**Building an Employee Churn Model in Python to Develop a Strategic Retention Plan**

# Contents

1. [Problem Definition](https://towardsdatascience.com/building-an-employee-churn-model-in-python-to-develop-a-strategic-retention-plan-57d5bd882c2d#c099)
2. [Data Analysis](https://towardsdatascience.com/building-an-employee-churn-model-in-python-to-develop-a-strategic-retention-plan-57d5bd882c2d#6270)
3. [EDA Concluding Remarks](https://towardsdatascience.com/building-an-employee-churn-model-in-python-to-develop-a-strategic-retention-plan-57d5bd882c2d#030a)
4. [Pre-processing Pipeline](https://towardsdatascience.com/building-an-employee-churn-model-in-python-to-develop-a-strategic-retention-plan-57d5bd882c2d#dd40)
5. [Building Machine Learning Models](https://towardsdatascience.com/building-an-employee-churn-model-in-python-to-develop-a-strategic-retention-plan-57d5bd882c2d#8341)
6. [Concluding Remarks](https://towardsdatascience.com/building-an-employee-churn-model-in-python-to-develop-a-strategic-retention-plan-57d5bd882c2d#13e2)

# 1. Problem Definition

Employee churn is a costly problem for companies. The true cost of replacing an employee can often be quite large.

A study by the [Center for American Progress](https://www.americanprogress.org/wp-content/uploads/2012/11/CostofTurnover.pdf) found that companies typically pay about one-fifth of an employee’s salary to replace that employee, and the cost can significantly increase if executives or highest-paid employees are to be replaced.

In other words, the cost of replacing employees for most employers remains significant. This is due to the amount of time spent to interview and find a replacement, sign-on bonuses, and the loss of productivity for several months while the new employee gets accustomed to the new role.

Understanding why and when employees are most likely to leave can lead to actions to improve employee retention as well as possibly planning new hiring in advance. I will be using a step-by-step systematic approach using a method that could be used for a variety of ML problems. This project would fall under what is commonly known as **HR .**

**Analytics** or **People Analytics**-

In this study, we will attempt to solve the following problem statement is:

* What is the **likelihood** of an active employee leaving the company?
* What are the key **indicators** of an employee leaving the company?
* What **strategies** can be adopted based on the results to improve employee retention?

Given that we have data on former employees, this is a **standard supervised classification problem** where the label is a binary variable, 0 (active employee), 1 (former employee). In this study, our target variable Y is the probability of an employee leaving the company.

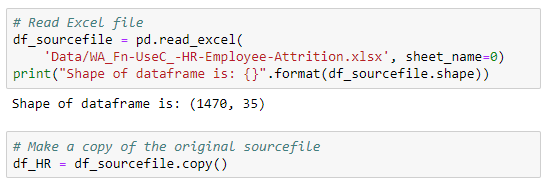
# 2. Data Analysis

In this case study, a HR dataset was sourced from [IBM HR Analytics Employee Attrition & Performance](https://www.ibm.com/communities/analytics/watson-analytics-blog/hr-employee-attrition/) which contains employee data for 1470 with various information about the employees. I will use this dataset to predict when employees are going to quit by understanding the main drivers of employee churn.

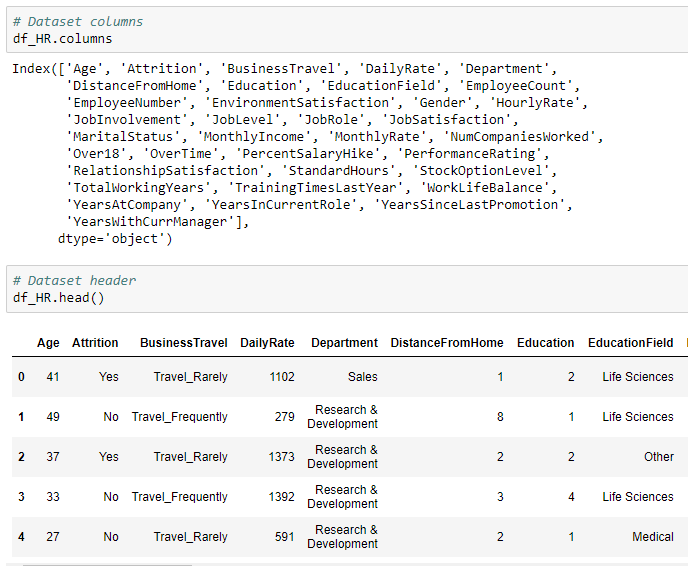
As stated on the [IBM website](https://www.ibm.com/communities/analytics/watson-analytics-blog/hr-employee-attrition/): “This is a fictional data set created by IBM data scientists. Its main purpose was to demonstrate the IBM Watson Analytics tool for employee attrition.”

## 2.1 Data Description and Exploratory Visualizations

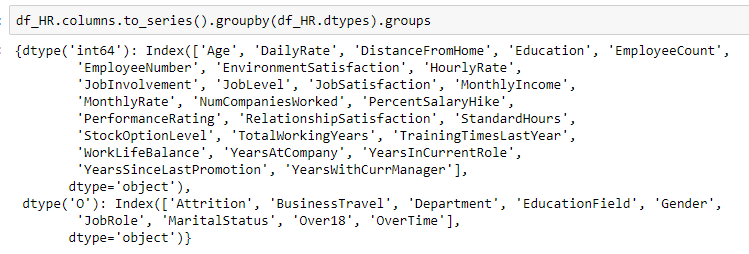
First, we import the dataset and make of a copy of the source file for this analysis. The dataset contains 1,470 rows and 35 columns.



The dataset contains several numerical and categorical columns providing various information on employee’s personal and employment details.



Let’s break down the columns by their type (i.e. int64, float64, object):



## 2.2 Data source

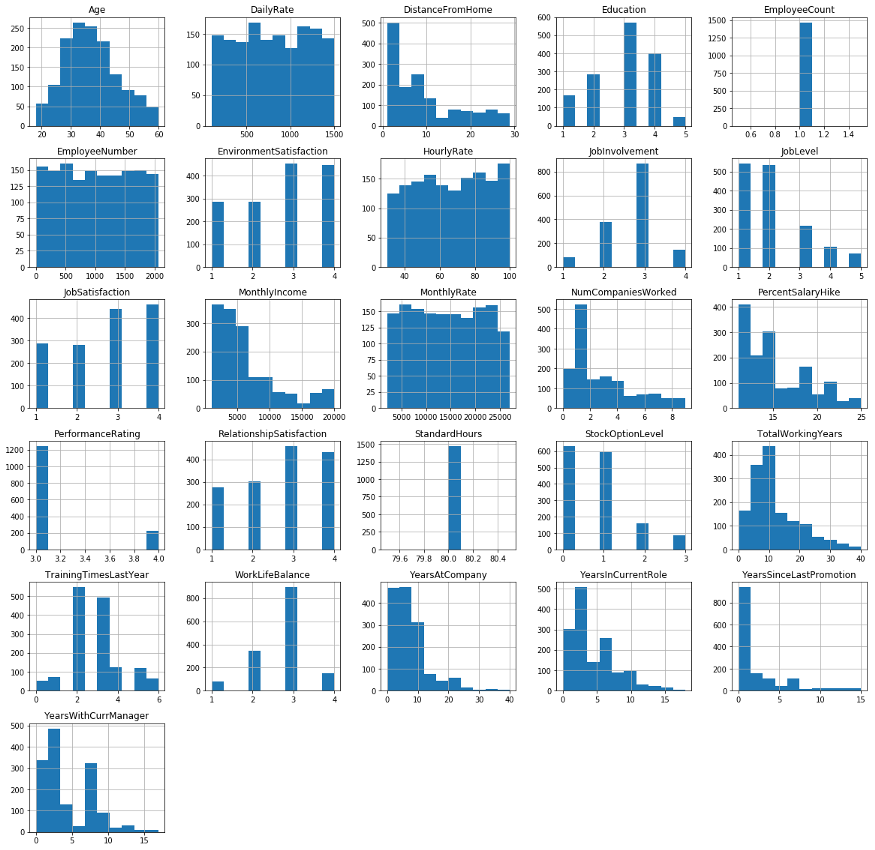
The data provided has no missing values. In HR Analytics, employee data is unlikely to feature large ratio of missing values as HR Departments typically have all personal and employment data on-file.

However, the type of documentation data is being kept in (i.e. whether it is paper-based, Excel spreadsheets, databases, etc) has a massive impact on the accuracy and the ease of access to the HR data.

