HR Analytics Project- Understanding the Attrition in HR

In this blog post, I’ll be going through the process of creating a machine learning model to understand the attrition in human resources. But first, let’s try to understand what ‘attrition’ actually means by going through its formal definition and try to understand its impact on companies.

# Problem Definition

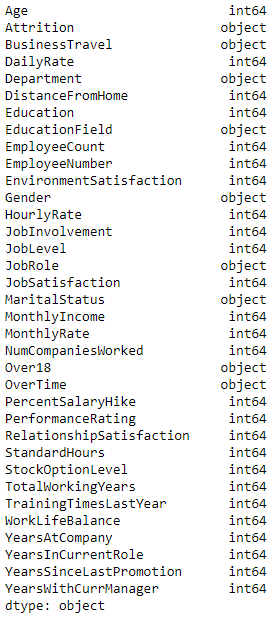
Attrition in human resources refers to the gradual loss of employees over time. Now the reason behind this loss may vary, it may be due to employees retiring due to old age or resigning to pursue different ventures. But how does it impact a company? Attrition is not good for a company, the main issue with a high attrition rate is that it cost the organization a lot. Job postings, hiring processes, paperwork, and new hire training are some of the common expenses of losing employees and replacing them. Additionally, regular employee turnover prohibits your organization from increasing its collective knowledge base and experience over time. This is especially concerning if your business is customer-facing, as customers often prefer to interact with familiar people. Errors and issues are more likely if you constantly have new workers.

Now that we have garnered some idea about what attrition actually means and know its impact on an organization, let’s dive into our dataset.

After importing the necessary libraries and loading the dataset, let’s take a look at the dataset and columns present in it.

# Data Analysis

The dataset consists of 1470 entries with 35 different columns which are a mixture of object and int data types with our target variable ‘Attrition’ being object data type, hence it’s a classification problem. We will go through the data and build a model which will predict whether the value for attrition will be ‘Yes’ or ‘No’, therefore, classification problem!

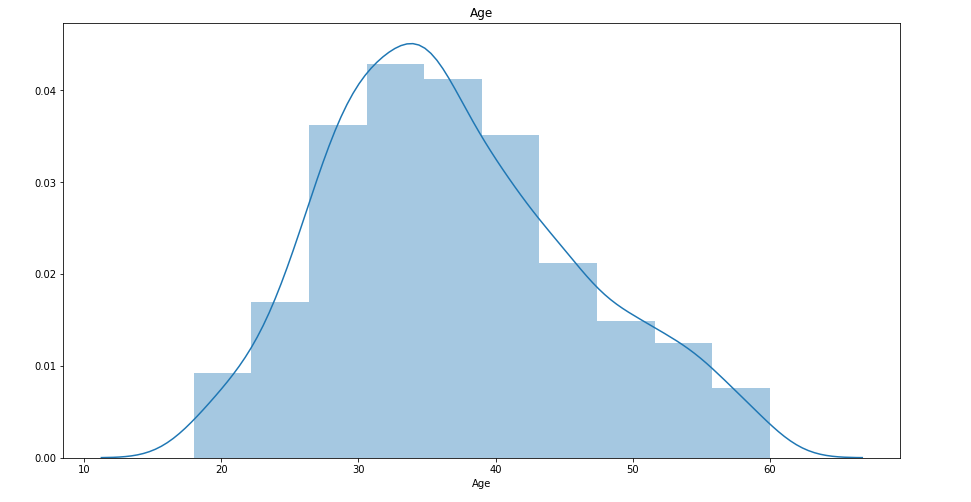


On further inspection of data, I found out that many int type columns are not continuous since they only hold few unique values and are instead categorical in nature. Columns like ‘EnvironmentSatisfaction’, ‘JobInvolvement’, ‘JobSatisfaction’, ‘ PerformanceRating’, ‘RelationshipSatisfaction’, etc. are just ratings for 1 - 4 depending on employee satisfaction level. Whereas ‘Education’ is just a label encoded column for the education level of an employee with 1 - ‘Below College’, 2 - ‘College’, 3 - ‘Bachelors’, 4 - ‘Masters’, 5 - ‘Doctor’.

