**Documentation for predicting of employee performance**

Human Resources Management (HRM) has become one of the essential interests of managers and decision makers in almost all types of businesses to adopt plans for correctly discovering highly qualified employees. Accordingly, managements become interested about the performance of these employees. Specially to ensure the appropriate person allocated to the convenient job at the right time.

From here, the interest of supervised machine learning specifically decision tree algorithm’s role has been growing that its objective is to discover knowledge that is used to predict employee’s performance in the organization from huge amounts of data sets.

In this documentation, decision tree algorithm techniques is utilized to build a classification model for predicting employees' performance using a precise and real dataset the decision tree algorithm technique is used for building the classification model and identifying the most effective factors that positively affect the performance.

Classification is the most commonly applied decision tree technique, which employs a set of pre-classified examples to develop a model that can classify the population of records at large. This approach frequently employs decision tree algorithms. The data classification process involves learning and classification. In Learning, the training data are analyzed by classification algorithm. In classification, test data are used to estimate the accuracy of the classification rules. The classifier-training algorithm uses these pre-classified examples to determine the set of parameters required for proper prediction. The algorithm then encodes these parameters into a model called a classifier.

Decision tree has been used for making meaningful decision for the Employee. Based on the employee’s performance results possible to take decision whether advanced training, talent enrichment or further qualification required or not. These applications also help administrative staff to enhance the quality of the organizations or companies. Each branch node represents a choice between a number of alternatives, and each leaf node represents a decision. A method based on this approach use an information theoretic measure, like entropy, for assessing the prediction power of each attribute. It is a tree-shaped structures that represent sets of decisions. These decisions generate rules for the classification of a dataset. There are two operations in decision tree as follows:

**Training**: The records of employee with known result is trained as attributes and values which is used for generating the decision tree based on the information gain of the attributes.

**Testing:** The unknown records of employee are tested with the decision tree developed from the trained data for determining the result.

Now let us proceed to some modules we have used to implement the data sets in the prediction of employee’s performance.

**Pandas:** it is a python module used to load the datasets from existing storage, storage can be SQL Database, CSV file, and Excel file. Pandas Data Frame can be created from the lists, dictionary, and from a list of dictionaries.

There are [multiple ways to select](http://pandas.pydata.org/pandas-docs/stable/indexing.html#different-choices-for-indexing) and index rows and columns from pandas data frame.

Each row in your data frame represents a data sample.

Each column is a variable, and is usually named.

  In this module there are options to achieve the selection and indexing activities in Pandas. these can be:

1. Selecting the data by row numbers(.iloc)
2. Selecting the data by label or by conditional statement(.loc)

**Matplotlib:** Matplotlib is an amazing visualization library in Python for 2D plots of arrays.

**Preprocessing:** The sklearn. preprocessing package provides several common utility functions and transformer classes to change raw feature vectors into a representation that is more suitable for the estimators.

 Whenever the data is gathered from different sources it is collected in raw format which is not feasible for the analysis. So we have to preprocess the data before applying it to the algorithm.

**LabelEncoder:** In ML models we are often required to convert the categorical i.e text features to its numeric representation. used to encode target values, i.e. y, and not the input X.

**Standardscaler:** The idea behind StandardScaler is that it is used to normalize the data set such that its distribution will have a mean value 0 and standard deviation of 1

**model\_selection:**  it is python module used for selecting data sets with best performance

**train\_test\_split:** used to split the data set into a training set and a test set.

**accuracy\_score: python module used to calculate accuracy of data sets.**

**classification\_report:** shows a representation of the main **classification** metrics on a per-class basis

**confusion matrix: it** 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. It allows the visualization of the performance of an algorithm.  
It allows easy identification of confusion between classes e.g. one class is commonly mislabeled as the other. Most performance measures are computed from the confusion matrix.

**DecisionTreeClassifier: it** is a flowchart-like tree structure where an internal node represents feature (or attribute), the branch represents a decision rule, and each leaf node represents the outcome