Turkish Economic Review
econsciences.com
Volume 12 June 2025 Issue 2

Forecasting inflation in Iran by applying machine learning algorithms to PPP lag

By Tal BOGER †

Abstract. Purchasing Power Parity (PPP) relates the prices of two countries by their exchange rates. Several economists use PPP to measure inflation in the absence of official and accu- rate government reports. In the case of Iran, the government's official inflation figures are significantly lower than what one would expect given their economic troubles; therefore, we apply PPP to measure inflation in Iran. Because of its volatility in the short-run, PPP is often used as a long-run economic indicator. The main cause for this is that PPP is a leading indicator, creating short-term inaccuracies. However, using machine learning algorithms, we forecast both the time until there is zero PPP lag (i.e. the official and implied inflation rates are equal) and the difference between the official and implied inflation rate (allowing us to predict official inflation rates) for Iran with minimal volatility. This allows us to use PPP accurately over both the short- and long-run.
Keywords: Purchasing Power Parity (PPP); Iranian inflation; Machine learning; Support vector machine; Random forest; k -nearest neighbors; Neural network
JEL. D30; D63; E21.

1. Introduction

Purchasing Power Parity (PPP) has been a popular method for measuring inflation in countries where official reports of inflation data either stopped or are inaccurate given the country's economic environment. For example, PPP was used to measure Zimbabwe's hy- perinflation once Zimbabwe's government ended the reporting of official inflation data (Hanke & Kwok, 2009).

Because PPP uses exchange rate data - which is available daily - to predict inflation, it serves as a leading indicator for official inflation. Realizing the short-term volatility of exchange rates, many economists reject PPP as an acceptable short-run indicator. This volatility creates a rift between the PPP implied inflation rate and the official inflation rate. According to Rogoff (1996), the difference between the PPP implied inflation rate and the official inflation rate decrease only 15 percent per year (Kenneth, 1996).

This slow mean reversion rate undermines the use of PPP in predicting inflation, espe- cially over short periods of time (such as in the case of Iran). However, by using machine learning (ML) algorithms, we can predict this PPP lag time and the deviation of PPP from official inflation rates.

Recently, machine learning has become significantly more popular, especially in the field of economics. Most econometric analysis and empirical economic analysis involves specifiying a model, evaluating confidence intervals for estimated parameters, and applying that model (Athey, 2018). It is important to note that ML models do not serve the same purpose of parameter esti- mation; while ML models can return regression coefficients, the results are rarely consistent. However, using ML in economics offers

† Johns Hopkins University Institute for Applied Economics, Global Health, and the Study of Business Enterprise, Baltimore, USA..

several advantages to this traditional approach, in- cluding uncovering patterns, fitting complex data in flexible functions, and finding functions that perform accurately out-of-sample. These are some of the main purposes of creating supervised ML algorithms (Mullainathan & Spiess, 2017).

2. The PPP Equation

Before continuing our discussion of PPP, it is important to understand the calculation of PPP. For our calculations, let:

- P_I = the Iran price level in Iranian rial (IRR)
- P_{US} = the United States price level in U.S. dollars (USD)
- $E_{IRR/USD}$ = the exchange rate of IRR:USD (IRR per unit of USD)

PPP, in a static sense, states that:

$$\frac{P_I}{P_{US}} = E_{IRR/USD}$$

PPP in a dynamic sense - which looks at the changes in price levels - states that:

$$\frac{1+\frac{\Delta P_I}{P_I}}{1+\frac{\Delta P_{US}}{P_{US}}} = 1+\frac{\Delta E_{IRR/USD}}{E_{IRR/USD}}$$

In countries which we suspect have high inflation, such as Iran, ΔPUS can be assumed to be 0, given that it is insignificant compared to ΔPI. Therefore:

$$\frac{\Delta P_I}{P_I} = \frac{\Delta E_{IRR/USD}}{E_{IRR/USD}}$$

The final PPP equation relates Iran's inflation to the IRR's exchange rate to the USD. Given the volatility of these exchange rates, many economists ignore PPP as an accurate measure of inflation in the short-run; many accept that PPP deviations have a 3- to 5- year half-life (Rogoff, 1996).

3. PPP Deviations in Iran

Our PPP data for Iran (beginning on 1/19/2012) took over a year and a half to reach equilibrium with the official inflation rate (i.e. official inflation rate - PPP implied inflation rate = 0). Following this point of inflection, the PPP implied inflation rate stayed relatively close to the official inflation rate. Recently, however, the deviation has grown significantly (see Figure 1).

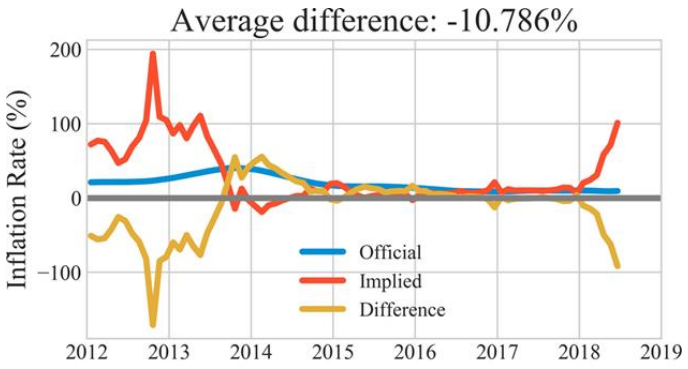


Figure 1. *Inflation rate*

On average over all our available data, the PPP implied inflation rate is 10.786% higher than the official inflation rate. This indicates the short-run inaccuracies of PPP implied inflation rates.

