Henry Harvin Education India LLP

Summer Internship Programme

Sector-2, Noida, U.P.-201306

Project Title – **HR ANALYTICS**

Mentor Name: Ms. Pooja Gupta (Designation)

Name: Abhijit Roy

Course: Summer Internship Programme (SIP) Python

Batch: June – July 2019

Job Title: Business Analyst Associate (Intern)

Institution: Lovely Professional University, Punjab

# 

# DECLARATION

I here by declare that the project report entitled of “**Car Price Prediction**” submitted by me to **HENRY HARVIN EDUCATION INDIA** is a record of bonafide project work carried out by me under the guidance of MS. POOJA GUPTA (Mentor). This project is an original report with references taken from websites and help from mentors and teachers.

Abhijit Roy

Date: 28 Jul 2019

SIP – Python

# Acknowledgements

In the accomplishment of this project successfully, many people have best owned upon me their blessings and the heart pledged support, this time I am utilizing to thank all the people who have been concerned with this project. Primarily I would thank god for being able to complete this project with success. Then I would like to thank my teachers MR. DHIRAJ UPADHYAYA and MR. ANIL JADON whose valuable guidance has been the ones that helped me patch this project and make it full proof success.

Their suggestions and instructions have served as the major contributor towards the completion of the project. I would like to thank my mentor MS. POOJA GUPTA (Mentor) for giving me this golden opportunity.

Then I would like to thank my parents and friends who have helped me with their valuable suggestions and guidance has been very helpful in various phases of the completion of the project. Last but not the least I would like to thank my batchmates who have helped me a lot.

Abhijit Roy

SIP-Python

# TABLE OF CONTENTS

Declaration 2

Acknowledgements 3

Abstract 5

1. Project Introduction 6

2. Project Data Introduction 7

3. Exploratory Data Analysis (EDA) 9

4. Model Building 31

5. Prediction and Evaluation 44

References 49

# Abstract

“Cars are being sold more than ever. Developing countries adopt the lease culture instead of buying a new car due to affordability. Therefore, the rise of used cars sales is exponentially increasing. Car sellers sometimes take advantage of this scenario by listing unrealistic prices owing to the demand. Therefore, arises a need for a model that can assign a price for a vehicle by evaluating its features taking the prices of other cars into consideration. In this paper, we use supervised learning method namely Random Forest to predict the prices of used cars. The model has been chosen after careful exploratory data analysis to determine the impact of each feature on price. A Random Forest was created to train the data. From experimental results, the linear regression accuracy was found out to be 64.377%, and the random forest accuracy was 76.147%. The model can predict the price of cars accurately by choosing the most correlated features.”

# 1. Project Introduction

The prices of cars in the industry is fixed by the manufacturer with some additional costs incurred by the Government in the form of taxes. So customers buying a new car can be assured of the money they invest to be worthy. But due to the increased price of new cars and the incapability of customers to buy new cars due to the lack of funds, used cars sales are on a global increase.

Predicting the prices of cars is an interesting and much-needed problem to be addressed. Customers can be widely exploited by fixing unrealistic prices for the used cars and many falls into this trap. Therefore, rises an absolute necessity of a car price prediction system to effectively determine the worthiness of the car using a variety of features.

Due to the adverse pricing of cars and the nomadic nature of people in developed countries, the cars are mostly bought on a lease basis, where there is an agreement between the buyer and seller. These cars upon completion of the agreement are resold. So reselling has become an essential part of today’s world.

Given the description of cars, the prediction of cars is not an easy task. There are a variety of features of a car like the age of the car, its make, the origin of the car (the original country of the manufacturer), its mileage (the number of kilometers it has run) and its horsepower. Due to rising fuel prices, fuel economy is also of prime importance. Other factors such as the type of fuel it uses, style, braking system, the volume of its cylinders (measured in cc), acceleration, the number of doors, safety index, size, weight, height, paint color, consumer reviews, prestigious awards won by the car manufacturer. Other options such as sound system, air conditioner, power steering, cosmic wheels, GPS navigator all may influence the price as well.

# 2. Project Data Introduction

This project is based on Predictive Analysis. This is a Python-based Project. This project was created via Spyder 3.3.5. IDE (Integrated Development Environment) using Python 3.7.3 and Ipython Console 7.4.0. The final outcome of this project is saved as a Jupyter Notebook v7.8.0. The libraries of python used in this project are:

1. NumPy

2. Pandas

3. Matplotlib

4. Seaborn

5. Statsmodels

6. Sci-kit Learn

This project is based on a data set provided by the teachers via GITHUB. The data used in the project is continuous, and hence, we are using “LINEAR REGRESSION” and “RANDOM FOREST REGRESSION” for predicting our data.

Here, the **target variable** is “LEFT”.

Data Set Dictionary:

|  |  |  |
| --- | --- | --- |
| **Name of Column** | **Description** | **Type** |
| satisfaction\_level | Satisfaction level of the employee | Numeric |
| last\_evaluation | Last evaluation of the employee | Numeric |
| Number\_projects | No of projects completed by the employee | Categorical |
| Average monthly hours | Average Monthly hours spent by the employee | Categorical |
| Time spend in company | Time spent by employee in the company | Categorical |
| **Name of Column** | **Description** | **Type** |
| Work accident | Accident happened to employee or not while working | Categorical |
| Left (target variables) | Employee left or not | Categorical |
| Promotion last 5 years | Promotion status in the last 5 years | Categorical |
| Department | Department of the employee | Categorical |
| Salary | Salary category | Categorical |

Data Set Size: 9653 rows, 10 columns

Categorical Variables:[number\_projects, work\_accident, left, promotion\_last5years, department, salary, time\_spend\_company, average\_monthly\_hours] = 8 features

Numeric Variables: [ satisfaction\_level, last\_evaluation] = 2 Features

This Data Set is present in the GITHUB Repository as follows:

<https://github.com/jitroy160/Final_Projects/blob/master/Final_Projects/HR_Analytics.xlsx>

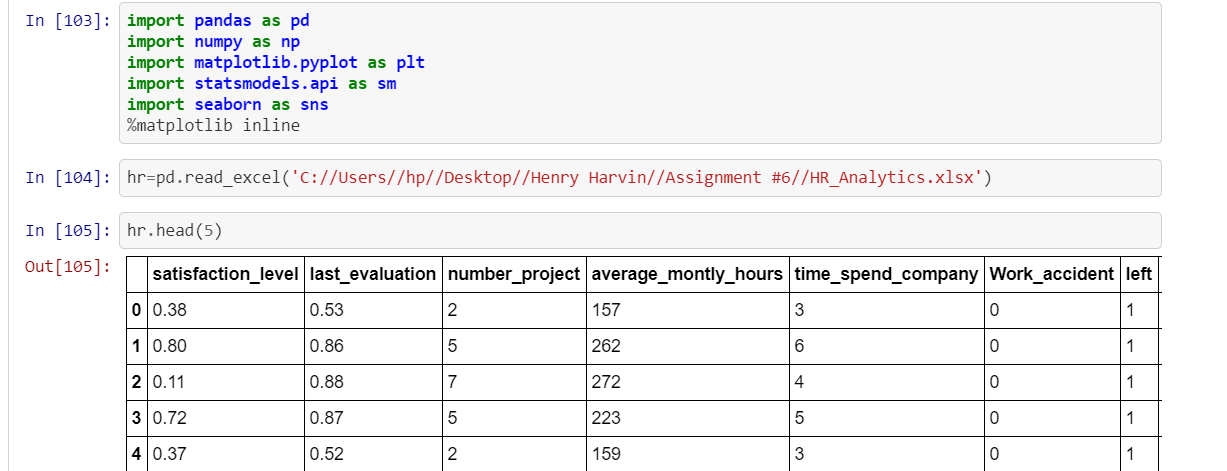
# 3. Exploratory Data Analysis (EDA)

In statistics, **exploratory data analysis** (**EDA**) is an approach to analyzing data sets to summarize their main characteristics, often with visual methods. A statistical model can be used or not, but primarily EDA is for seeing what the data can tell us beyond the formal modeling or hypothesis testing task. Exploratory data analysis was promoted by many to encourage statisticians to explore the data, and possibly formulate hypotheses that could lead to new data collection and experiments. EDA is different from initial data analysis (IDA), which focuses more narrowly on checking assumptions required for model fitting and hypothesis testing, and handling missing values and making transformations of variables as needed. EDA encompasses IDA.