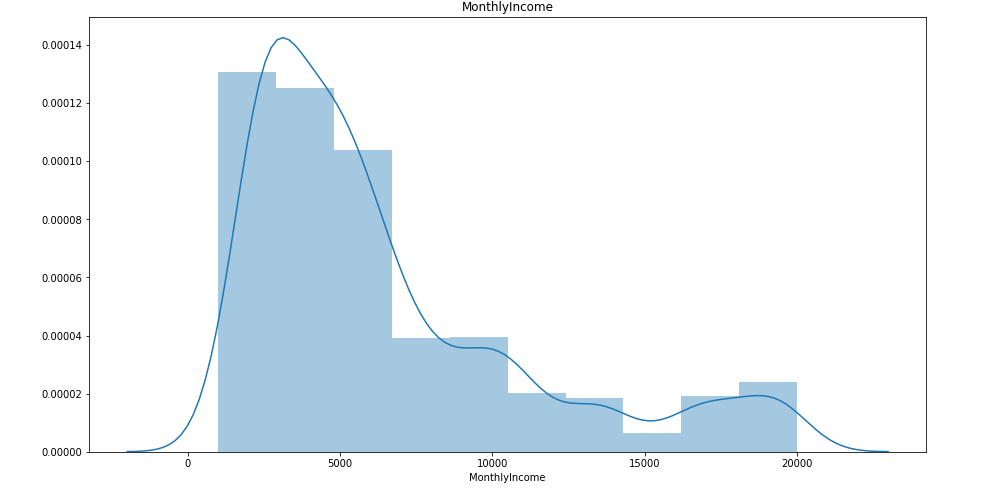
Also noticed some columns like ‘EmployeeCount’, ‘Over18’, ‘StandardHours’, ‘EmployeeNumber’ which only have 1 unique value so I dropped these columns from the dataset as they didn’t provide any helpful insight and variety to the data.

After dropping the unnecessary columns, I separated the categorical and continuous data from the dataset. As there are 30+ columns it will be easier to study and analyze the categorical and continuous data separately.

Let us first study the continuous data, on using the describe() function on it, the first thing to notice was that all the columns have the count of 1470, hence there are no missing values present in the continuous data. Most of the employees are around the age of 36 with a working experience of 10 years apart from this, they have been in this company for around 5 years and have been recently promoted in the past 2-3 years. While of course exceptions are present in this, with the maximum employee age registered is 60 years with work experience of 40 years. These exceptions also indicate the presence of skewness in the data, even though the age column is perfectly bell-shaped but the same cannot be said about other columns.



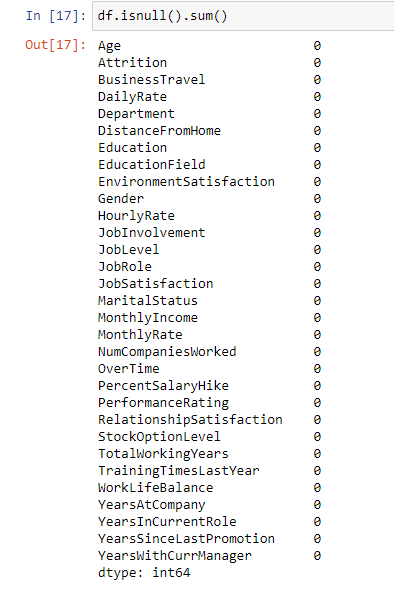
Another observation that I made from the data is the monthly income of employees, most of the employees earn around 4.7 - 5k per month (currency wasn’t mentioned anywhere). While the highest-earning employee earns 20k per month, which is 4 times the average income of other employees! Of course, this may be due to them being on higher posts and with more work experience.



Now let’s study the categorical data before we jump to bivariate analysis and the relationship between all the columns and attrition.

