Gaussian Mixture Model-Hidden Markov Model(GMM-HMM) vs Linear Regression(LR) for Cryptocurrencies Forecasting with new features extraction

Master's degree in Computer Engineering for Robotics and Smart Industry

COURSE: Machine Learning and Artificial Intelligence

ACADEMIC YEAR: 2020-2021

AUTHOR'S: Edoardo Fiorini and Antonino Parisi

September 15th 2021

Summary

1 Motivation and Rationale **3**

2 State of the Art **3**

3 Objectives **4**

4 Methodology **5**

4.1 Dataset 5

4.2 Features 6

4.3 Methods 6

4.3.1 Gaussian Mixture Model-Hidden Markov Model(GMM-HMM) 6

4.3.2 Linear Regression 7

4.4 Algorithms 7

4.5 Performance metrics 8

4.6 Computational tools 9

5 Experiments and Results **9**

6 Conclusions **14**

1 Motivation and Rationale

For the past 50 years a very hot topic has been trading with different stocks and raw material, but in the last decade the community has moved their interest to a different kind of coin called cryptocurrencies which have become a mutli-billion dollar market[1]. However, working on these new coins is not as easy as before because their behavior is affected by different events around the world, for instance the typical scenario is when a famous people buys one of these coins. High volatility, speculative forces and large dependence on social sentiment: these are the main features of crypto. In fact, due to their extreme variability traditional economic theories and standard financial models fail to capture their statistic attributes. For this reason, evaluating a cripto trend requires to add to a normal stochastic component a random one. Despite this well-known problem, people are implementing this new form of economy because of its cheapness, online, and anonymous means of exchange.
Trying to fill this random gap is very interesting. In fact, using artificial intelligence and analysing big data it is possible to predict it a priori or almost try to do it. Moreover, agent based artificial financial market for finding attractive technical patterns have also been proposed [2, 3]. This challenge is a multidisciplinary topic that requires different fields merged together. The economic knowledge can give a mean to each part of the model, for instance why there are some state or why there are some specific parameters, in this way we can avoid to deal with the problem in a completely empirical way. On the other hand, the computer science can provide the algorithm support to build the model.
Explore this topic has not only an educational aim(see how different department knowledge could be merged), but also a practical side. In fact, in this way people can easily trade in a difficult and challenging financial instrument like cryptocurrency.

2 State of the Art

As has been said in a previous section cryptocurrencies prediction is a very hot topic, for this reason the state of the art is full of works which try to give a different interpretation of the problem. In particular, they could be divided depending on how the huge data are used, which kind of model is exploited, whence the parameters are extracted and what the validation metric is. The main works are shown in following.
Kokia, Leonardos and Piliouras in 2020 have developed a very interesting work [4] in which have applied Hidden Markov Model(HMM) for predicting and explaining several cryptocurrencies as Bitcoin and Ripple. The models used are different both for the topology and number of state. They have underlined that each cryptocurrency has a different random behavior and for this reason the model to forecast is different for each coin. In particular, they have highlighted a sort of dynamic characterized by regime switches and frequent alternations. All these features are in line with the random component described in previous chapter. Another important work, always developed in 2020, is [5], in which has been used linear regression and neural networks models for forecasting Bitcoin closing price. Also in this case has been pointed out the existence of time regimes which could depend on the partitioning of dataset into shorter sequences. In