## 2.3 Numerical features overview

A few observations can be made based on the information and histograms for numerical features:

* Several numerical features are tail-heavy; indeed several distributions are right-skewed (e.g. MonthlyIncome DistanceFromHome, YearsAtCompany). Data transformation methods may be required to approach a normal distribution prior to fitting a model to the data.
* Age distribution is a slightly right-skewed normal distribution with the bulk of the staff between 25 and 45 years old.
* EmployeeCount and StandardHours are constant values for all employees. They’re likely to be redundant features.
* Employee Number is likely to be a unique identifier for employees given the feature’s quasi-uniform distribution.



Source code: df\_HR.hist () — isn’t Python a beautiful thing?

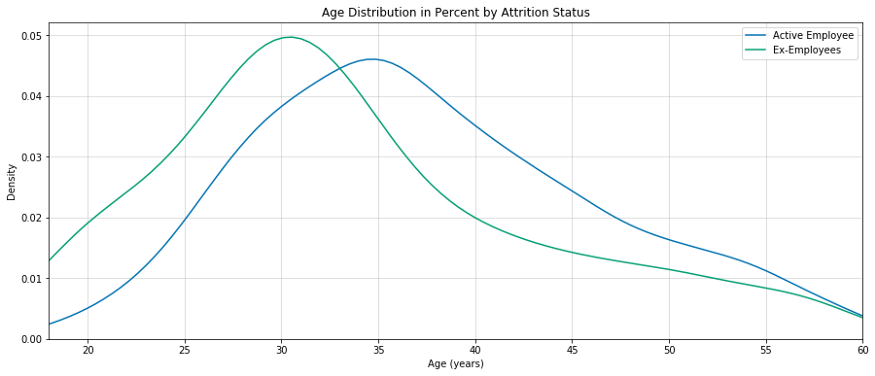
## 2.4 Feature distribution by target attribute

In this section, a more details Exploratory Data Analysis is performed.

**2.4.1 Age**

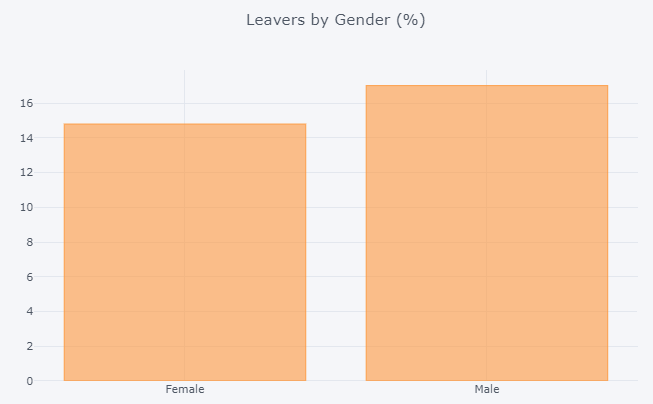
The age distributions for Active and Ex-employees only differ by one year; with the average age of ex-employees at 33.6 years old and 37.6 years old for current employees.

Let’s create a kernel density estimation (KDE) plot colored by the value of the target. A kernel density estimation (KDE) is a non-parametric way to estimate the probability density function of a random variable.



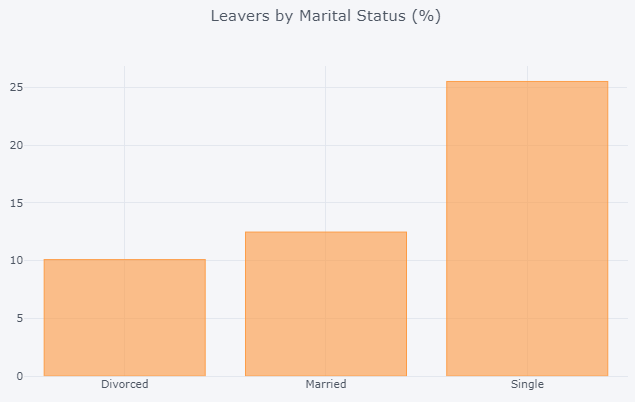
**2.4.2 Gender**

Gender distribution shows that the dataset features a higher relative proportion of male ex-employees than female ex-employees, with normalised gender distribution of ex-employees in the dataset at 17.0% for Males and 14.8% for Females.



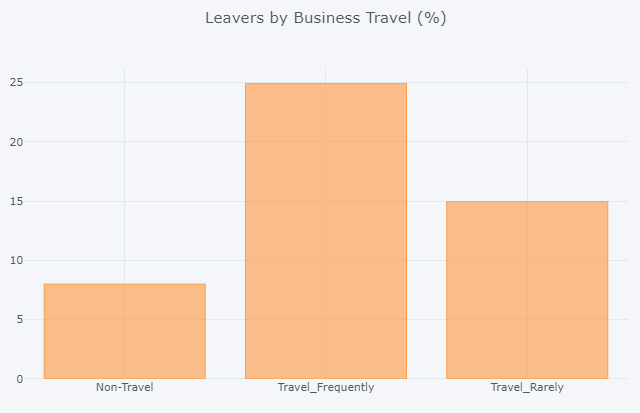
**2.4.3 Marital Status**

The dataset features three marital status: Married (673 employees), Single (470 employees), Divorced (327 employees). Single employees show the largest proportion of leavers at 25%.

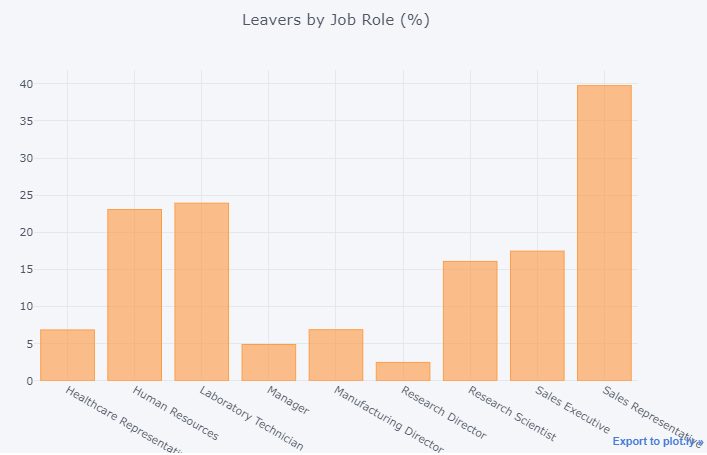


**2.4.4 Role and Work Conditions**

A preliminary look at the relationship between Business Travel frequency and Attrition Status shows that there is a largest normalized proportion of Leavers for employees that travel “frequently”. Travel metrics associated with Business Travel status were not disclosed (i.e. how many hours of Travel is considered “Frequent”).

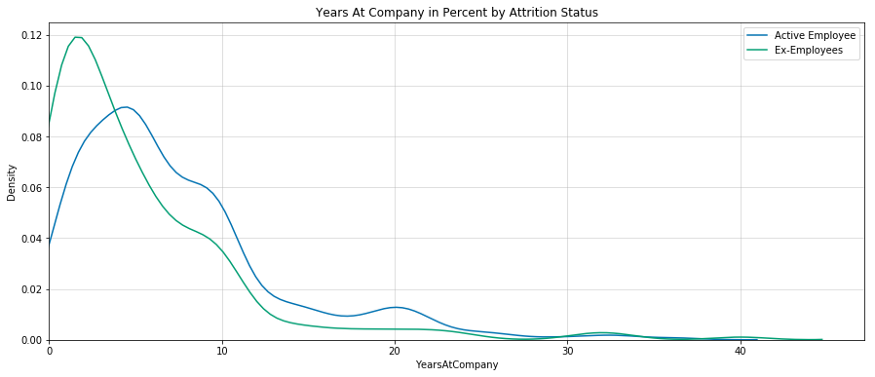


Several Job Roles are listed in the dataset: Sales Executive, Research Scientist, Laboratory Technician, Manufacturing Director, Healthcare Representative, Manager, Sales Representative, Research Director, Human Resources.



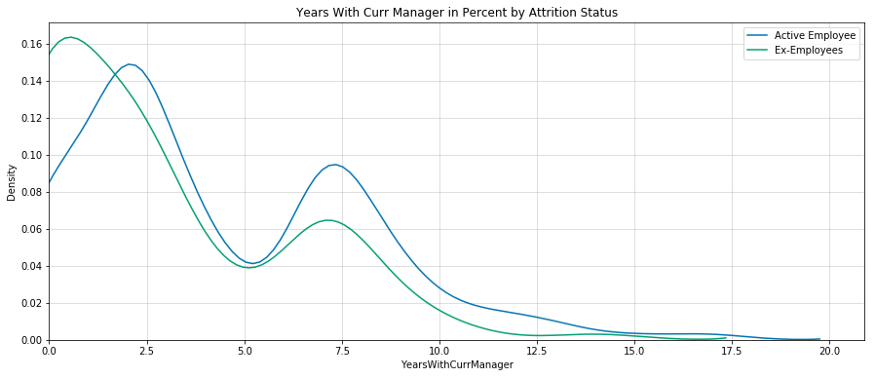
**2.4.5 Years at the Company and Since Last Promotion**

The average number of years at the company for currently active employees is 7.37 years and ex-employees is 5.13 years.



