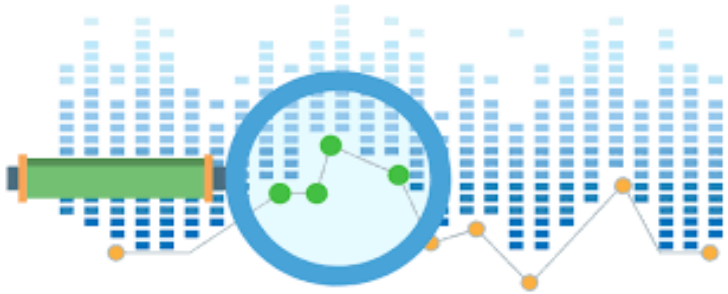
Methodology For Employee Attrition Control & Finance

TakenMind Global Data Analytics Internship Report, 2019

Name: Jimoh Abdulganiyu
Status: Graduate Student Intern
Tool: Python 2.7.16 & PyCharm
jimohabdulganiyuj@gmail.com

TakenMind Inc.
Financial Services
@Taken_Mind
https://takenmind.com/



Presentation Objectives

- Propose Methodology/Technique
  - **Select Target Variable**
  - **Get Historical Dataset**
  - **User-Defined Function**
  - **Build the Model (Random Forest, K-NN, & SVM)**
  - **Prediction algorithms Implementation & Data Analysis**
  - **Application of confusion matrices & probability**

SELECT TARGET VARIABLE

In the company X, the metric to be predict is what type of employee are leaving & employee that are prone to leave next. If one of three(3) algorithm to be implement is able to predict with higher accuracy measure, I will be getting back an exact employee that are prone to leave next. And with Python built-in analysis tools I will be able to know what type of employees are leaving.

In this case the value to be predict are "categorical target variable" resulting from a "discrete choice" or "continuous target variable" resulting from "numeric value"

GET HISTORICAL DATASET

The dataset comprises of existing employees (11,428) and employees who have already left (3,571). At this stage there is need to carry out cleaning and formatting of dataset provided.

USER-DEFINE FUNCTION

This will be two self define function which will be invoke for **“prediction”** through **“cross_validation_prediction”** built-in import from **“sklearn library”** and the **“accuracy measure function”** through **“metrics”** import from **“sklearn library”** as well.

BUILD THE MODELS

Model to build could be a regression or a classifier, it depends on the target data. This model is adapted from previous work done by authors.

- Random Forest Algorithm:
  - Splits are chosen according to a purity measure:
    - E.g. squared error (regression), Gini index or deviance (classification)
  - How to select N
    - Build trees until the error no longer decreases
  - How to select M
    - Try to recommend defaults, half of them and twice of them and pick the best

BUILD THE MODELS

- K-Nearest Neighbors Algorithm:
  - Calculate similarity based on distance function
  - Find K-Nearest Neighbors
  - Determine parameter K = number of nearest neighbors
  - Calculate the distance between the query-instance and all the training samples
  - Sort the distance and determine nearest neighbors based on the K-th minimum distance
  - Gather the category of the nearest neighbors
  - Use simple majority of the category of nearest neighbors as the prediction value of the query instance

BUILD THE MODELS

- Support Vector Machine Algorithm:
  - Choose the SVM class to be use
  - Train Test split
  - Training the Algorithm
  - Making Prediction
  - Evaluating the Algorithm: Confusion matrix, recall & F1
  - Show Results

PREDICTION ALGORITHMS IMPLEMENTATION AND DATA ANALYSIS

At this point the three prediction algorithms will be implement in Python programming language using already made built-in libraries of machine learning algorithm, then algorithms comparison will be done appropriately to know which one of the algorithms will have best accuracy measure while predicting the type of employee that are leaving and the employee that are prone to leave company X. Finally, analysis of the input and output files will be carry out appropriately.