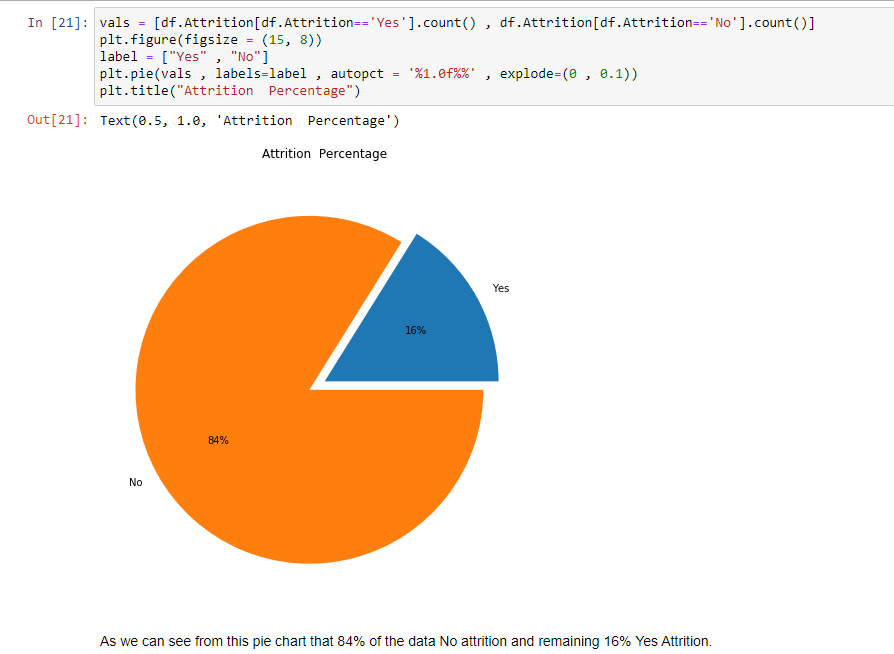
First, let us check if there are any missing values present in the categorical data. We already know that there are no missing values present in the continuous data as we made that observation from the describe () function.

It seems that there are no missing values present in the dataset whatsoever, as there are no missing values in the categorical data as well.



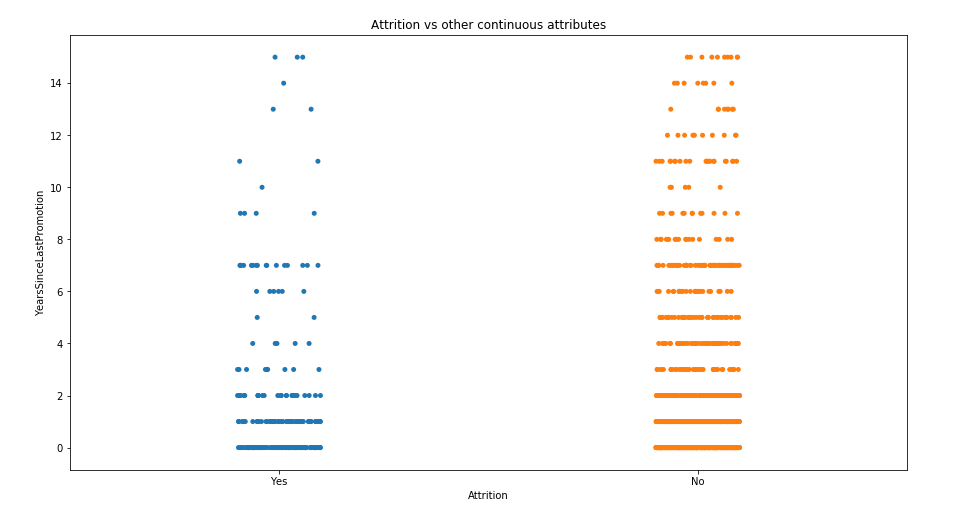
Employees rarely travel due to business and most of the employees are deployed in the Research & Development department with Life Science and Medical being the core fields of study. Most of the employees have their education level till Bachelors followed by Masters.

It seems to be a good work environment as most of the employees have rated '3' or higher in different fields like Environment Satisfaction, Job involvement, Job Satisfaction, Performance Rating, Relationship Satisfaction, etc. This may be the reason why the attrition rate is so low at only 16%.