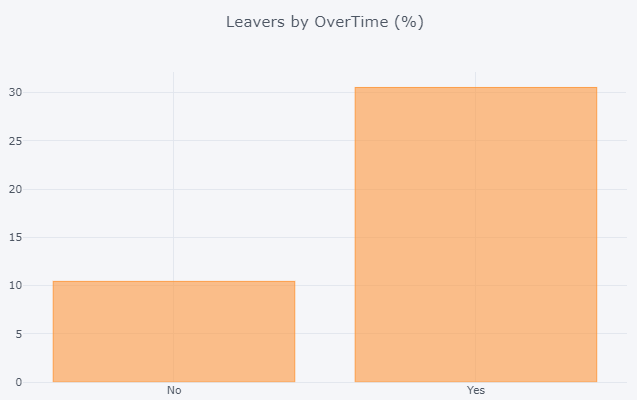
**2.4.6 Years with Current Manager**

The average number of year’s wit current manager for currently active employees is 4.37 years and ex-employees is 2.85 years.



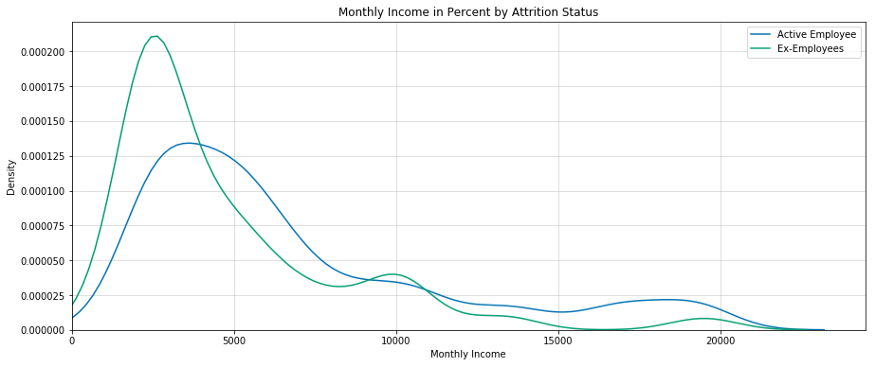
**2.4.7 Overtime**

Some employees have overtime commitments. The data clearly show that there is significant larger portion of employees with OT that have left the company.



**2.4.8 Monthly Income**

Employee Monthly Income varies from $1009 to $19999.

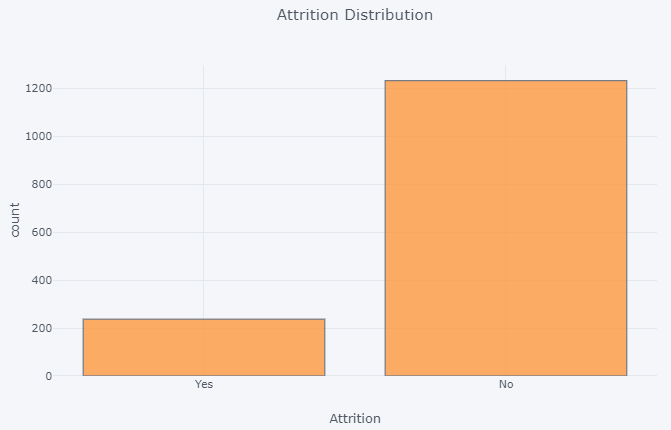


**2.4.9 Target Variable: Attrition**

The feature “**Attrition”** is what this Machine Learning problem is about. We are trying to predict the value of the feature ‘Attrition’ by using other related features associated with the employee’s personal and professional history.

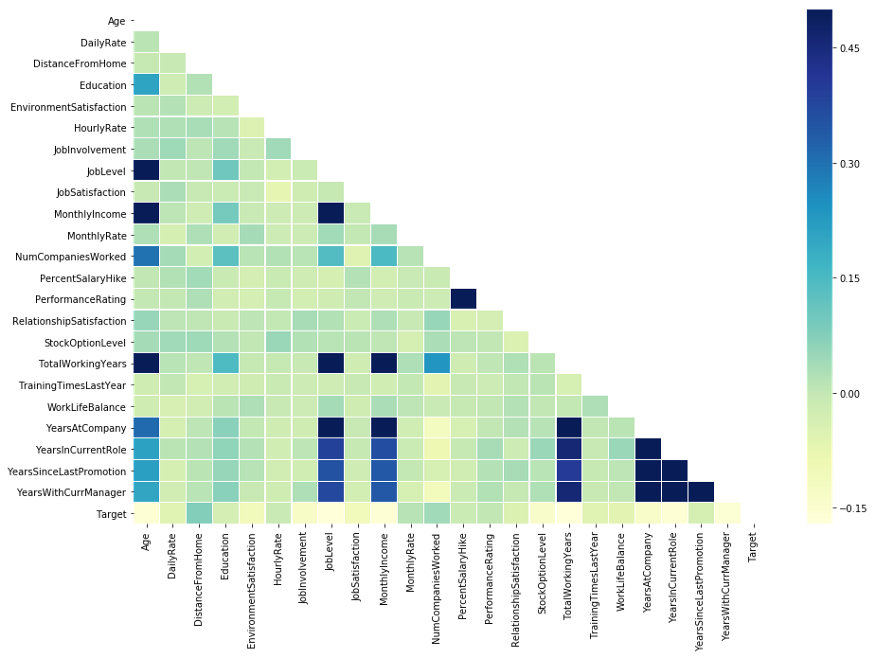
In the supplied dataset, the percentage of Current Employees is 83.9% and of Ex-employees is 16.1%. Hence, this is an **imbalanced class** problem.

Machine learning algorithms typically work best when the number of instances of each classes are roughly equal. We will have to address this target feature imbalance prior to implementing our Machine Learning algorithms.



## 2.5 Correlation

Let’s take a look at some of most significant correlations. It is worth remembering that correlation coefficients only measure linear correlations.



As shown above, “Monthly Rate”, “Number of Companies Worked” and “Distance From Home” are positively correlated to Attrition; while “Total Working Years”, “Job Level”, and “Years In Current Role” are negatively correlated to Attrition.

# 3. EDA Concluding Remarks

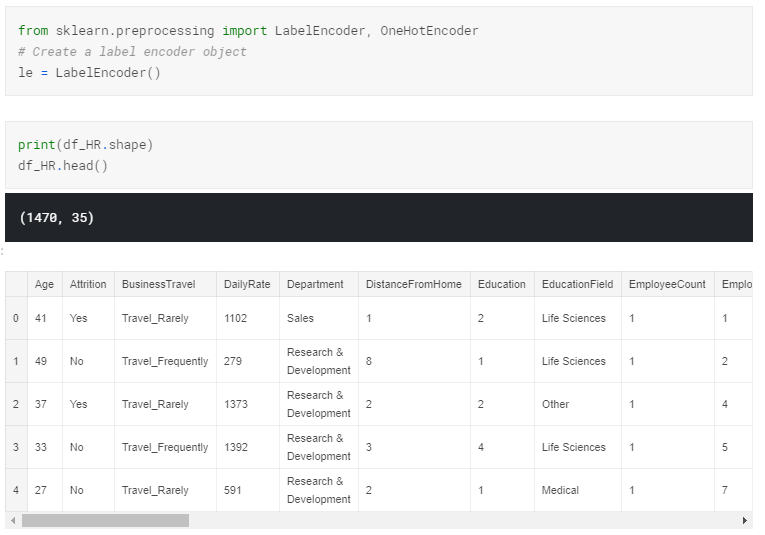
* The dataset does not feature any **missing or erroneous data values**, and all features are of the correct data type.
* The strongest **positive correlations** with the target features are: Performance Rating, Monthly Rate, Num Companies Worked, Distance From Home.
* The strongest **negative correlations** with the target features are: Total Working Years, Job Level, Years In Current Role, and Monthly Income.
* The dataset is **imbalanced**with the majority of observations describing Currently Active Employees.
* **Single employees** show the largest proportion of leavers, compared to Married and Divorced counterparts.
* About 10% of leavers left when they reach their **2-year anniversary** at the company.
* People who **live further away from their work** show higher proportion of leavers compared to their counterparts.
* People who **travel frequently** show higher proportion of leavers compared to their counterparts.
* People who have to work **overtime**show higher proportion of leavers compared to their counterparts.
* Employees that have already worked at several companies previously (already “bounced” between workplaces) show higher proportion of leavers compared to their counterparts.

