1. Of the attributes provided for each user, which subset of attributes is most predictive to estimate the Call Volume for the current month

On the basis ensemble method Extra Tree Regressor the feature importance along with the score are sorted:

1. feature 16 (0.079615)
2. feature 7 (0.076338)
3. feature 22 (0.075712)
4. feature 12 (0.067482)
5. feature 4 (0.056520)
6. feature 1 (0.050123)
7. feature 10 (0.046251)
8. feature 5 (0.044783)
9. feature 23 (0.042375)
10. feature 17 (0.041204)
11. feature 8 (0.038267)
12. feature 14 (0.033731)
13. feature 19 (0.033474)
14. feature 21 (0.031547)
15. feature 2 (0.031047)
16. feature 18 (0.026311)
17. feature 11 (0.025081)
18. Age (0.024633)
19. feature 15 (0.023325)
20. feature 24 (0.022353)
21. feature 8 (0.022349)
22. ID (0.021926)
23. feature 9 (0.021708)
24. feature 20 (0.021548)
25. feature 13 (0.021528)
26. feature 3 (0.020768)

After dropping the features having score < 0.02 a subset of features are used to fit the model.

NOTABLE MENTION: Even after selecting important feature and discarding lesser important ones there was impact on R-Squared value after fitting Linear Regression model.