fact, in this research, an important role is played both from how the data are merged and from which kind of features are used. In base of these, different models have been exploited to achieve some results, for instance Multiple Linear Regression(MLR), Multilayer Perceptron(MLP) and Long short-term memory (LSTM). The results have been compared with benchmark and the best are obtained using more instance for each feature. On the state of the art is possible to find also works out of the box as [6]. This project has been developed by a classical method of linear regression, but it has been applied also Support Vector Machine(SVM). This choice is pretty strange because SVM is a typical linear classification method. However, the results obtained are very good with an accuracy of 96.06%, which could be increased up to 99 % by adding features to the SVM method. This work is the proof of what it has been discussed before: the problem could have different interpretation. Also in [7] has been used a different classical approach as k-Nearest Neighbor (K-NN) and Ensemble method. In particular, this last method has been developed with multiple relative regression learner merged together, in this way it is possible to minimize the error and thus the desired output can be obtained through a voting process. It is considered as the best among all the models proposed with an accuracy of 92.4%. Since it is the age of deep learning this ensemble method has been revisited with this technique [8]. The analysis reported by the paper indicates that ensemble learning and deep learning can be efficiently beneficial to each other, for developing strong, stable, and reliable forecasting models. In fact, in this case the prediction is for each hour of the following day. It is important also to understand how the data that train the model are driven. For instance, in this work [9] to build the model is not directly used the data from each coin, but the prediction is given studying the dynamic under tweet volume and sentiment. It is considered a predictor of price direction with a great result. This quick overview of the state of the art point out that to adress this topic several methods could be applied. In fact, the community move from classical approach to more deep one, but altogether all these method works pretty good. However, after having analysed the literature it can be said that it is not a matter of method, but how the data are processed. Our work is focused on how to improve the features extraction rather than using directly the raw data as opening and closing price.

3 Objectives

The goal of this project is to apply machine learning techniques to build a model that is able to predict the behavior of a cryptocurrency. In particular, the purpose of this work is to forecast the next day's closing price of a cryptocurrency. Consulting other works we have understood that in this type of market it is already a good result have a model that describes well the trend in terms of rise or fall of the price, but we try to guess the price precisely. Our work could help to reduce the challenges and difficulties faced by people who invest in these coins. We have decided to deal with Ethereum(ETH), which is an historical cripto and it is not as affected as Bitcoin by events. In this way, the model has to describe almost only a stochastic component. Another important consideration we made is that on the state of the art are used several machine learning techniques and algorithms to achieve the prediction, but we have decided to use a classical machine learning method. In fact, to achieve our goal, we do not focus on develop a particular approach about how to build a model, but we want to give more importance to the data. All the methods work pretty well, but in our opinion what could give a better result is how the time series data are used. We avoid to use the raw data because they could mistake the model. The key point to solve this project is to find out different features from the classical opening, closing, high and low price. In our opinion, if this data processing is done in a smart way is not necessary a particular model to predict with a good accuracy because the data can better communicate each other. Of course, we want to compare the forecasting of two different classical methods: HMM and Linear Regression. Moreover, our results are compared with various benchmarks to make sure we are in line with the works already in literature.

4 Methodology

4.1 Dataset

The dataset is composed by week daily data from August 8th 2015 to July 6th 2021, which are available on any trading site, for instance https://coinmarketcap.com. The data includes seven attributes:

1. `Date`: day at which an order is executed in the market to purchase, sell or otherwise acquire a currency is performed
2. `Open price`: price at which a currency is first traded on a given trading day.
3. `Close price`: the final price at which a currency is traded on a given trading day.
4. `High`: the highest price at which a currency is traded on a given trading day
5. `Low`: the lowest price at which a currency is traded on a given trading day.
6. `Volume`: the total quantity of contracts traded for a specified currency.
7. `Market Capital`: the total dollar market value of a currency's outstanding contracts

The dataset has been divided into two subset: training and testing. Since the goal is to predict the next day's closing price, we have decided to create a testing dataset composed by the last 60 days of the initial data. In this way, we evaluate our model with several prediction rather than using only one. This choice allows us to have a better precision in the score valuation. On the other hand, the training dataset includes the data from May 5th 2021 to August 8th 2015. Of course, work on cryptocurrencies with such a large training set could distract the model because of the random component of this coins. For this reason, in this project we make a comparison between the whole training set,last year and only last 6 months.