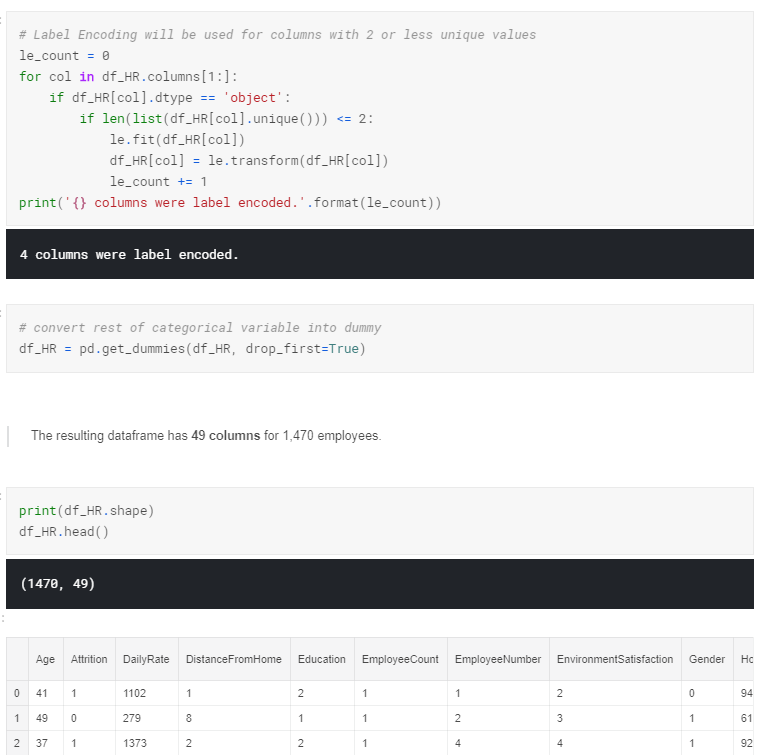
# 4. Pre-processing Pipeline

In this section, we undertake data pre-processing steps to prepare the datasets for Machine Learning algorithm implementation.

## 4.1 Encoding

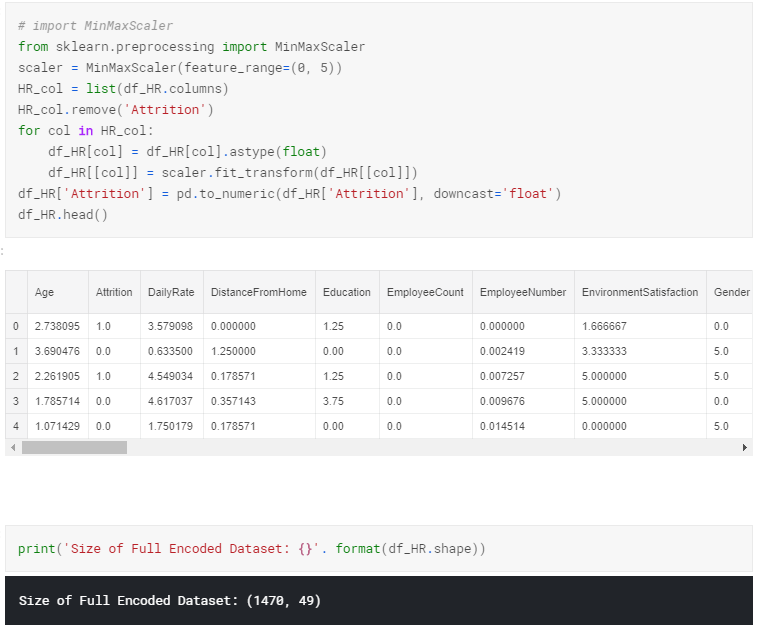
Machine Learning algorithms can typically only have numerical values as their predictor variables. Hence **Label Encoding** becomes necessary as they encode categorical labels with numerical values. To avoid introducing feature importance for categorical features with large numbers of unique values, we will use both **Label Encoding** and **One-Hot Encoding** as shown below.





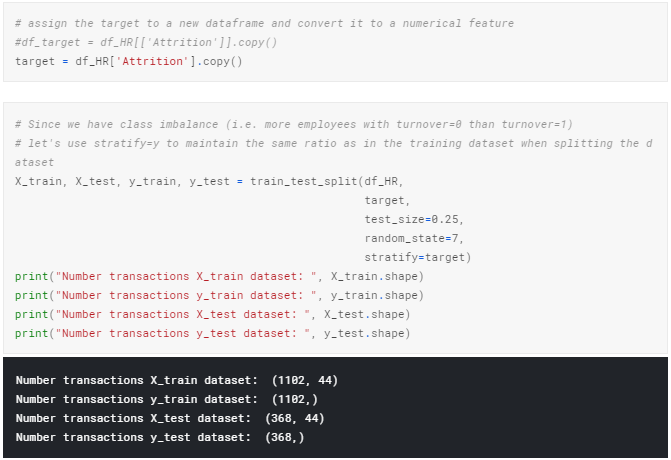
## 4.2 Feature Scaling

Feature Scaling using **MinMaxScaler**essentially shrinks the range such that the range is now between 0 and n. Machine Learning algorithms perform better when input numerical variables fall within a similar scale. In this case, we are scaling between 0 and 5.



## 4.3 Splitting data into training and testing sets

Prior to implementation or applying any Machine Learning algorithms, we must decouple training and testing dataframe from our master dataset.

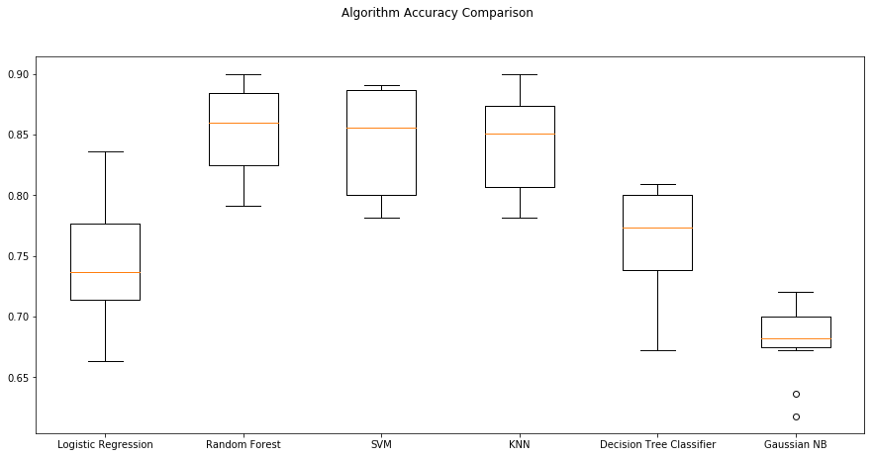


# 5. Building Machine Learning Models

## 5.1 Baseline Algorithms

Let’s first use a range of **baseline** algorithms (using out-of-the-box hyper-parameters) before we move on to more sophisticated solutions. The algorithms considered in this section are: **LogisticRegression**, **RandomForest**, **SVM**, **KNN**, **DecisionTreeClassifier**, **Gaussian NB.**

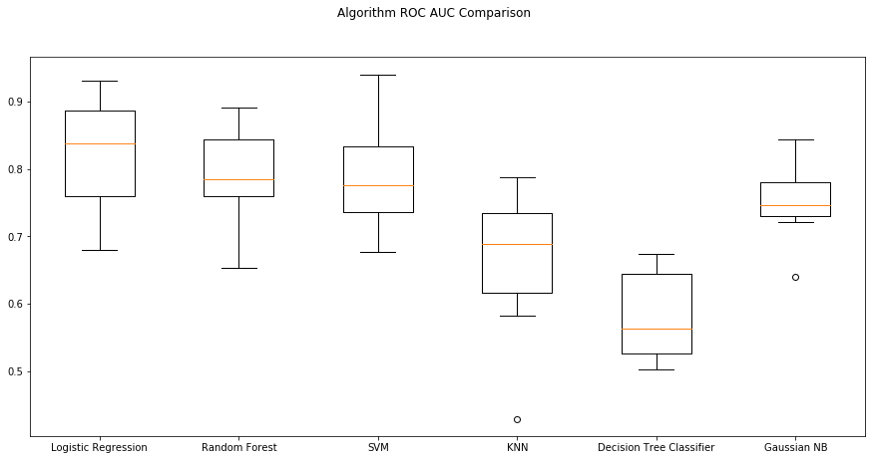
Let’s evaluate each model in turn and provide **accuracy**and **standard deviation scores**..



**Classification Accuracy** is the number of correct predictions made as a ratio of all predictions made. It is the most common evaluation metric for classification problems.

However, it is often **misused** as it is only really suitable when there are an **equal number of observations in each class** and all predictions and prediction errors are equally important. It is not the case in this project, so a different scoring metric may be more suitable.