Before creating models to limit this variance, we must first examine the relationships between the factors of PPP inflation and the PPP deviation. We test how the following factors correlate to the difference between the official and implied inflation rate (DoI) and the months to equilibrium (MtE; months until the two rates are the same):

1. Black market premium (%; $\frac{\text{Black market exchange rate}}{\text{Official exchange rate}} - 1$)
2. Official exchange rate

3.1. Difference between Official and Implied Inflation Rate (DoI)

Figure 2 shows the correlations between the black market premium and official exchange rate to the DoI.

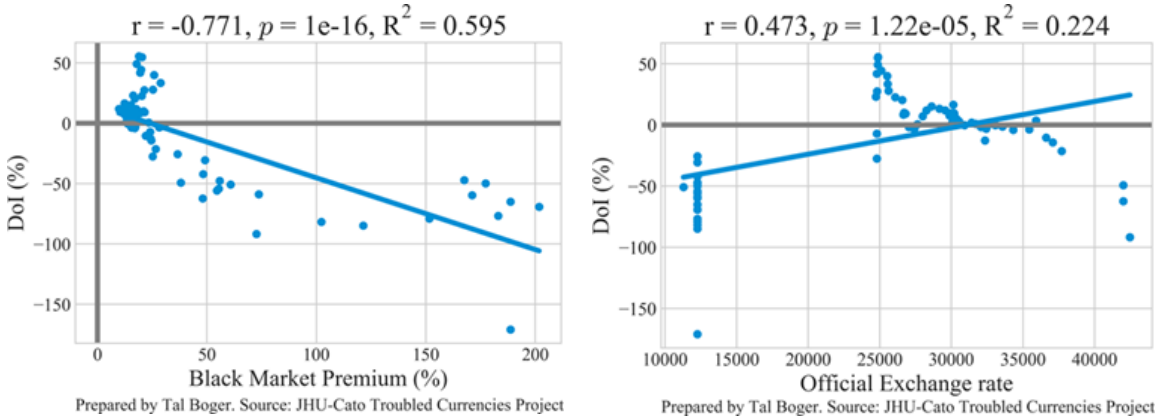


Figure 2. *(a) corr (Black market premium, DoI). (b) corr (Official exchange rate, DoI).*

Both regressions are statistically significant (p-value < 0.05). However, both show a weak correlation coefficient (r-value) and coefficient of determination (R^2). The models' R^2 values of 0.595 and 0.224, respectively, show the models are very weak.

3.2. Months to Equilibrium (MtE)

Figure 3 below shows the correlations between the black market premium and official exchange rate to the MtE.

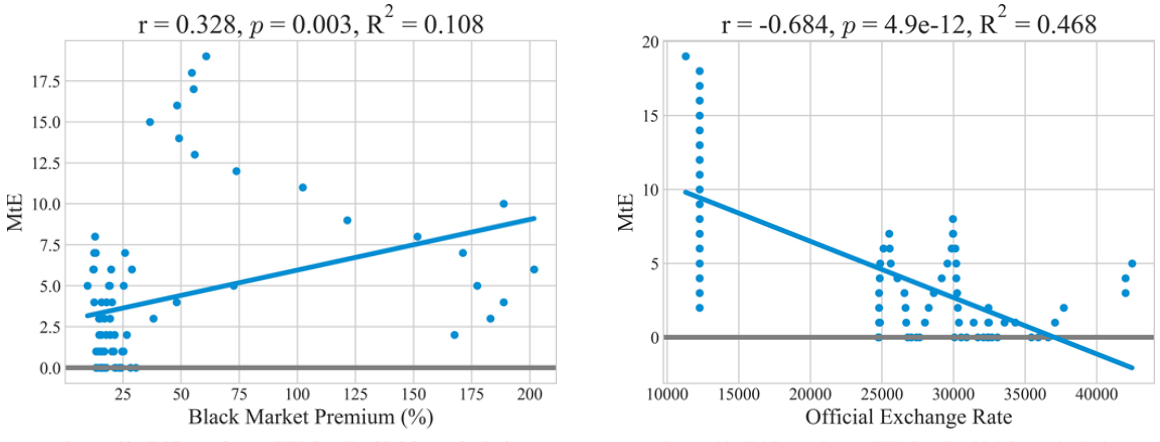


Figure 3. *(a) corr (Black market premium, MtE). (b) corr (Official exchange rate, MtE).*

Both these regressions are weak - again. Despite their statistical significance, the r-value and R^2 are too low to be able to make low-variance predictions about PPP lag using these linear regressions. The same goes for the DoI regressions.

4. Machine Learning Practices

Like most ML algorithm creators, we split the data into two components:

1. The training set. This is what the models use to "learn" the coefficients of the model
2. The test set. This is a set of known data for which the model predicts data. The predicted data is compared to the observed data to measure the model's accuracy.

Although ML algorithms have proven to be very powerful in fitting data, in many cases, they overfit the data to the specific training data. Overfitting creates a very high variance for the models when they are applied to data outside of the give training set (i.e. when applied to the most recent data outside the training set to predict PPP lag).

To examine the possibility of overfitting, we employ *k* -fold cross-validation (with k = 3). Cross-validation tests the model on difference splits of the dataset. In *k* -fold cross- validation, the original sample is randomly divided into *k* equal samples. This allows us to test the model's performance on a different test set to ensure that its accuracy holds when applied to different datasets of the same type.