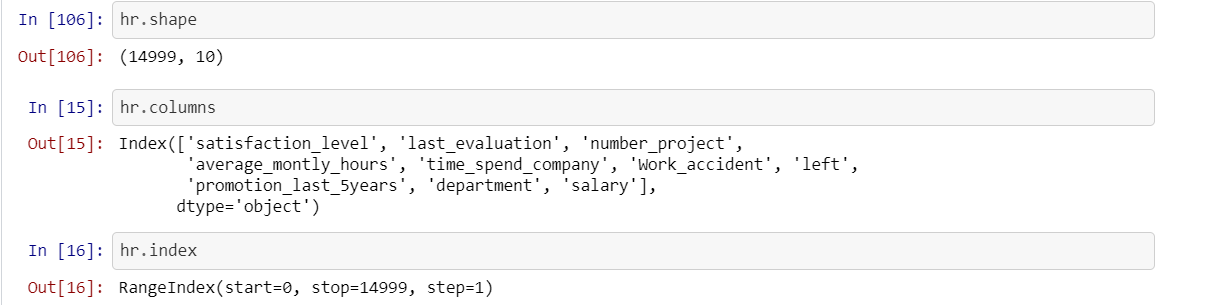
In this project, we used matplotlib, seaborn for EDA using python 3.7.3. It is as follows:

## 3.1. Data Understanding

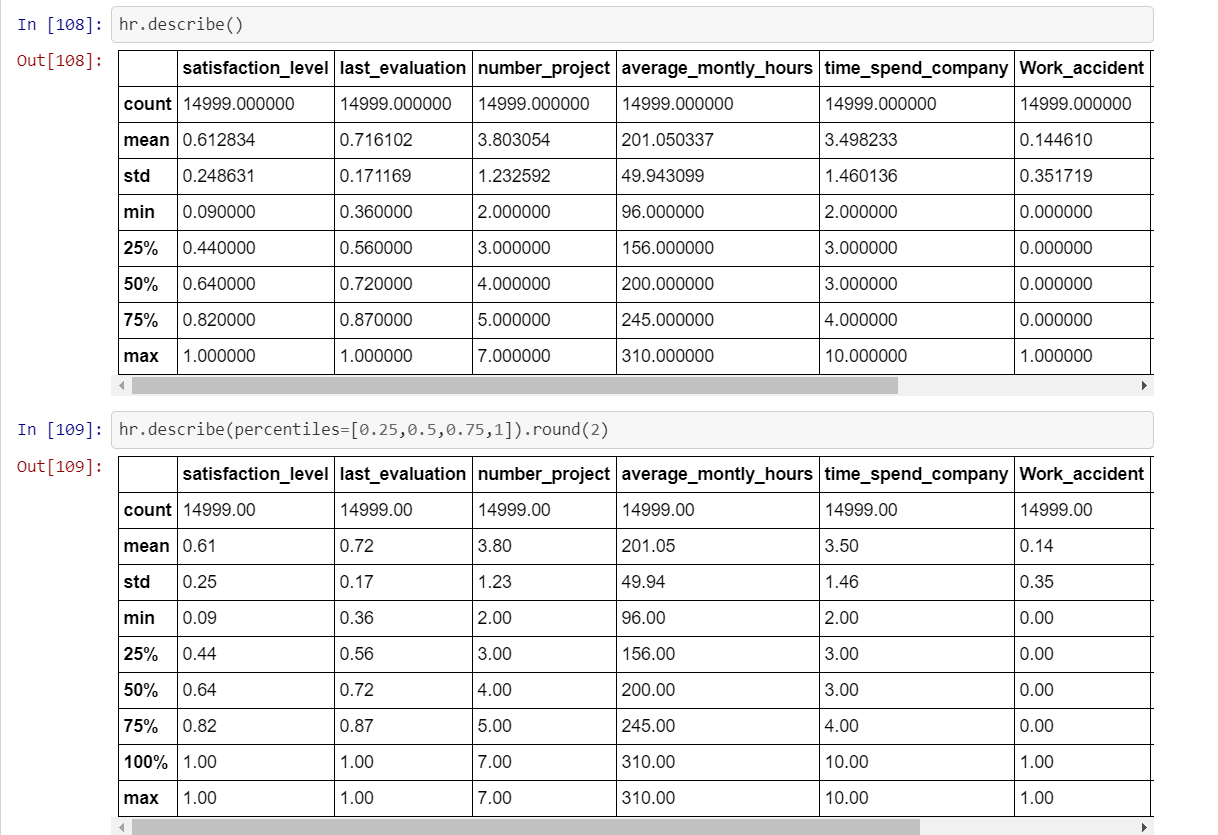
At first, I imported all the libraries initially required for EDA. Then, I imported the file saved in the repository link and displayed its data. The source code and output are:



Then, I used the **columns** and **shape** and **index** function to study the summary of the data( min, max, no of values etc.) The source code and output are the following:



Then, I used the describe() function and info() function to study my data set.

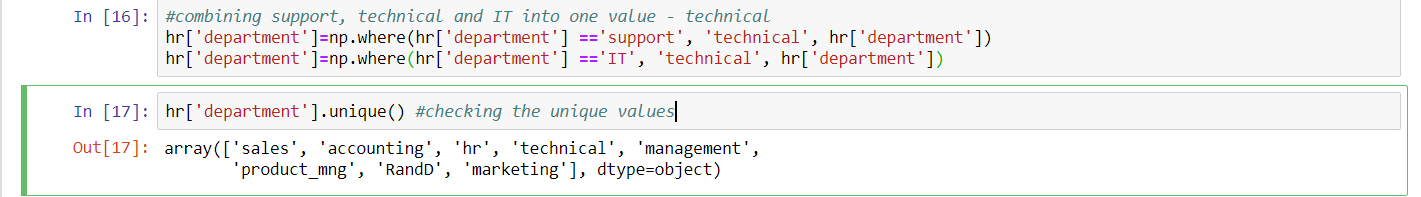


## 3.2. Data Cleaning and Preparation

Data Cleaning, as the name suggests, is to clean the data of any irregularities. By performing this step, we prepare our data for analysis. For this, we check for any spelling errors, empty values and duplicate values. The source code and output are:



Now, we have the columns ‘technical’, ‘IT’, ‘support’that mean the same. Hence, let us combine them to one column - ‘technical’. Check the values after doing it.

