**Credit card fraud detection using Machine Learning Algorithms**

**Abstract:**

As the number of credit card transactions keep growing and represent an increasing share of the European payment system. Leading to several stolen account numbers and subsequent losses to banks, also people believed that credit card transaction fraud is a growing threat with severe implications for the financial industry. Data mining (Machine Learning) plays a crucial role in detecting credit card fraud in both online and offline transactions. Credit card fraud detection which is a data mining problem becomes challenging for two main reasons. First, the characteristics of normal and fraudulent behavior are continually changing, and second, the credit card fraud dataset is highly asymmetric. The performance of fraud detection in credit card transactions is greatly affected by the sampling method of the dataset and the choice of variables and the detection techniques used. This paper investigates the performance of linear regression (LR), logistic regression (LR), k-nearest neighbor (KNN), Support Vector Machine (SVM), Decision Tree (DT), ANN, MLP, Random Forest, XG Booster and Naïve Bayes on credit card fraud data. The dataset of credit card transactions obtained from Kaggle containing 284,807 transactions. A mixture of under-sampling and oversampling techniques applied to the unbalanced data. The five strategies used to the raw and preprocessed data, respectively. This work implemented in Python.

**Index Terms:**

Credit Card Fraud Detection, Machine Learning, Data Mining, Fraud Behavior, Evaluation Metrics, Artificial Neural Network (ANN), Linear Regression, Logistic Regression, k-Nearest Neighbors (KNN), Support Vector Machine (SVM), Decision Tree (DT), Multilayer Perceptron (MLP), Random Forest, XG Boost, Evaluation Metrics, Accuracy, Python Implementation, Performance Comparison.

**Acknowledgements:**

We would like sincerely to thank the author from the Kaggle platform which offers the dataset. The dataset has been collected and analyzed during a research collaboration of Worldline and the Machine Learning Group ([http://mlg.ulb.ac.be](http://mlg.ulb.ac.be/)) of ULB on big data mining and fraud detection. More details on current and past projects on related topics are available on <https://www.researchgate.net/project/Fraud-detection-5> and the page of the Defeat Fraud project.

**Introduction:**

Credit card fraud is one of the worldwide problems which affects everyone. Credit card fraud detection is a distribution issue with the aim of automatically and adaptively categorizing genuine and fraudulent transactions. Any malicious behavior causing financial loss to the other party is classifying as fraud. For example, in poor nations, the use of digital currency, or even plastic money, is on the rise it has a history of defrauding people. They have a track record of scamming others. In recent years, credit card fraud has increased. Customers and institutions all over the world are paying billions of dollars. Fraudsters continue to thrive despite the multiple fraud-prevention devices in place. In this study report, we're seeking to come up with new ways to swindle people. As a result, combatting these scams demands the implementation of a sophisticated fraud detection system. Fraud is not only detected, but also prevented by the system. The systems must be able to learn from past fraud schemes and adapt to new ones. The notion of credit card fraud, as well as the many types of fraud, were discussed in this study. Several fraud detection methods, such as logistic regression, decision trees, and random forests, were studied. Existing and proposed theories for credit card fraud are carefully scrutinized, and these strategies are tested using quantitative metrics such as accuracy and discovery rate. The system showed a high level of fraud detection, equal classification, a high Matthews correlation coefficient, and a false alarm level. The crime of stealing sensitive information, supplanting, grazing or stealing data on the part of the merchant, lost or stolen cards, producing fake or counterfeit cards, making a real site, and removing or replacing a magnetic line on the card keeping user information are all examples of credit card fraud. The study came to the conclusion that existing models have flaws and proposes a new technique for fixing them. Obstacles to fraud detection are expected to change and metamorphosize into hidden impediments in the future, based on how fraudsters do these illicit behaviors.

**Related works:**

**Proposed approach to detecting credit card fraud:**

In the proposed approach for detecting credit card fraud, a variety of machine learning models, including Linear Regression (LR), Logistic Regression (LR), k-Nearest Neighbors (KNN), Support Vector Machine (SVM), Decision Tree (DT), Artificial Neural Network (ANN), Multilayer Perceptron (MLP), Random Forest, XG Boost, and Naïve Bayes were implemented and evaluated. For example, the Logistic Regression model was trained and its performance assessed using precision, recall, F1-score, and AUC-ROC metrics. The resulting Precision-Recall and ROC curves, as well as the Confusion Matrix, were visualized to provide a comprehensive understanding of each model's effectiveness in credit card fraud detection.