4.2 Features

This work wants to find out some new feature to reach a best accuracy. First of all, we do not consider all the seven attributes because information that are not so impressive could not add information and confuse the training of the model. Analyzing the opening, closing, high and low price we have understood that using raw data is not the best way to build a model. In fact, this data although they concern the same currency or are on the same day, they are rather unrelated to each other. Therefore, creating a model trained on these four features would lead to a poor result because it is based on data that cannot collaborate with each other. For this reason, we have decided to introduce three new features that can be seen as a sort of normalization of the previous ones. In this way the model can learn from the data all the information necessary to make a good prediction. This extraction has been done only to train better the model because in this way the data are more impressive. Once the model has predicted in terms of this features, it is easy to retrieve the next day's closing price that is the main goal of the project. The new features are shown in 1.

$$F_1 = \frac{CLOSE - OPEN}{OPEN}; F_2 = \frac{HIGH - OPEN}{OPEN}; F_3 \frac{OPEN - LOW}{OPEN} \tag{1}$$

4.3 Methods

This project is based on two classical methods: Hidden Markov Model and Linear Regression. As it has been underlined in previous sections, we do not focus on a particular approach because our goal is to see, even if we are not using a sophisticated approach, we can reach a good results with our new features. We can consider these two methods very clean, so if we overcame the already accuracy in literature we can give credit only to our features. In this way, machine learning techniques can be integrated into business intelligence systems for making real life decisions.

4.3.1 Gaussian Mixture Model-Hidden Markov Model(GMM-HMM)

Gaussian Mixture Model- Hidden Markov Model is a pretty common model that is useful for forecasting. The main difference between a classical HMM model and GMM-HMM is to change the point of view from discrete observations to continuous observations. In fact, HMM usually tries to divide the observations in different discrete range. In this way, if the variance of the predictions is very small then every score is given to the same observations label. Instead, using GMM-HMM we can consider also tiny margin. A vector observation sequence is created to train the model.In particular, each observation at time t has the shape 2, where the fractional represents the features shown in 1.

$$O_t = (fractional - Close, fractional - High, fractional - Low) \tag{2}$$

We used Viterbi algorithm to train the model with a fixed number of iterations, but different numbers of mixture and state. It is not necessary to impose some initial conditions because the training converge always in the same model and we cannot exploit the data to imposed some initial conditions. For prediction, we imposed different range as possible observations and we used the function 3 to compute most likely prediction, where in 3 λ represents the trained GMM-HMM model. This function tries to select the next day's observation prediction starting from some previous day real observations. The best results is choose using log probability under the trained model.

$$O_{d+1} = argmaxO_{d+1}[P(O1, O2, \ldots, Od, Od_{+1}|\lambda)] \tag{3}$$

4.3.2 Linear Regression

Linear Regression is a typical model used for prediction in several domains. It is a set of statistical processes for estimating the relationships between a dependent variable and one or more independent variables. The main goal to build the model is to find out a line that is able to fit the data in such a manner that the error is minimal. The line is computed by the function 4, where X is the independent variable, which are observed in data (features) and are often represented by a vector, β is the dependent variable which is observed in data and is often represented by a scalar and ϵ which represents the error.

$$\begin{aligned}&Given\{y_i, x_{i1}, ..., x_{ip}\}_i^n{=}1,\\&y_i = \beta_0 + \beta_1 x_{i1} + ... + \beta_p x_{ip} + \epsilon_i,\\&y = X\beta + \epsilon\end{aligned} \tag{4}$$

4.4 Algorithms

In this section we are going to show the methodology and approach to make a forecast with HMM, because with regression the prediction is made in analytical way. Our heuristic is simple and works well, and can be resumed in this way, we have built three sets with a large number of samples between a range, the three sets are the three features used for training, after making these three sets we make a observation set with all the possible combinations of the three features made before, and the last thing is create a N sets made with the previous combinations of features and each set has K records as the K days of forecast wanted. The last thing is concatenate the last D days to the observation set and choose the best observation set which gives the maximum likelihood in the score function of the model. We describe better in a procedural way in following section;

```
real_sequence = last 5 records (days) of features
observation_set = Array()
feature_1 = range(-0.1,0.1,#samples)
feature_2 = range(0,0.1,#samples)
feature_3 = range(0,0.1,#samples)


all_combinations = build_all_possible_combinations(feature_1,
feature_2,feature_3)

all_combinations.shuffle()

for k in k_observations:
    for d in range(days_of_forecast):
        index = random_int(0,length(all_combinations))
        observation.append(all_combinations[index])
    observation_set.append(observation)
    observation.clear()


max_score = -inf
best_sequence = []
for obs in observation_set:
    sequence = [real_sequence,obs]
    score = model.score(sequence)
    if(score > max_score):
        max_score = score
        best_sequence = obs
```