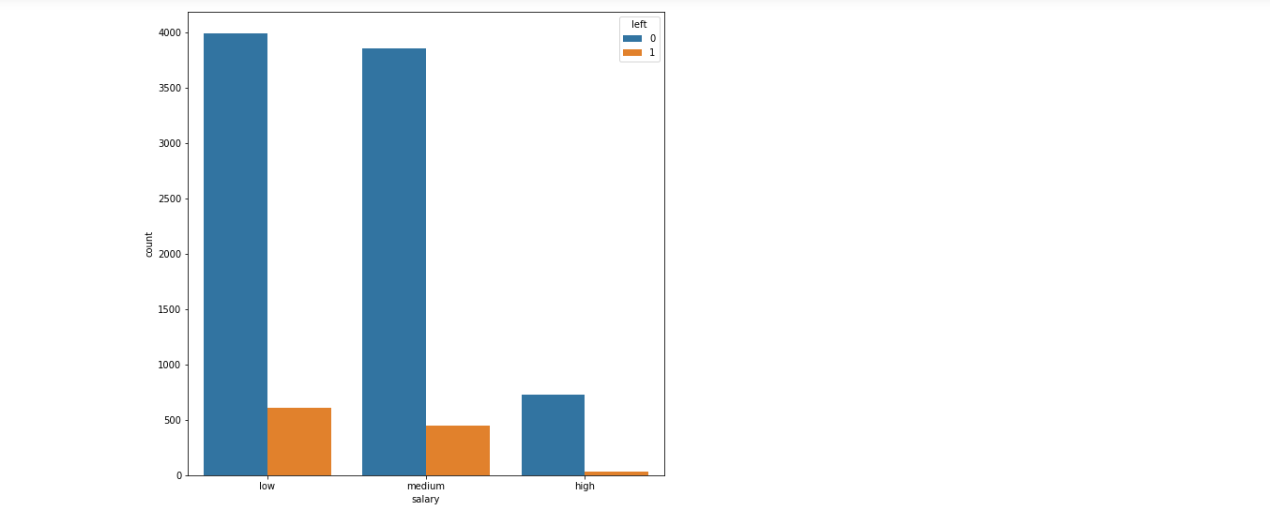


Our data is officially clean. It’s time for the final step of EDA: Visualization.

## 3.3. Visualization

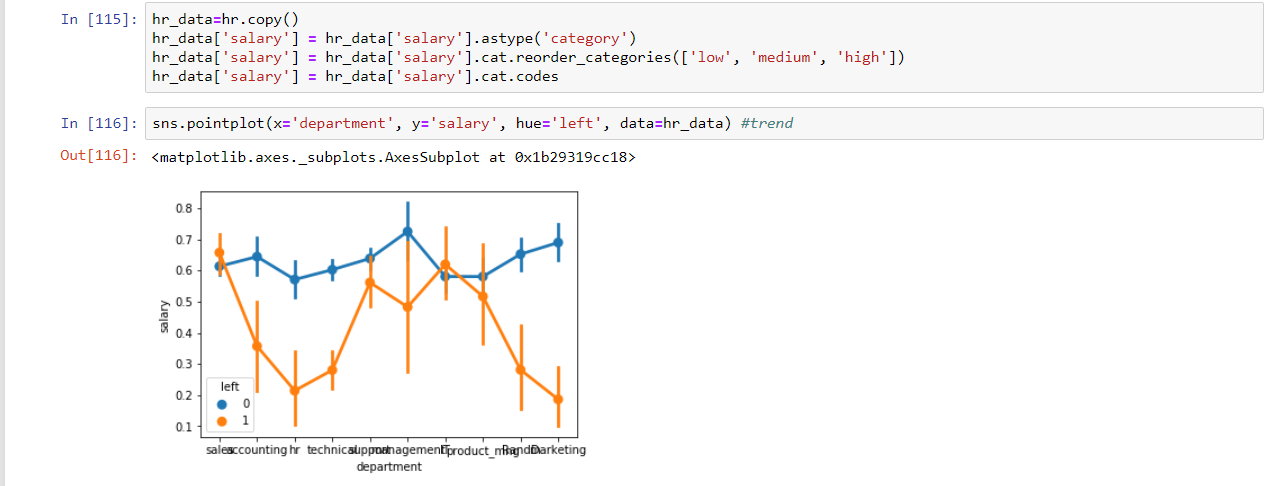
Visualization refers to the term that gives a picture to our information. We can describe our data by drawing graphs and charts to check different parameters that, in the end, might help us choose features for our analysis. In python, we use matplotlib and seaborn for visualization. These two libraries are efficient enough to give us an output that gives us an idea about our data set. The source code and output are :





We can see that technical has most of its employees leaving. Also, low and medium salary employees are leaving more than others.

Now, lets check the salary trend by converting the salary category to codes. For that, we will use cat function.



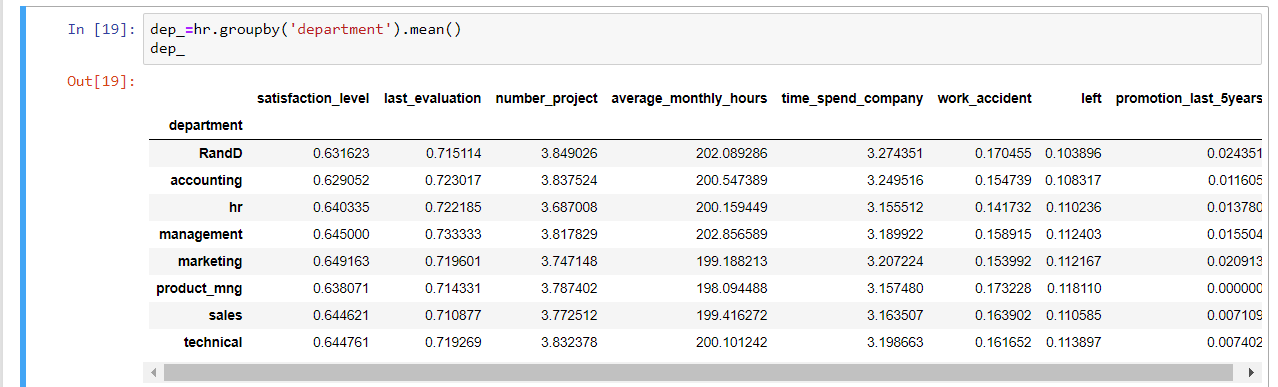
Now, we need to visualize our features to select the most significant of them for our analysis for better results. Hence, let’s first visualize the categorical variables.  
  
Categorical Variables: [ number\_projects, work\_accident, left, promotion\_last5years, department, salary, time\_spend\_company, average\_monthly\_hours] = 8 features

**1. Department**

First, let us see what the department category has to offer. We can check the turnover frequency for each department. Also, we can check different values for each department.

dep\_=hr\_data.groupby('department').mean()

dep\_



Now, let us plot a graph for department turnover frequency.