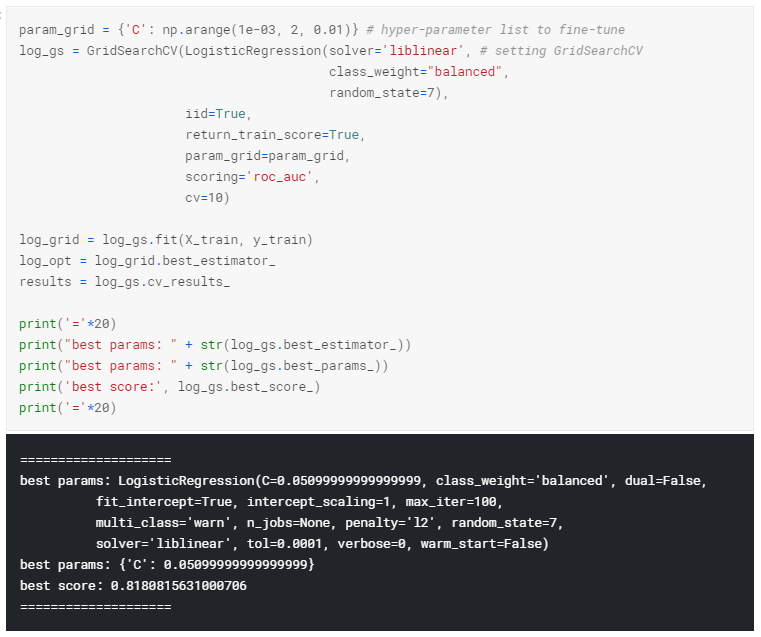
**Area under ROC Curve** (or AUC for short) is a performance metric for binary classification problems. The AUC represents a **model’s ability to discriminate between positive and negative classes**, and is better suited to this project. An area of 1.0 represents a model that made all predictions perfectly. An area of 0.5 represents a model as good as random.



Based on our ROC AUC comparison analysis, **Logistic Regression** and **Random Forest** show the highest mean AUC scores. We will shortlist these two algorithms for further analysis. See below for more details on these two algos.

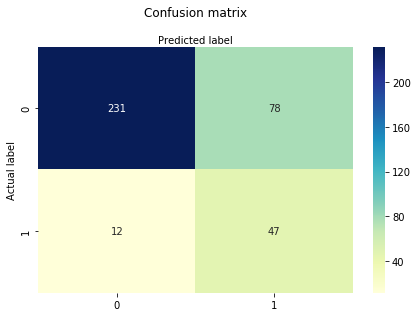
## 5.2 Logistic Regression

GridSearchCV allows us to fine-tune hyper-parameters by searching over specified parameter values for an estimator. As shown below, the results from GridSearchCV provided us with fine-tuned hyper-parameter using ROC\_AUC as the scoring metric.



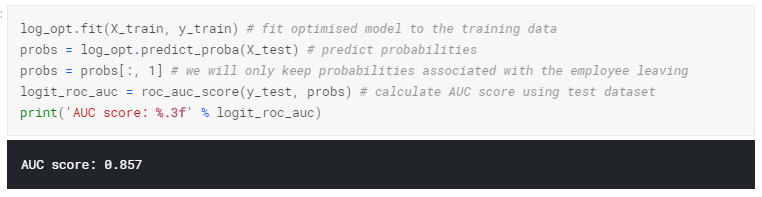
## 5.3 Confusion Matrix

The Confusion matrix provides us with a much more detailed representation of the accuracy score and of what’s going on with our labels — we know exactly which/how labels were correctly and incorrectly predicted. The accuracy of the Logistic Regression Classifier on test set is 75.54.



## 5.4 Label Probability

Instead of getting binary estimated target features (0 or 1), a probability can be associated with the predicted target. The output provides a first index referring to the probability that the data belong to **class 0** (employee not leaving), and the second refers to the probability that the data belong to **class 1** (employee leaving). Predicting probabilities of a particular label provides us with a measure of how likely an employee is to leave the company.

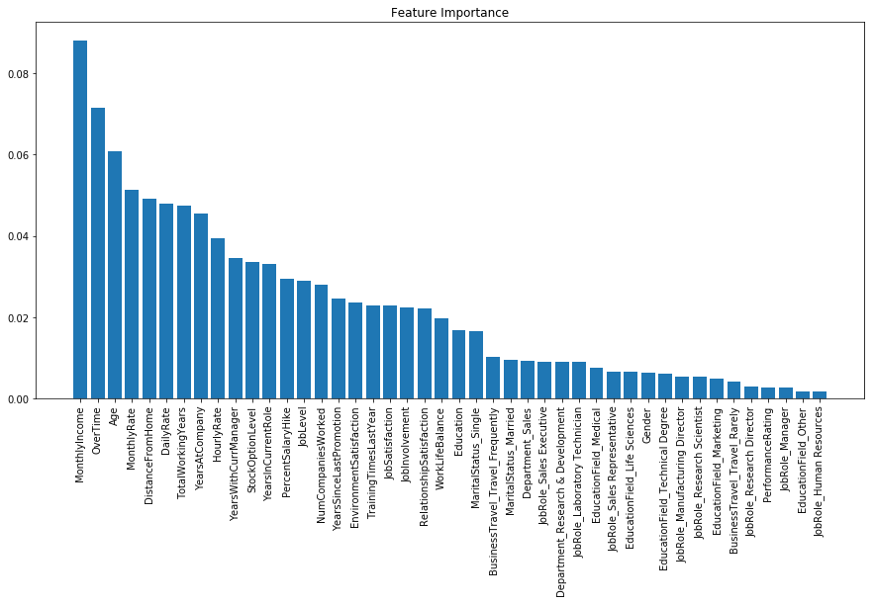


## 5.5 Random Forest Classifier

Let’s take a closer look at using the Random Forest algorithm. I’ll fine-tune the Random Forest algorithm’s hyper-parameters by cross-validation against the AUC score.

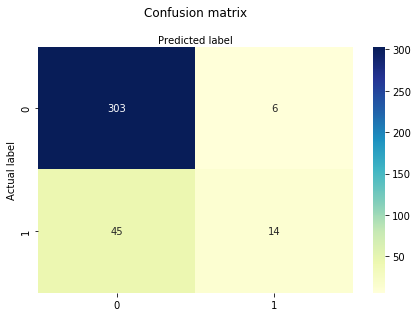


Random Forest allows us to know which features are of the most importance in predicting the target feature (“Attrition” in this project). Below, we plot features by their importance.

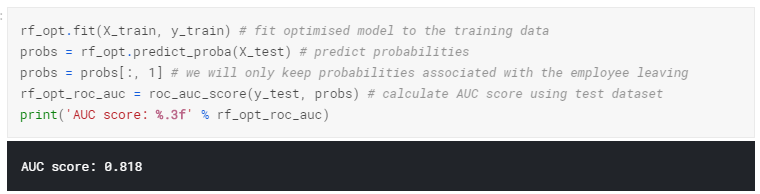


Random Forest helped us identify the Top 10 most important indicators (ranked in the table below) as: (1) MonthlyIncome, (2) OverTime, (3) Age, (4) MonthlyRate, (5) DistanceFromHome, (6) DailyRate, (7) TotalWorkingYears, (8) YearsAtCompany, (9) HourlyRate, (10) YearsWithCurrManager.

The accuracy of the RandomForest Regression Classifier on test set is 86.14. Below the corresponding Confusion Matrix is shown.

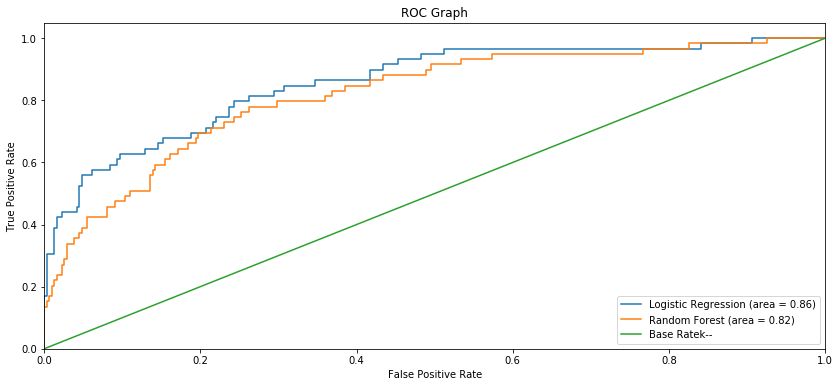


Predicting probabilities of a particular label provides us with a measure of how likely an employee is to leave the company. The AUC when predicting probabilities using RandomForestClassifier is 0.818.