4.5 Performance metrics

The performance metrics we have used in this work is explained variance regression score. This is much more indicative than the RMSE because we are dealing with numbers that change by a few cents. In this way small gap are not weighted too much. Its formula is 5, where Var is the square of the standard deviation.

$$1-\frac{Var\{Real-Forecast\}}{Var\{Real\}} \tag{5}$$

Moreover, we have used another performance metrics called R^2 score. It provides an indication of goodness of fit and therefore a measure of how well unseen samples are likely to be predicted by the model, through the proportion of explained variance. This is more indicative than explained variance because it uses the data directly. Its formula is 6, where $y = \frac{1}{n}\sum_{i=1}^{n} y_i$ with y_i the corresponding true value for total samples.

$$\frac{\sum_{i=1}^{n}(Real_i - Forecast_i)^2}{\sum_{i=1}^{n}(Real_i - y)^2} \tag{6}$$

Of course, in these cases the value of performance could be very low because we have to compare prices that can be very far numerically but very close if we normalize the value such as we have done for the feature extraction; so we have decided to compute the accuracy and error with classical methods and with our performance indicator in base of our knowledge of the problem.

Our performance indicator is described by the formula 7 and in this case we have a better way to interpret the forecast value.

$$\nu = \frac{(Forecast - Real)}{Real} \tag{7}$$

4.6 Computational tools

All the project has been developed in python. We have used hmmlearn library to build the GMMHMM model. In fact, there is a class called GaussianHMM which allows to initialize, fit and predict the model. We have easily setted the parameters (number of state, tollerance and iteration) and create the model. For the linear regression model sklearn library has been used. Preprocessing, model fitting and metrics class have been imported to find the best prediction. Moreover, due to the kind of data some other library have been exploit to deal with csv file. Matplot library plots the results, and also numpy allows us to manage data and prediction.

5 Experiments and Results

This section explains all the traials we have done to achieve the best result. In particular, we have made a comparison of models with different parameters. Before showing the results, we want to underlined another key point of our work: how we extracted the next day's closing price starting from our new features. In fact, what we are trying to predict depends on the opening price that is known and is equal to the closing price of the previous day. For this reason, in the inverse formula to retrieve the next day's closing price we have used the closing data of the previous day. There could be some gap between the closing price of the previous day and the opening price of the day after due to some random components. Instead, we believe it is more correct to use the data of the day before because we assume that we have no information on the day we are predicting.

Regarding the GMM-HMM we have done 3 sets of tests.
First of all, we have compared models with different numbers of states and with a training dataset ranging from 6 months up to all available years. As table 1 shows, for each configuration the metrics have been computed and the best result is a model with 3 states and trained over 6 months.

states/months	6	12	full dataset
3	R2 : 0.9564 EV : 0.9589 Idx : 0.012	R2 : 0.9426 EV : 0.9571 Idx : 0.014	R2 : 0.9368 EV : 0.9427 Idx : 0.015
4	R2 : 0.9564 EV : 0.9589 Idx : 0.012	#	R2 : 0.9378 EV : 0.9393 Idx : 0.01
5	#	R2 : 0.9337 EV : 0.9417 Idx : 0.022	R2 : 0.9296 EV : 0.9365 Idx : 0.018

Table 1: Comparison between states of GMM and Dataset length, the number of mixtures are fixed to 1. EV is explained variance, the best value is 1, R2 is sklearn index the best value is 1, Idx is our index the best value is 0.

Then we have made a comparison between models with different covariance matrices and always with different training dataset. Table 2 shows the results, in which we have underlined that for the time series data that we have the covariance matrices is not decisive.

covariance/months	6	12	full dataset
full	R2 : 0.9426 EV : 0.9471 Idx : 0.014	R2 : 0.9426 EV : 0.9471 Idx : 0.014	R2 : 0.9415 EV : 0.9415 Idx : -0.001
diag	R2 : 0.9426 EV : 0.9471 Idx : 0.014	R2 : 0.9426 EV : 0.9471 Idx : 0.014	R2 : 0.9368 EV : 0.9427 Idx : 0.015
spherical	R2 : 0.9426 EV : 0.9471 Idx : 0.014	R2 : 0.9426 EV : 0.9471 Idx : 0.014	R2 : 0.9415 EV : 0.9415 Idx : -0.001

Table 2: Comparison between covariance of GMM and Dataset length, the number of states are fixed to 3. EV is explained variance, the best value is 1, R2 is sklearn index the best value is 1, Idx is our index the best value is 0.

