My Approach for solving 1st Problem of the HackerEarth ML Challenge (Jun-Jul, 2020)

Author: Niket Ganatra (GitHub: github.com/dev-niket)

Approach:

First of all, I replaced all of the NaN values with the mean of the respective columns. Thereafter, I tried to use various regression algorithms separately, just to get an understanding of how the different Regression Algorithms were performing (especially Random Forrest).

Previously, I did not have much knowledge of stacking but I did learn it for this challenge and then implemented a single layered stacking with a Linear Regression model as a meta layer but tried to continuously improve it and finally ended up with 2 good possibilities, first, a 2-layered stacking with the first layer containing K-Neighbors Regressor, Decision Tree Regressor and XGB Regressor and then in the second layer I had SVR and Random Forrest Regressor with the Ridge regressor as the meta layer and the second method, a single layered stack containing the 5 regression models mentioned earlier with Ridge regressor as the meta layer. Finally, I opted for the later one since it gave a better result upon testing on the website as well as a slightly better result when I split the 'Train.csv' into training and testing datasets and evaluated the results. The reason for choosing Ridge Regressor was that there were less than 100K records and all the necessary features were already selected by me while Feature Selection so it didn't make sense to use something like a Lasso Regressor or an ElasticNet regressor.

After the model was trained and the predicted values were obtained, a data frame was created with the Employee ID and the predicted Attribution Rate and this was exported to a csv file which was the required format for submission.

Feature Selection:

A very simple approach was applied. I just produced a heat map of correlations between various columns and dropped the appropriate columns. All the columns had a very low correlation with the Attribution rate but some were exceptionally low which were dropped. After this, amongst others, we still had Time of Service, Time Since Promotion and Age. Age had very high values of correlation (compared to the other values in the heatmap) with the other two mentioned and hence was dropped since it could negatively affect the model's performance.

Programming Language Used: Python

Tool Used: Google Colab

Libraries used: pandas, numpy, matplotlib, seaborn, google.colab (for handling files), io, sklearn, xgboost