**HR Analytics Project- Understanding the Attrition in HR**

**Problem Definition:**

Every year a lot of companies hire a number of employees. The companies invest time and money in training those employees, not just this but there are training programs within the companies for their existing employees as well. The aim of these programs is to increase the effectiveness of their employees.

Human resource analytics (HR analytics) is an area in the field of analytics that refers to applying analytic processes to the human resource department of an organization in the hope of improving employee performance and therefore getting a better return on investment. HR analytics does not just deal with gathering data on employee efficiency. Instead, it aims to provide insight into each process by gathering data and then using it to make relevant decisions about how to improve these processes.

HR Analytics is the method of collecting and evaluating Human Resource data to improve an organization’s culture, workers performance, retention etc.

**Attrition in HR:**

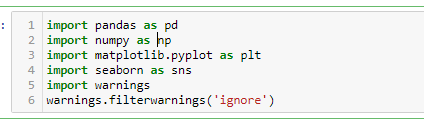
Attrition in human resources refers to the gradual loss of employees overtime. In general, relatively high attrition is problematic for companies. HR professionals often assume a leadership role in designing company compensation programs, work culture, and motivation systems that help the organization retain top employees.

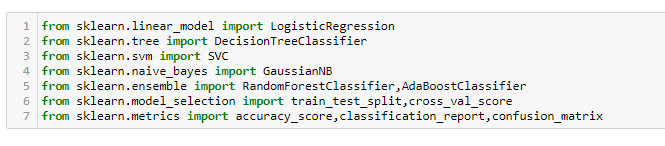
**Attrition affecting Companies:**

A major problem in high employee attrition is its cost to an organization. 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.

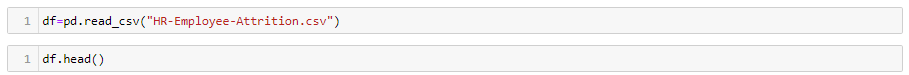
**Data Analysis:**

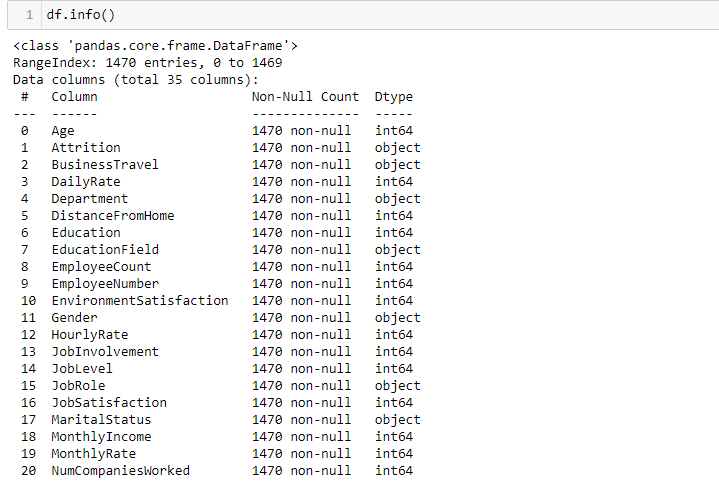
In Data analysis, we have check data types, missing values, and many more things.so let’s do it one by one.

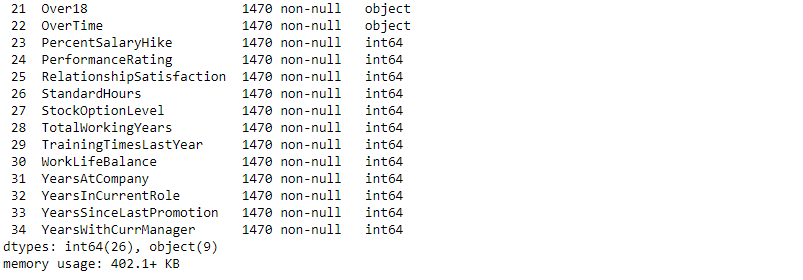
First importing all necessary libraries.