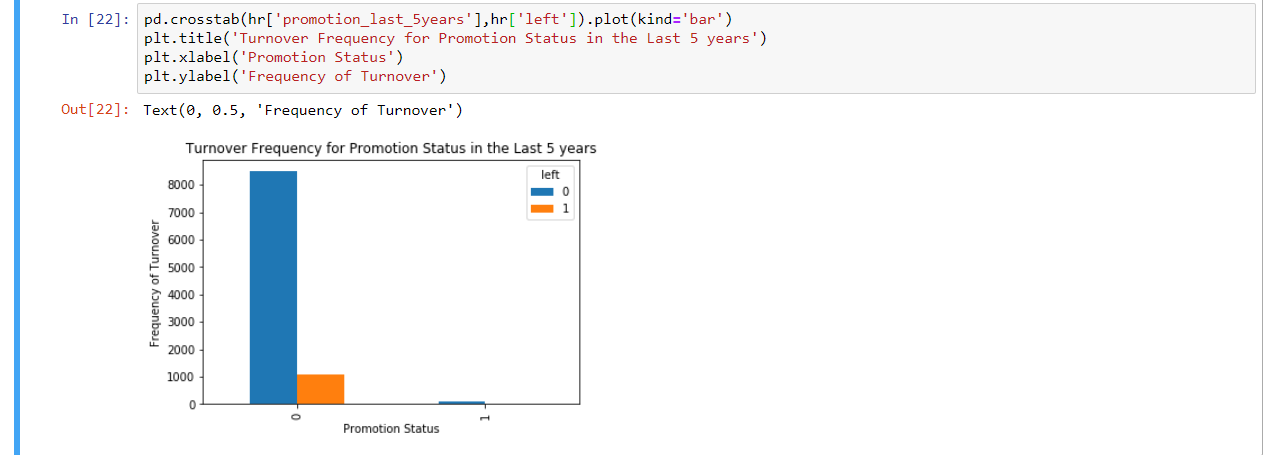


**2. Salary**

Salary is one of the most important features that decide the employee turnover of the company. Let us look at it’s data.  
  

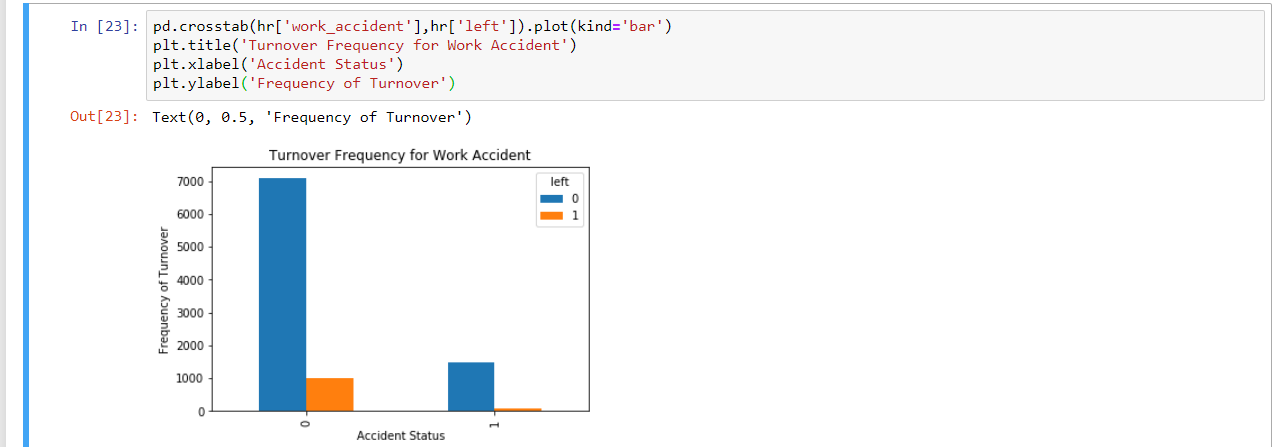

This clearly shows that medium salary and low salary people are leaving more frequently.

**3. Promotion Last 5 Years**

This is a very important variable for the analysis. Promotion becomes an essential feature when it comes to employee leaving or staying in the company.  
  


This shows that people who didn’t get promotion in the last 5 years were the most to leave.

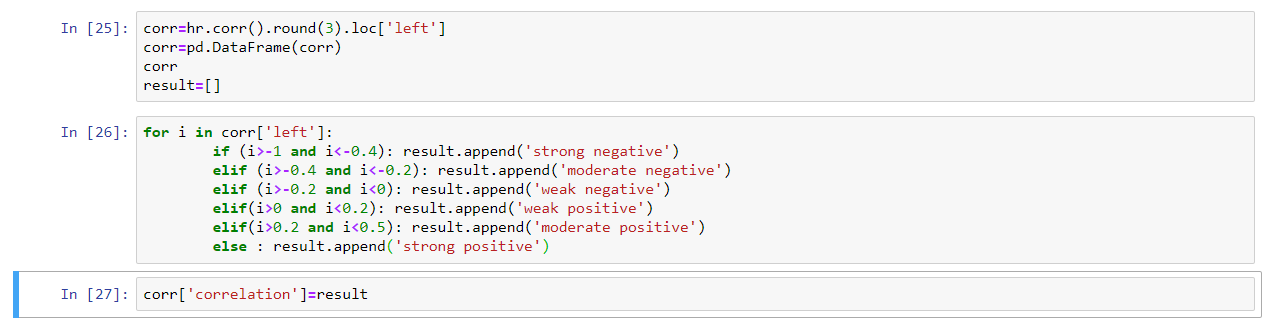
**4. Work Accident**

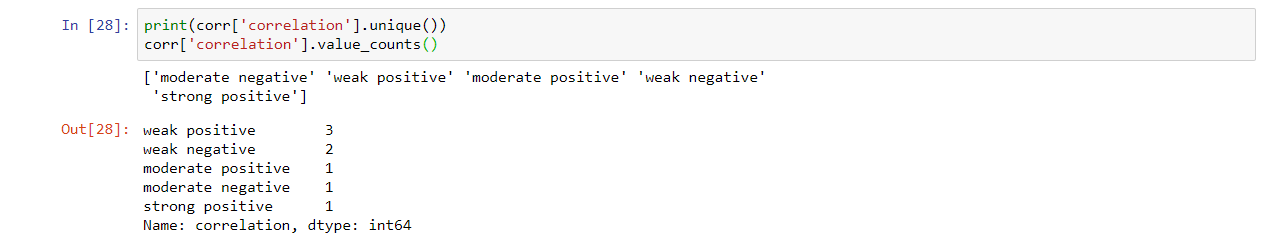
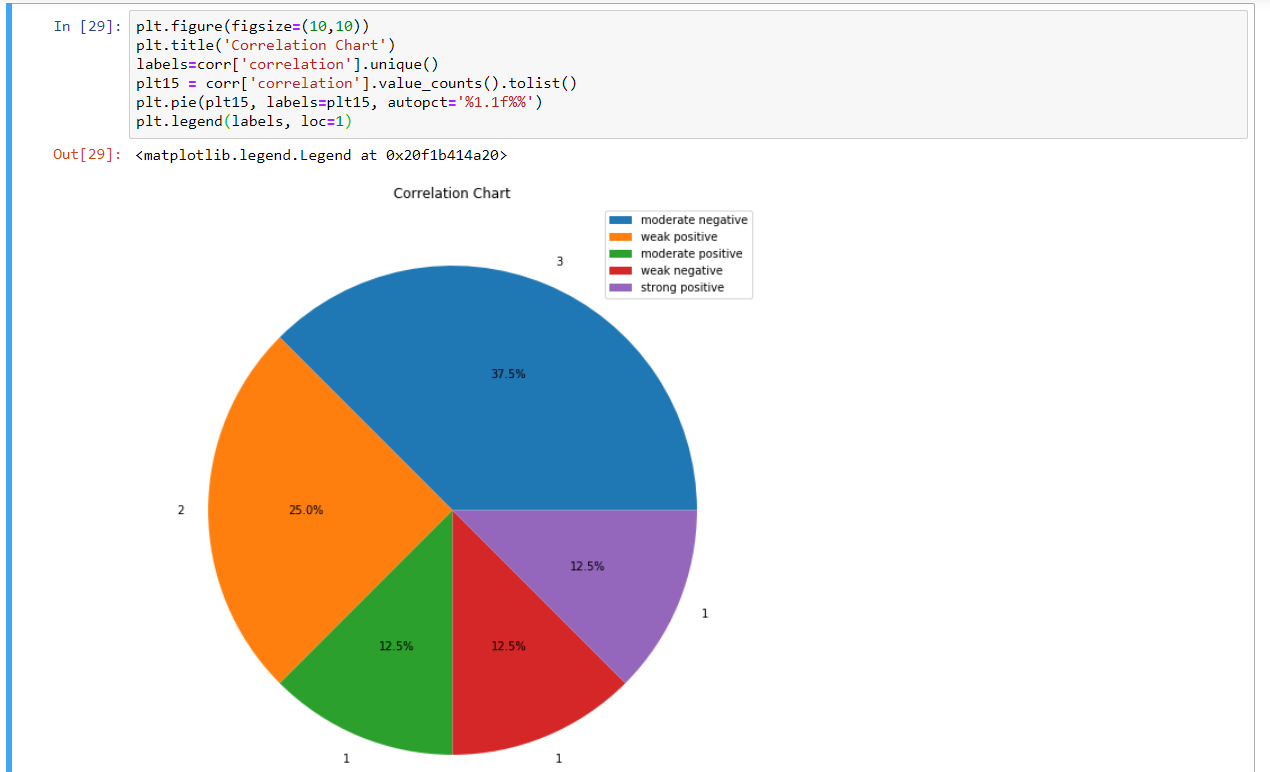
It can be another significant variable in predicting whether the employee will leave or not. Hence, let us check the employee turnover.  
  