The last set of trials regarding GMM-HMM tries to find out the relation between different numbers of mixture and different training dataset. Results are shown by table 3. We have highlighted with the character # that with a mixture number greater than 2 the model is unable to train. The best result is with 2 mixtures and the full dataset.

mixtures/months	6	12	full dataset
2	R2 : 0.9310 EV : 0.9389 Idx : 0.024	R2 : 0.9401 EV : 0.9404 Idx : -0.003	R2 : 0.9415 EV : 0.9415 Idx : -0.001
3	#	#	#
4	#	#	#

Table 3: Comparison between mixtures of GMM and Dataset length, the number of states are fixed to 3 and covariance type is set to spherical. EV is explained variance, the best value is 1, R2 is sklearn index the best value is 1, Idx is our index the best value is 0.

Summarizing the experiments done, the best model is the one that has one mixture and a training set of the last 6 months. In our opinion, this is due to the random component that affects the cryptocurrencies. In figure 2 is shown the plot of the best model which is really close to the real value, the trend is really confident, but few times the predicted trend is inverted. To remark the result we have in this case the R^2 is 0.9426 and EV is 0.9471. Instead in figure 1 is shown the behavior of the model trained by the whole dataset.

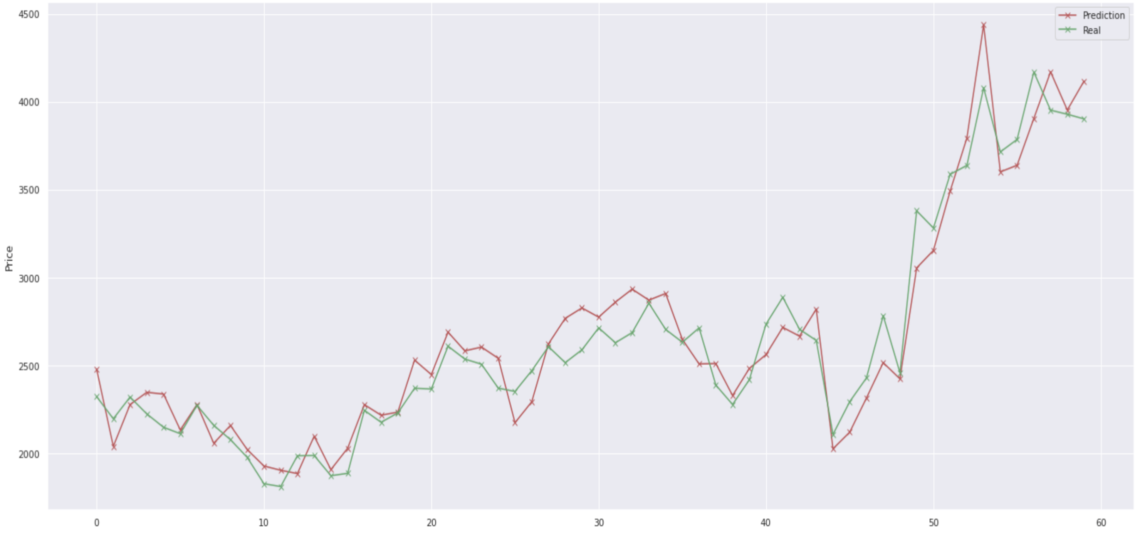

Figure 1: Forecast for next 60 days with the whole dataset as training set (without the 60 days used for forecasting obv.)