## 5.6 ROC Graphs

AUC — ROC curve is a performance measurement for classification problem at various thresholds settings. ROC is a probability curve and AUC represents degree or measure of separability. It tells how much model is capable of distinguishing between classes. The green line represents the ROC curve of a purely random classifier; a good classifier stays as far away from that line as possible (toward the top-left corner).



As shown above, the fine-tuned Logistic Regression model showed a higher AUC score compared to the Random Forest Classifier.

# 6. Concluding Remarks

## 6.1 Risk Score

As the company generates more data on its employees (on New Joiners and recent Leavers) the algorithm can be re-trained using the additional data and theoretically generate more accurate predictions to identify **high-risk employees** of leaving based on the probabilistic label assigned to each feature variable (i.e. employee) by the algorithm.

Employees can be assigning a “**Risk Score**” based on the predicted label such that:

* **Low-risk** for employees with label < 0.6
* **Medium-risk** for employees with label between 0.6 and 0.8
* **High-risk** for employees with label > 0.8

## 6.2 Indicators and Strategic Retention Plan

The stronger indicators of people leaving include:

* **Monthly Income**: people on higher wages are less likely to leave the company. Hence, efforts should be made to gather information on industry benchmarks in the current local market to determine if the company is providing competitive wages.
* **Over Time**: people who work overtime are more likely to leave the company. Hence efforts must be taken to appropriately scope projects upfront with adequate support and manpower so as to reduce the use of overtime.
* **Age**: Employees in relatively young age bracket 25–35 are more likely to leave. Hence, efforts should be made to clearly articulate the long-term vision of the company and young employees fit in that vision, as well as provide incentives in the form of clear paths to promotion for instance.
* **DistanceFromHome**: Employees who live further from home are more likely to leave the company. Hence, efforts should be made to provide support in the form of company transportation for clusters of employees leaving the same area, or in the form of Transportation Allowance. Initial screening of employees based on their home location is probably not recommended as it would be regarded as a form of discrimination as long as employees make it to work on time every day.
* **TotalWorkingYears**: The more experienced employees are less likely to leave. Employees who have between 5–8 years of experience should be identified as potentially having a higher-risk of leaving.
* **YearsAtCompany**: Loyal companies are less likely to leave. Employees who hit their two-year anniversary should be identified as potentially having a higher-risk of leaving.
* **YearsWithCurrManager**: A large number of leavers leave 6 months after their Current Managers. By using Line Manager details for each employee, one can determine which Manager have experienced the largest numbers of employees resigning over the past year.

Several metrics can be used here to determine whether action should be taken with a **Line Manager**:

* # of years the Line Manager has been in a particular position: this may indicate that the employees may need management training or be assigned a mentor (ideally an Executive) in the organization
* Patterns in the employees who have resigned: this may indicate recurring patterns in employees leaving in which case action may be taken accordingly.

## ****6.3 Final Thoughts****

A strategic retention plan can be drawn for each **Risk Score**group. In addition to the suggested steps for each feature listed above, face-to-face meetings between a HR representative and employees can be initiated for **medium-** and **high-risk employees** to discuss work conditions. Also, a meeting with those employee’s Line Manager would allow to discuss the work environment within the team and whether steps can be taken to improve it.

# [Loan Application Status Prediction](https://medium.com/@dip.neepco/loan-application-status-prediction-e3b17b536b2?source=user_profile---------0----------------------------)

# Build predictive models to automate the process of predicting, if the loan of the applicant will be approved or not.

# Table of Contents:

1. Problem Definition.

2. Data Analysis.

3. EDA Concluding Remark.

4. Pre-Processing Pipeline.

5. Building Machine Learning Models.

6. Concluding Remarks.

## Problem Definition

Sometimes it’s hard to get a loan approved after going through many manual documentations processes & it consumes a lot of time to apply for a loan. Here, we have a data set that includes details of applicants who have applied for loans. The dataset includes details like credit history, loan amount, income, dependents, etc.

Objective: Here, with the help of the applicant’s details, we would be able to predict whether the loan of the applicant will be approved…

**Type of problem:**

The above problem statement clearly explains that the target variable is continuous & it’s a regression problem as we need to predict the price of the flight tickets. So this can be solved by any of the below Regression Machine learning algorithms:

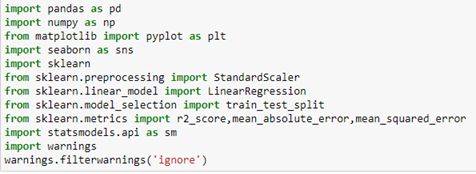
1. Linear Regression.

2. Decision Tree Regressor.

3. Random Forest Regressor.

I have mentioned only a few. We would be performing each of the techniques later in this blog.

Importing libraries:



# Data Analysis

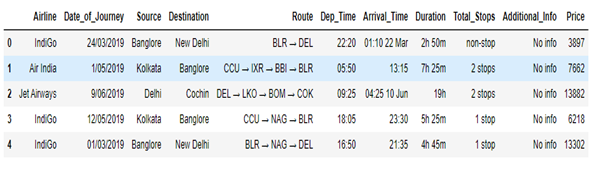
# Data Set Description:

There are 2 data sets that are given. One is training data and one is testing data.

1) Train file will be used for training the model, i.e. the model will learn from this file. It contains all the independent variables and the target variable. Size of training set: 10683 records.

2) Test file contains all the independent variables, but not the target variable. We will apply the model to predict the target variable for the test data. Size of test set: 2671 records.

# Read the excel file and convert it into a data frame:



**DATA SET**

**Airline**: Name of the airline used for traveling

**Date\_of\_Journey**: Date at which a person traveled

**Source**: Starting location of flight

**Destination**: Ending location of flight

**Route**: This contains information on starting and ending location of the journey in the standard format used by airlines.

**Dep\_Time**: Departure time of flight from starting location

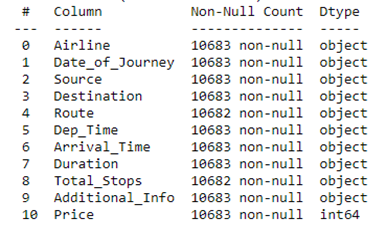
**Arrival\_Time**: Arrival time of flight at destination

**Duration**: Duration of flight in hours/minutes

**Total\_Stops**: Number of total stops flight took before landing at the destination.

**Additional\_Info**: Shown any additional information about a flight

**Price**: Price of the flight.



* We have 10683 rows & 11 columns in the dataset.
* We have maximum columns as categorical, which is an independent feature.

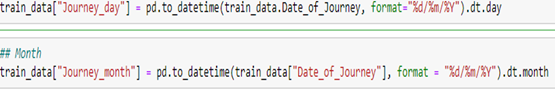
## Handling Missing values:

* Out of all these columns, we can see 1 missing value for Route & Total Stops column. As we have a huge dataset & out of which only 1 value is missing, rather than filling it, dropping would be the better option.
* After removing the null values, we have 10682 rows & 11 columns.

## Let’s check for each independent feature & convert the data type into the required Data type:

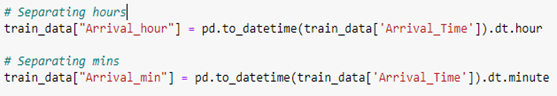
Training data set is having all independent features as categorical and the only target variable is numerical, also we can see some special character also being used because of that we need to perform data transformation on it before building the model.

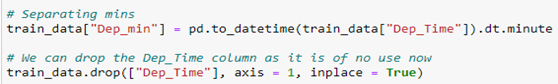
1. Date\_of\_Journey: it is given as day-month-year, so we need to use DateTime to convert it & separate it with day & month.



**DateTime Code**

2) Similarly, we can separate the hour & mins for Dep\_Time & Arrival Time.

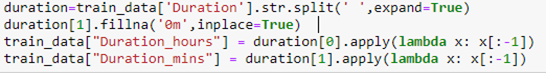




After dropping unwanted columns as it is of no use now: Arrival Time, Dep\_Time & Date\_of\_Journey, we have:

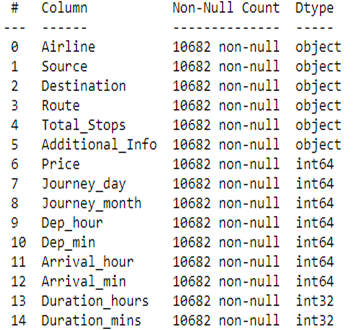


3) For the duration column, if we see above in the dataset, it is basically the difference between arrival & departure time & it is given as string data type. If we could split then it will be again in string format. So we need to split it & add 0 hours & mins.