From the graph, we can say that the employees that were not involved in a work accident were leaving more frequently.  
  
**5. Number of Projects , Time Spent in Company**

Number of Projects is a significant variable in the prediction of employee leaving or not. So is Time spent in company. Let us check.  
  


This shows that number of projects 4 were completed most frequently. Also, most employees spent 3 hours in the company.

With this conclusion, the categorical variables visualization is complete. Now, let us make a correlation dataframe for left.  
  


Now, let us print the unique values of the correlation. Also, display the frequency for each category.  
  
  
  
Let us create a pie chart for this correlation. It will tell us how much which category is contributing to the data.  
  
This correlation chart shows that most of the categories are moderately positively correlated with the target variable (‘left’).

After the visualization, the variables selected for analysis are:

1. Satisfaction Level

2. Time Spend Company

3. Last Evaluation

4. Number of Projects

5. Work Accident

6. Promotion last 5 years

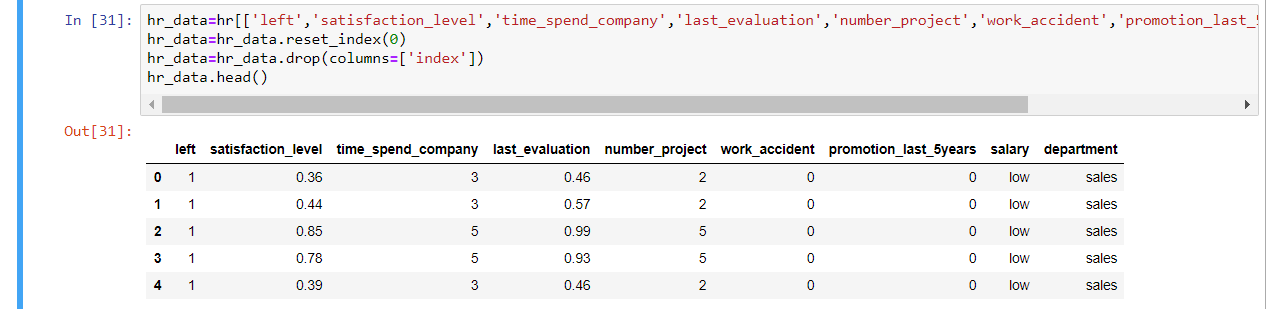
7. Salary

8. Department

# 4. Model Building

Now that we have selected which features to be chosen, it is time to make a model for our analysis. First of all, what is a model?

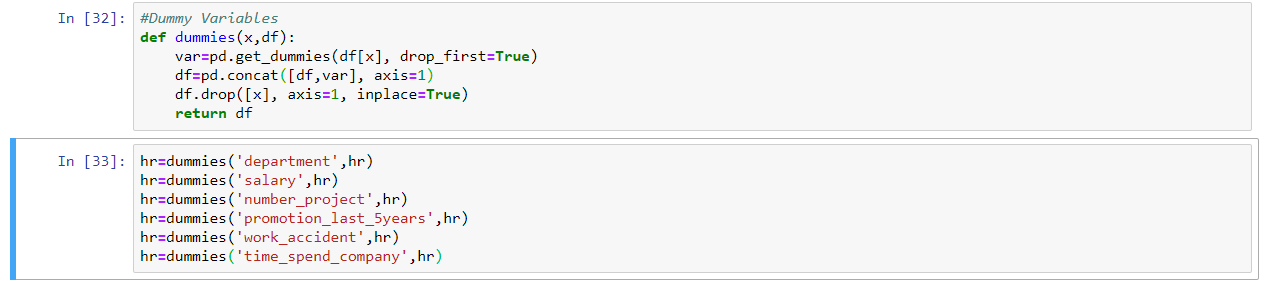
“A statistical model is usually specified as a mathematical relationship between one or more random variables and other non-random variables. As such, a statistical model is "a formal representation of a theory.”

Now, to be precise, we need to create a model. For that, we need a data set that has only values of the features we selected through visualization. Let’s do that. The code and output are:  
  


This is the data set, that we need to work on. Now, moving on further, we need to create dummy variables. What are dummy variables?

“A dummy variable is one that takes the value 0 or 1 to indicate the absence or presence of some categorical effect that may be expected to shift the outcome. Dummy variables are used as devices to sort data into mutually exclusive categories.”

Now, lets create these dummy variables. Remember, we need to create dummy variable for categorical variables only. Lets do this.



Now, lets create the target and independent variables

hr\_var=hr.columns.tolist()