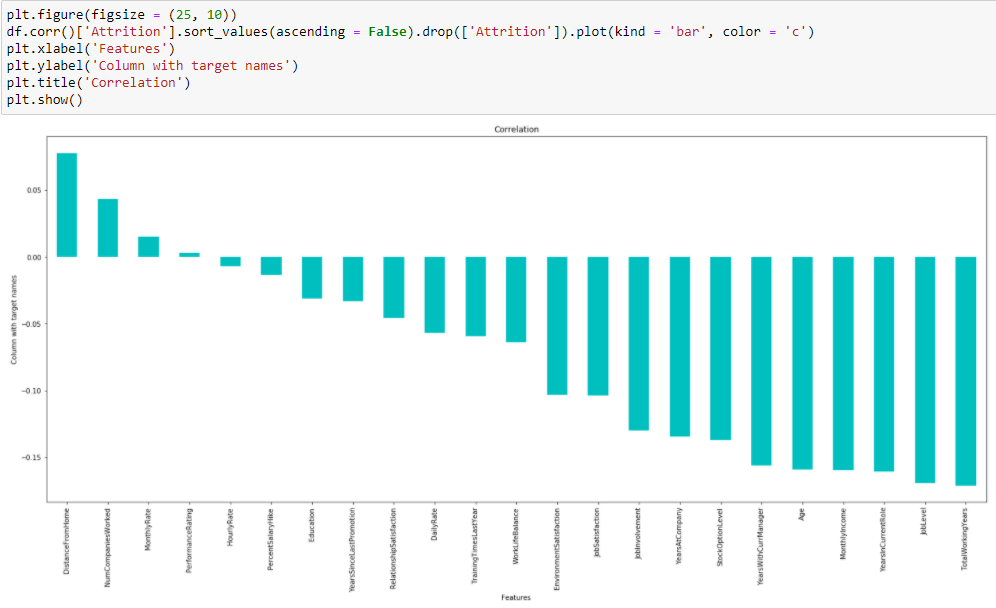


On bivariate analysis, I noticed that most of the attrition that took place was with the employees with low monthly income, low salary hike, low work experience, and years with the company.

Another surprising observation was that the attrition rate is higher with employees who recently got a promotion.



Now let us take a look at the correlation between the columns and attrition.



Most of the features are negatively correlated with attrition, this is due to 84% of attrition value being No. Total working years seems to have the highest negative correlation with attrition at -0.17, while there are others like job level, years in the current role, monthly income, age, years with current manager, etc which also seems to have a very high negative correlation with attrition.

# EDA Concluding Remark

In conclusion, the attrition rate of this organization is fairly low, this is due to high satisfaction levels of employees in different fields like work environment, work-life balance, job satisfaction, etc. The only reason for attrition even though it is fairly low is due to low monthly income, inexperience, and low years with the company.

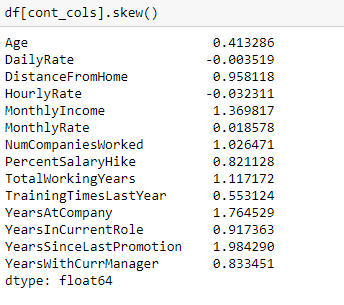
# Outliers and Skewness

We already sensed the presence of outliers and skewness in the data when we were analyzing the continuous data, let us further take a look at it and see if they need any treatment or not. For checking outliers we will use boxplot, a simple for loop over the continuous column will fetch us the box plot for all the columns separately for us to analyze.



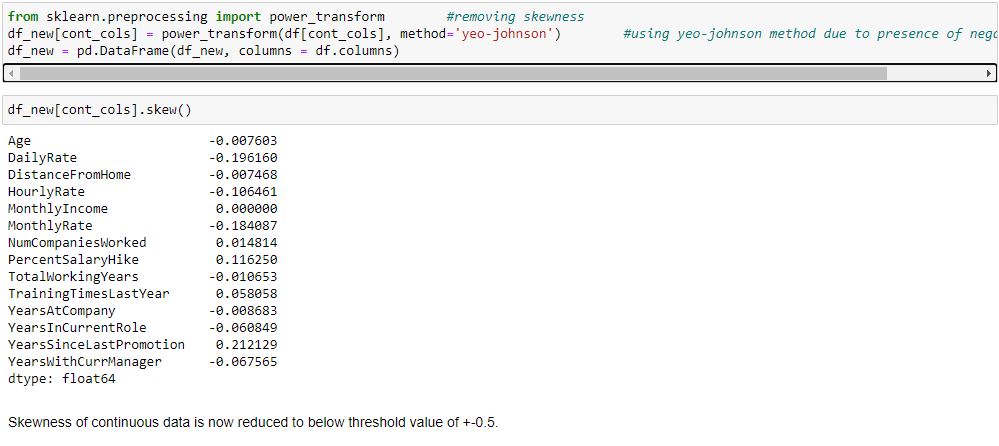
Surprisingly there are not many outliers present in the data apart from the 2-3 columns that we discussed earlier like monthly income, total working years, total years at the company, etc. There isn't any need for treatment of these outliers as they are not too fishy and are actually relevant to the data.