**Get dataset:**

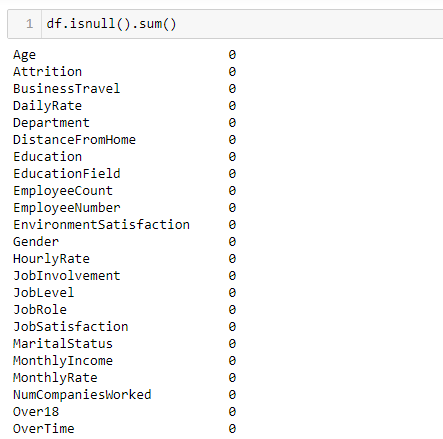
**Short Info about data:**

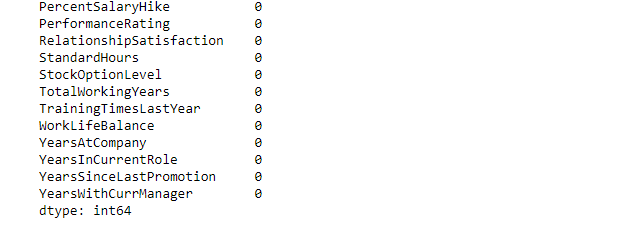




**The data set has 1470 examples and 34 features with the target variable**. 26 of the features are int, 9 are objects.

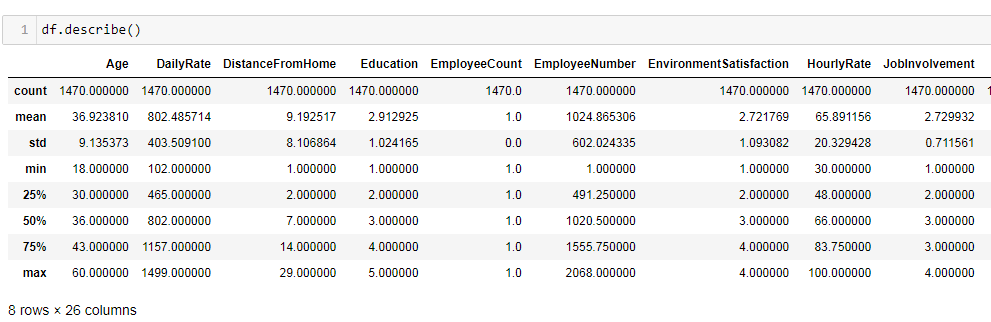
**Checking missing values**:

Now there is need to check whether data set have missing values or not. 

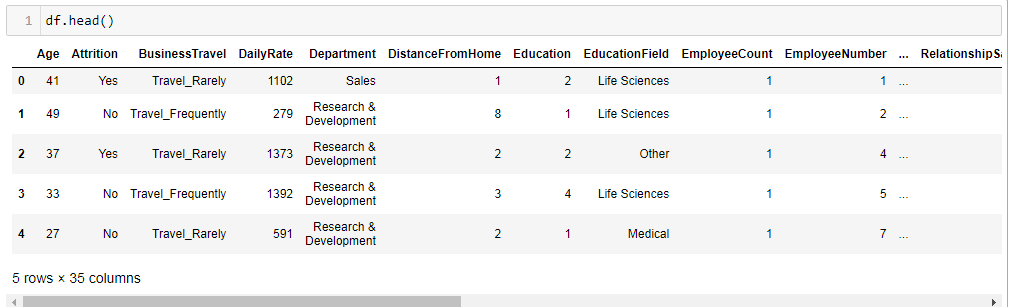


Our data set don’t have any missing value.

**Stastical summary:**



This function gives much information about data set such as count function tells that there are no missing values. Also it provides mean, median, max values from which we can derive many conclusions

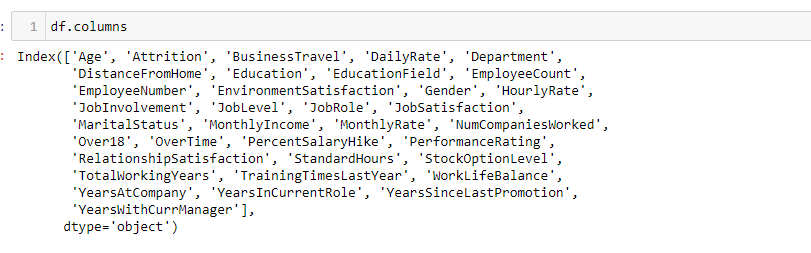


From the table above, we can say that there are many features which are in categorical form and we need to **convert**

**that features into numeric** ones later on, so that the machine

Learning algorithms can process them.

**Lets look total columns into data set:**



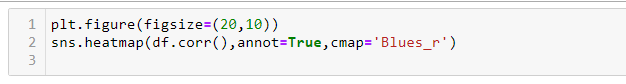
**Checking correlation**:

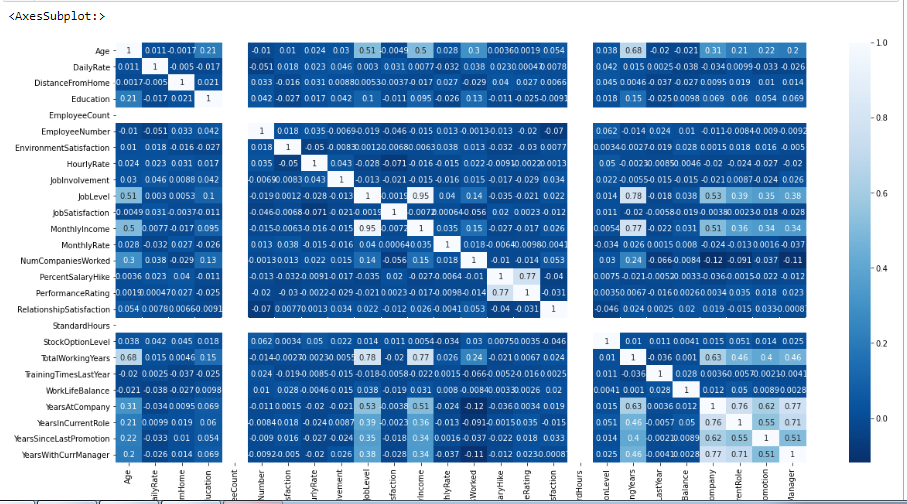
Correlation function gives information about in which manner features are correlated with each other.There may be