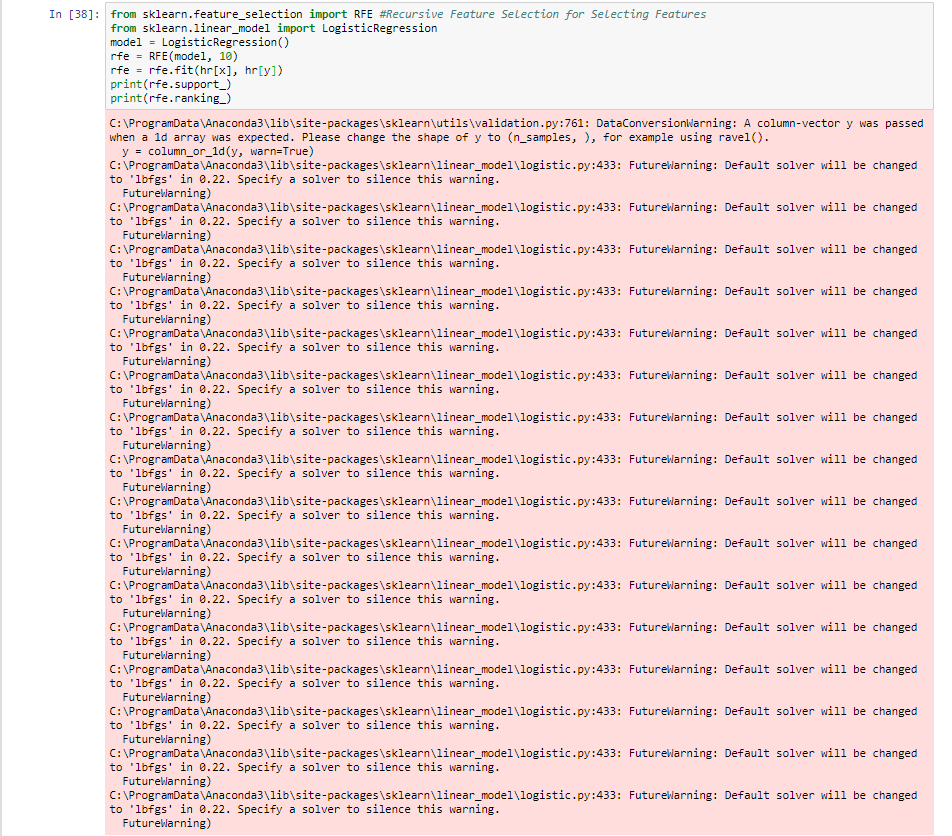
y=['left']

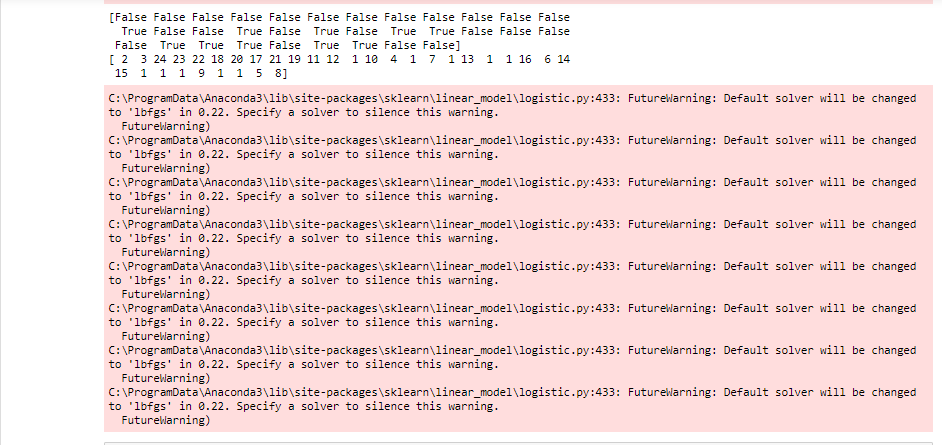
x=[var for var in hr\_var if var not in y]

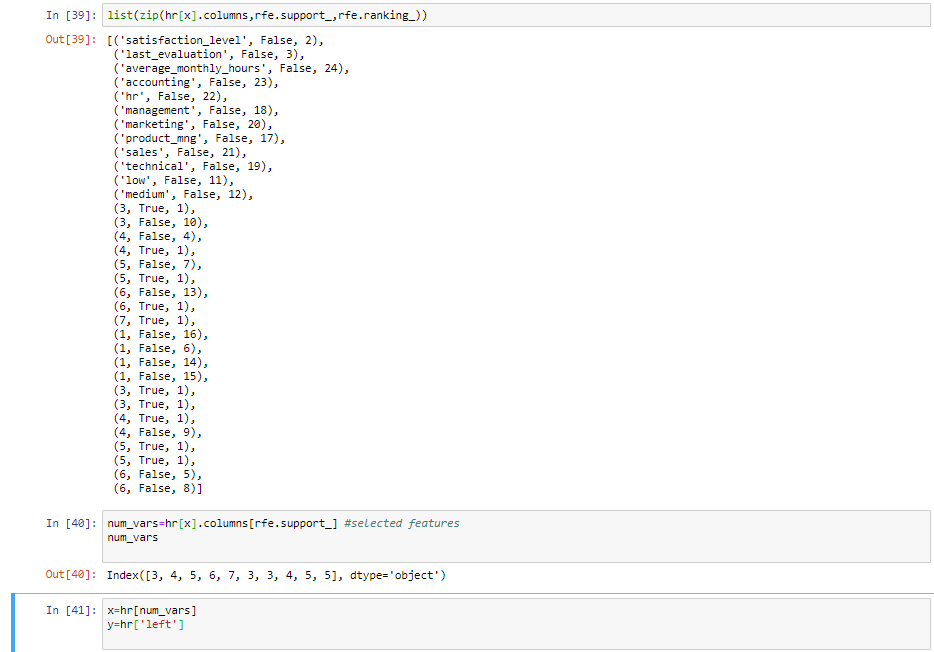
Now, it is time to build our model. For that, we use RFE (Recursive Feature Engineering) to select ‘n’ no. of variables from our already selected variables. Lets do that.

RFE selects ‘n’ no. of variables for your model on its own. You don’t need to select the variables. However, if you want to, you can do it by not using RFE. I have used it because it selects variables after running some tests on those variables.

Now, lets create our model.







The rfe.support\_ function gives the value ‘True’ to each feature selected. The rfe.ranking\_ gives the value 1 to each feature selected. Then, we are overwriting the response variable(s), i.e. x to the dataframe with the features selected and the target variable i.e. y to the left column of the dataframe.

Now comes the most important part. We will be splitting our data set into training set and test set.What are these sets? Why do we do this? Split the data set?

First of all, if you run the model on the whole data set and predict from the same, you will get accuracy too high, which would be invalid because your dependent variable will be included in the data set in which you are predicting your values.

Secondly, if the model fails, the data set has to be re-loaded from the beginning. Hence, we first train our model with the training set, and when our model runs perfectly, we use it on our test set to predict values. Remember, training set should always be greater than test set. The more you train your model, the better it will predict. Let’s do this…

from sklearn.model\_selection import train\_test\_split

x\_train, x\_test, y\_train, y\_test = train\_test\_split(x, y, test\_size=0.2, random\_state=0)

from sklearn import metrics

logreg = LogisticRegression()

logreg.fit(x\_train, y\_train)

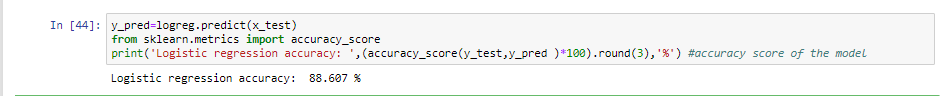
Here, we will use logistic regression to predict values.Why logistic regression? What is logistic regression?

Logistic regression is a statistical model that in its basic form uses a logistic function to model a binary dependent variable, although many more complex extensions exist. In regression analysis, logistic regression (or logit regression) is estimating the parameters of a logistic model (a form of binary regression). Mathematically, a binary logistic model has a dependent variable with two possible values, such as pass/fail which is represented by an indicator variable, where the two values are labeled "0" and "1".

