**Company X’s Attrition Control Analysis using Random Forest Classifier**

**Project Report**

**By**

**Abdulkabir Tola Ojulari**

**Submitted to Hash Analytics**

**On June 18, 2020**

**Preface**

The analysis was carried out to determine which of the input variables influence the attrition in the company X.

In this project, Python was considered as language for the analysis and Spyder IDE was used to write the code.

The analysis was conducted by comparing the output result of Decision Tree Classifier with Random Forest Classifier, from which I reached a conclusion that the Random Forest Method was efficient with the accuracy of 99% and error rate of 1% compared with the Decision Tree Model with 95% accuracy rate and 5% error rate. It is very essential to compare models to be sure of accuracy and performance of the algorithm. Even though the result of the two model indicate Satisfaction Level contributes most to the factors for those who have left and prone to leave, it is required to go with the model with high performance and accuracy rate.

**In this report, the two analyses were divided into two phases; phase one explains the result of the Decision Tree Classification Model and phase two explains the Random Forest Classification Model.**

**Objective**

To conduct an analysis on the company X data to find out which of the factors that contribute to the attrition and find out the type of the employees that are prone to leave next.

**PHASE ONE**

**Preliminary Hypothesis**

Reading through the problem definition, I understand it is a classification problem (i.e. leaving or not leaving based on input values) and what came to my mind was to try using Decision Tree Classifier. The goal is to create a model that predicts the value of a target variable by learning simple **decision** rules inferred from the data features.

Having got 95% from **Decision Tree Classifier**, I know the model can be accentuated by exploring a better performer by using ensemble classifier

**Decision Tree Model**

**Confusion Matrix**

**A picture containing clock

Description automatically generated**

**Accuracy Rate** = 0.95

**Error Rate** = 0.05

**Type 1 Error** = 133

**Type 2 Error** = 89

Accuracy 

**Feature Importance**

**A screenshot of a cell phone

Description automatically generated**

**A screenshot of a cell phone

Description automatically generated**

**Bar chart Diagram Illustration from Decision Tree Classifier Model**

As depicted on the bar chart, **Satisfaction Level** plays most significance in the company X’s attrition, followed by **“Time spend Company”, “Last Evaluation”, “Number project”.** There is a clear distinction from the two models, even though the results are relatively close

**PHASE TWO**

**Pre-processing**

It behooves data quality is taken into consideration in this phase to ascertain, every necessary input variable is captured, cleaned and processed for the machine learning algorithm.

Import libraries – there are standard libraries built on python. In this project, Sci-Kit Learn was considered because of its wide range of relevance in machine learning analysis.

A picture containing drawing, flower

Description automatically generated

Dataset was divided into categories (original file) i.e. **Existing Employees sheet** and **Employees who have left sheet**. A new column was introduced to each of the sheets of the excel file (dataset) with a distinct name; **attrition\_status** andthe sheets were appended and randomized. **Existing Employee** with an added column; attrition\_status populated with 0s to denote “not left” and Employee Who have left’s attrition status is denoted with 1s

A picture containing screen, monitor, holding, phone

Description automatically generated

A picture containing screenshot, building

Description automatically generated

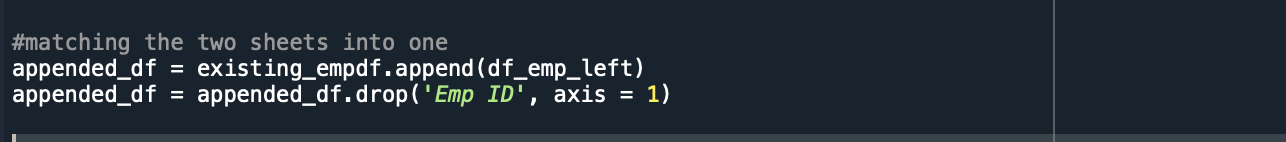
Fig 1: DataFrame of Existing Employee

A picture containing building, laptop

Description automatically generated

Fig 1b: DataFrame of Employees who have left

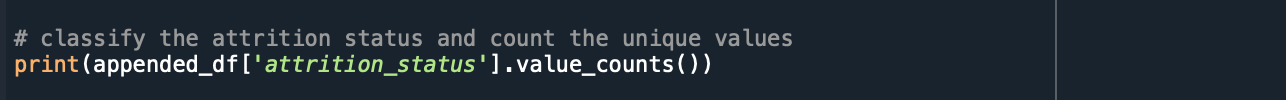
At this time, I have a single DataFrame named **appended\_df**



**Drop Unnecessary Column**

I dropped the column that may not have an impact of the analysis to narrow my focus on specificity.

