Fraud detection Through Machine Learning Algorithm
Sumit Kumar¹ Pankaj Rodage² Rrushikesh Randive³ Prof. Yogesh Kadam⁴

¹,²,³,⁴Department of Computer Engineering
Bharati Vidyapeeth College of Engineering Lavale, India

Abstract— Machine Learning (ML) has unfold from the Artificial Intelligence, a field of computer science. Machine Learning (ML) is multidisciplinary field, a combination of statistics and computer science algorithms which is widely used in predictive analyses and classification. Over the past decades, Artificial intelligence (AI) stream has become the broad and exciting field in computer science as it prepare the machines to perform the tasks that human being may do, and it aims to train the computers to solve real world problems with the maximum success rate. As perceiving scientific growth and advancement in technology AI systems are now capable to learn and improve through past experiences without explicitly assistance code if they exposed to new data. Eventually it leads to technology of Machine learning (ML) which uses learning algorithms to learn from the data available. Machine Learning uses data mining techniques to extract the information from the huge size datasets. ML and Data Mining techniques explore data from end to end to find the hidden patterns inside dataset. Machine Learning and data mining algorithms has been deployed in various fields such as Computer networking, travel and tourism industry, finance forecasting, telecommunication industry and electric load forecasting and so on at least.

Keywords: Machine Learning, Deep Learning, SVM, Naïve Bayes, Fraud Detection, Clustering

I. INTRODUCTION

Machine Learning has always been useful for solving real-world problems. Nowadays, it is widely used in every field such as medical, e-commerce, banking, insurance companies, etc. Earlier, all the reviewing tasks were accomplished manually. But with the increase in the processing power of systems and the advancement in statistical modelling, the acceptance of Machine Learning in every sector has increased. In this blog, we will see fraud detection algorithms using Machine Learning. For years, fraud has been a major issue in sectors like banking, medical, insurance, and many others. Due to the increase in online transactions through different payment options, such as credit/debit cards, PhonePe, Gpay, Paytm, etc., fraudulent activities have also increased. Moreover, fraudsters or criminals have become very skilled in finding escapes so that they can loot more. Since no system is perfect and there is always a loophole them, it has become a challenging task to make a secure system for authentication and preventing

II. MOTIVATION

1) Massive Data Consumption from Unlimited Sources
2) Rapid Analysis Prediction and Processing
3) Improves Precision of Financial Rules and Data

III. PROBLEM STATEMENT

In 2000, Enron was one of the largest companies in the United States. By 2002, it had collapsed into bankruptcy due to widespread corporate fraud. In the resulting Federal investigation, there was a significant amount of typically confidential information entered into public record, including tens of thousands of emails and detailed financial data for top executives. In this project, put your new skills to use by building a person of interest identifier based on financial and email data made public as a result of the Enron scandal. Using machine learning algorithm to use by building an algorithm to identify Enron Employees who may have committed fraud based on the public Enron financial and email dataset. Machine learning is an application of artificial intelligence (AI) that provides systems the ability to automatically learn and improve from experience without being explicitly programmed. Machine learning focuses on the development of computer programs that can access data and use it learn for themselves.

IV. REQUIREMENT ANALYSIS

We should have python and sklear running on your computer, as well as the starter code (both python scripts and the Enron dataset). The starter code can be found in the final project directory of the codebase that you downloaded for use with the mini-projects. Some relevant files: 1) poi id.py: starter code for the POI identifier, you will write your analysis here 2) final project dataset.pkl: the dataset for the project, more details below 3) tester.py: when we will turn in your analysis for evaluation by a Udacity evaluator, you will submit the algorithm, dataset and list of features that you use (these are created automatically in poi id.py). The evaluator will then use this code to test your result, to make sure we see performance that’s similar to what you report. Transparency and for your reference. 4) emails by address : this directory contains many text files, each of which contains all the messages to or from a particular email address. It is for your reference, if you want to create more advanced features based on the details of the emails dataset.
V. ANALYSIS PHASE

1) The first step in any machine learning problem is to identify the data source and clearly defined questions.
2) The larger the size of the dataset, the greater the quality of the result.
3) In the majority of cases, increasing the size of the dataset yields better results than fine tuning the algorithms.
4) Train/Test dataset split: the dataset is divided into training and testing sets in order to estimate the algorithm’s performance on an independent dataset.
5) Identify potential over-fitting scenarios. The dataset should be sampled randomly in order to bypass situations where similar training points are sequentially ordered.
6) It is important to maximize the size of both the training set (for the best learning result) and the testing set (for the best validation). The K-Fold validation is used for this purpose.
7) K-Fold validation process: Divide dataset into K equal bins. Run K learning experiments using 1 training bin and K-1 testing bins. Average the test result for the K experiments.
8) Features are the properties of a dataset (e.g., name, age, salary) that give information about the training points (e.g., employees of the company).
9) Naive Bayes Classification: Naive Bayes methods are a set of supervised learning algorithms based on applying Bayes’ theorem with the “naive” assumption of conditional independence between every pair of features given the value of the class variable. Bayes’ theorem states the following relationship, given class variable and dependent feature vector through P(y | x1, ..., xn) = P(y)P(x1,...,xn|y)P(x1,...,xn). GaussianNB implements the Gaussian Naive Bayes algorithm for classification. The likelihood of the features is assumed to be Gaussian: P(xi | y) = \frac{1}{\sqrt{2\pi\sigma^2_y}} \exp\left(-\frac{(x_i-\mu_y)^2}{2\sigma^2_y}\right). From sklearn.datasets import load_iris
From sklearn.model_selection import train_test_split
From sklearn.naive_bayes import GaussianNB
X, y = load_iris(return_X_y = True) Xtrain, Xtest, ytrain, ytest = train_test_split(X, y, testsize = 0.5, randomstate = 0) gnb = GaussianNB() ypred = gnb.fit(Xtrain, ytrain).predict(Xtest)
print("Number of mislabeled points out of a total 75 points: ", (Xtest != ypred).sum())

VI. PROTOTYPE

Feature processing after cleaning the data from outliers, I had to pick the most sensible features to use. First, I picked ‘from poi to this person’ and ‘from this person to poi’ but there is no strong pattern when I plotted the data so I used fractions for both features of “from/to poi messages” and “total from/to messages”.

VII. CONCLUSION

Naive Bayes gives the highest and best recall value; that is, the highest probability of getting a person of interest. However, we considered Decision Tree a better algorithm because it gives the most stable average result for accuracy, precision, and recall for all the three experiments and feature sets. Although, Naive Bayes algorithm gives the highest recall, it also seems heavily flawed with bias relative to Decision Tree algorithm. These numbers are quite good but we still can improve the strategy. One of the possible paths to improvement is digging in to the emails data more. The email features in the starter dataset were aggregated over all the messages for a given person. By digging into the text of each individual’s messages, it’s possible that more detailed patterns.

REFERENCES