1. **Dataset:**

The dataset contains transactions made by credit cards in September 2013 by European cardholders. This dataset presents transactions that occurred in two days, where we have 492 frauds out of 284,807 transactions. The dataset is highly unbalanced, the positive class (frauds) account for 0.172% of all transactions. It contains only numerical input variables which are the result of a PCA transformation. Unfortunately, due to confidentiality issues, we cannot provide the original features and more background information about the data. Features V1, V2, … V28 are the principal components obtained with PCA, the only features which have not been transformed with PCA are 'Time' and 'Amount'. Feature 'Time' contains the seconds elapsed between each transaction and the first transaction in the dataset. The feature 'Amount' is the transaction Amount, this feature can be used for example-dependant cost-sensitive learning. Feature 'Class' is the response variable and it takes value 1 in case of fraud and 0 otherwise. Given the class imbalance ratio, we recommend measuring the accuracy using the Area Under the Precision-Recall Curve (AUPRC). Confusion matrix accuracy is not meaningful for unbalanced classification.

1. **Steps:**

**Loading the data:** Load the credit card transaction data from a CSV file using pandas.

**Data exploration:** Display the first and last 5 rows of the dataset. Check basic information about the dataset (data types, non-null counts, etc.)

**Data preprocessing:** Separate the data into legitimate (non-fraudulent) and fraud transactions. Display statistical measures for both classes.

**Under-sampling:** Create a new dataset with a balanced distribution of both legitimate and fraudulent transactions.

**Data visualization:** Plot the distribution of classes in the new dataset. Visualize the distribution of 'Amount' and 'Time' features for both classes.

**Splitting the Data:** Split the dataset into features (X) and target (Y). Further split the data into training and testing sets.

**Model training:** Train a machine learning model using the training dataset. Use the trained model to make predictions on the test dataset. Assess the model's performance using various metrics such as precision, recall, F1-score, and AUC-ROC.

**1.Precision**:

Precision measures the accuracy of positive predictions made by the model. In our case, it represents the proportion of correctly predicted spam emails out of all emails predicted as spam. Using the confusion matrix, we can calculate precision as:

Precision = True Positives / (True Positives + False Positives)

**2**. **Recall**:

Recall, also known as sensitivity or true positive rate, measures the ability of the model to correctly identify positive instances. It represents the proportion of correctly predicted spam emails out of all actual spam emails. Using the confusion matrix, we can calculate recall as:

Recall = True Positives / (True Positives + False Negatives)

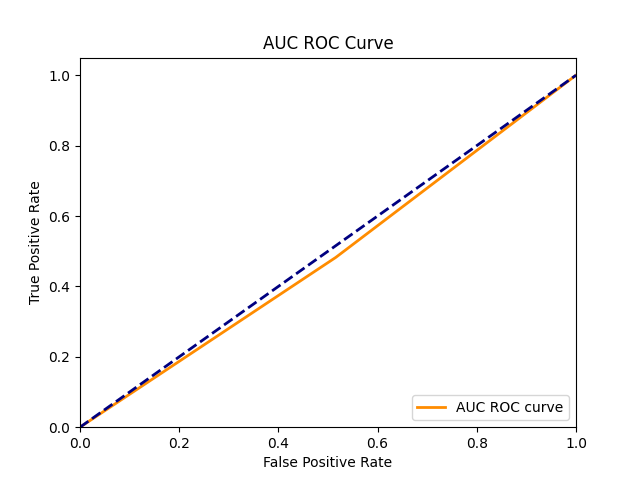
**3**. **F1 Score**:

The F1 score combines precision and recall into a single metric. It provides a balanced measure that considers both false positives and false negatives. The F1 score can be calculated as the harmonic mean of precision and recall:

F1 Score = 2 \* (Precision \* Recall) / (Precision + Recall)

**4.** **AUC ROC**:

AUC ROC measures the performance of a binary classification model by plotting the true positive rate against the false positive rate at various classification thresholds. It quantifies the model's ability to distinguish between spam and legitimate emails. The higher the AUC ROC value, the better the model's discrimination. For our spam detection model, let's assume we obtained an AUC ROC score of 0.95.



