Technology Bucket : Finance
Company Name : FIS SOLUTIONS(INDIA)
Team Leader Name : Prabhu Sharan Singh

Category: Software
Problem Statement ID : AK2
College Code : 1-3513287662

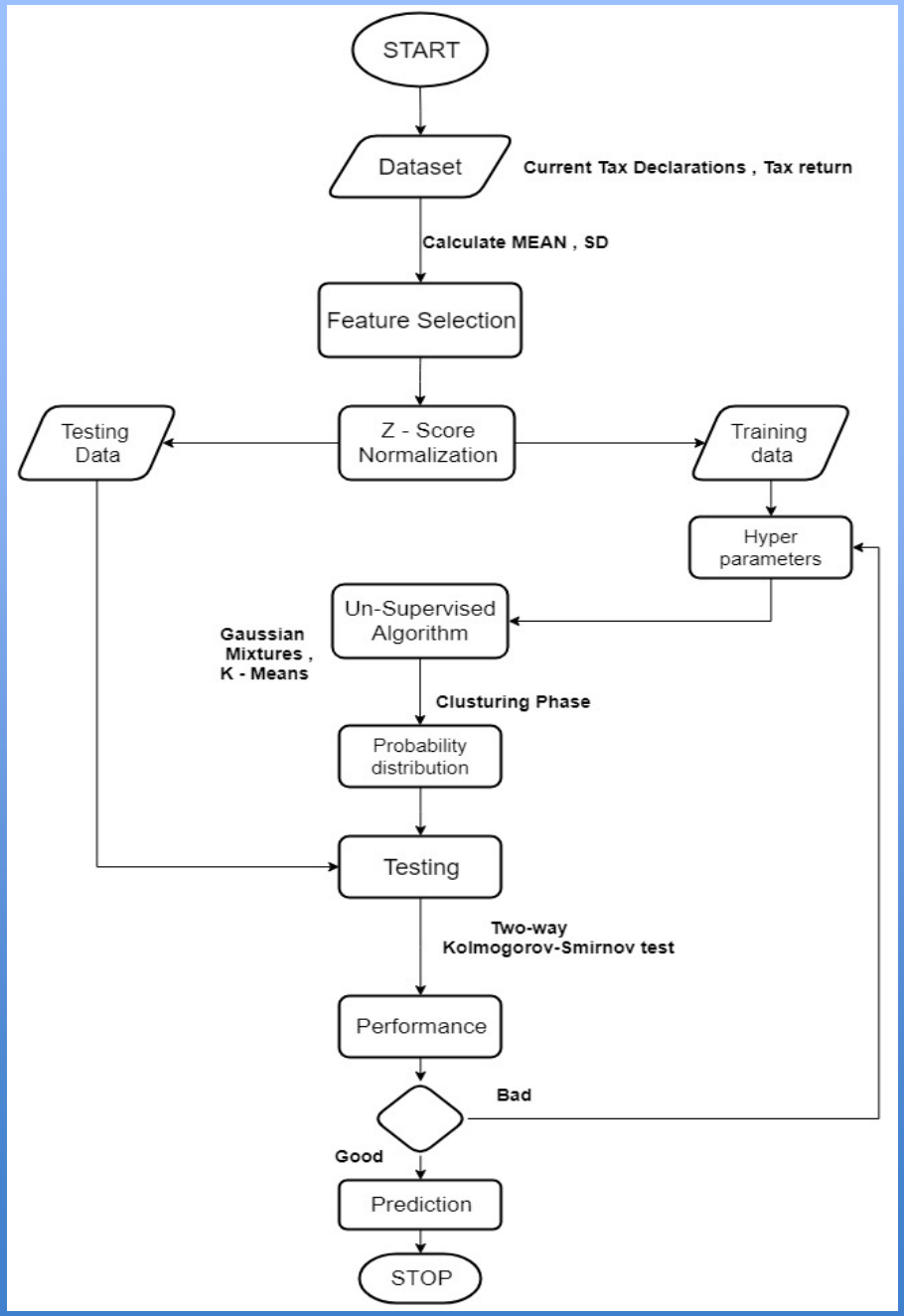
IDEA

In many scenarios, tax payers must declare the amount according to the tax base in some process and pay a percentage of that amount. This implies that many tax payers under-report earnings to reduce their taxes, as they have no incentive to report the actual amount.

What we are going to build at this hackathon is a software that uses *Unsupervised* methodology to detect and score tax payers under-reporting their tax base in order to pay less taxes than they should. The main assumption of this strategy is that similar tax declarations according to their features should pay the same amount of money.

This software will work in a three-fold process:
1. First, a clustering phase is made, grouping similar tax declarations according to the values of their features.
2. Second, we adjust a probability distribution to the tax bases reported in each cluster.
3. Finally, we detect suspicious declarations using a quantile of the adjusted distribution.

The flow-chart of our idea implementation is given below:


START
Dataset
Current Tax Declarations , Tax return
Calculate MEAN , SD
Feature Selection
Testing Data
Z - Score Normalization
Training data
Hyper parameters
Un-Supervised Algorithm
Gaussian Mixtures , K - Means
Clusturing Phase
Probability distribution
Testing
Two-way Kolmogorov-Smirnov test
Performance
Bad
Good
Prediction
STOP

TECHNOLOGY STACK

Python is the language of choice for this software because of its ease of use and extensive ecosystem of libraries in the field of machine learning. The libraries I am going to use:

Sci-kit learn: For implementation of Supervised and Unsupervised learning algorithms (like Decision Trees, K-Means, Gaussian mixture models, Bayesian Gaussian Model)

Pandas: For analysing and managing data

Matplotlib: For plotting and showing different graphs/images etc.

Numpy: For numerical calculation on arrays

Desired results/accuracy is hoped to achieve with help of Sci-kit learn and Python. May pivot to state of the art Neural Networks like Stochastic Maximum likelihood learning, Bernoulli Restricted Boltzmann machines for better results/accuracy.

USE CASE

Tax fraud is a global phenomenon, affecting society as a whole. Recent studies have estimated that governments around the world lose approximately US$500 billion annually. As a result, mid-income countries like India that have a greater reliance on tax revenues for fiscal planning, are largely affected by budget shortages, limiting the reach of their public investment.

Considering these stats, it seems like we could use a tool/software which would help us to distinguish between fraudulent and non-fraudulent activities, thereby classifying fraudulent tax payers or activities and enabling tax authorities to take actions to decrease the impact of fraud.

DEPENDENCIES

Our software will fundamentally depend upon the tax declaration and lifestyle information (data) of the potential tax evaders.

Features selection is based on economical environment which relates to the source of income of tax payers.

Data should be normalized after feature selection to improve the quality of information and decrease the fluctuation.

SHOW STOPPER

Supervised machine learning techniques tend to fail in the context of tax fraud detection since tax authorities, have extremely low amounts of historic labeled data due to the high cost in time and resources of auditing. This greatly hinders the generalization power of supervised algorithms and thus their usefulness.

Our un-supervised machine learning model eliminates the use of labeled historic data and make use of tax declarations and lifestyle information of individual tax payers which is much easy to obtain. It will take much less time for classification of fraudulent behavior in Tax payers than normal supervised learning algorithms.

Future work includes the evaluation of our model in a scenario where tax declarations are better characterized by their features and where more data is available.

Results obtained from our software would be without any false identification (false positive) or false exclusion (false negative) results.