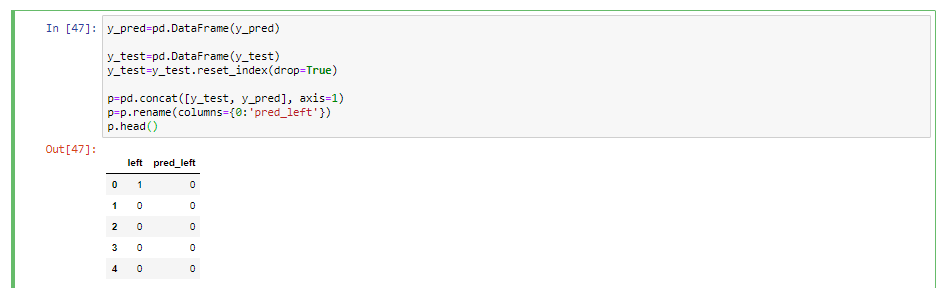
We are using logistic regression because our target variable is categorical or binary (0,1) and is not continuous.

# 5. Prediction and Evaluation

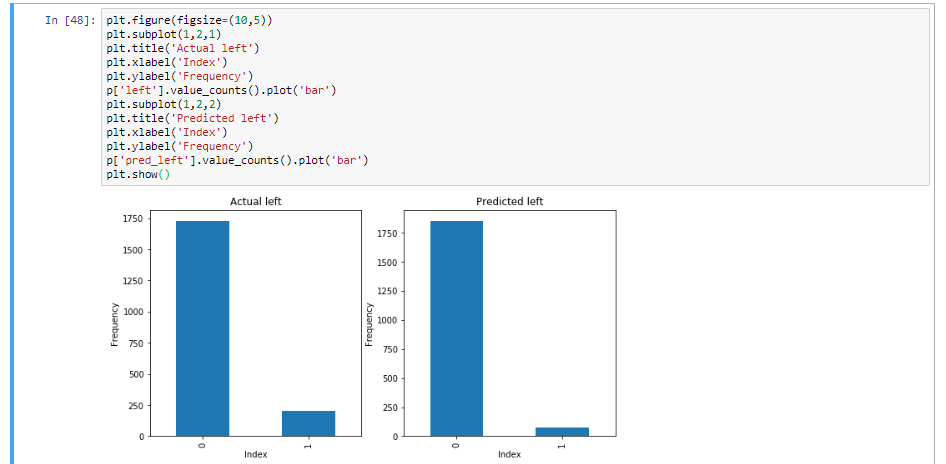
We are at the final stage of our project. Time to predict values!



Our model accuracy is 88.607%. It’s perfect for logistic regression. Now, let us check the actual and predicted values.



Time to plot this. Remember, these two need to be plot seperately. Let us try this and check how accurate our accuracy is.

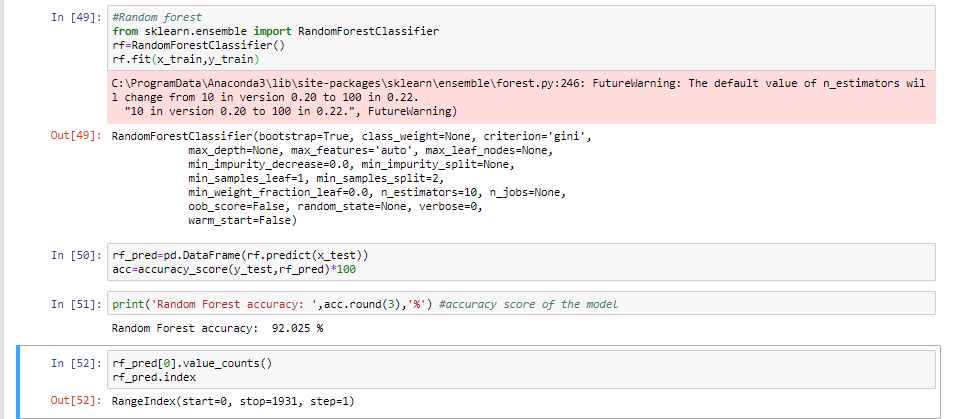
****

Let us check the random forest classifier for this model. What is Random Forest?

Random forests or random decision forests are an ensemble learning method for classification, regression and other tasks that operates by constructing a multitude of decision trees at training time and outputting the class that is the mode of the classes (classification) or mean prediction (regression) of the individual trees.

Now, let us use Random Forest Classifier for our analysis and predict values using random forest classifier and check the accuracy score.

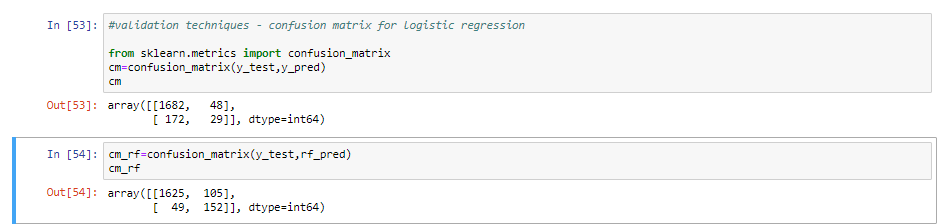
The code and output are:



The random forest accuracy is 92.025%.This certainly means that random forest is a better method for this data. Let us validate our analysis using confusion matrix and k-fold cross validation.

**Confusion Matrix**: A confusion matrix is a table that is often used to describe the performance of a classification model (or "classifier") on a set of test data for which the true values are known. The confusion matrix itself is relatively simple to understand, but the related terminology can be confusing.

Now, let us create a confusion matrix for the same.



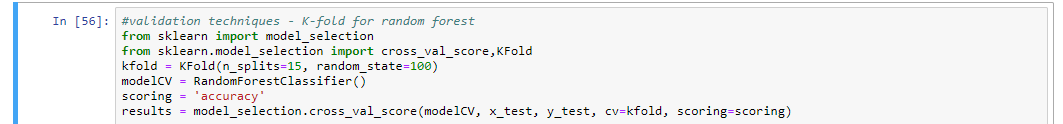
**K-Fold Cross Validation**: Cross-validation is a statistical method used to estimate the skill of machine learning models. It is commonly used in applied machine learning to compare and select a model for a given predictive modeling problem because it is easy to understand, easy to implement, and results in skill estimates that generally have a lower bias than other methods.

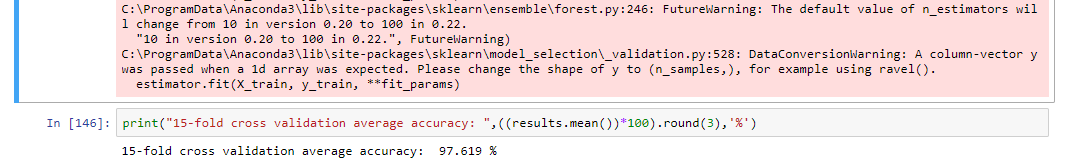
The procedure has a single parameter called k that refers to the number of groups that a given data sample is to be split into. As such, the procedure is often called k-fold cross-validation. When a specific value for k is chosen, it may be used in place of k in the reference to the model, such as k=10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

The general procedure is as follows:

1. Shuffle the dataset randomly.
2. Split the dataset into k groups
3. For each unique group:
   1. Take the group as a hold out or test data set
   2. Take the remaining groups as a training data set
   3. Fit a model on the training set and evaluate it on the test set
   4. Retain the evaluation score and discard the model
   5. Summarize the skill of the model using the sample of model evaluation scores