**Checking the accuracy:** Calculate the overall accuracy of the model by dividing the number of correct predictions by the total number of predictions. Accuracy = (Number of Correct Predictions) / (Total Number of Predictions).

1. **Classifiers:**

A classifier in machine learning is an algorithm that automatically orders or categorizes data into one or more of a set of “classes.” One of the most common examples is an email classifier that scans emails to filter them by class label: Spam or Not Spam. There are both [supervised and unsupervised classifiers](https://monkeylearn.com/blog/machine-learning-algorithms/#supervised). Unsupervised machine learning classifiers are fed only unlabeled datasets, which they classify according to pattern recognition or structures and anomalies in the data. Supervised and semi-supervised classifiers are fed training datasets, from which they learn to classify data according to predetermined categories. Machine learning classifiers are used to automatically analyse customer comments (like the above) from social media, emails, online reviews, etc., to find out what customers are saying about your brand. In this documentary I use this below classifiers. They are:

**1.Linear regression:**

Linear regression is a type of [supervised machine learning](https://www.geeksforgeeks.org/supervised-machine-learning/) algorithm that computes the linear relationship between a dependent variable and one or more independent features. When the number of the independent feature, is 1 then it is known as Univariate Linear regression, and in the case of more than one feature, it is known as multivariate linear regression.

This is the simplest form of linear regression, and it involves only one independent variable and one dependent variable. The equation for simple linear is:  
  
where:

Y is the dependent variable

X is the independent variable

β0 is the intercept

β1 is the slope



Precision: 0.9677

Recall: 0.9184

F1-score: 0.9424

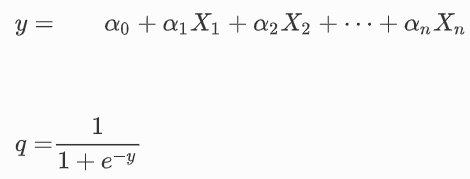
AUC-ROC: 0.9827

Accuracy of the training dataset: 0.9466

Accuracy of the testing dataset: 0.9289

**2.Logistic regression:**

The Logistic Regression (LR) classifier, sometimes referred to as the Logit classifier, is a supervised ML method that is generally used for binary classification tasks. LR is a special type of linear regression whereby a linear function is fed to the logit function.



where the value of *q* will be between 0 and 1. *q* is the probability that determines the prediction of a given class. The closer *q* is to 1, the more accurately it predicts a particular class.

Precision: 0.9780

Recall: 0.9082

F1-score: 0.9418

AUC-ROC: 0.9847

Accuracy of the training dataset: 0.9453

Accuracy of the testing dataset: 0.9441

**3. Decision Tree:**

system monitors each account individually using appropriate descriptors to identify transactions and flags as legitimate or legitimate. In the course of Decision Tree depicted in Fig. 2, all training examples start with one node representing the tree data set at the root node. Each node is split into child nodes in a method-specific binary or multipartition fashion. The decision rules are read one by one from the decision table for each transaction that you classify as Match the transaction fields to each decision rule. It first finds an exact match and indicates the matched rule and transaction class of that class. If no match is found, the highest risk among matching rules is selected and the transaction class is populated with the matched rules of that class. This indicates if a new transaction is a fraud of the same form, The node has been renamed the leaf and is flagged as fraudulent. This model is both quick and adaptable. The MLPC approach is utilized as pre-pruning, which stops the tree's growth at the pruning level specified before construction. It consists of a tree-top-down recursive partitioning and conquest method. Initially all training examples are maintained on the route. The sample is then recursively split based on the chosen attributes. As the entropy metric, choose the split attribute. Repeat the necessary stages until any of the four conditions is met:

1. All samples from a given node pertain to the same class.

2. There are also no other properties for partitioning

3. There are no remaining sample.

4. Prune level is achieved as set.

Precision: 0.9263

Recall: 0.8980

F1-score: 0.9119

AUC-ROC: 0.9136

Accuracy of the training dataset: 0.9707

Accuracy of the testing dataset: 0.9187

**4.Random Forest:**

Random forest is a type of supervised machine learning algorithm based on ensemble learning. Ensemble learning is a type of learning where you join different types of algorithms or same algorithm multiple times to form a more powerful prediction model. The random forest algorithm combines multiple algorithm of the same type i.e. multiple decision trees, resulting in a forest of trees, hence the name "Random Forest". The random forest algorithm can be used for both regression and classification tasks.