5. Support Vector Machines (SVMs)

The support vector machine was first invented by Vapnik & Lerner (1963). It seeks to find supporting vectors that create the maximum distance between groups, or maximize *margin* (defined as the distance between supporting hyperplanes).

Because SVMs create separations that are less influenced by large outliers, they can often be more accurate than other regressions. Furthermore, SVMs can be made with both a linear kernel and a radial basis function kernel. This non-linear regression is called the *kernel trick*, which allows us to map inputs into features of higher dimensions, allowing for further optimization of the model.

6. Random Forests (RFs)

Random Forests (RFs) were first introduced by Tin Kam Ho (1995). Random Forests create several decision trees from the training set, and output the mean prediction of the trees. The primary advantage of a random forest over a normal decision tree is that random forests reduce overfitting, thereby increasing their accuracy.

7. *k*-Nearest Neighbors (*k*-NNs)

k -Nearest Neighbors were invented by Evelyn Fix and J.L. Hodges in an unpublished military report in 1951 (Fix & Hodges, 1989). The model predicts a value for an input based on the average its *k* nearest neighbors.

One advantage of the *k* -Nearest Neighbors regressor is that it is a simple lazy-learner algorithm, meaning that the function is calculated locally and values are predicted only once test data is put in (as opposed to eager learners, who learn from the training data). Furthermore, because it predicts values

based on data similar to the input, it works very well on both noisy and simple data.

8. Deep Neural Networks (DNNs)

Artificial neural networks were first introduced by Warren McCulloch & Walter Pitts (1943). Artificial neural networks are based around computing values similar to an animal brain; they create several connected nodes called artificial neurons, which can process and send signals. These connections are called "edges." Both the neurons and the edges have set weights that change as the network learns.

This layer of neurons is called the "hidden layer." A DNN has multiple hidden layers. Because they have the structure of a brain instead of a defined structure (i.e. linear), neural networks are robust in approximating functions regardless of their structure.

9. Hyperparameter Search Space

We created our models using scikit-learn in Python.[11] To optimize the accuracy of the models, we tested the models using varying hyperparameters, and chose the ones that yielded the highest accuracy. We tested:

SVM
- *γ* (gamma; kernel coefficient) - *{*0.000001, 0.00001, 0.0001, 0.001, 0.01, 0.1, 1, 10, 1/n features (scikit-learn's automatic value)*}*
- C (penalty parameter of the error term) - *{*0.001, 0.01, 0.1, 1, 10, 100, 1000*}*
- kernel (the kernel type to be used in the algorithm) - *{*linear, radial basis function, polynomial, sigmoid*}*
- *E* (epsilon; the epsilon-tube within which no penalty is associated in the training loss function with points predicted within a distance epsilon from the actual value)
- *{*.001, .01, .1, 1*}*

RF
- Number of trees - *{*10, 50, 100, 200, 300*}*
- criterion to measure the quality of a split - *{*mean squared error, mean absolute error*}*

k-NN
- n neighbors (number of neighbors to call) - *{*3, 4, 5, 8, 10, 15*}*
- _ weights - *{*uniform, distance*}*

DNN
- hidden layers -*{*25, 50, 75, 100, 150, 200*}*
- activation function for the hidden layer - *{*identity (no-op activation), logistic sigmoid function, hyperbolic tan function, rectified linear unit function*}*
- solver for weight optimization - *{*LBFGS (quasi-Newton method), "adam" stochas- tic gradient-based optimized*}*

10. Model Creation

Our models accept the black market premium, official exchange rate, and PPP implied inflation rate as its inputs. They predict the DoI and MtE with these inputs.

Along with optimizing the hyperparameters for accuracy, we tested different splits of the train and test set to see which yielded the optimal accuracy. We found the following parameters yielded the most accurate results:

- Models predicting DoI (difference of inflation, or reported - implied inflation rate)
  - Training dataset size: 60%, testing dataset size: 40%
  - SVM
    - γ (gamma) = 0.0001
    - C = 100
    - kernel = radial basis function
    - E (epsilon) = 0.01
  - RF
    - number of trees = 50
    - criterion = mean absolute error
  - k-NN
    - n neighbors = 4
    - weights = distance
  - DNN
    - hidden layers = 50
    - activation function = identity (no-op activation)
    - solver for weight optimization = LBFGS (quasi-Newton method)
- Models predicting MtE (months to equilibrium, or months until DoI = 0)
  - Training dataset size: 70%, testing dataset size: 40%
  - SVM
    - γ (gamma) = 0.000001
    - C = 100
    - kernel = radial basis function
    - E (epsilon) = 0.01
  - RF
    - number of trees = 50
    - criterion = mean squared error
  - k-NN
    - n neighbors = 5
    - weights = distance
  - DNN
    - hidden layers = 50
    - activation function = identity (no-op activation)
    - solver for weight optimization = LBFGS (quasi-Newton method)

11. Description of DoI and MtE Data

11.1. Distribution of Data

Before analyzing the accuracy of the different models, it is important to understand the distribution and autocorrelation of the outputs (DoI and MtE).

Though we expect the DoI data may be normally distributed, we do not expect the MtE data to be normally distributed. This is because the inflation rates reach equilibrium multiple times throughout the dataset. Therefore,

after the first time when the inflation rates are at equilibrium, most of the MtE values will be low. Therefore, we expect the MtE distribution to have higher frequencies at lower MtE values. See Figure 4 on the next page.