strong positive corelation or negative corelation between

variables.From this we can understand which features are

strongly corelated to target variable.



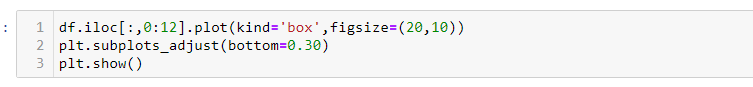


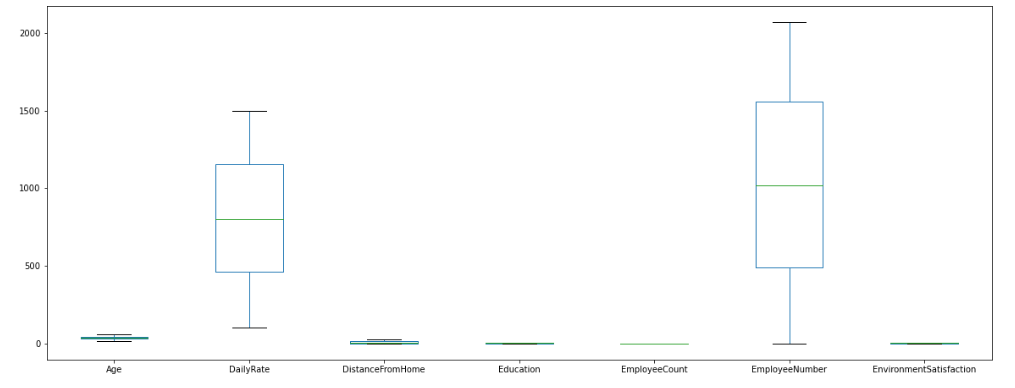
Above heatmap shows the correlation between variables.lighter shades describes positive corelation and negative

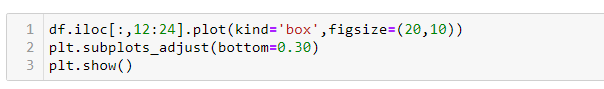
corelation shown by darker shade.

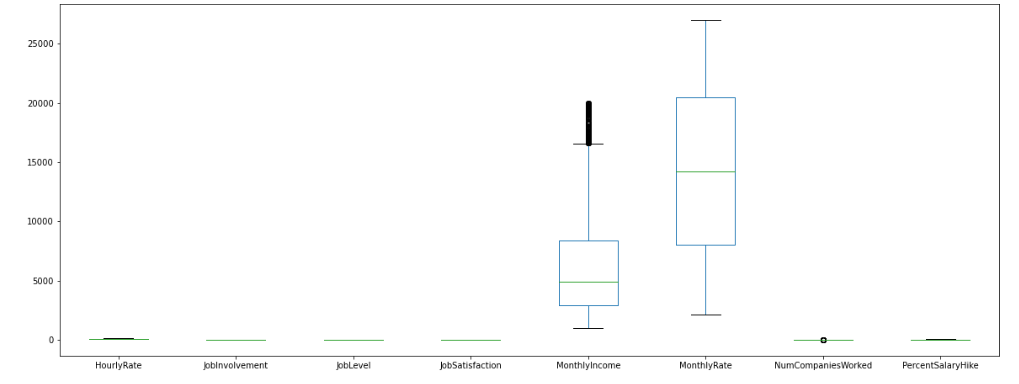
**Outliers detection:**

In this data set some outliers present. But this data set has categorical data so we can’t remove it