APPLICATION OF CONFUSION MATRICES & PROBABILITY

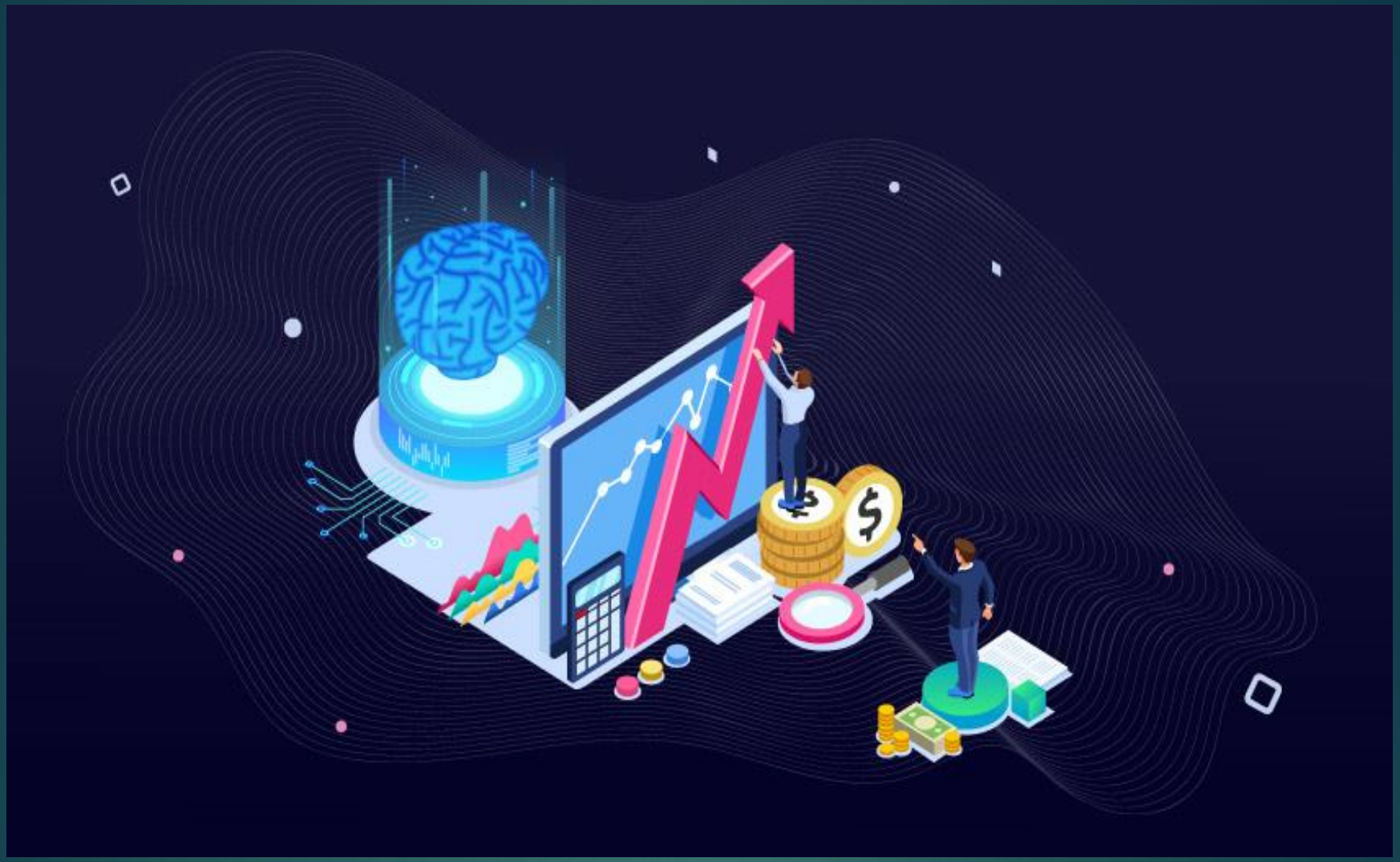
a. Confusion matrices will be calculated for each algorithm models.
b. Confusion matrices will also be draw as well for heatmap plot visualization.
c. Probabilities technique will be use to know the exact probability of employees that are prone to leave company X.

CONCLUSIONS

RF is fast to build. Even faster to predict! Practically speaking, not requiring cross-validation alone for model selection significantly speeds training by 10x-100x or more.

K-NN is robust to noisy training data & effective if the training data is large.

SVM is effective in high dimensional spaces. Still effective in cases where number of dimensions is greater than the number of samples.



Thank You!