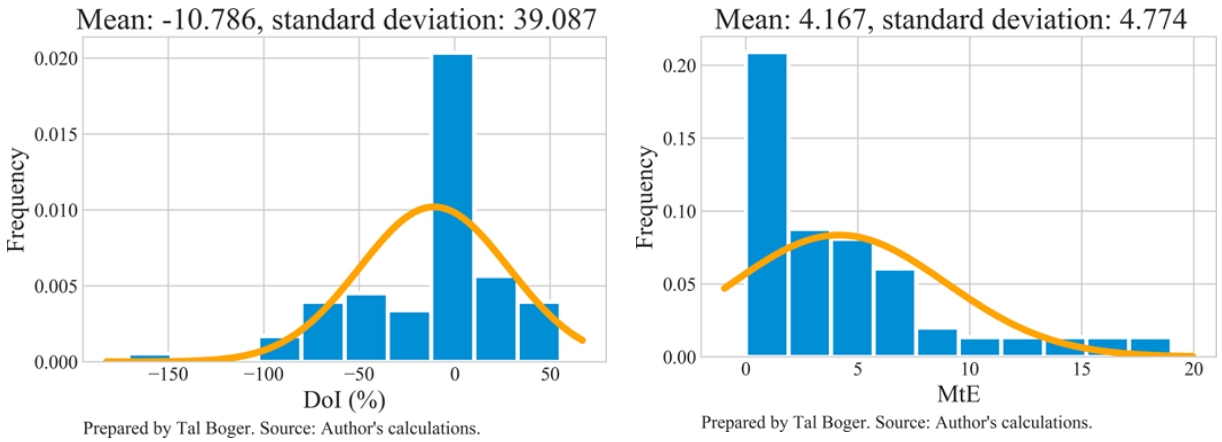


Figure 4. *(a) Histogram of DoI data. (b) Histogram of MtE data.*

As expected, the distribution for the DoI appears close to normal, while the MtE distri- bution is far from it. Though it is important to note that the data itself does not come from a normal distribution, it does not affect our model creation.

11.2. Autocorrelation of Data

We expect the DoI data to have high autocorrelation, given that the PPP implied in- flation rate and official inflation rate approach each other when the DoI is high (i.e. mean reversion). Therefore, the DoI data will have a high autocorrelation in the beginning of the data, which will level out after the inflation rates reach equilibrium for the first time. This same phenomenon will likely apply to the MtE data. At the beginning of the dataset (i.e. before the first time DoI = 0), the MtE is always approaching 0. Therefore, it will have high autocorrelation in the beginning of its data. See Figure 5.

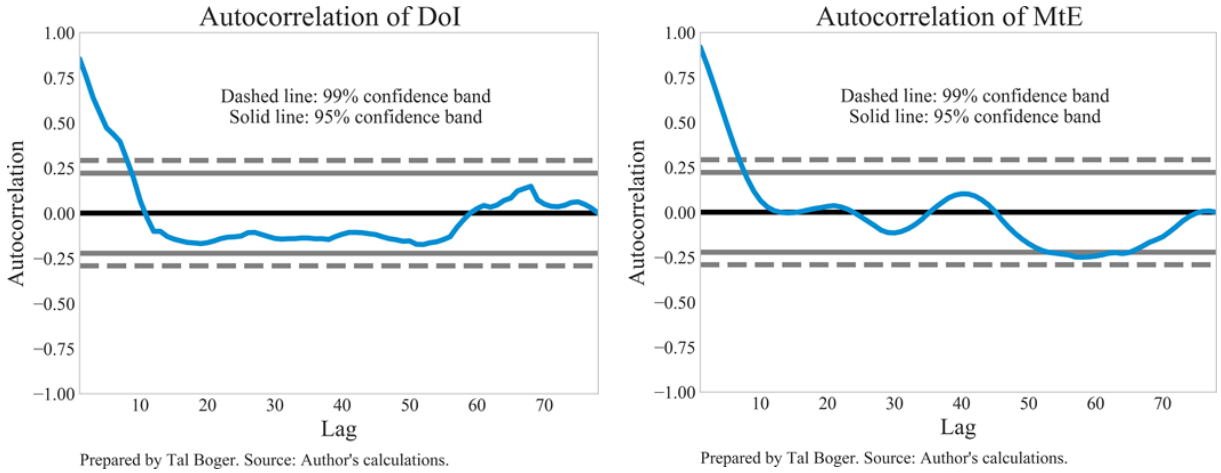


Figure 5. *(a) Autocorrelation plot of DoI data. (b) Autocorrelation plot of MtE data.*

As expected, both datasets have some autocorrelation. Datasets with minimal or no autocorrelation will have autocorrelation values near 0 for the majority of the dataset. The high autocorrelation values in the first ~10 lags, along

with the negative autocorrelation after the first time DoI = 0 shows that both the DoI and MtE data have autocorrelation.

12. Model Analysis

12.1. DoI Models

Because we are primarily interested in finding and analyzing the PPP lag time, our discussion of the DoI models will be limited to simple goodness-of-fit metrics - R^2 and mean squared error. Both these metrics give an estimate of how well the model fits the data.

Table 1 below shows the R^2 and mean squared error for the DoI models.

Table 1. *R2 and mean squared error for the DoI models*

Model	R^2	Mean Squ. Error
SVM	0.938	81.004
RF	0.973	35.309
k-NN	0.914	112.929
DNN	0.961	50.970

All four models have very strong accuracy according to their R^2 and mean squared error. Furthermore, the models have a significantly lower variance than the simple linear models (made in Sections 3.1 and 3.2).