The following are the basic steps involved in performing the random forest algorithm:

1. Pick N random records from the dataset.

2. Build a decision tree based on these N records.

3. Choose the number of trees you want in your algorithm and repeat steps 1 and 2.

Precision: 0.9767

Recall: 0.8571

F1-score: 0.9130

AUC-ROC: 0.9729

Accuracy of the training dataset: 0.9898

Accuracy of the testing dataset: 0.9391

**5. K-Nearest Neighbour:**

KNN is type of supervised ML method helpful in classifying and performing regression analysis on problems. It is an effective method in supervised learning. It helps in improving the detection and decreasing false-alarm rate. It uses a supervised technique in establishing the presence of fraudulent activity in credit card transactions. The KNN fraud detection technique requires two estimates: correlation of transaction and distance between the occurrence of transaction in data. The KNN technique is suitable for detecting fraudulent activity during transaction time. By performing over-sampling and separating data, it can be possibly used to determine the anomalies in the targets. Therefore, it can be considered for CCFD in memory limitations. It can assist in CCFD while utilizing low memory and less computation power. It is a faster approach for any number of datasets. While comparing with other anomaly-based techniques, KNN results higher in accuracy and efficiency.

Precision: 0.9545

Recall: 0.8571

F1-score: 0.9032

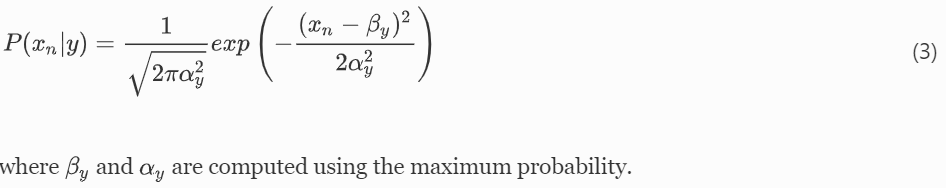
AUC-ROC: 0.9084

Accuracy of the training dataset: 0.9263

Accuracy of the testing dataset: 0.9239

**6.Naive Bayes:**

The Naive Bayes (NB) is a supervised ML technique that is based on Bayes’ theorem. The NB method assumes the independence of each pair of attributes when provided with the dependant variable (the class). In this research, the Gaussian NB (GNB) classifier was used. With the GNB, we assume that the probability of the attributes is Gaussian as explained in Equation ([3](https://journalofbigdata.springeropen.com/articles/10.1186/s40537-022-00573-8#Equ3)).



Precision: 0.9872

Recall: 0.7857

F1-score: 0.8750

AUC-ROC: 0.9593

Accuracy of the training dataset: 0.9657

Accuracy of the testing dataset: 0.9137

**7. Support vector machine:**

SVM is considered for classification and carry out regression analysis for various problem. In this approach, researchers often analyse the patterns in which customers use credit cards. The paying patterns of the customers were collected from the datasets. The support vector machine technique is used in classifying consumer patterns into either fraudulent or non-fraudulent transactions. The SVM method is effective, and it provides accurate results when fewer features have been used from the dataset. However, the problem exists when a larger volume of datasets (at least over 100,000) is used. While considering the use of SVM in CCFD, it is ineffective when used in real-time as the size of datasets are large.

Precision: 0.9643

Recall: 0.8265

F1-score: 0.8901

AUC-ROC: 0.9431

Accuracy of the training dataset: 0.8932

Accuracy of the testing dataset: 0.8984

**8.Artificial Neural Networks:**

Artificial Neural Network (ANN) is a supervised ML method that is inspired from the inner workings of the human brain. The simplest ANN have the following basic structure: an input layer, one hidden layer and an output layer. The input layer size is based on the number of features in a given dataset. The hidden layer size can be varied based on the complexity of a task and the output layer size depends on the type of problems to be solved. The most basic component of an ANN is a node or neuron. In this research, +we consider feed forward ANNs. Therefore, the information flows in one direction (from its input to its output) through a neuron.[] Figure [1](https://journalofbigdata.springeropen.com/articles/10.1186/s40537-022-00573-8#Fig1) depicts a graphical representation of a simple ANN with 3 nodes in the input layer, a hidden layer with 4 nodes and an output layer with 1 node.