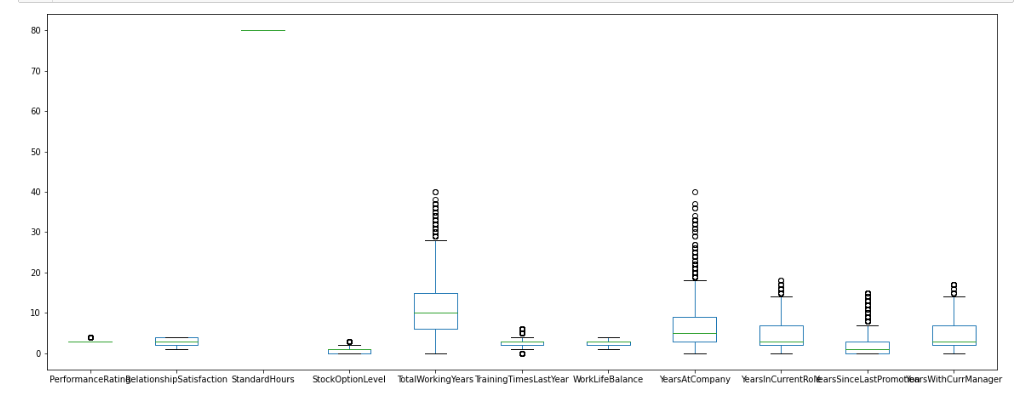






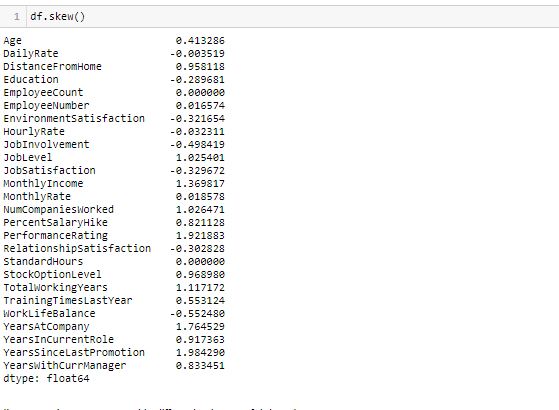




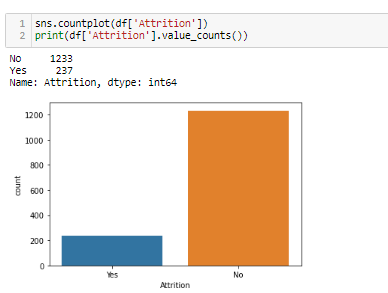


**Skewness in data set:**

If skew of data columns is above 0.5 and less than -0.5 then we have to reduce it. But if data is categorical then we can’t remove.

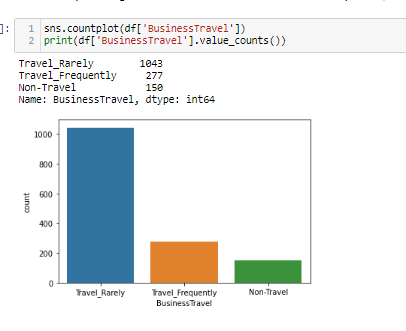
****

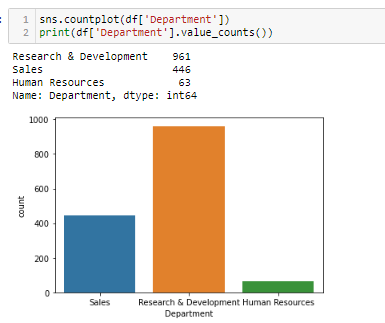
**Visualization of categorical data:**

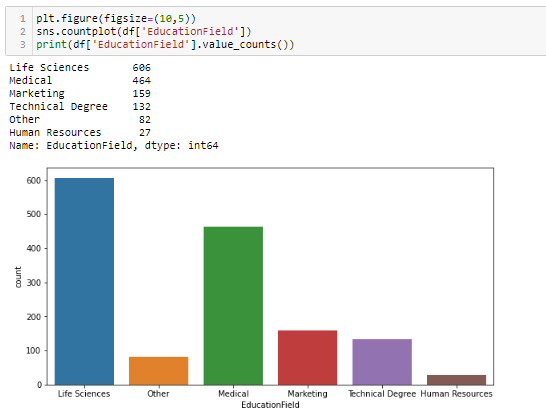


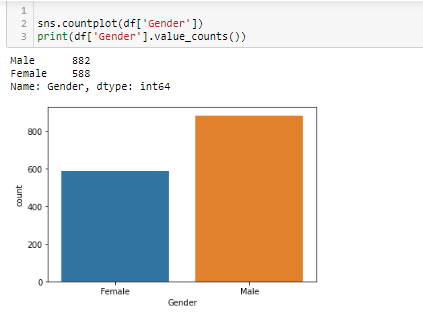
Above countplot of target variable shows that there is class imbalance problem that we need to handle it later on.