12.2. MtE Models

As we did with the DoI models, we will first examine the R^2 and mean squared error of the models.

After measuring these basic goodness-of-fit metrics, we will then perform *k* -fold cross- validation, along with various analyses on the models' residuals to both further examine their accuracy by testing their randomness and normality.

12.2.1. R^2 and Mean Squared Error

Table 2 on the next page shows the R^2 and mean squared error of the models.

Table 2. *R^2 and mean squared error of MtE models.*

Model	R^2	Mean Squ. Error
SVM	0.902	2.181
RF	0.883	2.607
k-NN	0.860	3.113
DNN	0.747	5.615

The models' R^2 values are relatively strong, and the mean squared error for all models is low. All models have a mean squared error of less than 6 months, indicating that our estimates for PPP lag will likely have an error of less than half a year.

12.2.2. k-Fold Cross-Validation

To examine that our models are not overfitting (meaning they would have very high variance when applied to other data), we performed *k* -fold cross-validation for the models' R^2 scores.

Table 3 below shows the cross-validation scores and the 95% confidence intervals of the scores.

Table 3. *Cross-validation scores and 95% confidence interval for R^2*

Model	Cross-Validation Score for R2	95% Confidence Interval
SVM	0.874	±0.103
RF	0.710	±0.268
k-NN	0.768	±0.096
DNN	0.468	±0.344

Only the DNNs cross-validation score is significantly lower than its R^2. This combined with the fact that its 95% confidence interval has the largest range of the regressions indicates that the DNN is likely overfitting to some extent. However, the DNN had the lowest R^2 and highest mean squared error of the 4 regressions, so we already suspected it would be the least accurate. This assumption will be tested more later.

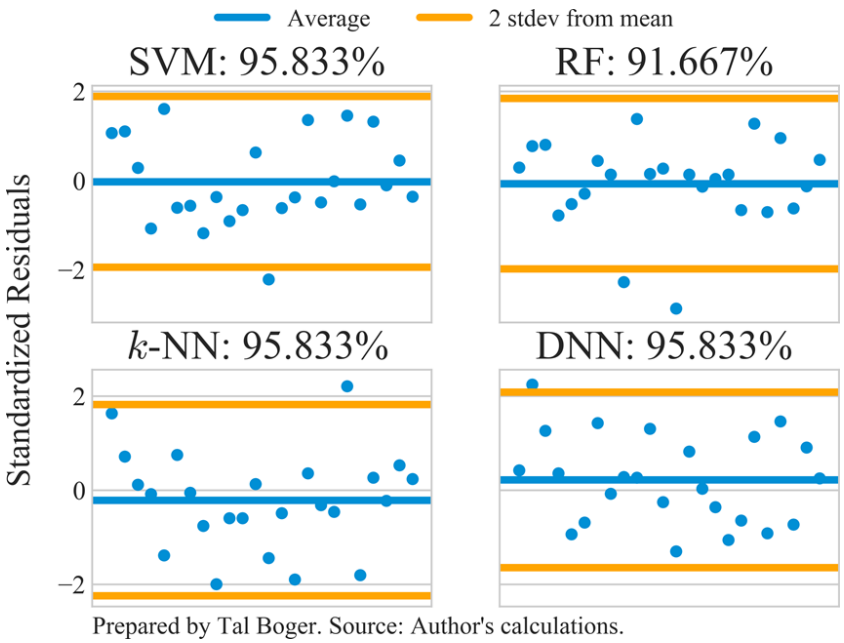
The other three regressions have cross-validation scores only marginally lower than their R^2, and their 95% confidence intervals have small ranges. The R^2 for the RF and *k*-NN is at the upper end of the 95% confidence interval for the cross-validation score. Meanwhile, the SVM's R2 is very close to its cross-validation score. Therefore, the SVM is likely the most accurate regression. The RF and k-NN regressions are similarly accurate, and the DNN is the least accurate.

12.2.3. Standardized Residuals Test

To prove that the models' errors are truly random and to account for the fact that variances for different observations differ, we standardized the residuals of all four models. Because the number of samples (n) in our test set is less than 30, we used the following equation:

$$\text{Standardized Residual}_i = \frac{\text{observed - expected value}_i}{\sqrt{\frac{(\sum_{i=1}^{n} \text{observed - expected value}_i)^2}{n-2}}}$$

95% of the standardized residuals should be within two standard deviations of the mean, and any standardized residual with an absolute value greater than 2 is an outlier. See Figure 6 below.



Prepared by Tal Boger. Source: Author's calculations.

