Introduction to Machine Learning (ELL - 784)

Assignment – 1

ROHAN KUMAR BOHARA (2021AMA2095)

Part 1 a):

Polynomial Curve Fitting In this part of assignment, we had to implement the concepts of Linear Regression to solve the problem of polynomial curve fitting. The end goal is to identify the underlying polynomial (both the degree and the coefficients), as well as to obtain an estimate of the noise variance.

We assume the data to be from a polynomial of degree M and then generate a design matrix containing feature vectors (1, x1 , x2 , ..., xM) for each x in the data set. In the data distribution, the range of x is from around -1 to 2, hence in case of higher degree of polynomials, say 10, the feature vector would be containing values of orders ranging from less than 10^{-10} to around 10^{3} . This would prevent the proper convergence of gradient descent. Hence, the data has been normalized to have zero mean and unit variance. We then minimise the least-squares error, i.e., $\sum^{n}_{i=0}(h\theta(xi) - ti)^2$ using the above mentioned two methods. Using the implementation of the above methods done using numpy, I have obtained the following results and observations.

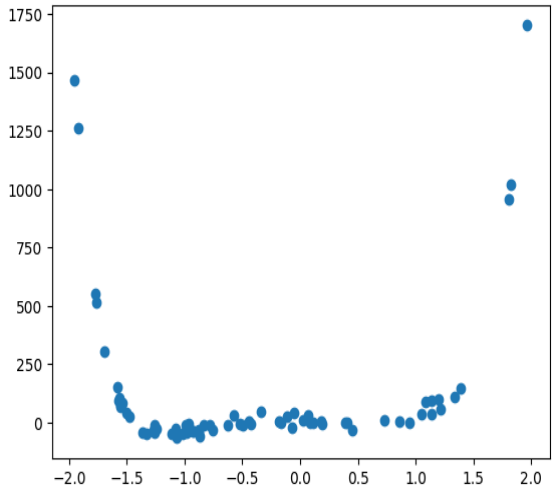


Fig.: Data distribution of full data set

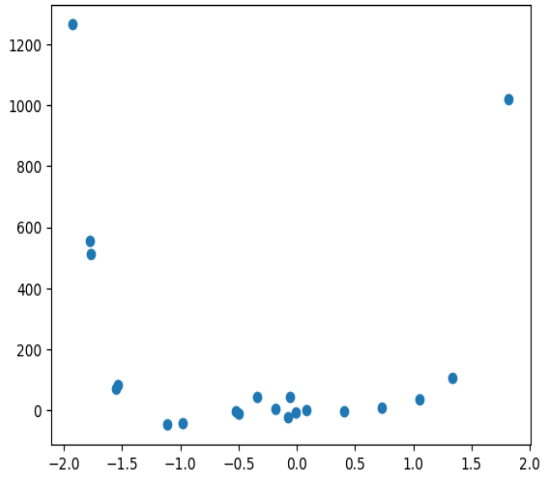


Fig.: Data distribution of first 20 data points

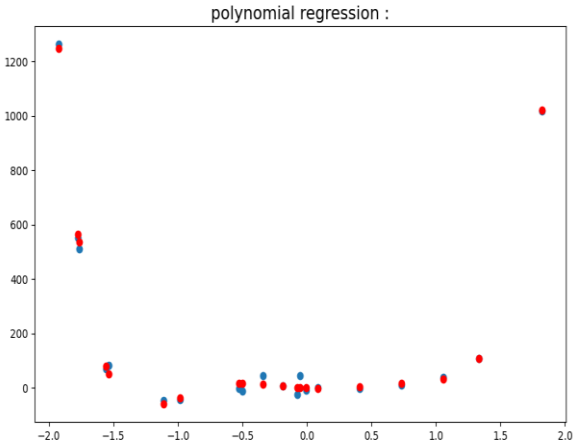


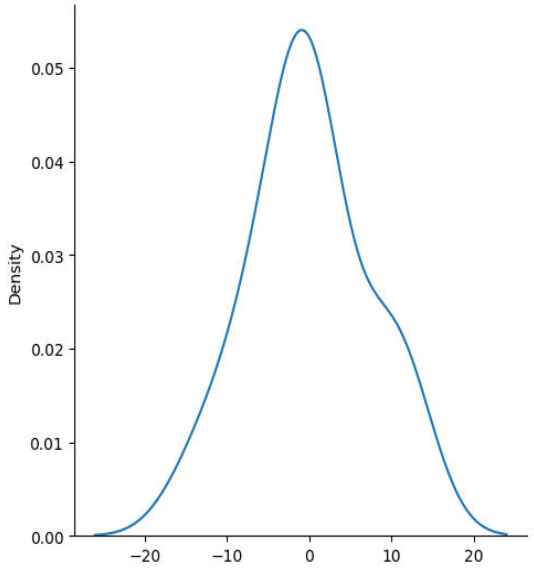
Fig.: Polynomial Curve fitting of 20 data points

Estimate of underlying Polynomial As discussed earlier, from Figure , the underlying polynomial must be of degree 11. For fitting degree 11 polynomial on the entire data set, we can take $\lambda = 0$ as we have already picked the least complex model. After running the least-squares regression on the entire data.

Solved the curve fitting regression problem using error function minimisation by both the analytic and gradient descent approaches.

Estimated variance and mean square error for both training and testing data.

Estimated noise variance for LASSO, RIDGE, ANALYTIC cases.



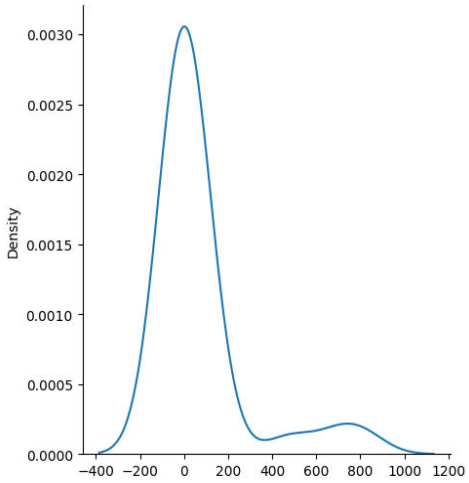
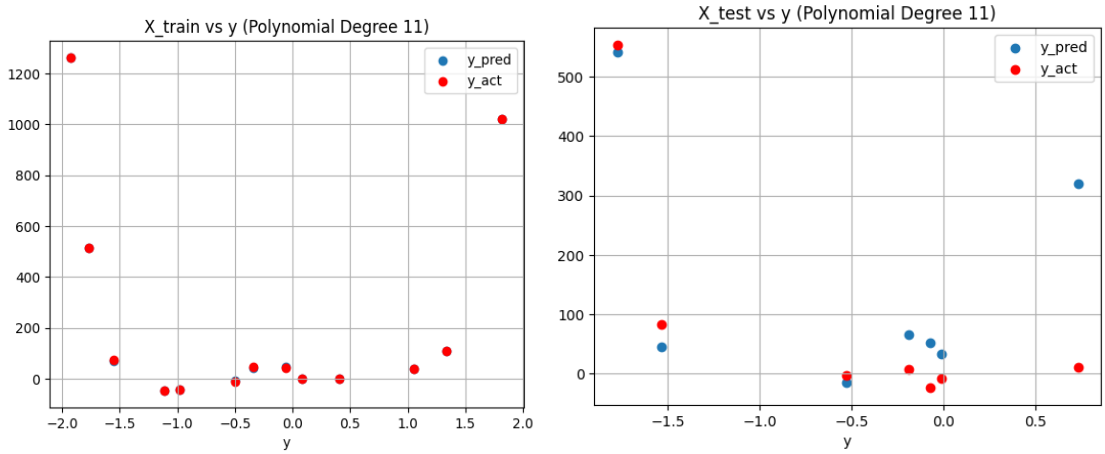
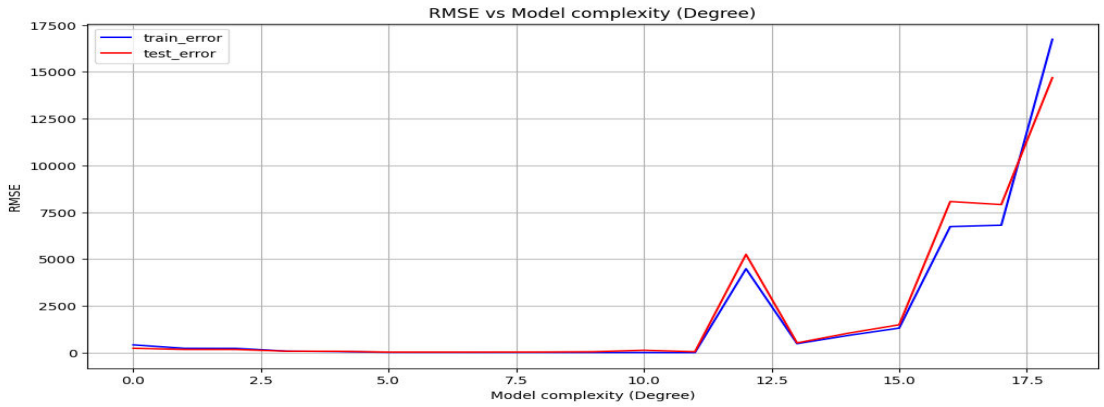


