

Solution Sheet

1. Which model have you used for probability prediction? Explain your model.

After imputing the missing values and eliminating the zero correlated features (ie: Name, people_ID). I have applied the Encoding to encode the categorical variables into numerical values. After all the necessary pre-processing I've applied used Deep Neural Network(DNN) as regressor but result wasn't promising. Then opted ensemble methods to perform the regression task and results were reasonable good. I have fit the data on two ensemble methods:

1. **GradientBoostingRegressor** : GB builds an additive model in a forward stage-wise fashion; it allows for the optimization of arbitrary differentiable loss functions. In each stage a regression tree is fit on the negative gradient of the given loss function.

2. **AdaBoostRegressor**: An AdaBoost regressor is a meta-estimator that begins by fitting a regressor on the original dataset and then fits additional copies of the regressor on the same dataset but where the weights of instances are adjusted according to the error of the current prediction. As such, subsequent regressors focus more on difficult cases.

2. Which model have you used for Diuresis Time series prediction? Explain your model.

I have used LSTM(Long Short Term Memory) which is RNN(Recurrent Neural Network with memory cell) for time series predictions. Time series prediction problems are a difficult type of predictive modeling problem. Unlike regression technique, time series also adds the complexity of a sequence dependence among the input variables. A powerful type of neural network designed to handle sequence dependence is called recurrent neural networks. The Long Short-Term Memory network or LSTM network is a type of recurrent neural network used in deep learning because very large architectures can be successfully trained. The Long Short-Term Memory network, or LSTM network, is a recurrent neural network that is trained using Back-propagation Through Time and overcomes the vanishing gradient problem. LSTM networks can be stacked in Keras in the same way that other layer types can be stacked. One addition to the configuration that is required is that an LSTM layer prior to each subsequent LSTM layer must return the sequence. This can be done by setting the return_sequences parameter on the layer to True.