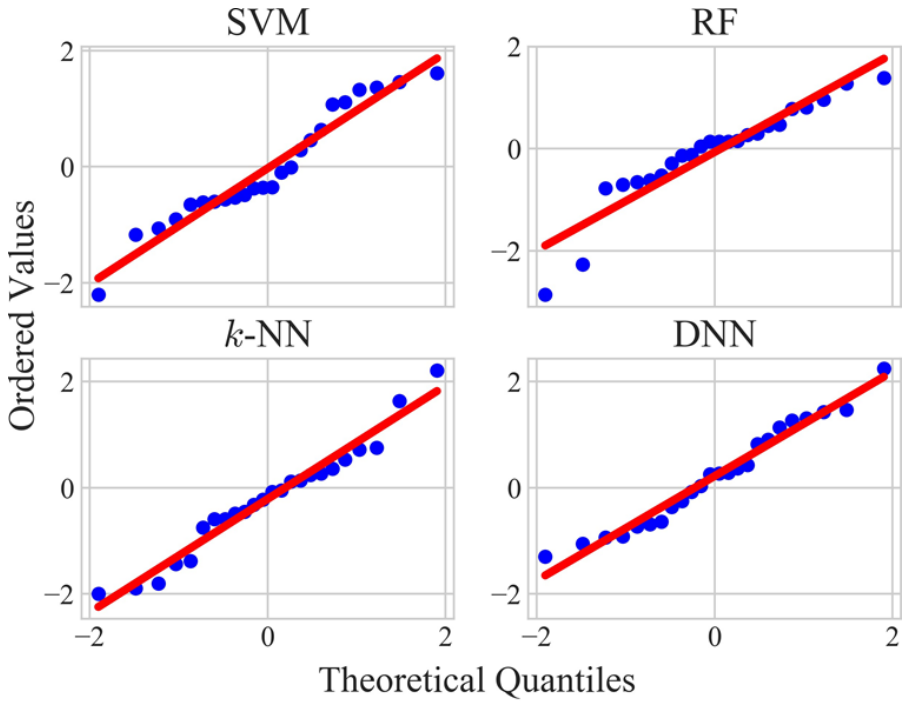
Figure 6. *Standardized residuals test results for all four models.*

Only the RF fails to pass the 95% threshold for standardized residuals within two stan- dard deviations of the mean. Furthermore, it has two standardized residuals with absolute value greater than 2. Therefore, its errors are not truly random.

Along with passing the 95% threshold, each of the other three models have only one standardized residual with absolute value greater than 2. Furthermore, their standardized residuals plots have no visible trend, further indicating that the models' errors are random. Therefore, the standardized residuals test confirms that the errors of the SVM, *k*-NN, and DNN are random and likely normally distributed. The results question the accuracy of the RF model, given that it fails some qualifications of the statistical test.

12.2.4. Q-Q Plot for Standardized Residuals

To further examine the distribution of the standardized residuals for the models, we created a Q-Q plot (quantile-quantile). A Q-Q plot plots the quantiles of our sample data against the quantiles of a normal distribution. A normally distributed sample has a Q-Q plot with a line y = x, with most of the points close to said line. See Figure 7 below.



Prepared by Tal Boger. Source: Author's calculations.

Figure 7. *Q-Q plot of standardized residuals for all four models.*

All the Q-Q plots seem to have a line with a 45° angle. Furthermore, they all appear to have a line close to y = x, though the DNN's line is slightly elevated.

The points for the SVM, *k* -NN, and DNN appear very close to the line with no significant jumps. Though most of the RF's points fall close to the line, the two left-most points have a significant jump between them and the rest of the graph.

Therefore, the SVM, *k* -NN, and DNN standardized residuals likely follow a normal dis- tribution according to the Q-Q plot, while the RF standardized residuals likely follow a bimodal distribution because of the jump in the data.

12.2.5. Shapiro-Wilk Test for Normality of Standardized Residuals

To further test whether the models' standardized residuals are normally distributed, we performed a Shapiro-Wilk test on the standardized residuals.

The test returns a W-value between 0 and 1. W = 1 when the data is perfectly normally distributed. Smaller values for W indicate the data is less likely to be normally distributed. The test also returns a *p*-value. Note, however, that the H_0 (null hypothesis) of a Shapiro- Wilk test is that the data is not normally distributed. Therefore, higher *p*-values indicate a higher likelihood that the data is normally distributed, as a higher *p*-value signals a lower probability of rejecting H_0. See Table 4 below.

Table 4. *Shapiro-Wilk test for normality of standardized residuals results.*

Model	W-value	*p*-value
SVM	0.943	0.193
RF	0.895	0.016
k-NN	0.965	0.556
DNN	0.965	0.550

The Shapiro-Wilk test confirms our suspicion from the Q-Q plot that the RF's standard- ized residuals are not normally distributed. The RF had the lowest W-value of the four models, and its *p*-value of 0.016 indicates we can reject H_0. Therefore, because the *p*-value allows us to reject H_0, we conclude that the RF's standardized residuals are not normally distributed.

The test's result for the other three models' standardized residuals are encouraging. The SVM has the lowest *p*-value of the three, at 0.193. However, because $p > 0.05$, we cannot reject the H_0 that the SVM's standardized residuals are normally distributed.

Because the *k* -NN and DNN have W-values close to 1 and high *p*-values, it is very likely *k* -NN and DNN's standardized residuals are normally distributed. Despite the SVM's high W-value, its low *p*-value indicates there is a higher chance of rejecting H_0 for the SVM than for the *k* -NN and DNN. Therefore, though it is likely the SVM's standardized residuals are normally distributed, we cannot come to a certain conclusion. We can, however, conclude with high certainty that the RF's standardized residuals are not normally distributed, given that $p < 0.05$.

12.2.6. Durbin-Watson Test for Autocorrelation of Standardized Residuals

Along with testing that the models' standardized residuals are normally distributed, it is important to test that they have minimal autocorrelation. Any significant autocorrelation in a model's errors indicates it makes the same mistake several times and is, therefore, underfitting to an extent (i.e. the model is not complex enough to accurately understand a relationship between our inputs and outputs).

To test for autocorrelation, we performed a Durbin-Watson test. A Durbin-Watson test returns a Durbin-Watson test statistic (DW-value) between 0 and 4. A value of 2 indicates no autocorrelation. Values between 0 and 2 indicate positive autocorrelation, and values between 2 and 4 indicate negative autocorrelation. See Table 5 below.