Without discarding features
R-Sqaured = 0.73

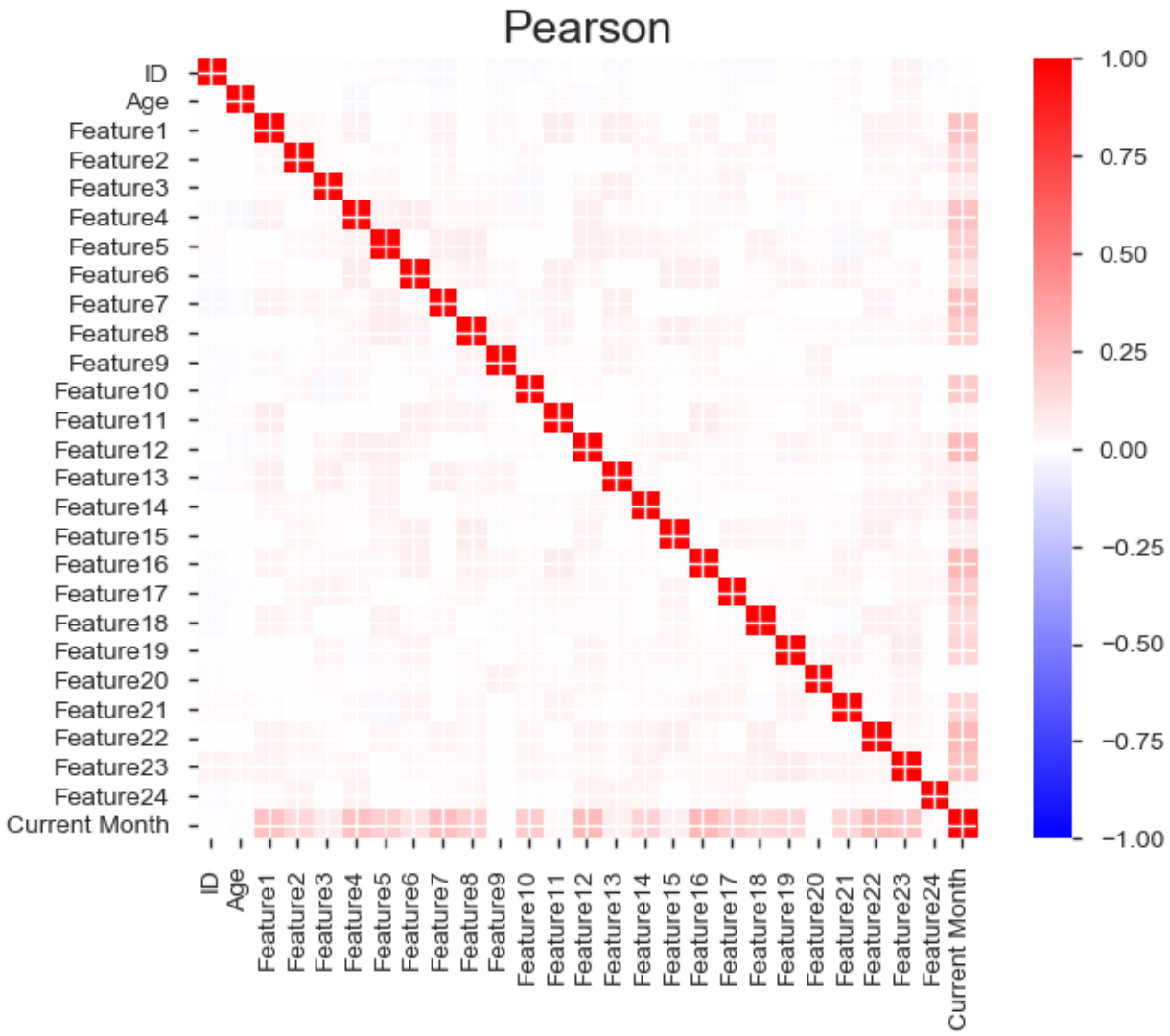
RMSE = 438.092

After discarding lesser important feature:
R-Squared = 0.72

RMSE = 443.45

2. To better predict the Call Volume, which set of given attributes would you explore to make new features and why?

To check the co-relation among features for further reduction and also to check with the linear co-linearity of such feature to target feature, Current Month following heat map is produced using Pearson's coefficients:



This grid signifies the features aren't correlated much among themselves. So dimensionality reduction isn't a good choice.
Also, Low correlation means there's no *linear* relationship; it doesn't mean there's no information in the feature that predicts the target.

To generate a higher order equation we can add powers of the original features as new features. The linear model,

$$Y = \theta_0 + \theta_1 x$$

can be transformed to

$$Y = \theta_0 + \theta_1 x + \theta_2 x^2$$

To fit such a model polynomial feature is used.

To check if such high order features from the existing ones improve R-squared, RMSE metrics, degree 2 and 3 polynomial regression is applied with following results:

Degree 2
R-square: 0.67

RMSE: 474.73

Degree 3:
R-squared: negative values implied poor model choice
RMSE: 1337.03

There is a significant decrease in metrics, R-squared, as more complex features ar e added from the existing ones and Linear model surpasses the performance.

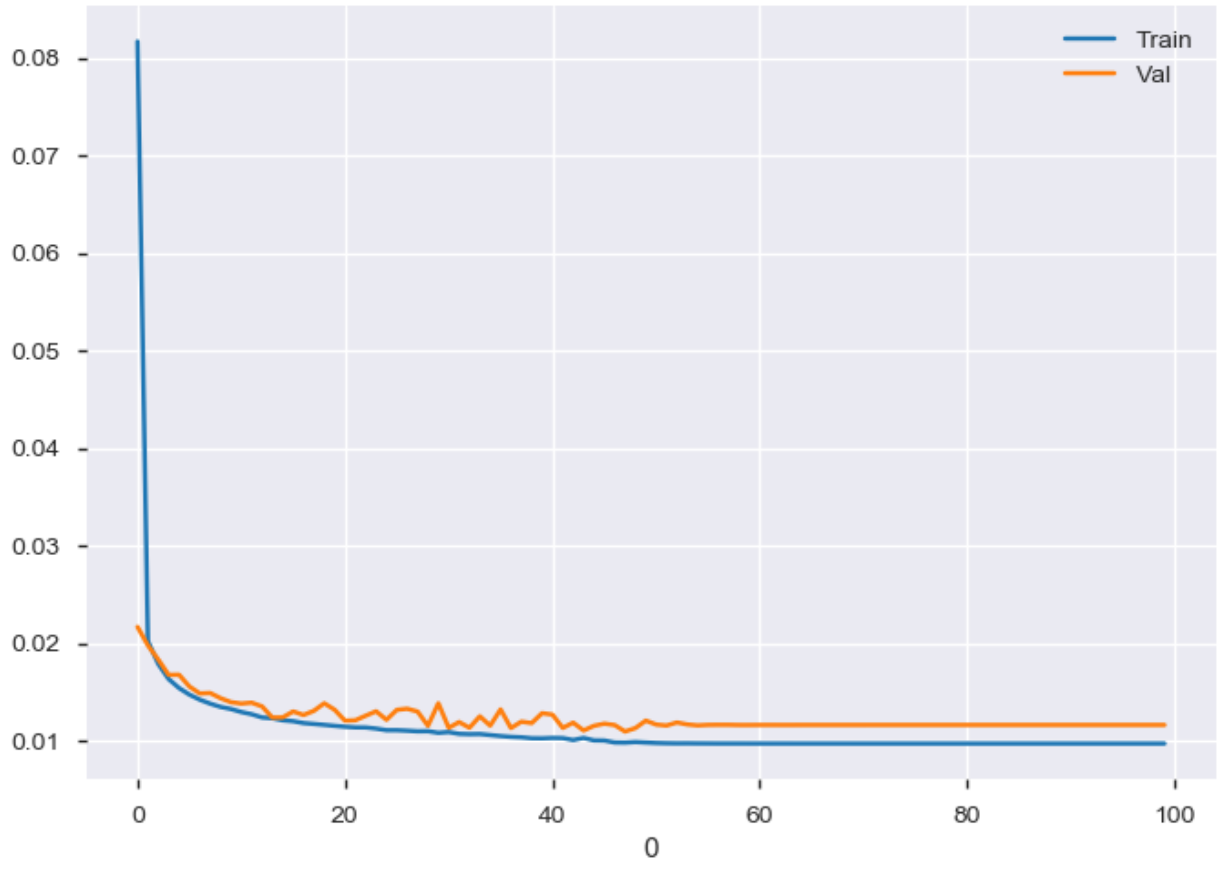
3. Which different models did you applied and which one performed the best to estimate the Call Volume?