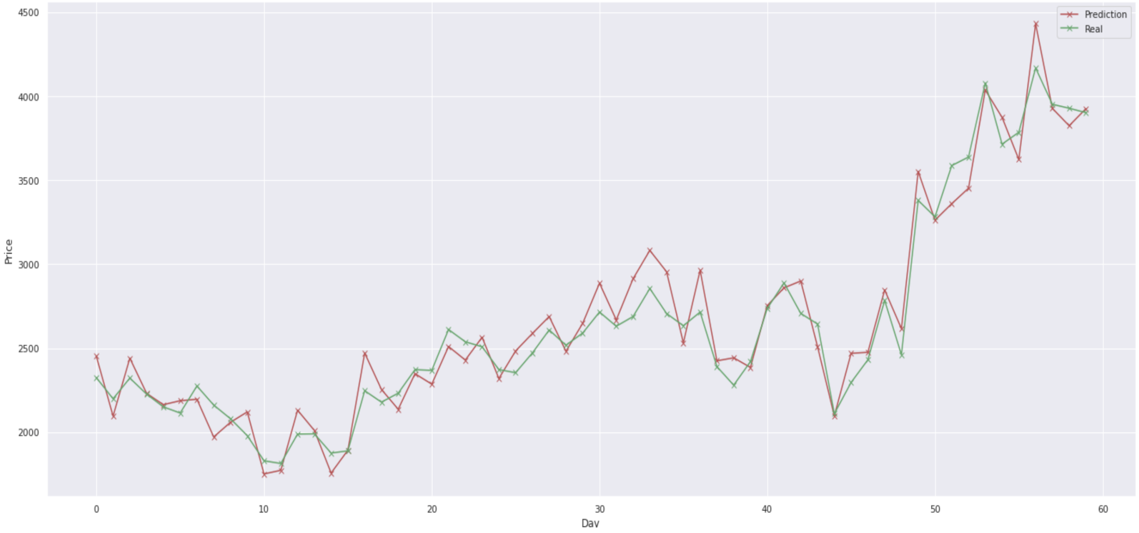

Figure 2: Forecast for next 60 days with the last 6 months of dataset records as training set (without the 60 days used for forecasting obv.)

In case of the linear regression the results are quite different. The result of regression with least square is worse, if we look at figure 3 , the predicted values and the real values are not really close as the GMMHMM model, but if we look the error during the training is pretty perfect4.

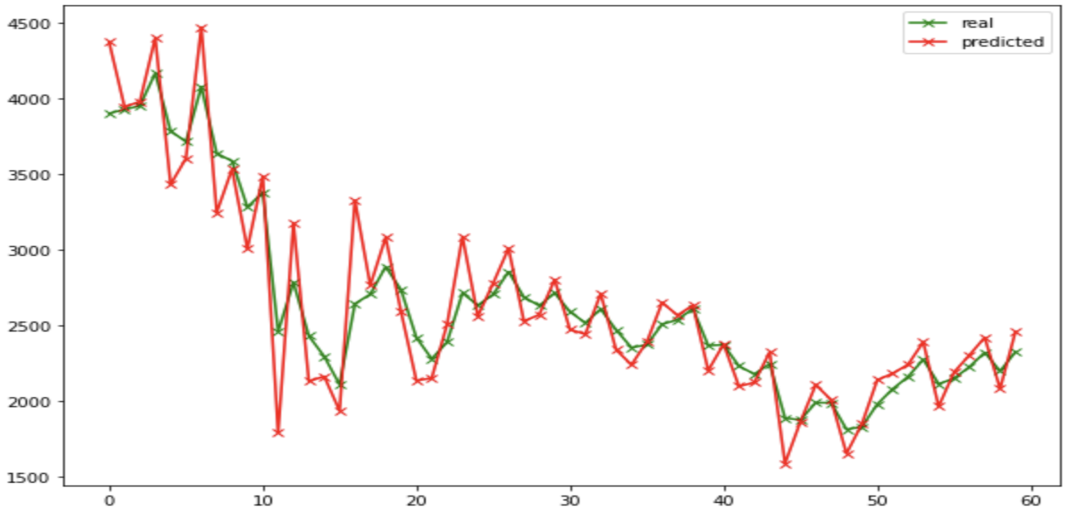

Figure 3: Least square forecast for the next 60 days after the training on the whole dataset.