Trying to understand the **attrition\_status** value, the count and classification based on Existing Employees and Employees who have left. This is a check to know if the record is absolutely intact as it before the merging.



Output 

Describing the dataset



The dataset contained the following fields

* Satisfaction Level
* Last Evaluation
* Number of projects
* Average Monthly Hours
* Time spend Company
* Dept
* Salary
* Work accident
* Attrition status

A picture containing black

Description automatically generated

Check column(s) with categorical values

A picture containing holding, player

Description automatically generated

Output

A screenshot of a cell phone

Description automatically generated

There are two columns with non-numeric values, the summary of the values is given in the figure above.

Looking out for the correlation of the column across the DataFrame



**Technique and methodology Adopted**

In this project, I tried to adopt the combination of descriptive analytics and prescriptive analytics.

* Descriptive analytic approach – what are the observations to consider for various hypothesis about employee attrition
* Prescriptive analytic approach – What type of employees are prone to leave

Transforming categorical columns into numerical values (machine learning only deals with numerical input). In this project, I opted for LabelEncoder for fast and easy transformation. LabelEncoder was imported from Scikit-learn

A screenshot of a cell phone

Description automatically generated

Splitting the dataset into training and test sets. In case, the independent variables are connoted with X\_train and X\_test while the target variable is connoted with Y\_train, Y\_test.

With 30% test\_size which divided the dataset in proportion of 30:70 (test set is 30%, training set is 70%). It’s practice-recommendation to have higher percentage for the training set, where several variables will be readily made available to train the machine learning algorithm.

A picture containing food

Description automatically generated

# ****Train Model****

After all the work of data pre-processing, the model is created and trained using Scikit-learn. The model was instantiated and fitted on the training data. (Again setting the random state for reproducible results).

**Fitting random forest classifier**

A screenshot of a cell phone

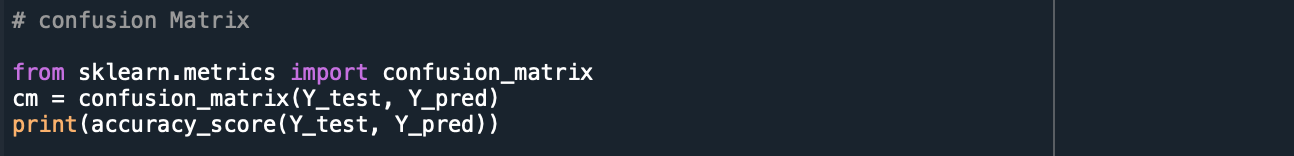
Description automatically generated

The model has now been trained to learn the relationships between the features and the targets.

**Determine Performance Metrics**

To put our predictions in perspective, we can calculate an accuracy using confusion Matrix, accuracy score and classification report

**Confusion Matrix**



Output

Confusion matrix was used to test the accuracy of the prediction

Predicted values

A picture containing object, clock

Description automatically generated

Actual Value

(TP + TN) / (TP + TN + FN + FP)

**Type 1 error**: 6

**Type 2 error**: 35

**Percentage of Accuracy**: (3423 + 1031) / (3423 + 8 + 38 + 1031) = 0.9898

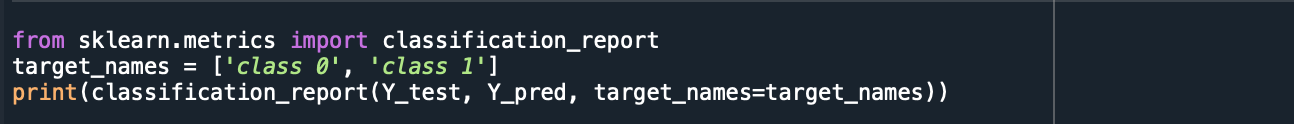
**Percentage of Error**: (38 + 8) / (3423 + 8 + 38 + 1031) = 0.0102

**Accuracy score**



The model measures high accuracy rate which implies the machine learning algorithm is efficient.