Choice of Models

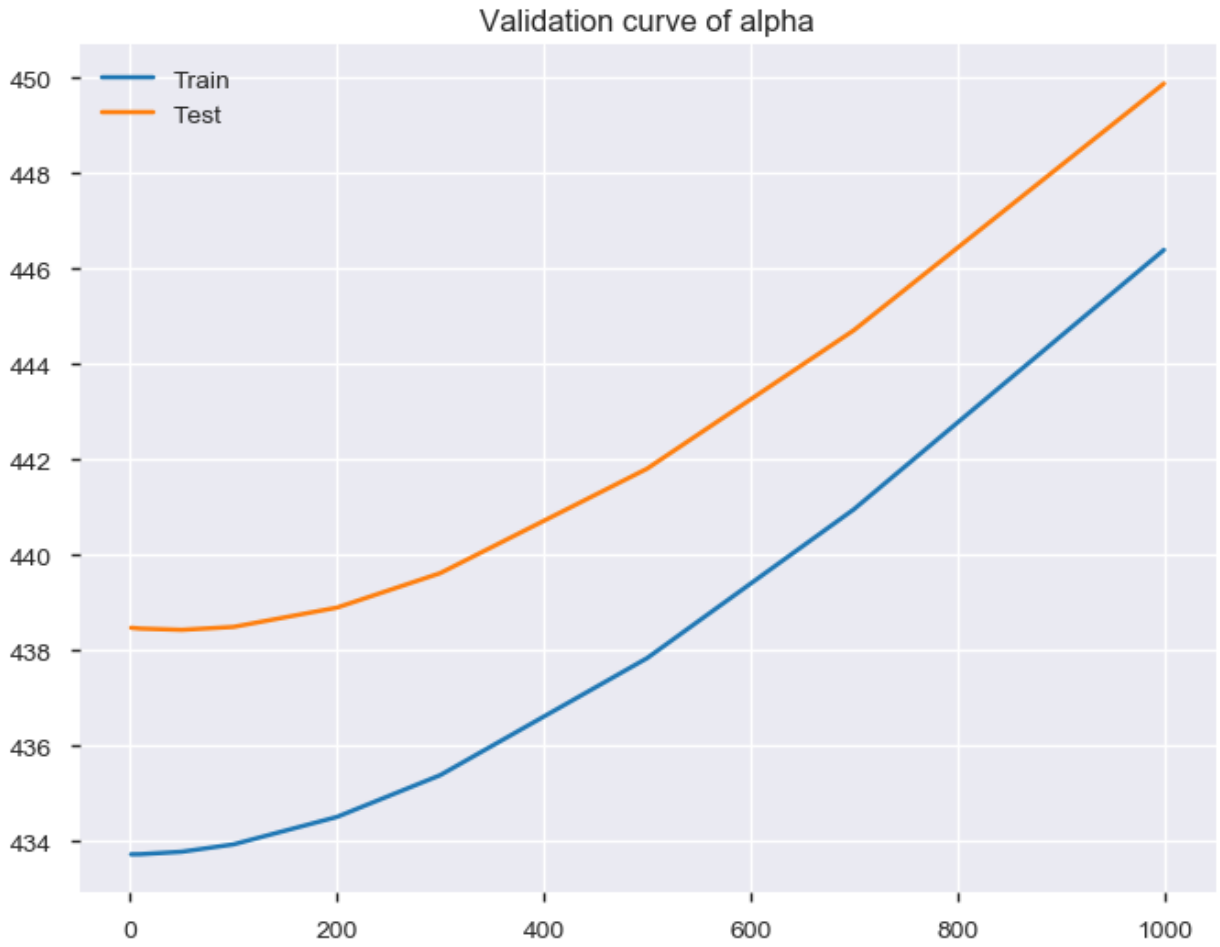
A. LINEAR REGRESSION
B. RIDGE REGRESSION TO COMBAT OVERFITTING
C. POLYNOIMIAL REGRESSION (DEGREE 2 & 3)
D. EXTRA TREE REGRESSOR TO FIND IMPORTANT FEATURES
E. NEURAL NETWORK WITH 3-4 DENSELY CONNECTED LAYERS

The best performance was achieved by 4 Layer sequential network having 3 densely connected layer and one neuron in the output layer with MSE loss 0.0091.
Configuration used:
Prepressing: MinMax Scaling Features, Dropping Name and Address columns