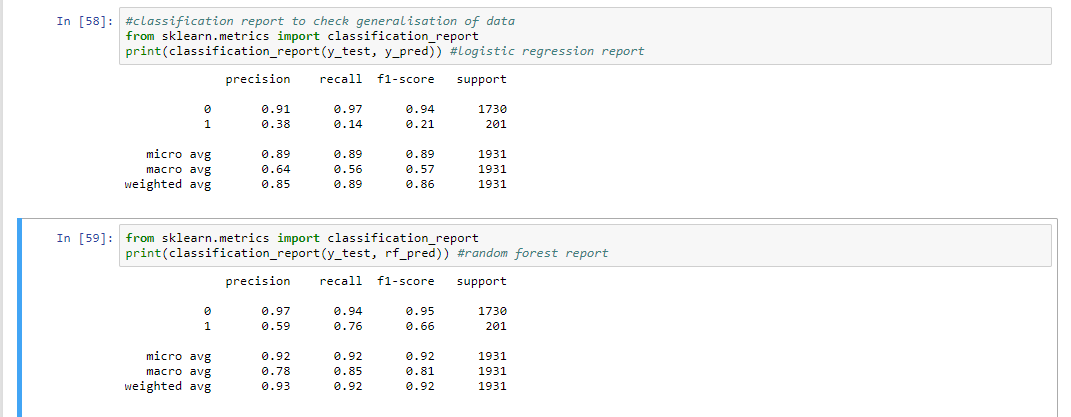
Now, let us use K-fold Cross-Validation to check overfitting of model. The code and output are:





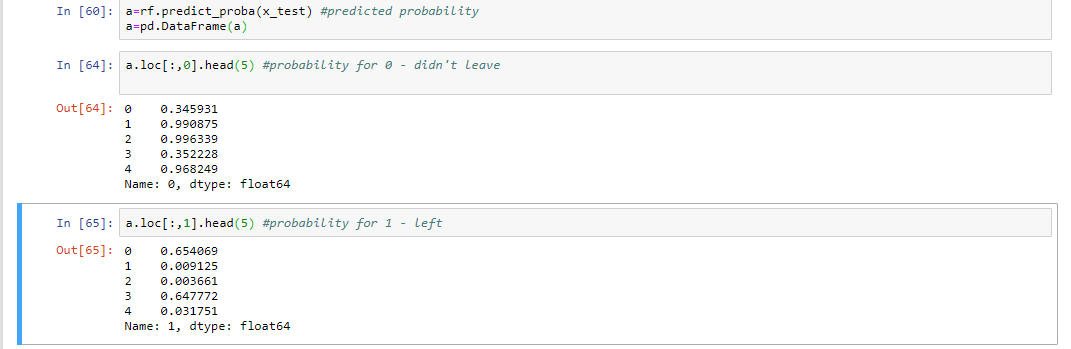
We can see that our cross validation accuracy is **97.619%**, which is almost as much as our random forest accuracy. Hence, we can say that there is negligible overfitting of our model.

Now, let us see a classification report to check the generalization of our data. This will tell if our analysis is generalized to the whole data set or not.

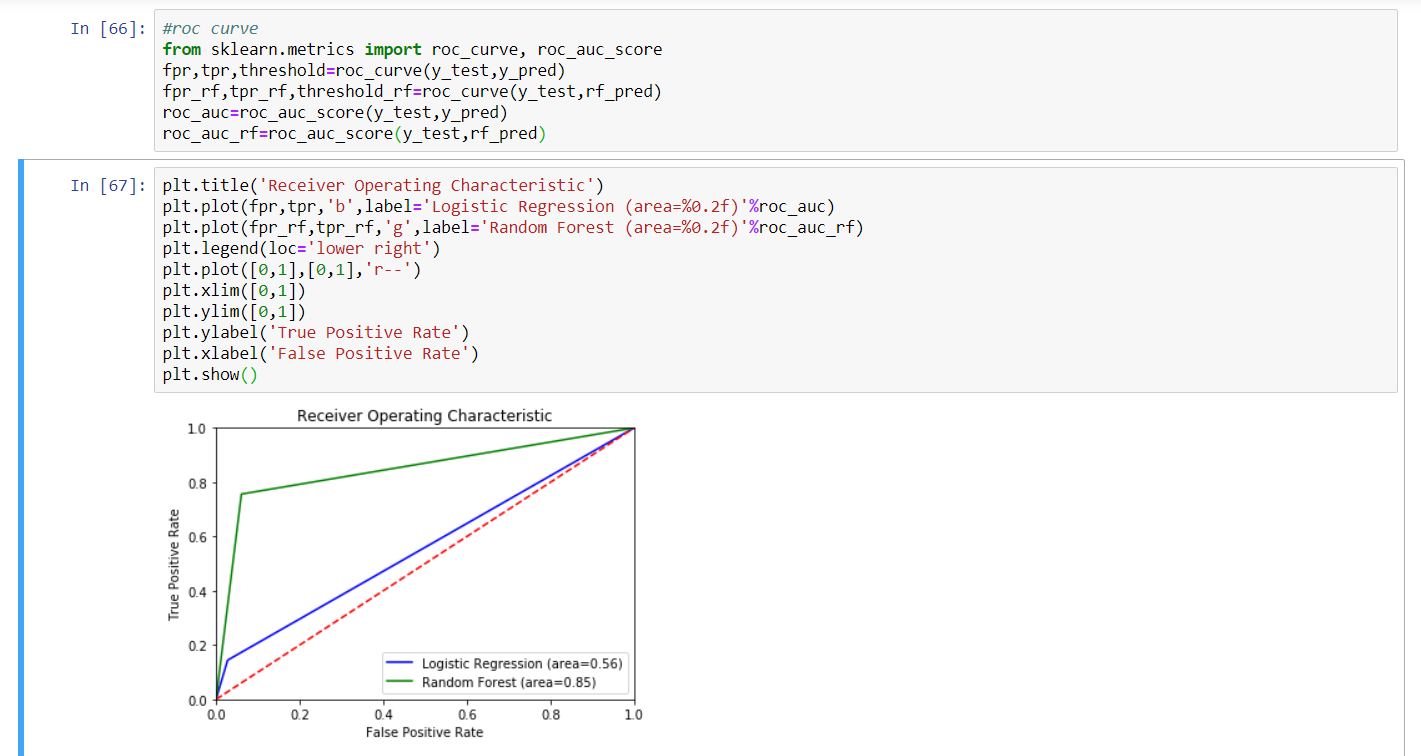


According to our report, the recall value for logistic regression is 0.97, which means it is more generalized than not, whereas the recall value for random forest is 0.94 which means it is 94% generalized. Hence, we can conclude that both our models are pretty generalized.

Let us check the predicted probability for both the cases- 0 for did not leave and 1 for left. The code and output are:



Now, let us make an ROC (Receiver Operating Characteristic) Curve to understand which method is better for this data set – logistic regression or random forest.



Here, we can infer that Random Forest is better than Logistic Regression in this case.

# References :

* <https://www.academia.edu/39249197/HR_ANALYTICS_A_MODERN_TOOL_IN_HR_FOR_PREDICTIVE_DECISION_MAKING>
* <https://github.com/akjadon/Finalprojects_DS/tree/master/HR_Analytics>
* <https://www.dataschool.io/simple-guide-to-confusion-matrix-terminology/>
* <https://machinelearningmastery.com/k-fold-cross-validation/>
* <https://www.cgi.com/sites/default/files/hr_whitepaper.pdf>
* <https://en.wikipedia.org/wiki/Logistic_regression>

\*------This Project is available to be seen on GITHUB follow the below link------\*

<https://github.com/jitroy160/Final_Projects/blob/master/Final_Projects/Final_Project_CarPrice_Prediction.ipynb>