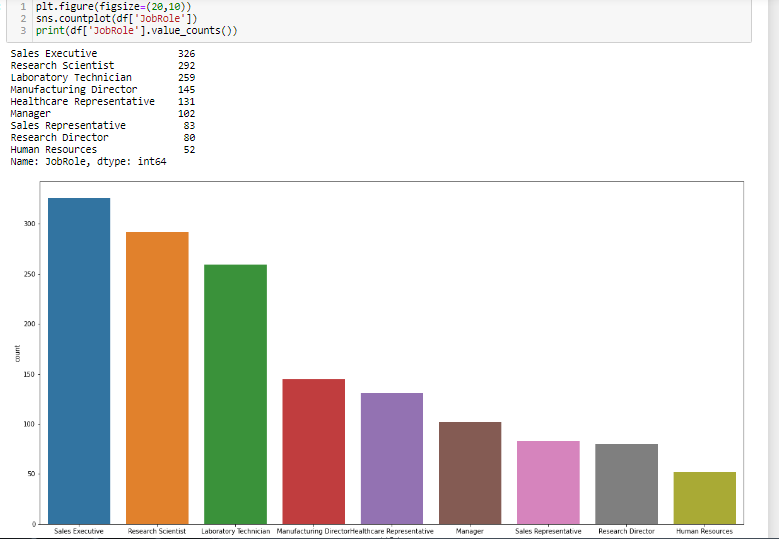
Following are countplots of input variables which are categorical in nature.

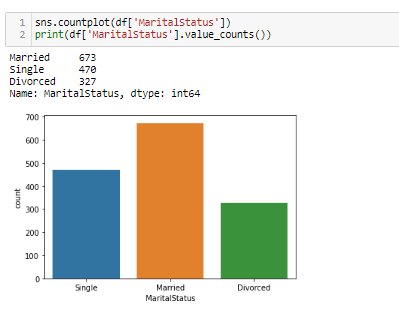


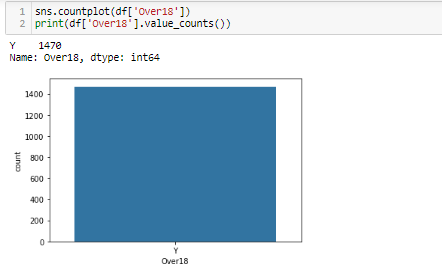


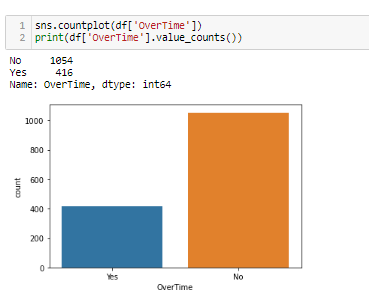












**Pre-Processing:**

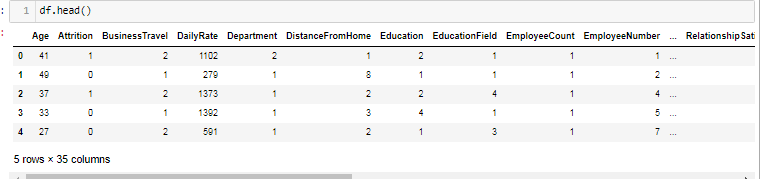
Encoding: First we have to encode the categorical data into numerical data. There are different techniques of encoding:

* One Hot Encoder: Encode categorical integer features using a one-hot aka one-of-K scheme. The input to this transformer should be a matrix of integers, denoting the values taken on by categorical (discrete) features. The output will be a sparse matrix where each column corresponds to one possible value of one feature.
* Label Encoder:Label Encoding refers to converting the labels into numeric form so as to convert it into the machine-readable form. Machine learning algorithms can then decide in a better way on how those labels must be operated. It is an important pre-processing step for the structured dataset in supervised learning.
* OrdinalEncoder: In ordinal encoding, each unique category value is assigned an integer value. For example, “red” is 1, “green” is 2, and “blue” is 3. This is called an ordinal encoding or an integer encoding and is easily reversible. Often, integer values starting at zero are used.

In this project Label Encoder is used to encoder the categorical data into numerical data.

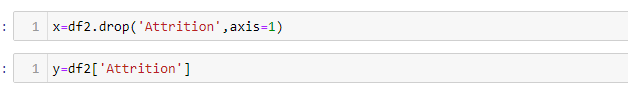


After performing encoding all categorical data transform into numerical form as below:



After encoding ,we have to separate input features and target

feature.



Above fig shows x variable contains all input variable and y variable contains target variable.

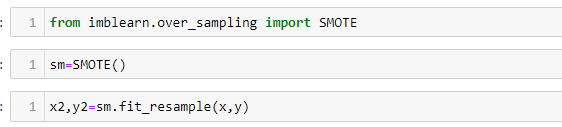
**Class Imbalance Problem:**

It is the problem in machine learning where the total number of a class of data (positive) is far less than the total number of another class of data (negative)

Our target variable is unbalanced means there are more values of ‘No’ than ‘yes’ as shown in above attritution count plot. So we have to handle it. Tis problem is solve using either by oversampling or undersampling.