Number of layers: 4
Number of Nodes in densely connected layers: 64
Number of nodes in output layer: 1
Hidden layer activation function: ReLU
Output Layer Activation Function: Linear
Optmtimizer: Adam
Initial Learning Rate: 0.0001
Metrics: Mean Absolute Error
Loss: Mean Squared Error
Epochs: 100



4. Can you show that you are not overfitting? What steps did you take to avoid overfitting.



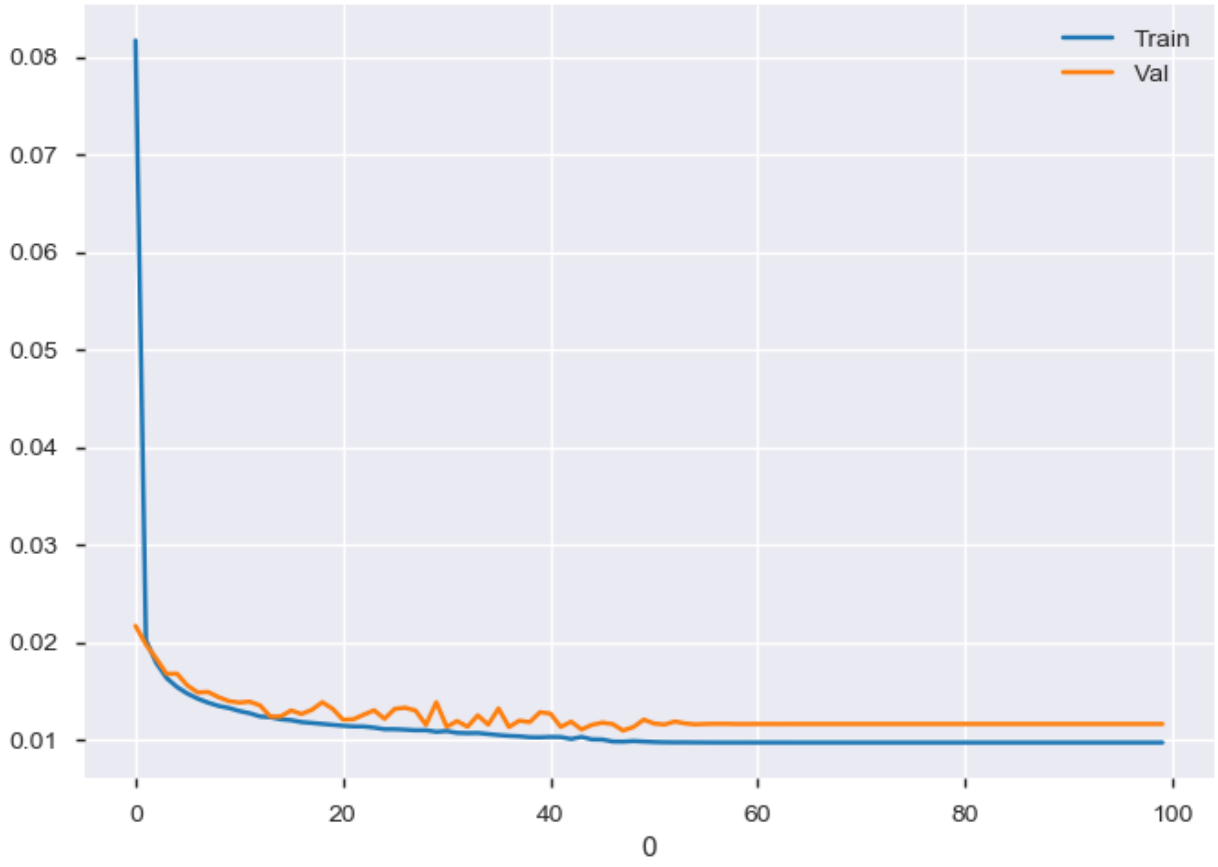
A. After fitting a Linear regression model, Ridge regression is applied to check for various values of alpha if the model is overfitting/underfitting.

   From above plot It can be seen that Train and Test scores are increasing after alpha=600.

   Even at alpha=0, difference between RMSE is not much, so there is no objectionable overfitting. As alpha is increased, bias is increasing and so the RMSE. Alpha=0 signifies a linear regression with no penalty to coefficients of features.

   As linear regression is not overfitting the model, using Ridge regression is not necessary.

Also, the neural network produced the following train loss and validation loss curves



After 50 epochs the train loss and val loss plateaus and reach a minimum and validation loss never increases. This trend of loss decay shows there is no overfittiing. Also, after checking with such results there seems to be no requirement of droput, early stopping techniques.

But definitely, as the size of the dataset grows it will learn better and perform better on unseen data.

For testing: Please use the same preprocessing. The model architecture (architecture.json) and weights after 100 epoch (best_weights.hdf5) are provided in the bundle.