Table 5. *Durbin-Watson test for autocorrelation of standardized residuals results.*

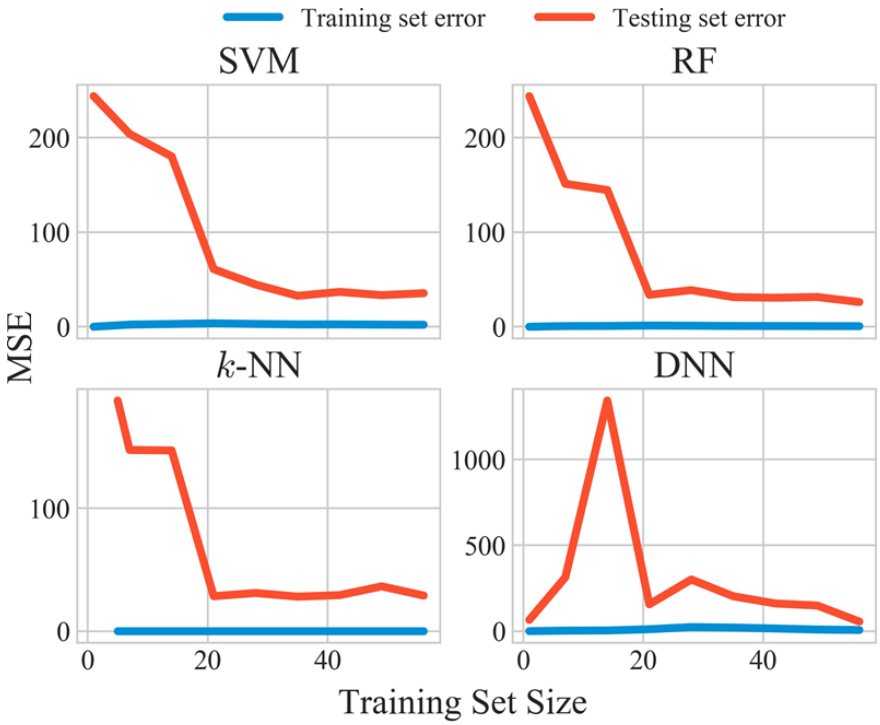
Model	DW-value
SVM	2.159
RF	2.600
k-NN	2.009
DNN	2.054

The Durbin-Watson test confirms our results from the other tests in terms of which models are most usable. The *k*-NN and DNN have DW-values closest to 2, with the SVM close behind them. The RF has a DW-value significantly farther from 2 than the other three models. Therefore, the Durbin-Watson test helps prove that the *k* -NN and DNN are the most accurate models, followed by the SVM, and then the RF.

13. Improving Accuracy for Future Models

Though our models are already very robust, we suspect they can be improved with more data. Our data extends back to only January 2012. Because we only included points for which the official inflation rate was available, we only have one data point per month.

To examine whether adding more data would increase the models' accuracy, we plotted the models' learning curves (their training and testing set error with varying training set sizes). The trend of the error lines indicates not only whether future data will improve the model accuracy but also if there is a high bias or error in the models. We can examine these lines for both mean squared error (MSE) and R^2. See Figure 8 below.



Prepared by Tal Boger. Source: Author's calculations.

Figure 8. *Learning curve of all four models for MSE.*

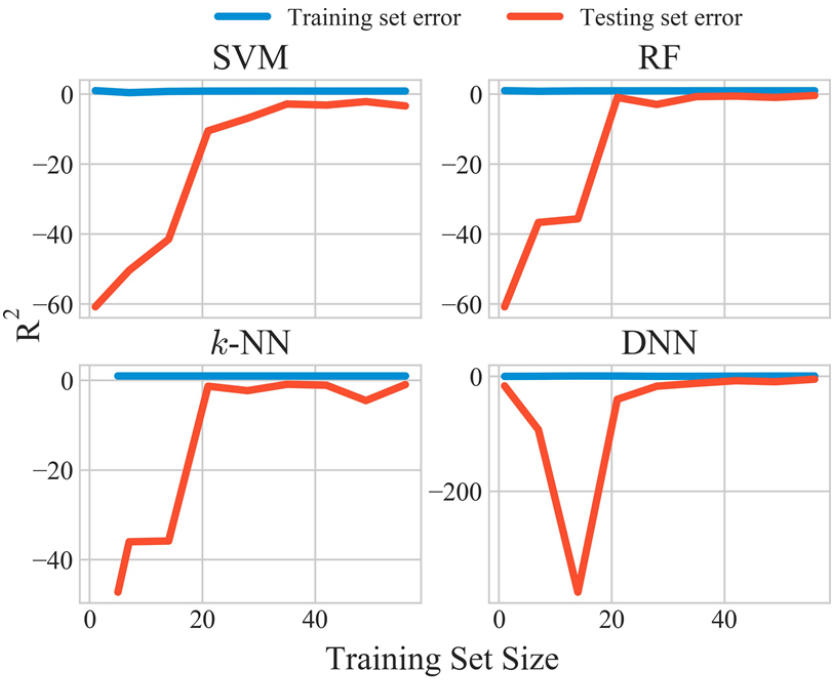
Note that the *k* -NN data starts at a training set size of 5 because the *k* -NN is constructed using the 5 nearest neighbors, and *n samples* ≥ *n neighbors*.

The noticeable difference between the training set error and testing set error lines for the RF and *k*-NN indicate that adding more training data is very likely to increase accuracy.