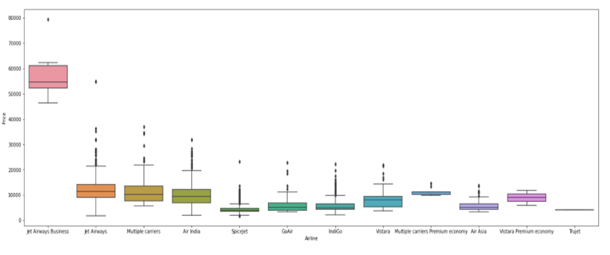
## EDA Concluding Remark

## Handling categorical columns:



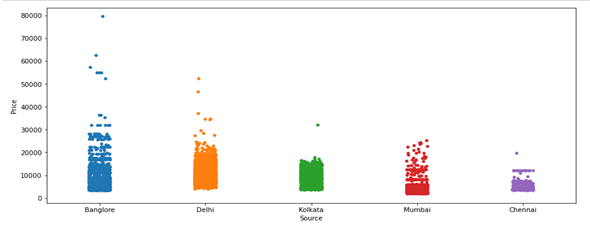
So, we have Airline, Source, Destination, Route, Total Stops & Additional Info as categorical columns. Let’s check the relation for each feature with Price.

1. **Airline Vs Price**



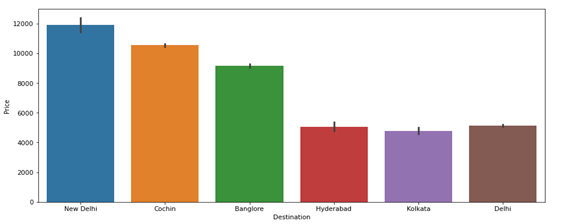
Jet Airways Business is selling tickets with maximum fare followed by jet airways & multiple careers & least is giving True.

**2) Source vs Price**



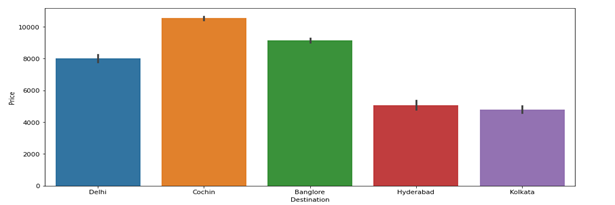
The maximum fare of the flights is departing from Bangalore followed by Delhi.

**3) Destination Vs Price**

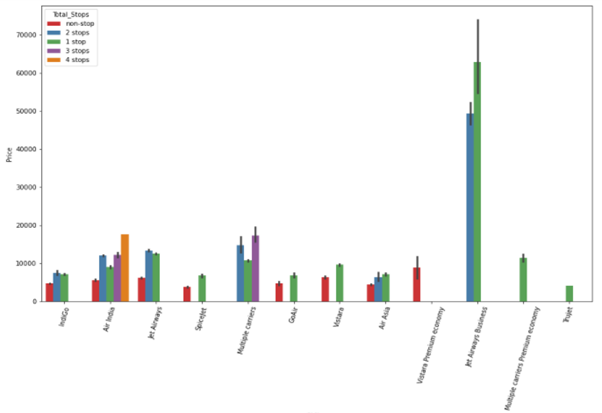


New Delhi is showing the highest average price in terms of destination.

Once we combine New Delhi & Delhi as both are in the same destination city, we got cochin as the highest average price.



**4) Total Stops(Airlines) Vs Price**



Jet Airways Business is showing the highest price having a single stop followed by 2 stops.

## Action Taken based on observation:

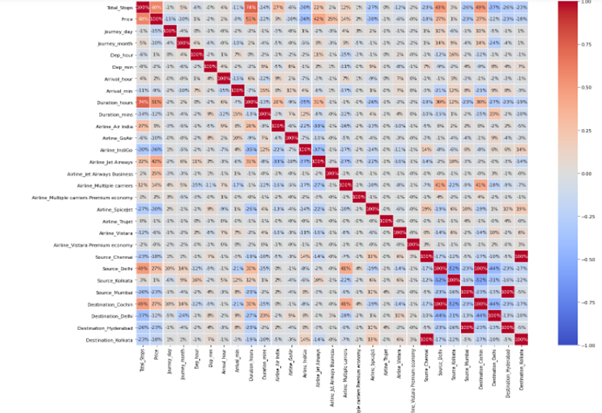
We have seen how categorical data is distributed & correlated with the target variable. Also, I have mentioned the findings below each plot.

Dropped columns Route & Additional info as the column ‘Route’ is related to total stops & column ‘additional info’ is showing no information for a maximum number of rows.

Performed One Hot Encoding for Nominal categorical columns: Airline, Source, Destination & performed label encoding for an ordinal categorical column: Total Stops.

No need to check for outliers/skewness as all the independent features are categorical.

## Visualizing the correlation in a heat map to check if there is any coefficient of multicollinearity



1)Total stops are more correlated with the target variable.

2)Multicollinearity: total stops & destination hours.

# Pre-Processing Pipeline

# Divide the data set into test and train

1. Now that we have converted all the categorical columns into numerical using the encoding technique. The next step is to split the data into test and train and drop the price column from the data set as we need to predict the price.

https://miro.medium.com/max/463/1*xnX3bTf-PJkor59C-YGB6Q.png

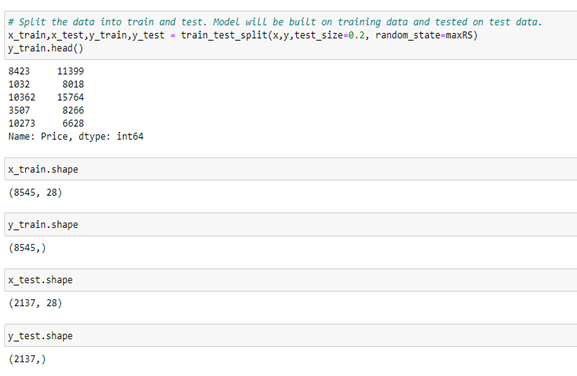
## 2) Finding the best Random\_ state: We have used Random Forest Regressor to obtain the best

## random state.

## 3) Final dimension of data: 10682 rows & 29 columns.

## 4) Created train test split: We have split the train & test data in 0.2 test size with the best random

## state.

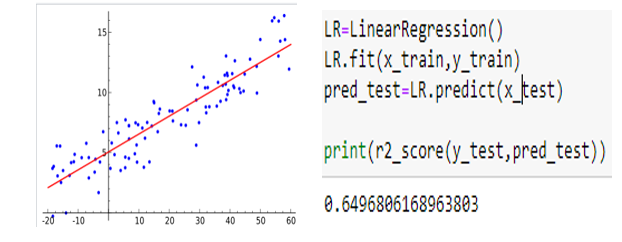


## Building Machine Learning Models

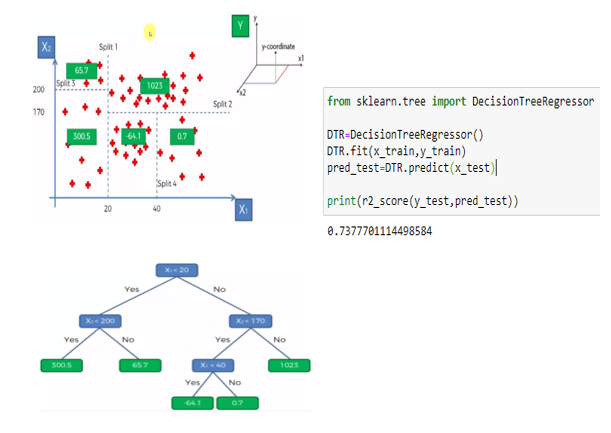
# Model Creation:

Now, after performing the train test split, we have x\_train, x\_test, y\_train & y\_test, which are required to build Machine learning models. We would build multiple regressor models to get the best R2(coefficient of determination) regression score function out of all the models.

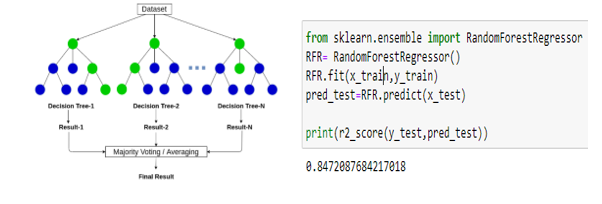
1. Linear Regression: It is a linear model, e.g. a model that assumes a linear relationship between the input variables (x) and the single output variable (y). More specifically, that y can be calculated from a linear combination of the input variables (x).



