjusts the L1 penalty. For l1_ratio=0, Elastic net is as good as Least squares with Ridge Regularization and for l1_ratio=1, it turns into Lasso Regularization. Again, we predict the mean of Bitcoin value on the given day depending upon the number of days/open value of Bitcoin.

The optimization objective of Lasso and Ridge is:

$$\min_w \frac{1}{2n_{samples}} \|X_w - Y\|_2^2 + \alpha\rho\|W\|_1 + \frac{\alpha(1-\rho)}{2} \|W\|_2^2$$

A combination of both is used in Elastic Net, as the regularization factor.

4.3. Polynomial Regression:

We create polynomial features and apply Linear Regression on the same. Grid search is applied to the parameter degree to predict the mean value of Bitcoin.

4.4. Bayesian Regression:

Bayesian regression techniques can be used to include regularization parameters in the estimation procedure: the regularization parameter is not set in a hard sense but tuned to the data at hand. Mathematically, to obtain a fully probabilistic model the response y is assumed to be Gaussian distributed around Xw as follows:

$$p(y|X, w, \alpha) = \mathcal{N}(y|Xw, \alpha)$$

Where alpha is a hyper-parameter for the Gamma distribution prior.

4.5. ARIMA/SARIMAX:

The Autoregressive Integrated Moving Average (ARIMA) [12] is a method that fully ignores independent variables in forecasting [13]. The ARIMA method is capable of predicting historical data with the influence of data that is difficult to understand technically and has a high degree of accuracy in short-term forecasting and is capable of dealing with seasonal data fluctuations.

The ARIMA method is divided into 4 groups, namely Autoregressive (AR), Moving Average (MA), Autoregressive Moving Average (ARMA) and Autoregressive Integrated Moving Average (ARIMA).

The AR Model has the assumption that data at present is affected by previous period data. AR stands for autoregressive, which denotes that this model is trained against the previous values of the variable itself.

The AR method is used to determine the order value of the coefficient p that indicates the dependency of a value with the previous closest value.

The general form of AR Model with order p (AR (p)) or ARIMA model (p, 0, 0) is stated as follows:

$$X_t = \mu + \emptyset_1 X_{t-1} + \emptyset_2 X_{t-2} + \cdots + \emptyset_p X_{t-p} + e_t$$

The MA method is used to determine the value of the order coefficient q which explains the variable movement of the previous residual value.

The general form of MA model with order q (MA (q)) or ARIMA model (0, 0, q) is stated as follows:

$$X_t = \mu + e_t - \emptyset_1 e_{t-1} - \emptyset_2 e_{t-2} + \cdots + \emptyset_q e_{t-q}$$

The ARMA model is a composite of the AR and MA models.

Common forms of the model of the AR and MA or ARIMA processes (p, 0, q) are stated as follows:

$$X_t = \mu + \emptyset_1 X_{t-1} + \emptyset_2 X_{t-2} + \cdots + \emptyset_p X_{t-p} + e_t - \emptyset_1 e_{t-1} - \emptyset_2 e_{t-2} - \cdots - \emptyset_q e_{t-q} \quad (5)$$

The ARIMA model assumes that the data used must be

stationary which means the average variation of the data in consideration is constant. ARIMA method is represented by three parameters, the first of the data AR process of the past period is taken and maintained later in the Integrated process (I) makes the data become to facilitate the predictive process. The common form of the ARIMA model (p, d, q) is stated as follows:

$$X_t = \mu + X_{t-1} + X_{t-d} + \cdots + \emptyset_1 X_{t-1} + \emptyset_2 X_{t-2} + \cdots + \emptyset_p X_{t-p} + e_t - \emptyset_1 e_{t-1} - \emptyset_2 e_{t-2} - \cdots - \emptyset_q e_{t-q} \quad (6)$$

Exogenous variables that directly or indirectly affect the time series variable under observation often contribute to the strengthening of a model. A SARIMAX model uses the seasonality and exogenous variables along with the existing properties of an ARIMA model to make even more accurate predictions [14].

4.6. ARCH/GARCH:

Autoregressive conditional heteroskedastic (ARCH) models assume the variance of the current error term to function the actual sizes of the previous periods' error terms. Often, the variance is related to the squares of the previous innovations. ARCH models are generally employed in modelling financial time series that exhibit time-varying volatility clustering. If an ARMA model is assumed for the error variance, the model is called a generalized ARCH or GARCH model [15].

The GARCH (1, 1) model is defined as:

$$\epsilon_t^2 = \omega + (\alpha + \beta)\epsilon_{t-1}^2 + v_t - \beta\, v_{t-1}$$

Where et is the error, alpha and beta are constants while vt are values of original data.

GARCH can also be represented as:

$$Y_t = X_t^{'}\theta + \epsilon_t$$
$$Y_t = \omega + \alpha\, \epsilon_{t-1}^2 + \beta\, \sigma_{t-1}^2$$

Where sigma is conditional variance. We used GARCH (1, 1) model on the residuals from SARIMAX models to calculate the error specified above.