Fig.: Train noise

fig.: test noise

For analytic case:





mean: 1.6373746798756463e-06, standard deviation: 6.863787433897781

variance: 47.11157793773309

Calculated optimised weights in analytic, ridge and lasso model.

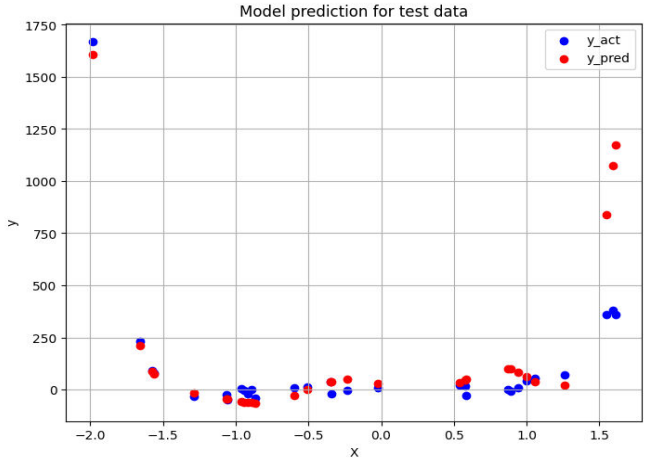


Fig.: Analytic

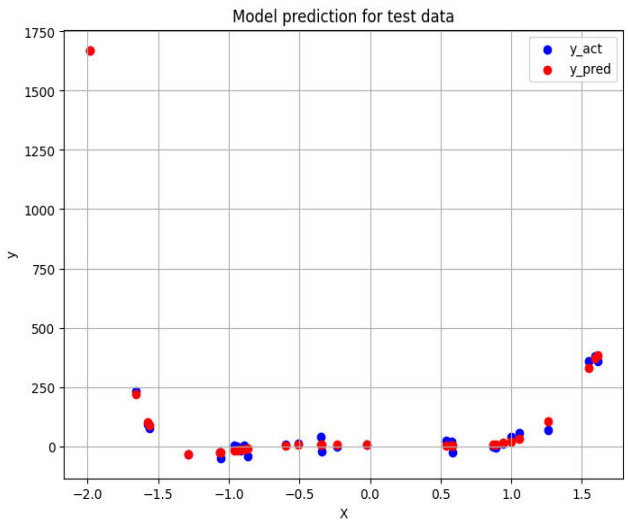


Fig.: Ridge model

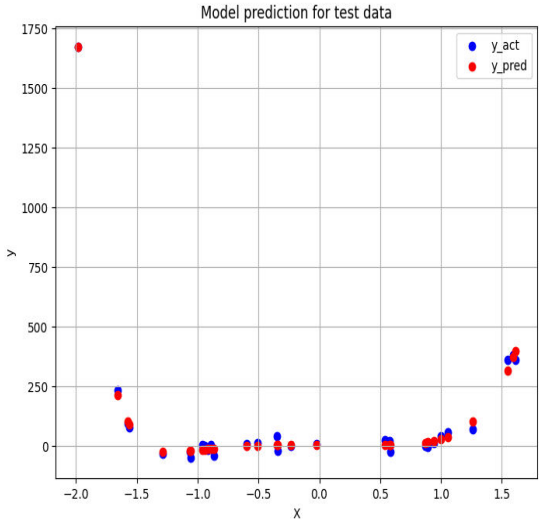


Fig.: Lasso model

Part 1 b):