Though the training and testing set lines appear very close together for the SVM and DNN, there is still a significant margin between the final result. The difference between the testing set error and training set error for the SVM and DNN is about 33.336 and 48.960, respectively. Therefore, all four models are likely to have a decreased MSE if more training data was added.

The minimal changes in the training set error indicate that the models are in a low bias, high variance state. We think this is the ideal case in the bias-variance tradeoff for our models, given that additional data will have similar qualities to our current training data.

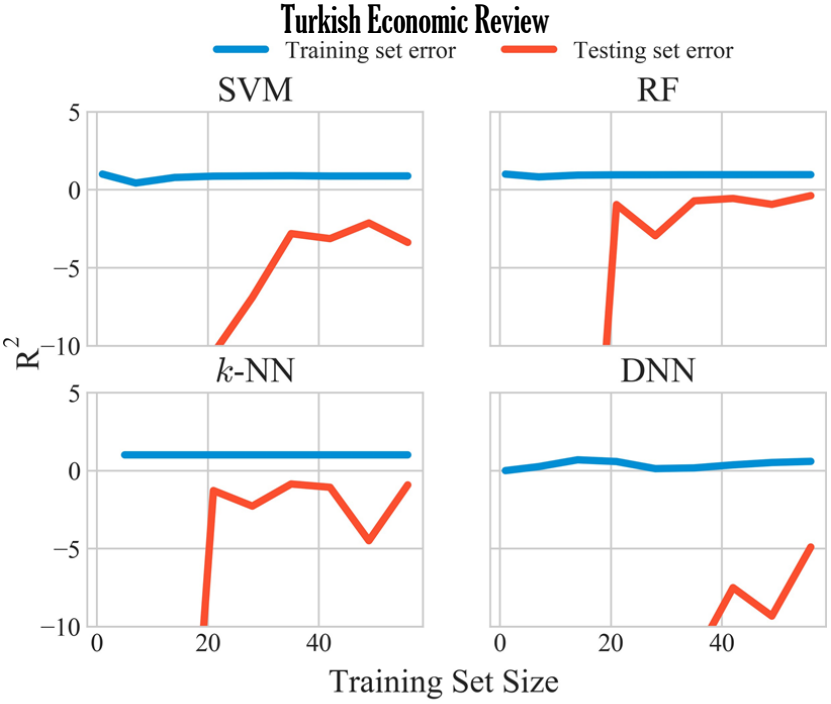
We also made a learning curve for the models' R^2. See Figure 9 below.



Prepared by Tal Boger. Source: Author's calculations.

Figure 9: *Learning curve of all four models for R^2.*

These learning curves also show a low bias state. Because the training set's R^2 is between 0 and 1, it is difficult to see how close the testing set error line is to the training set error line. See Figure 10 below for the learning curves with a smaller y-axis range, such that the difference between the training set error and testing set error lines is clear.



Prepared by Tal Boger. Source: Author's calculations.

Figure 10. *Learning curve of all four models for R² with y-axis range of (-10, 5).*

This shows that though the R² of the testing set is approaching that of the training set, there is still a sizable difference between the two. Therefore, the models' R² would increase if more training data was added.

14. Conclusion

By applying ML algorithms to Iran's inflation data, we can accurately measure the PPP lag time, and how long it will take for there to be no PPP lag. Though we only minimally discussed the DoI models, they can be used to predict Iran's official inflation rate with very high accuracy (at least by basic goodness-of-fit metrics). Our MtE models are also very robust. They not only have strong goodness-of-fit metrics, but also pass various other tests examining whether the errors are normally distributed and random. Therefore, they can be used to measure PPP lag times with minimal variance and high accuracy.

As more data becomes available, the models' accuracy will improve, as shown by the learning curves. This will make even lower variance predictions of PPP lag. Furthermore, these same methods can be applied to other countries with enough data to predict their PPP lag.

References

Athey, S. (2018). The impact of machine learning on economics. In A. K. Agrawal, J. Gans, & A. Goldfarb (Eds.), *The economics of artificial intelligence: An agenda* (pp. 503–543). University of Chicago Press.

Fix, E., & Hodges, J. L. (1989). Discriminatory analysis. Nonparametric discrimination: Consistency properties. *International Statistical Review / Revue Internationale De Statistique*, *57*(3), 238–247.

Hanke, S. H., & Kwok, A. K. F. (2009). On the measurement of Zimbabwe's hyperinflation. *Cato Journal*, *29*(2), 353–364. https://object.cato.org/sites/cato.org/files/serials/files/cato-journal/2009/5/cj29n2-8.pdf

Ho, T. K. (1995). Random decision forests. *Proceedings of the 3rd International Conference on Document Analysis and Recognition* (pp. 278–282). IEEE.

McCulloch, W., & Pitts, W. (1943). A logical calculus of ideas immanent in nervous activity. *Bulletin of Mathematical Biophysics*, *5*(4), 115–133. https://doi.org/10.1007/BF02478259

Mullainathan, S., & Spiess, J. (2017). Machine learning: An applied econometric approach. *Journal of Economic Perspectives*, *31*(2), 87–106. https://pubs.aeaweb.org/doi/pdfplus/10.1257/jep.31.2.87

Rogoff, K. (1996). The purchasing power parity puzzle. *Journal of Economic Literature*, *34*(2), 647–668.

Vapnik, V. N., & Lerner, A. (1963). Pattern recognition using generalized portrait method. *Automation and Remote Control*, *24*, 774–780.

SCAN ME