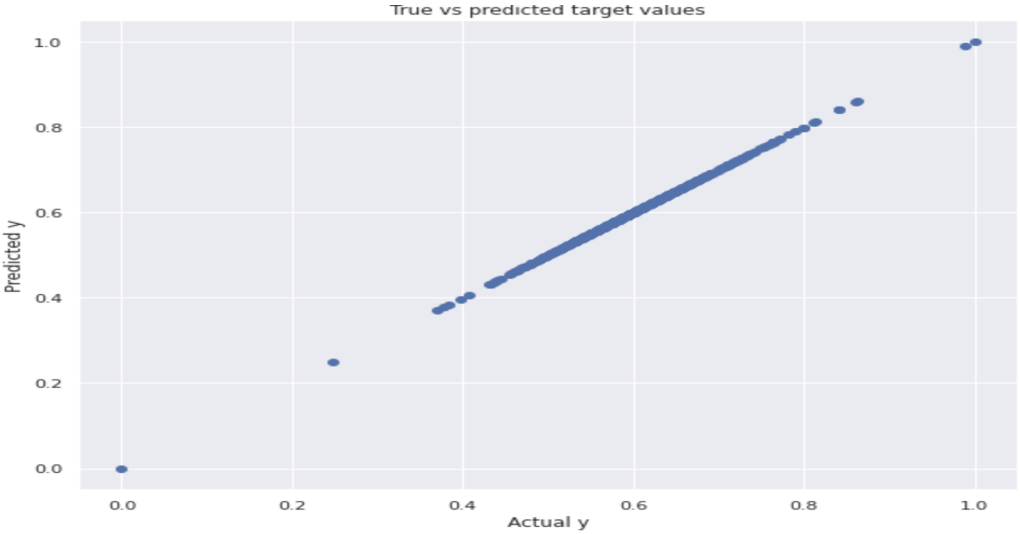

Figure 4: Regression training results with LSE.

The results are a little bit better if we take into account a regression made with Gradient descend method. In fact the predicted values are quite similar and it follows the trend of the value, as it is shown in figure 5.

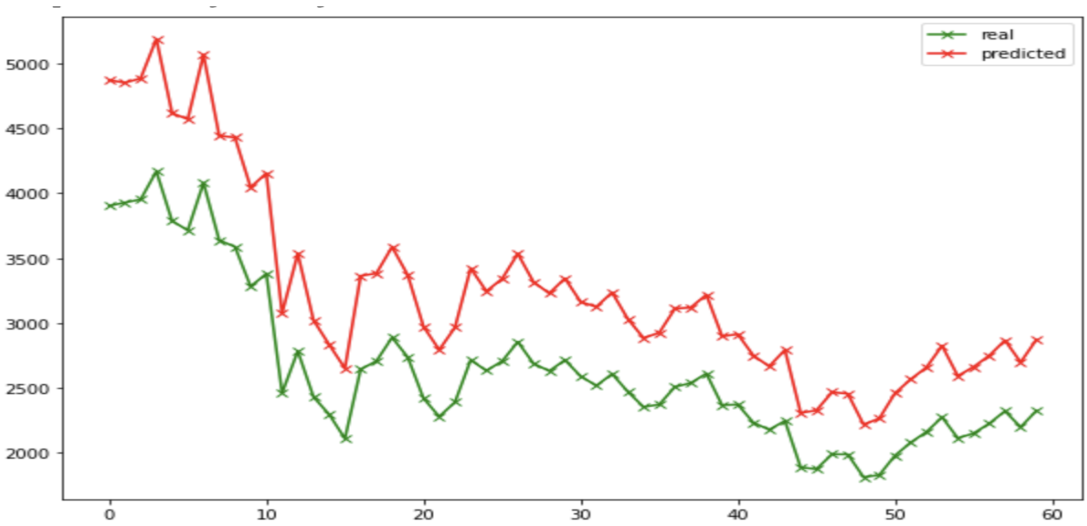

Figure 5: Gradient descend forecast for the next 60 days after the training on the whole dataset.

If we consider a reduced version of the dataset we can find similar performance of training for LSE approach, but for the Gradient Descend method the result of tests are always worst than LSE. In fact, the performance computed with R^2 score is much more indicative.