The next thing we’ll do is take a look at the skewness of the data.



The threshold value for acceptable skewness that I usually work with is +-0.5 anything higher than 0.5 or lower than -0.5 gets treated. And as we can see most of the columns have skewness higher than the threshold value.

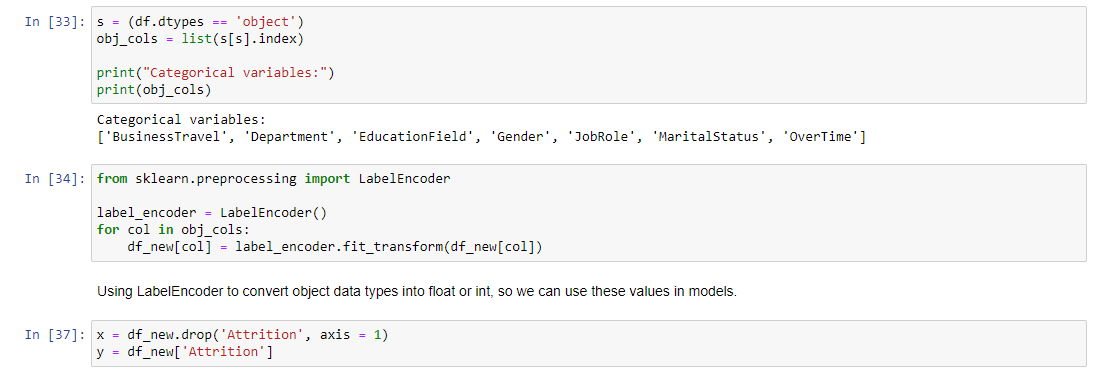
There are many ways to treat skewness, you can either use log transformation or you can use square root transformation or any other method. In this case, I used power transform to treat the skewness. Another thing, I used the ‘yeo - johnson’ method because there were negative values for skewness present in the data, hence I could not use the ‘box - cox’ method since it only works with strictly positive data. For more information and usage of power transform, you can read its documentation on scikit-learn.org.



As we can see now the skewness for all the columns is below the threshold value of 0.5.

# Building Machine Learning Models

Now that we have analyzed the data, cleaned it to remove skewness and outliers, it is time for us to build the model. But before building the model we will label encode the object data types and separate the independent and dependent variables.

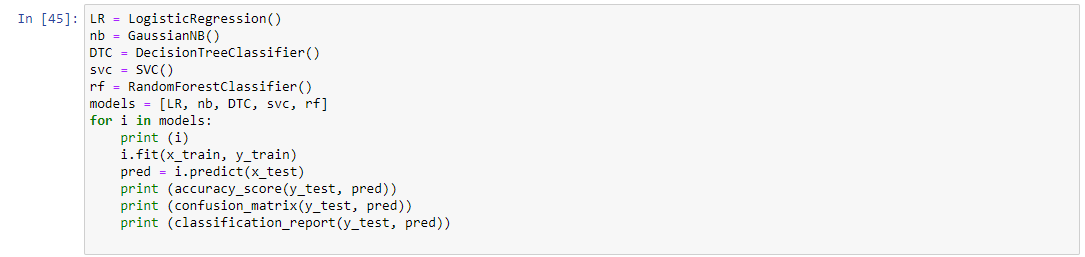


Now we will split the data set into train and test so we can build our model and train and test it. Before that I usually use a for loop to find the best random state for the train test split function, it is totally optional but it sort of gives the sense of the accuracy of the data.



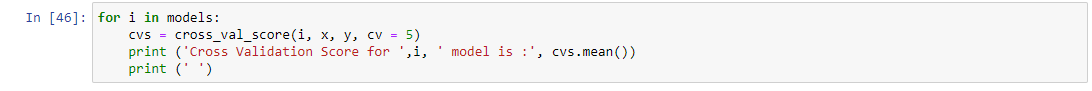
After finding the best random state we’ll create training and testing datasets and use other models like GaussianNB, Decision Tree, Support Vector, and Random Forest to find the best model for our dataset. For evaluation of our models, we will use accuracy score, classification report, and confusion matrix. Instead of writing separate codes for all these models, I usually use a for loop over these models and use a single block of code to find the best model.