Here, the noise in the data set provided is of some non-Gaussian distribution. The data given for this task is shown below in the scatter plot:

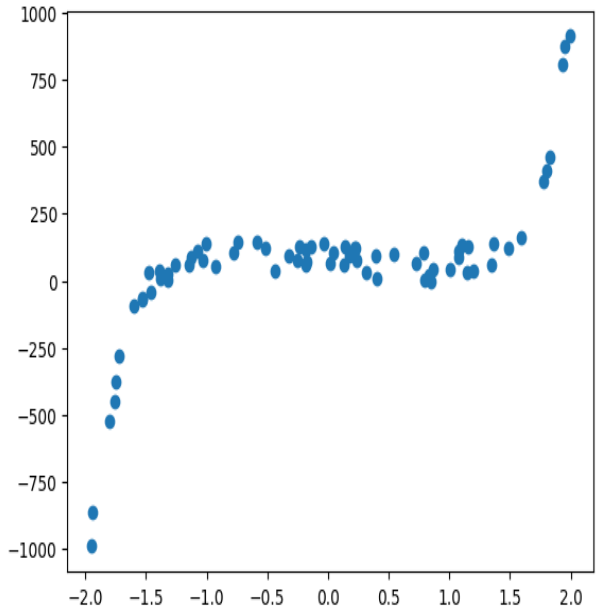


Fig.: Data Distribution of full dataset points

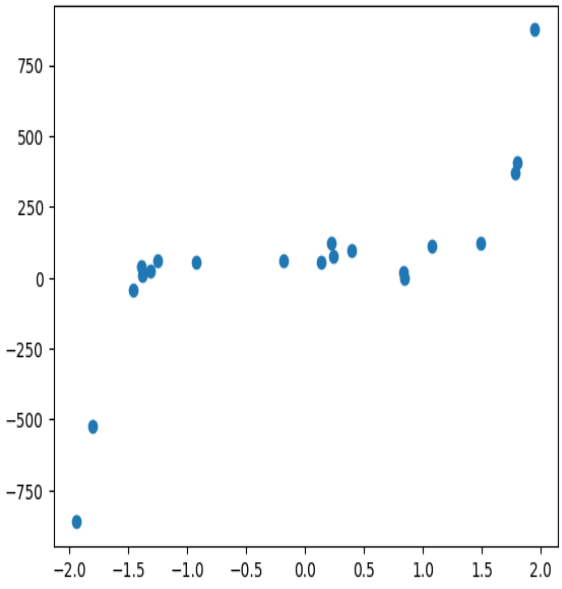


Fig.: Data Distribution of first 20 data

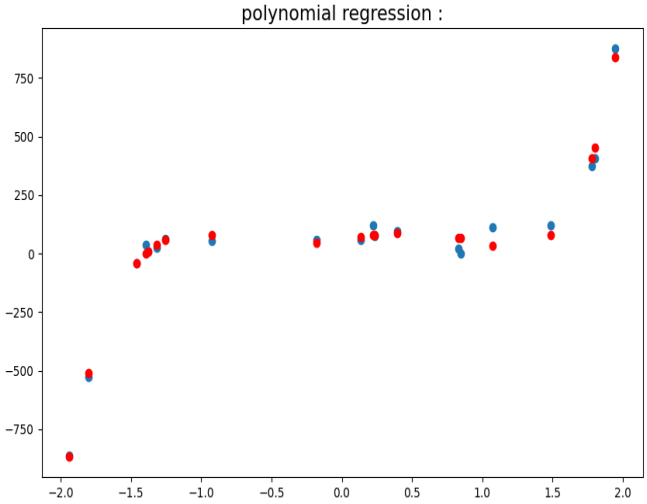


Fig.: Polynomial fitting

Done all the analysis which is required for part b) for non - gaussian dataset.