These observation are also confirmed by the table of experiments (see 4). The LSE method in this case has very lower performance than GMM-HMM

, we cannot assure the perfect prediction for the next day, but we want to highlight that this method is good if the model rules are respected and also sometimes can occur big drops and overshot because these cryptovalues are also influenced by external events that are very difficult to model.

method/months	6	12	full dataset
LSE	R2 : 0.8729 EV : 0.8729 Idx : -1.0214	R2 : 0.8729 EV : 0.8729 Idx : -1.0214	R2 : 0.8729 EV : 0.8729 Idx : -1.0214
Gradient descend	R2 : -0.10480 EV : 0.9411 Idx : -0.5027	R2 : -0.0158 EV : 0.9451 Idx : -0.5234	R2 : 0.1820 EV : 0.9531 Idx : -0.5735

Table 4: Comparison between two approaches to linear regression, LSE and Gradient descend.

6 Conclusions

This project wants to use classical methods with new features for cryptocurrency forecasting. One of the goals is to highlight that to achieve best results it is more important how the data is handled than to create an innovative model consisting of several simple models put together. The new features 1 have been extracted from the raw data trying to guarantee a greater flow of information during the training of the model. This sort of normalization is necessary because the data are very similar to each other, for example opening and closing price, and could mistake the model. We have made a comparison among Gaussian Mixture Model-Hidden Markov Model and Linear Regression. The results showed that GMM-HMM's prediction is much better than linear regression because R^2 score has more importance than the other metrics. To make sure we are in line with the state of the art works, we have compared our accuracy with the major benchmarks. All the models have been evaluated by explained variance regression score, R^2 score and our performance indicator. The firs two are for the next day's closing price, the other for the new features. In our results, we have shown that the trend is predicted pretty good.
At the end of this work we can conclude that it is possible to achieve excellent results also using classical models because the fundamental role is played by features which train the model. The next step could be to use these features in more complex models, for instance in the field of deep learning, to achieve even higher accuracy. Moreover, it would be interesting to apply these models to more difficult cryptocurrencies such as bitcoin.

References

[1] Hu, A.S., Parlour, C.A., Rajan, U., 2019. Cryptocurrencies: Stylized facts on a new investible instrument, Financial Management 48, 10491068.

[2] L. Cocco, G. Concas, M. Marchesi, Using an artificial financial market for studying a cryptocurrency market, J. Econ. Interact. Coord. 12 (2) (2015) 345–365.

[3] S. Ha, B. Moon, Finding attractive technical patterns in cryptocurrency markets, Memet. Comput. 10 (3)

[4] Koki, C., Leonardos, S., Piliouras, G. (2020). Exploring the Predictability of Cryptocurrencies via Bayesian Hidden Markov Models. arXiv: Applications.

[5] Uras N, Marchesi L, Marchesi M, Tonelli R. 2020. Forecasting Bitcoin closing price series using linear regression and neural networks models. PeerJ Computer Science 6:e279.

[6] Poongodi M., Ashutosh Sharma, Vijayakumar V., Vaibhav Bhardwaj, Abhinav Parkash Sharma, Razi Iqbal, Rajiv Kumar, Prediction of the price of Ethereum blockchain cryptocurrency in an industrial finance system, Computers Electrical Engineering, Volume 81, 2020, 106527, ISSN 0045-7906.

[7] Reaz Chowdhury, M. Arifur Rahman, M. Sohel Rahman, M.R.C. Mahdy, An approach to predict and forecast the price of constituents and index of cryptocurrency using machine learning, Physica A: Statistical Mechanics and its Applications, Volume 551, 2020, 124569, ISSN 0378-4371.

[8] Livieris, I.E.; Pintelas, E.; Stavroyiannis, S.; Pintelas, P. Ensemble Deep Learning Models for Forecasting Cryptocurrency Time-Series. Algorithms 2020, 13, 121.

[9] Abraham, Jethin; Higdon, Daniel; Nelson, John; and Ibarra, Juan (2018) ”Cryptocurrency Price Prediction Using Tweet Volumes and Sentiment Analysis,” SMU Data Science Review: Vol. 1 : No. 3 , Article 1.