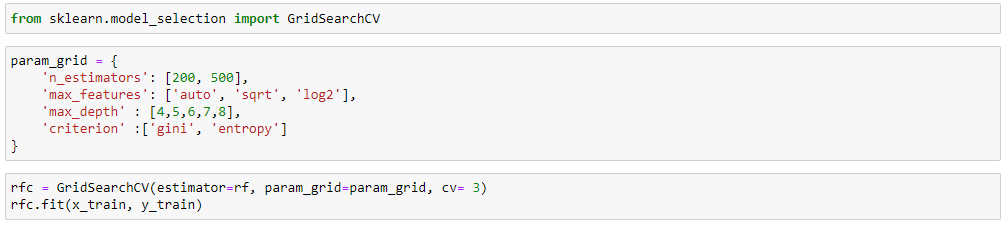
The Logistic Regression model had the highest accuracy as we saw above of 0.92, while accuracy for GaussianNB, Decision Tree, Support Vector, and Random Forest was 0.84, 0.77, 0.88, and 0.90 respectively.

Before we go on and save the logistic regression model, we will check for overfitting by using cross\_val\_score. Again for this, I’ve used a for loop over all the models and used a single block of code to check cross val score for all the models.

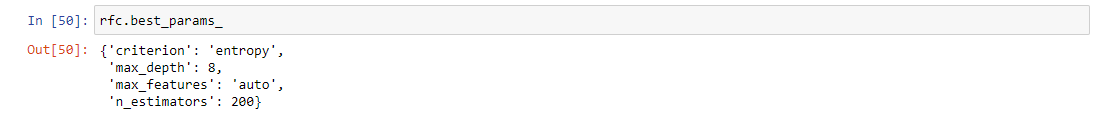


The cross val score for the logistic regression model was 0.87 while the score for the random forest model was 0.85. Even though both the differences are 0.05 but I preferred to use the random forest model since the difference was slightly lower and there is still some chance of slightly improving the model accuracy after hypertuning.

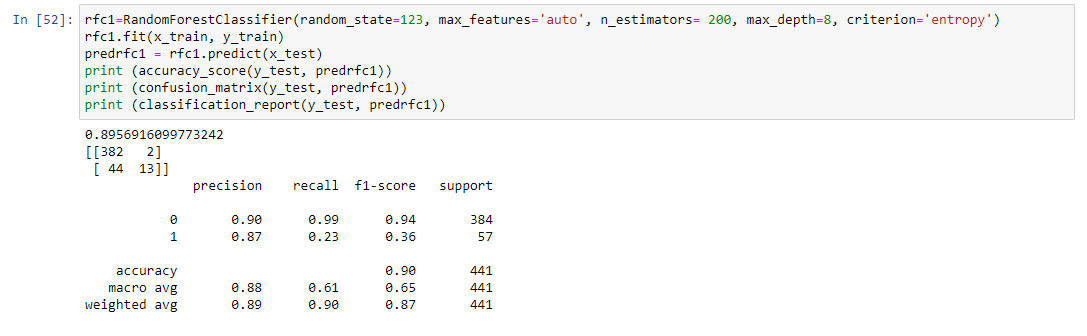
For hypertuning we used GridSearchCV from sklearn.model\_selection. Hypertuning is a time-consuming method with time-varying from around 10-15 minutes to over 4-5 hours, all depending on the size of the dataset, number of parameters passed in param\_grid, and the value of cv. This happens because in gridsearchcv the model passes a different combination of parameters listed in param\_grid into your selected model over the dataset which further gets cross-validated based on the value of cv.



After hypertuning, we use best\_params to get the best parameters for our model and rebuild our model with these parameters.



Now fitting these parameters into our new random forest model.



After hypertuning with gridsearchcv our model accuracy remained the same. A bit anticlimactic end to all this waiting time, to be honest, still an accuracy score of 0.90 is really good and there wasn't much room for improvement. Time to save our model using pickle and move on.



# Conclusion

While working on this model we learned about attrition and how employee data like their work experience, income, years with the company, and their review of the company affect the attrition rate in the organization. It is usually low-income employees with few years of experience and low years with the company who prefers to leave the company for new ventures while high experienced and employees who have served the company for a longer duration are much treasured by the organization and they also repay the organizations trust by staying loyal to the company.