We have to try and characterise the noise and find out what kind of noise it is. We know, that values ti − y(xi) comprise the noise in the data. Hence, we can to identify the noise by analysing these values. For this, first we have to fit a polynomial of suitable degree on this data. To find the suitable degree of polynomial, we plot the various polynomials found at various values of M.

It can be clearly seen that the loss decreases by a considerable amount for M = 6 and remains mostly constant afterwards. This is a reconfirmation to our claim that the polynomial is of degree 6. Using this information, we then find an estimate for the underlying polynomial by fitting a 6 degree curve to the data. This polynomial found can be used as a good estimate for y(xi), and can be used to find the noise on each data point.

Estimated through L1 and L2 regularisation.

The guess for the noise type would be Poisson distribution with a shifted mean so as to incorporate negative values as well.

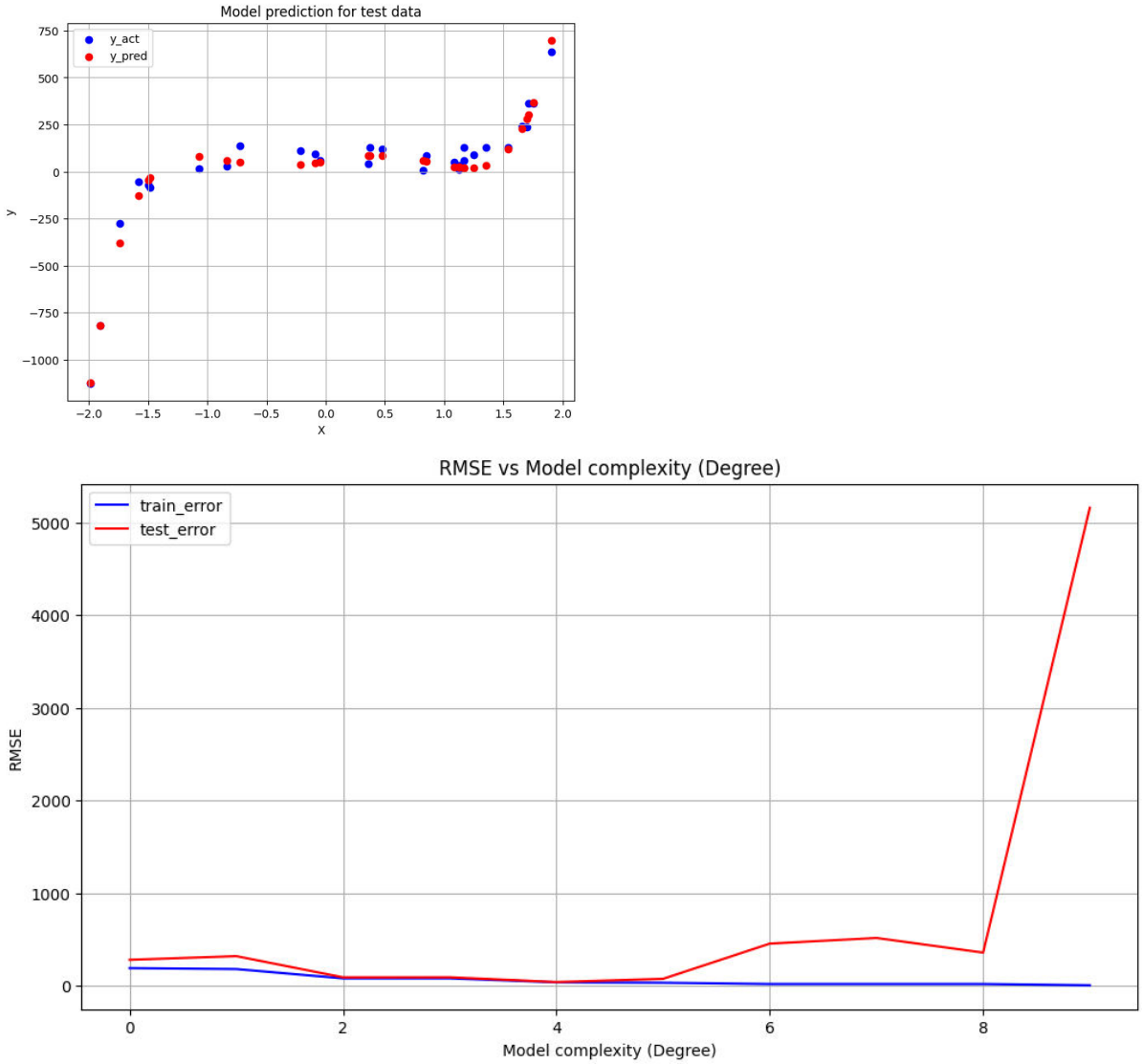


Fig.: Analytic method

Part 2:

Real Time series Data

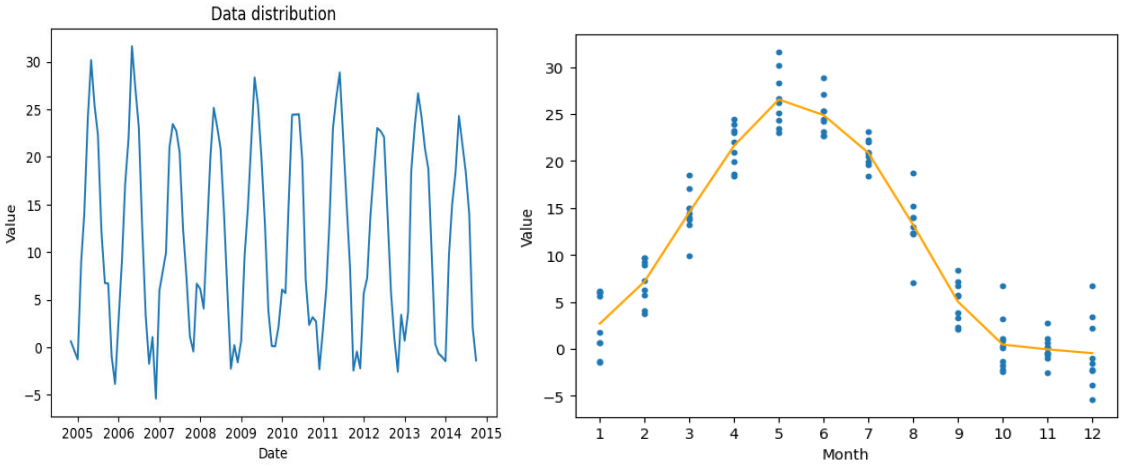
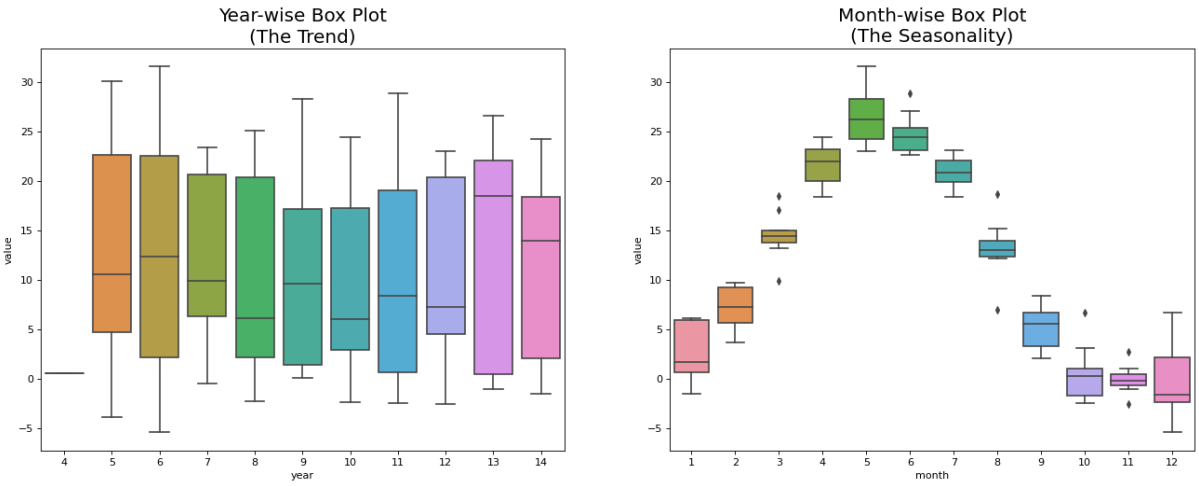