* Over\_sampling: Random oversampling involves randomly selecting examples from the minority class, with replacement, and adding them to the training dataset. Random undersampling involves randomly selecting examples from the majority class and deleting them from the training dataset
* Under\_Sampling:Undersampling refers to a group of techniques designed to balance the class distribution for a classification dataset that has a skewed class distribution.

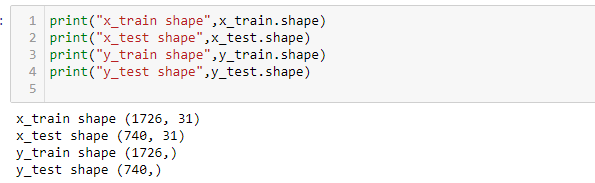
We solve this problem using oversampling method. For this we have to import necessary library



No lets divide data for training and testing

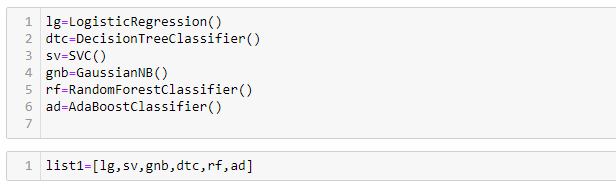


Shape of the training and testing data:

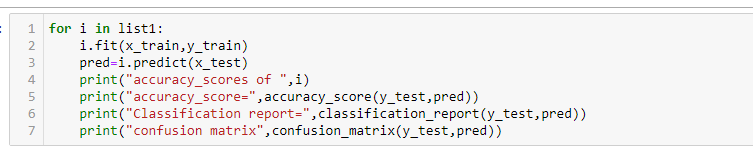


Now data is ready to fit into different models. As this is classification problem we have to use classification algorithms.

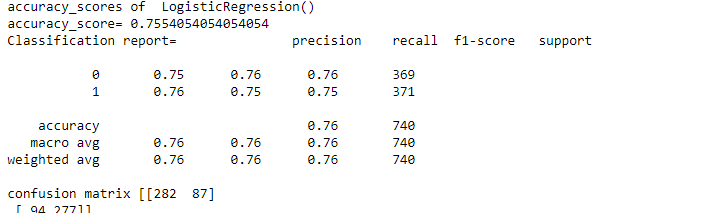
As shown above we have imported classification algorithm libraries, now we have to create instance of that.



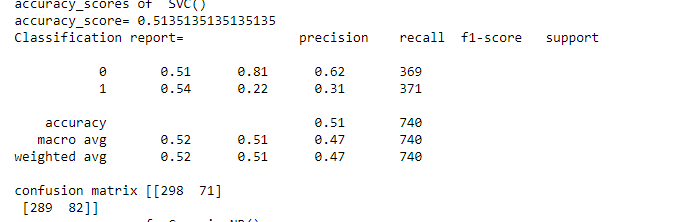
Now let’s fit data into models and predict values and derive accuracy scores:



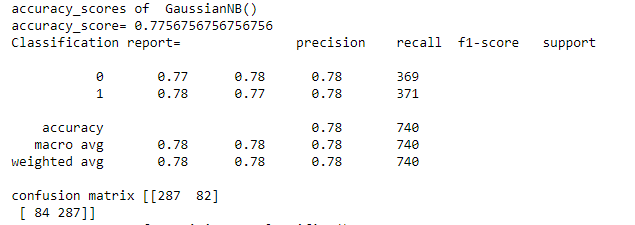
**Accuracy\_scores of LogisticRegression:**



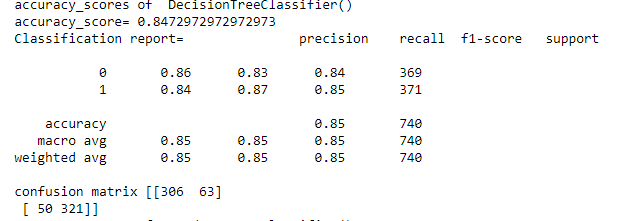
**Accuracy\_scores of Support Vector Machine**



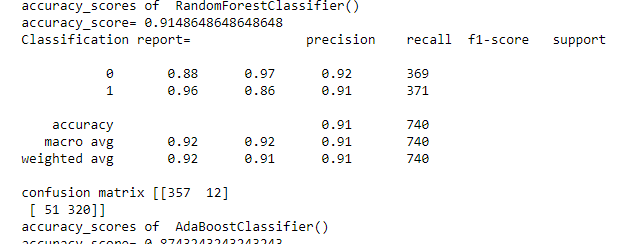
**Accuracy\_scores of GaussianNB:**



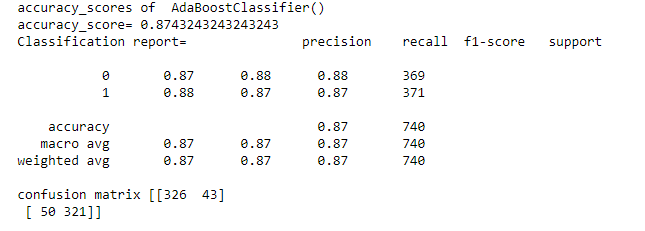
**Accuracy\_scores of DecisionTreeclassifier:**



**Accuracy\_scores of RandomforestClassifier**



**Accuracy\_scores AdaBoostClassifier:**



As we can see, the Random Forest classifier gives highest accuracy. But first, let us check, how random-forest performs, when we use cross validation.