5. Results and Analysis:

A 90:10 train-test split was created. Thus train data contained 2440 rows of data while testing was for last 272 days

5.1. Regression and its Derivatives:

For all the regression algorithms like linear regression, lasso and ridge regression and Bayesian regression, we evaluated in two different ways:

1) Predicting mean value from the open value of the same day, i.e. these predictions are valid only for that day.

2) Predicting mean value depending on the number of days, mathematically prediction variable depending upon itself.

We used RMSE as the evaluation metric, which is the sum of squares of the differences between predicted values and actual values. We use RMSE and not any other metric because we wanted to penalize far values more.

TABLE I: Results for Regression

Model	Attribute	RMSE value
Linear Regression	Mean value using open value	362.83365
	Mean value using number of days	2190.2021
Elastic-Net regression	Mean value using open value	203.8337
	Mean value using number of days	2090.2043
Polynomial Regression (degree 2)	Mean value using open value	324.1615
	Mean value using number of days	1293.7657
Bayesian Regression	Mean value using open value	2191.2366
	Mean value using number of days	682.8338

For Elastic-Net Regression, we aim at finding the ratio of how much we should apply L1 penalization and L2 penalization, we apply grid search on the parameter alpha, and get alpha =0.8 on which results are given in table I. Applying Linear Regression after creating polynomial features and applying Grid Search resulted in the following RMSE

Degree= 2 RMSE: 1293.765 Degree= 3 RMSE: 1721.389 Degree= 4 RMSE: 2212.316 Degree= 5 RMSE: 1778.283

Thus Degree 2 was chosen and the evaluations are present in Table I. For Bayesian Regression, The estimation of the model is done by iteratively maximizing the marginal log-likelihood of the observations. The values obtained are present in Table I.

Under Linear models, we find a better solution when we predict Mean value w.r.t. to the open value on a given day. This is due to the direct relationship between the two. To establish a relationship between Linear Models and Time Series, we manipulated Linear Models to predict the mean value of a given day depending on the previous values of mean.

These models tried above does not give promising results, because it's able to capture the trend but lacks to capture seasonality and residuals. Thus, it was need of the hour to use the time series models, which are specially made to handle the forecasting scenarios.

5.2. ARIMA and its derivatives:

TABLE II: Results for ARIMA

Attribute	Model	RMSE value
Mean value using Open value for the previous day	AR	3489.97707212
	ARMA	181.94589519
	SARIMAX	196.22975044
	SARIMAX+ GARCH	188.35273047
Mean value using Open, Close, Low, High, Volume from the previous day	AR	3482.34921955
	ARMA	179.17065121
	ARIMA	9234.29311001
	SARIMAX	156.81198263
	SARIMAX+ GARCH	154.327596

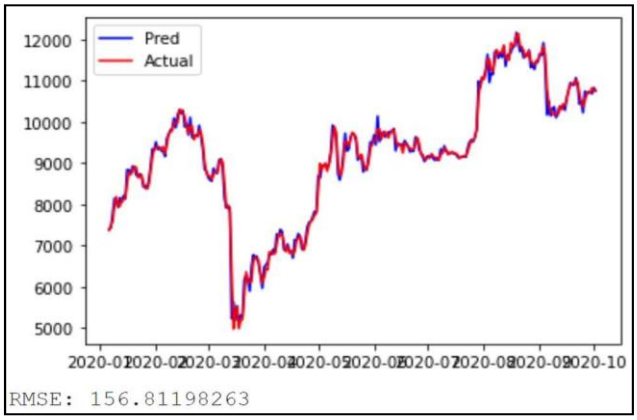

Figure 3: SARIMAX model with RMSE 156.811

We used the min-max normalized dataset with default differencing, i.e. the order of differencing equal to 1 applied on the variable to be forecasted. We used AR, MA, ARIMA, ARMA, SARIMAX with Open, close, high and low as exogenous variables and seasonal, yearly data(m=12) as shown in Table II. The graph in figure 3 shows the plot of predicted values vs actual for SARIMAX model.

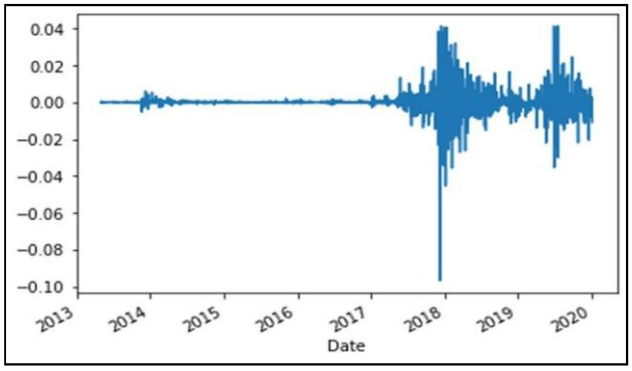

Figure 4: Residuals from SARIMAX models

However, SARIMAX models take care only for the seasonal data and trend; thus, the residuals as shown in figure 4, are ignored while predicted. So, we try to utilize the residuals and calculate volatility and variance of the model through GARCH with (p, q) = (1, 1) on the residuals from the SARIMAX models and the added the error to the predicted values. It led to a better model with reducing RMSE to 154.32, which is better than SARIMAX alone. Figure 5 shows the graph of the best results.