Fig.: Plot on the basis of month

Visualizing the trends and seasonality of the series. The box plot tells that there is not much variation of the series over the years, but is periodic every 12 months. Observation: The range of values in a month is of length at max 5.




```
x_t = np.array(((x_y)*12 +x_m),dtype=int)
```

Training Using above index value for each date is calculated. Then we train the dates with same months with input as their index value and output as the label. Polyfit is implemeted on these models

Predictions Depending on the month, the value is predicted on date's month. The model were trained using regularization, coupled with cross-validation.

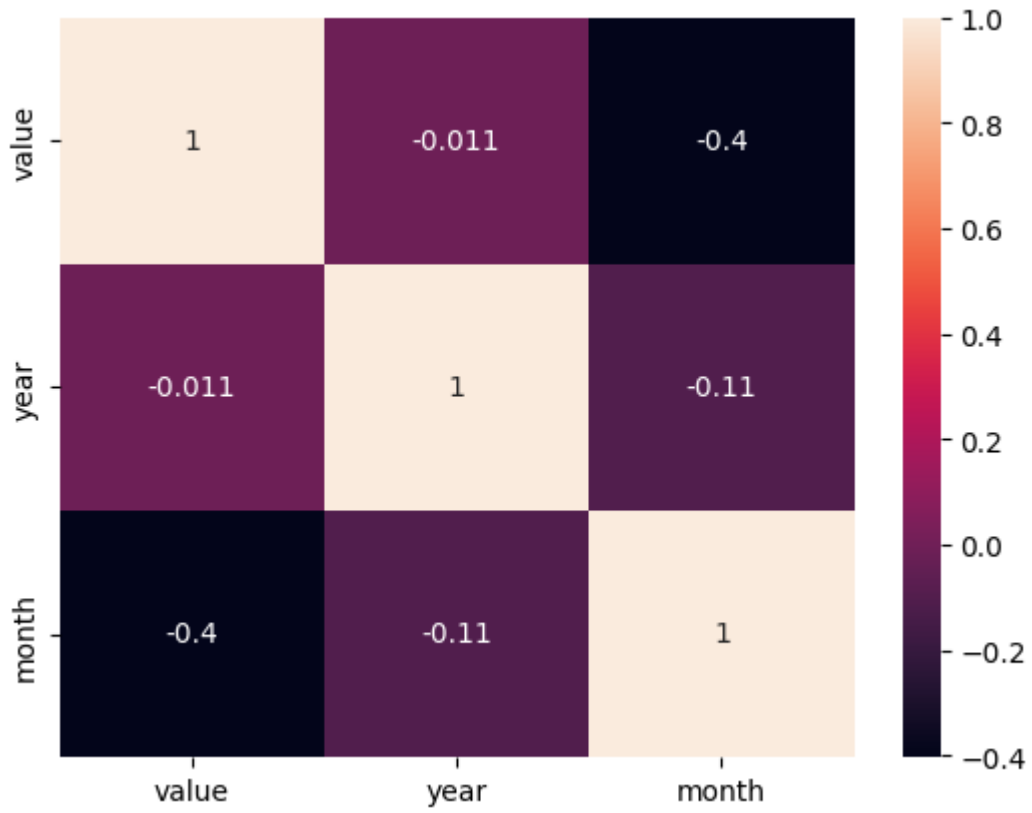


Fig.: correlation matrix