**Classification Report**



A Classification report was used to measure the quality of predictions from a classification algorithm. How many predictions are True and how many are False? More specifically, True Positives, False Positives, True negatives and False Negatives are used to predict the metrics of a classification report as shown below

A picture containing outdoor, black, monitor, side

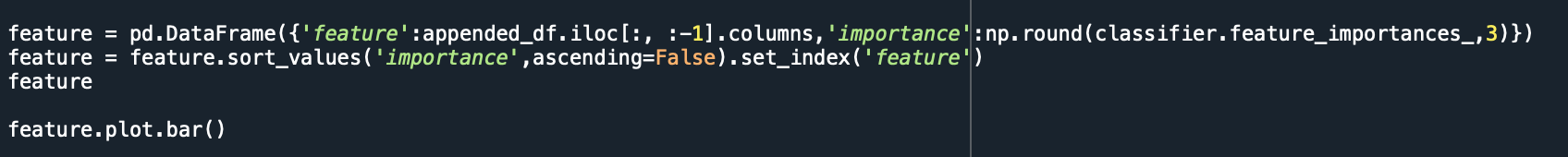
Description automatically generated

The report shows the main classification metrics precision, recall and f1-score on a per-class basis. The metrics are calculated by using true and false positives, true and false negatives. Positive and negative in this case are generic names for the predicted classes. There are four ways to check if the predictions are right or wrong:

1. **TN / True Negative:**when a case was negative and predicted negative
2. **TP / True Positive:**when a case was positive and predicted positive
3. **FN / False Negative:**when a case was positive but predicted negative
4. **FP / False Positive:**when a case was negative but predicted positive

**Feature Importance**

In order to quantify the usefulness of all the variables in the entire random forest, the relative importances of the variables are considered for exploration. The importances returned in Skicit-learn represent how much including a particular variable improves the prediction.



**Summary of the report**

A screenshot of a cell phone

Description automatically generated

**Satisfaction Level** appears to be the most important feature followed by the **Time Spend Company**, **Number of Project**, **Average Monthly Hour**, **Last Evaluation**

It is very interesting that “Promotion Last 5 Years” plays least significance in the determination of the reason why the employees are prone to leave and the basis for those who have left.

Employees that fall under **Satisfaction level**, **Time spend company**, **number of project**, **average monthly hour** and **last evaluation** are prone to leave and these factors have contributed to the reasons for those who have left with satisfaction level playing the most important role.

By looking at the **Satisfaction Level column**, we can easily deduce that with exceptions, the majority of the employees

are below 1 (i.e fall between between 0.09 – 0.99) while few are ranked on 1

**Time Spend Company**, looking at the two sheets, the employees who have left fall between 2 – 6 years which also

forms second most importance input value.

**Number of project** also plays an important role which implies that the categories of employees involved are prone to leave and are among those that have left.

**Average Monthly Hour** also contributes to the reason employees are prone to leave the company, looking at the antecedence of the employees who have left, the average monthly hours increased to highest threshold compared to the existing employees, which implies when they reach the peak of intolerable average monthly hour they may leave

**Last Evaluation as** indicated on the bar chart has a close relationship with the **average monthly hour**, this also plays an

important role in determination of the reason why employees left and may leave.

In future implementations of the model, we can remove those variables that have no importance to the prediction so that we can get a better performance.

## **Visualizations**

The chart below is a simple bar plot of the feature importances to illustrate the disparities in the relative significance of the variables.

A screenshot of a cell phone

Description automatically generated

Fig The Bar chart representing the features against the importance level.

**Conclusion**

Taking into account the comparison of the two models, Decision Tree Classification Model and Random Forest Classification, it is obvious that the Random Forest Model performed better with accuracy of 99% and error rate of 1%. In this project, the process controls for overfitting and produced a very robust, high-performing model.

Random forest uses decision trees as base learners. The base learners are high variance, low bias models. The variance of the overall model is reduced by aggregating the decisions taken by all base learners to predict the response variable. The idea is to ensure that each base learner learns a different aspect of data. This is achieved via both row and column sampling.