Figure 5: SARIMAX+GARCH best model with RMSE 154.32

We also used VAR to predict bitcoin prices, but it did not work out, and we received the RMSE value of 3476.53, which is way larger to even include in this paper. It did not work because it requires more than one endogenous variable, but it was only one, namely the 'mean' in our case. Moreover, it does not provide satisfactory results with highly seasonal data, and our data was yearly seasonal, as shown in the seasonal decomposition. Thus, VAR and VECM analysis could not perform well.

6. Conclusion:

In conclusion, there has been much learning achieved by this project, not only related to course work but in a different direction in the field of machine learning. This project expanded the knowledge on topics like linear regression and regularization and made us learn the difference between routine regression problems and the prediction/ forecasting problems. It made us encounter algorithms specially made for forecasting, i.e. time series algorithms like ARIMA, VAR and GARCH. It also taught us how to analyze time-series data with seasonal decomposition, further breaking down to a better understanding of the SARIMAX model, which uses this functionality to predict the best values. These algorithms work on the concept of assigning weights on new and old data, which is different from the weights assigned by regressive algorithms on features.

As Time Series algorithms provide better results than simple regression models, we have used the best model

ARIMA made until forecasting in machine learning. It motivates us to expand further in the direction and apply DL techniques like CNN and LSTM to obtain accurate predictions for a more considerable time in future.

Contribution of each member:

Dushyant-Extracting Dataset, Preprocessing Data, Time series algorithm such as AR and MA, Parameter Tuning, GitHub Repo maintenance.

Ishan- Preprocessing Data, Time series algorithm such as ARIMA, SARIMAX, Literature survey, Analyzing results, ARIMA+GARCH and evaluation of residuals.

Sajag- Feature extraction, linear models and relation, comparison with Time series algorithms, analyzing results, VAR, VECM analysis.

References

[1] Rob J Hyndman, "Forecasting: Principles and Practice," no. September, 2014.

[2] Agrawal J, Chourasia V, Mittra A. 2013. State-of-the-art in stock prediction techniques. International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering 2(4):1360–1366.

[3] Kazem A, Sharifi E, Hussain FK, Morteza S, Hussain OK. 2013. Support vector regression with chaos-based firefly algorithm for stock market price forecasting. Applied soft computing 13(2):947–958.

[4] R. A. Juanda, Jondri, and A. A. Rohmawati, "Bitcoin Price Prediction by Using Recurrent Neural Network," e-Proceeding Eng., vol. 5, no. 2, pp. 3682–3690, 2018.

[5] R. Albariqi, "Bitcoin Price Change Prediction Using Artificial Neural Network," 2018.

[6] Naimy VY, Hayek MR. 2018. Modelling and predicting the Bitcoin volatility using GARCH models. International Journal of Mathematical Modelling and Numerical Optimisation 8:197–215 DOI 10.1504/IJMMNO.2018.088994.

[7] G. S. Lilipaly, D. Hatidja, and J. S. Kekenusa, "Stake Price Prediction Using ARIMA Method at PT. BRI, TBK.," J. Ilm. Sains, vol. 14, pp. 61–66, 2014.

[8] Bakar N, Rosbi S. 2017. Autoregressive Integrated Moving Average (ARIMA) model for forecasting cryptocurrency exchange rate in high volatility environment: a new insight of Bitcoin transaction. International Journal of Advanced Engineering Research and Science 4(11):130–137 DOI 10.22161/ijaers.4.11.20.

[9] Bitcoin Price Prediction using Machine Learning part 1 (only part 1 is published till yet which proposes of techniques to make clean data using data mining) ,Siddhi Velankar*, Sakshi Valecha*, Shreya Maji* *Department of Electronics & Telecommunication, Pune Institute of Computer Technology, Pune, Maharashtra,India409-415.

[10] A. Qonita, A. G. Pertiwi, and T. Widiyaningtyas, "Prediction of Rupiah Against Us Dollar by Using ARIMA," Int. Conf. Electr. Eng. Comput. Sci. Informatics, vol. 4, no. September, pp. 746–750, 2017.

[11] Coinmarketcap API - https://web-api.coinmarketcap.com/v1/cryptocurrency/ohlcv/historical?convert=USD&slug=bitcoin&time_end=1601510400&time_start=1367107200

[12] A. Hendranata, "ARIMA (Autoregressive Integrated Moving Average)," 2003.

[13] Y. S. Lee and L. I. Tong, "Forecasting time series using a methodology based on autoregressive integrated moving average and genetic programming," Knowledge-Based Syst., vol. 24, no. 1, pp. 66–72, Feb,2011.

[14] Hillmer, S. Craig, and George C. Tiao, "An ARIMA-Model-Based Approach to Seasonal Adjustment," vol. 10, no. 1, pp. 5–24, 2017.

[15] Bollerslev, T., 1986. Generalized autoregressive conditional heteroskedasticity. *Journal of Econometrics*, 31(3), pp.307-327.