# K-Fold Cross Validation:

Cross-validation is a resampling procedure used to evaluate machine learning models on a limited data sample.

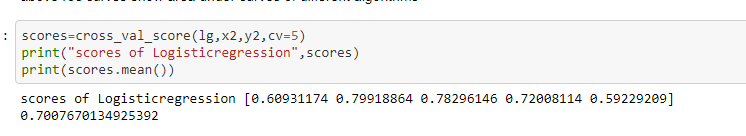
The procedure has a single parameter called k that refers to the number of groups that a given data sample is to be split into. As such, the procedure is often called k-fold cross-validation. When a specific value for k is chosen, it may be used in place of k in the reference to the model, such as k=10 becoming 10-fold cross-validation.

Cross-validation is primarily used in applied machine learning to estimate the skill of a machine learning model on unseen data. That is, to use a limited sample in order to estimate how the model is expected to perform in general when used to make predictions on data not used during the training of the model.

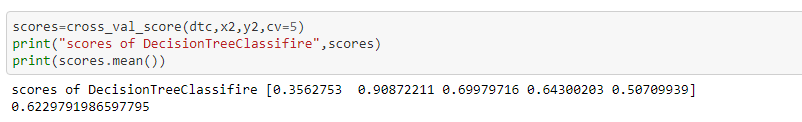
It is a popular method because it is simple to understand and because it generally results in a less biased or less optimistic estimate of the model skill than other methods, such as a simple train/test split.

Following are cross\_val\_scores of above classification algorithms:

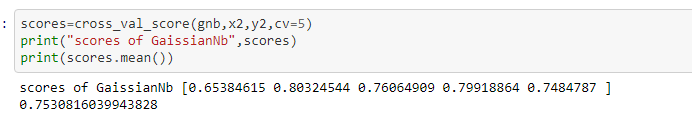
**Logistic Regression:**



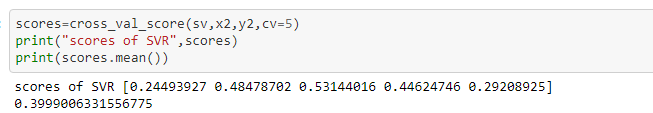
**DecisionTreeClassifier:**



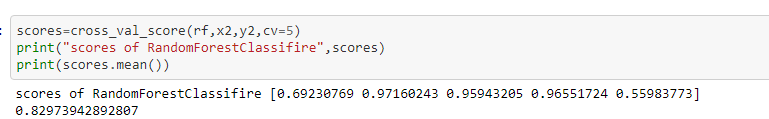
**GaussianNB**:



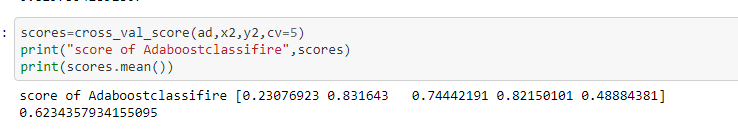
**Support Vector Machine**:



**RandomForestClassifier:**



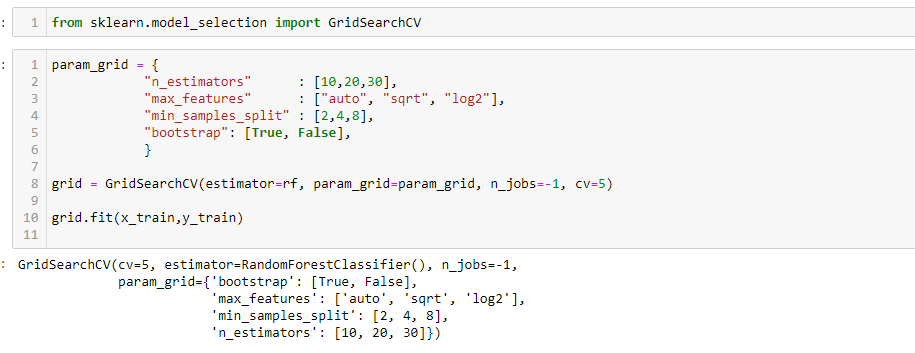
**AdaBoostClassifier:**

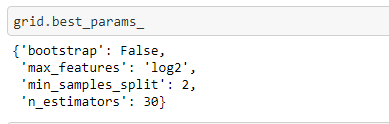


As we can see in above scores, we can conclude that Random Forest gives better performance than others.