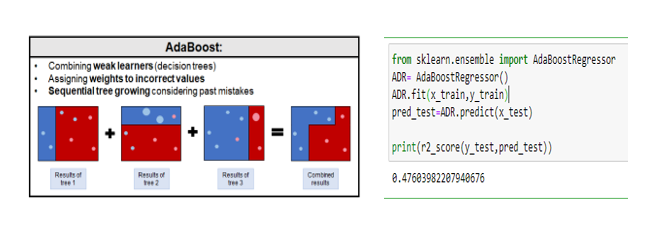
2) Decision Tree Regression: Decision Regression trees are needed when the response variable is numeric or continuous. It breaks down a dataset into smaller and smaller subsets while at the same time an associated decision tree is incrementally developed. The final result is a tree with decision nodes and leaf nodes.



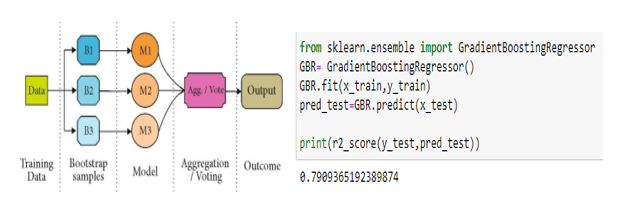
3) Random Forest Regressor: It uses the ensemble learning method for regression. The ensemble learning method is a technique that combines predictions from multiple machine learning algorithms to make a more accurate prediction based on majority voting/averaging.



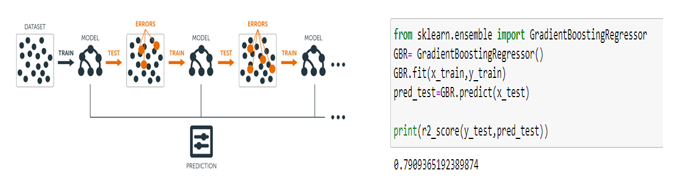
4) AdaBoost Regressor: It is used to boost the performance of any machine learning algorithm. It helps you combine multiple “weak classifiers” into a single “strong classifier”.



5) Bagging Regressor: It is an ensemble meta-estimator that fits base regressors each on random subsets of the original dataset and then aggregates their individual predictions (either by voting or by averaging) to form a final prediction.



6) Gradient Boosting Regressor: It produces a prediction model in the form of an ensemble of weak prediction models, typically decision trees.

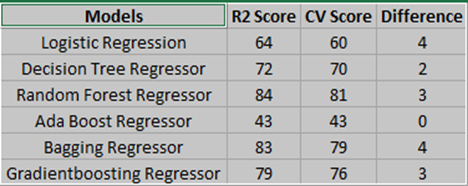


## 1) Observation**:**

For every model, we have train the data (x\_train, y\_train) & predict with the help of x\_test. Now, with the help of the y\_test & prediction test we got the R2 score. So, based on every model we have the best R2 score for Random Forest Regressor with an 84% score, however, this could be because of overfitting. Hence we need to perform cross-validation to check if the model is overfitted or not.

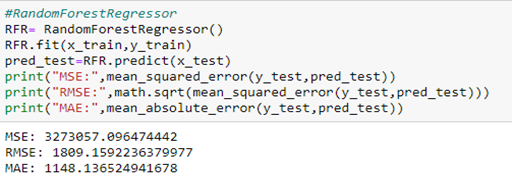
# Cross-Validation:

We would perform cross-validation for every model & compare it with the R2 score, whichever model gives less difference between the Cross-validation score & R2 score is the best fit model.



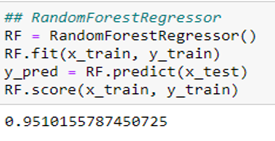
As we have seen, the Ada boost regressor is giving 0 difference but it is having a very low R2 Score. Random Forest Regressor is giving best R2 score with less difference of R2 Score & CV Score, however, let's proceed with model evaluation to get the MSE(mean squared error), RMSE(root mean squared error), & MAE( mean absolute error). These evaluation metrics will help to decide the best model.

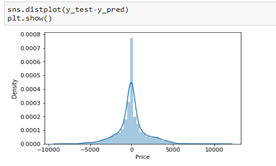
**Model Evaluation:**



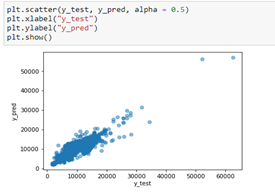
Random Forest Regressor is giving the best result as compare to other models. Let's check its test & train score.

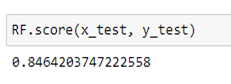
**Train Score:**





**Test Score:**



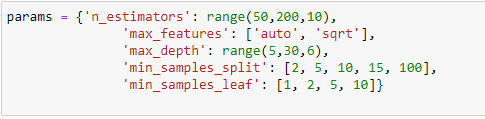


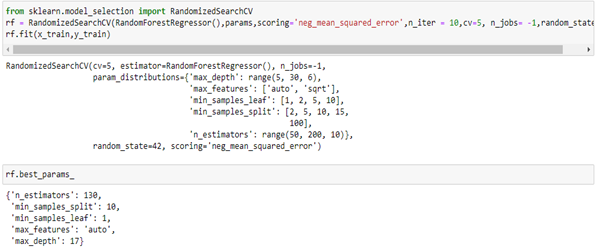
As we have seen, we have less difference of train & test score, and the predicted value & test value is normally distributed, Also, in the scatter plot the test value & the prediction value is linearly distributed. The test & prediction values are almost close to each other.

Since Random Forest Regressor is the best model in terms of model score, cross-validation difference, test & train r2 score difference, also as per the evaluation metrics, we choose Random Forest Regressor to be the final model. Let's see if we can increase the score by using hyper-parameter tuning.

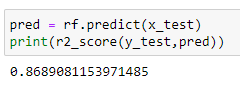
## Hyper Parameter Tuning:

We would try to increase the model by giving the best parameters (params as seen below) of Random Forest Regressor & again train to get an increased R2 score.

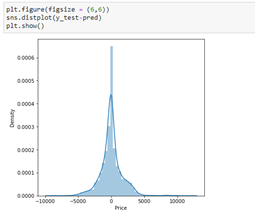


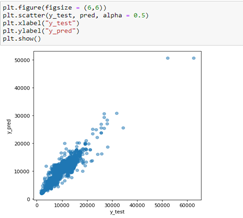


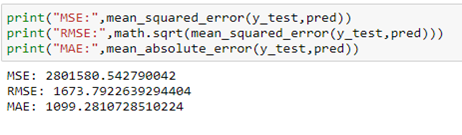
## Prediction:



As we can see, Score has increased to 87%.

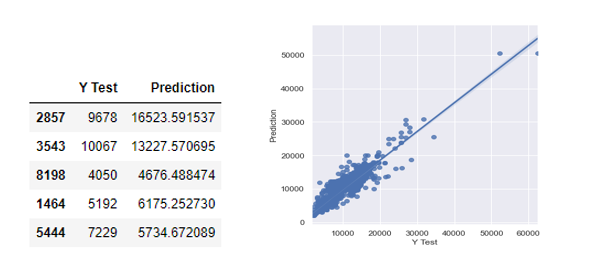






After performing Hyper Parameter tuning, we have increased the Score, evaluation metrics are showing better results, also test & prediction value is normally distributed. The Test & Prediction value is also linearly related to each other.

## Let’s predict & compare the results:

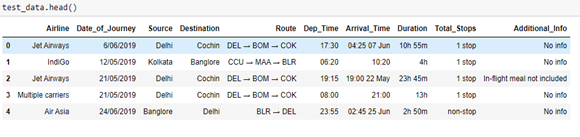


# Concluding Remarks

1) Saving the model: The model is ready & we have saved the model in ‘pkl’ format by using

“joblib”.

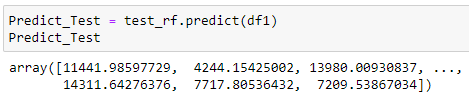
2) Test data: It is having 2672 rows & 10 columns.



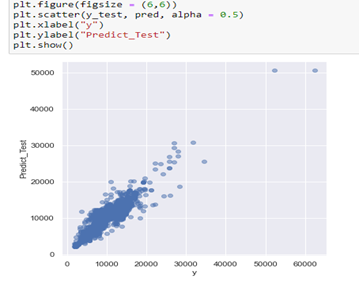
3) We need to perform all the same steps as we performed for the training dataset, which includes Data analysis, EDA, & Pre-processing. The test dataset is having all the same columns except the target variable.

4) We don’t need to build any model using the test dataset. Hence no need to perform a train test split, only cleaning the data is required.

5) Once the test dataset is ready, we can load the train data set & predict the price.



6) Comparing the actual target value from the training dataset & predicted test value.



## Final conclusion:

As we have seen, the prediction is showing a similar relationship with the actual price from the train data set, which means the model predicted correctly & this could help airlines by predicting what prices they can maintain. It could also help customers to predict future flight prices and plan the journey accordingly because it is difficult for airlines to maintain prices since it changes dynamically due to different conditions. Hence by using Machine learning techniques we can solve this problem.