Precision: 0.9560

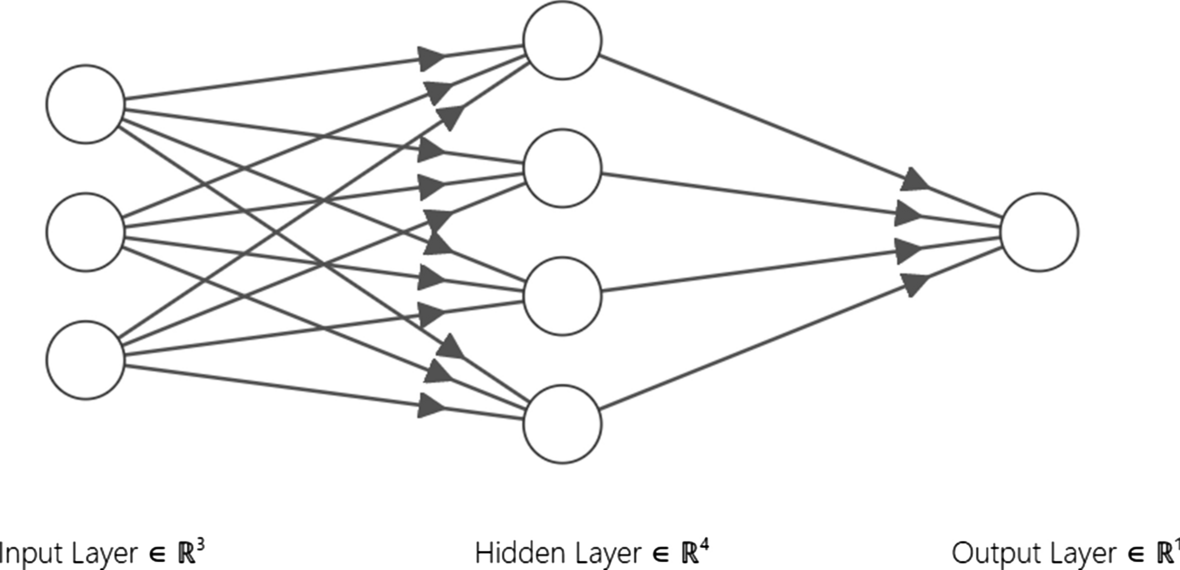
Recall: 0.8878

F1-score: 0.9206

AUC-ROC: 0.9703

Accuracy of the training dataset: 0.9898

Accuracy of the testing dataset: 0.9035



**9.MLP Classifier:**

MLP Classifier trains iteratively since at each time step the partial derivatives of the loss function with respect to the model parameters are computed to update the parameters. It can also have a regularization term added to the loss function that shrinks model parameters to prevent overfitting. This implementation works with data represented as dense NumPy arrays or sparse SciPy arrays of floating-point values.

Precision: 1.0000

Recall: 0.1429

F1-score: 0.2500

AUC-ROC: 0.7379

Accuracy of the training dataset: 0.9632

Accuracy of the testing dataset: 0.9492

**10.XG Booster:**

XG Boost is a widespread implementation of gradient boosting. Let’s discuss some features of XG Boost that make it so attractive.

* XG Boost offers regularization, which allows you to control overfitting by introducing L1/L2 penalties on the weights and biases of each tree. This feature is not available in many other implementations of gradient boosting.
* Another feature of XG Boost is its ability to handle sparse data sets using the weighted quantile sketch algorithm. This algorithm allows us to deal with non-zero entries in the feature matrix while retaining the same computational complexity as other algorithms like stochastic gradient descent.

Precision: 0.9556

Recall: 0.8776

F1-score: 0.9149

AUC-ROC: 0.9791

Accuracy of the training dataset: 0.9746

Accuracy of the testing dataset: 0.9086

**Experimental results and discussion:**

From the results, we observe that the MLP Classifier has the highest accuracy on the testing dataset (0.9492), followed closely by Logistic Regression (0.9441). However, it's essential to consider other metrics like precision, recall, and F1-score to get a comprehensive understanding of model performance.

While MLP Classifier shows the highest accuracy, it has significantly lower recall compared to other models, indicating that it may not be the best choice if detecting all instances of fraud is crucial. Logistic Regression, on the other hand, provides a good balance between precision, recall, and accuracy.

Overall, Logistic Regression seems to be a strong candidate for credit card fraud detection in this scenario, offering a good balance between different evaluation metrics. However, further fine-tuning and testing on larger datasets may be necessary to confirm its effectiveness in real-world applications.