**HyperParameterTunning:**

In [machine learning](https://en.wikipedia.org/wiki/Machine_learning), hyperparameter optimization or tuning is the problem of choosing a set of optimal [hyperparameters](https://en.wikipedia.org/wiki/Hyperparameter_(machine_learning)" \o "Hyperparameter (machine learning)) for a learning algorithm. A hyperparameter is a [parameter](https://en.wikipedia.org/wiki/Parameter) whose value is used to control the learning process. By contrast, the values of other parameters (typically node weights) are learned.The same kind of machine learning model can require different constraints, weights or learning rates to generalize different data patterns. These measures are called hyperparameters, and have to be tuned so that the model can optimally solve the machine learning problem. Hyperparameter optimization finds a tuple of hyperparameters that yields an optimal model which minimizes a predefined [loss function](https://en.wikipedia.org/wiki/Loss_function) on given independent data. The objective function takes a tuple of hyperparameters and returns the associated loss.

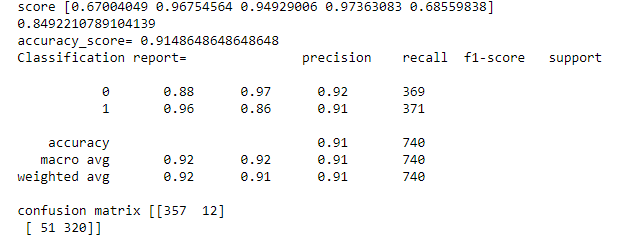
So now let’s do hyper parameter tuning for RandomForestclassifier 

**Derive best parameters**: 

Now again fit data into RandomForestClassifier using this parameters.

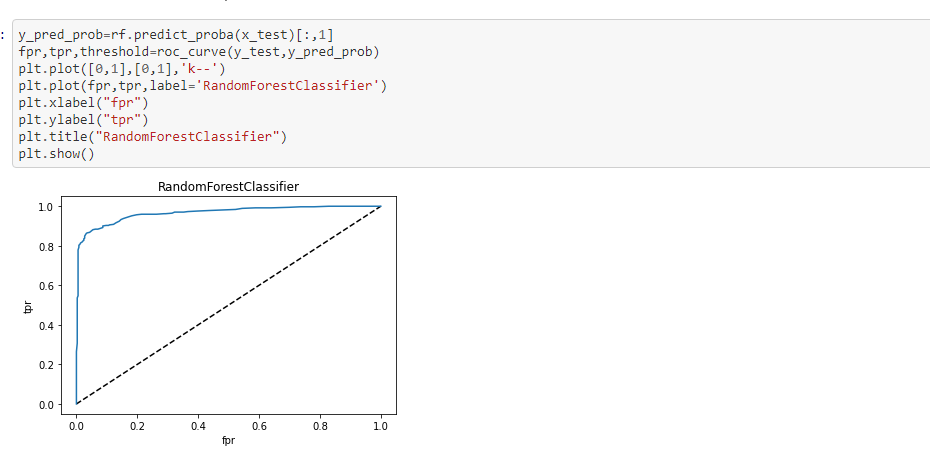


Accuracy\_scores:



**AOC-ROC Curve:**

AUC - ROC curve is a performance measurement for the classification problems at various threshold settings. ROC is a probability curve and AUC represents the degree or measure of separability. It tells how much the model is capable of distinguishing between classes. Higher the AUC, the better the model is at predicting 0 classes as 0 and 1 classes as 1



Above figure shows AOC\_ROC Curve of random forest Classifier.

**RandomForestClassifier**

Random Forest Classifier is ensemble algorithm.  **Ensembled algorithms** are those which combines more than one algorithms of same or different kind for classifying objects. For example, running prediction over Naive Bayes, SVM and Decision Tree and then taking vote for final consideration of class for test object.

One big advantage of random forest is, that it can be used for both classification and regression problems, which form the majority of current machine learning systems. With a few exceptions a random-forest classifier has all the hyperparameters of a decision-tree classifier and also all the hyperparameters of a bagging classifier, to control the ensemble itself.

The random-forest algorithm brings extra randomness into the model, when it is growing the trees. Instead of searching for the best feature while splitting a node, it searches for the best feature among a random subset of features. This process creates a wide diversity, which generally results in a better model. Therefore when you are growing a tree in random forest, only a random subset of the features is considered for splitting a node. You can even make trees more random, by using random thresholds on top of it, for each feature rather than searching for the best possible thresholds (like a normal decision tree does).

**Conclusion:**

Throughout this, we saw Data is important in Human Resource department. We used Random forest and learned how it can be very advantageous over other available machine learning algorithm. Most of all we found factors which are most important to employees and if are not fulfilled might lead to Attrition.