Graphical user interface

Description automatically generated with low confidence

Term Project Report for

**Comparison of Classification Algorithms in Credit Card Fraud Detection**

by

Akshay Hebbar ( [ahebbar@syr.edu](mailto:ahebbar@syr.edu) )

Raghunandan Gajanan Bhat ( [rgbhat@syr.edu](mailto:rgbhat@syr.edu) )

**CIS 563 Introduction to Data Science – Fall 2022**

**Syracuse University**

**New York**

1. **Introduction**

Credit card fraud is a growing concern for both cardholders and financial institutions. It can result in financial losses and damage to a person's credit score and reputation. With the increasing use of credit cards for online transactions, the risk of fraud has also increased. Traditional methods for detecting fraud, such as manually reviewing transactions, are no longer sufficient to keep pace with the volume and complexity of modern transactions.

Machine learning algorithms offer a promising solution to this problem. These algorithms can analyze large amounts of data, identify patterns, and make predictions about the likelihood of fraud. By training these algorithms on large datasets of past fraudulent and non-fraudulent transactions, they can learn to recognize the characteristics that are common to fraudulent transactions.

There are several types of machine learning algorithms that can be used for credit card fraud detection. These include supervised learning algorithms, which require labeled data to learn from, and unsupervised learning algorithms, which can learn from data without explicit labels. However, the number of fraudulent transactions compared to the number of normal transactions is very small. This imbalance makes it difficult to identify fraudulent transactions.

In this project, we attempt to apply different techniques of classification to identify fraudulent transactions. We balanced the dataset using different sampling techniques. Then we applied decision trees, random forests, support vector machines and logistic regression algorithms to classify the transactions and compare their performances.

Overall, the use of machine learning algorithms has the potential to significantly improve the accuracy and efficiency of credit card fraud detection. By automating the process and leveraging the power of these algorithms, financial institutions can better protect their customers and reduce their own losses from fraud.

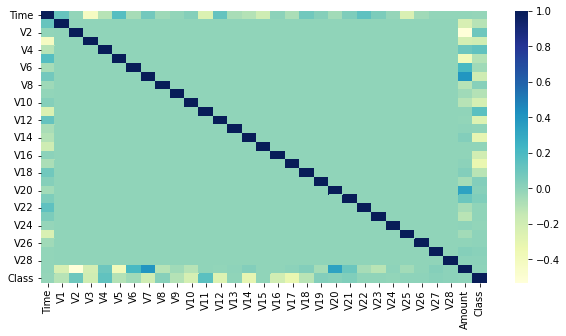
1. **Prior Work**

Credit card fraud detection has been a topic of interest for researchers and industry practitioners for many years. Earlier methods for detecting credit card fraud were based on the analysis of historical transaction data to identify patterns and anomalies that might indicate fraudulent activity. More recent approaches have utilized machine learning algorithms, such as decision trees, random forests, and support vector machines, to improve the accuracy and efficiency of fraud detection.

* 1. [Credit Card Fraud Detection System based on Operational & Transaction features using SVM and Random Forest Classifiers](https://ieeexplore-ieee-org.libezproxy2.syr.edu/document/9357709). In this paper the major focus was on the development of classifiers using the random forest and SVM’s where they have shown improvement in precision, recall and accuracy scores with both these methods. Our implementation also reflects this behavior as displayed in the results section below. Thus, we are able to achieve better results by using the methods mentioned in this paper.

1. **Dataset and Methods  
     
   Dataset**

The credit card fraud dataset used in this project is available on Kaggle. The dataset has been collected and analyzed during a research collaboration of Worldline and the Machine Learning Group(http://mlg.ulb.ac.be) of ULB (Université Libre de Bruxelles) on big data mining and fraud detection. It contains transactions made by European card holders in September 2013 and it is highly unbalanced with 492 fraud transactions and 284,807 normal transactions. Most of the features (V1-V28;except Time, Amount and Class) are already transformed using PCA which is evident from the correlation heatmap of the dataset. The feature-‘Class’ is the target variable which takes the value 0 (for non-fraud) or 1 (fraud).



*Fig 1. Correlation heatmap of the dataset*

Since the dataset is highly unbalanced, training the classifiers on this dataset will result in models that are skewed towards the majority class.

Decision Trees and Random Forest:

To eliminate this problem, we first sampled the dataset using both oversampling and undersampling methods and then used the sampled data to train and test the classifiers.

To perform sampling, we used ‘imbalance-learn’- an open source library relying on Scikit-learn, which provides tools when dealing with classification with imbalanced classes.

SVM and Logistic Regression:

The nature of the given dataset is imbalanced, where the positive class (non-fraud transactions) are much larger than the negative class (fraudulent transaction). Thus, to balance this data we employ a sampling strategy in such a way that the positive and negative classes do not induce bias in the training of the model and thus the resulting trained model does not misclassify instances.

For the sampling strategy we have have defined 3 types of sampling strategies which are as follows:

1. Simple Sampling:   
   This technique will sample the non-fraudulent results based on the index of the data. Hence we pick the first 492 records of non fraud data and combine it.
2. Shuffled Sampling:   
   This is an extension of the simple sampling technique described above with the addition of data shuffling to make sure the data is sampled randomly.
3. Reservoir Sampling:   
   Reservoir sampling is a sampling technique used to randomly select a subset of items from a larger population of items, where the subset size is fixed and predetermined. The technique is useful when it is not possible or practical to randomly sample the entire population, or when the population is too large to fit in memory.  
   The k value is 492 for this case study as we want to balance the fraud dataset equally. However, we could also scale the data with twice or thrice the number of the fraud data to verify results.

Note: The default values for sample selection is kept as Reservoir Sampling with k=492 and scaling =1.

**Methods**

* 1. **Decision Trees**

Decision tree algorithms are a type of machine learning algorithm that is used to build models in the form of a tree structure. These algorithms can be used for classification or regression tasks. In a decision tree, the data is split into different branches based on certain decision rules. Each internal node in the tree represents a feature or attribute of the data, and the leaf nodes represent the class labels or target values.

The decision tree algorithm works by starting at the root node and then making a series of decisions based on the feature values of the data points. At each decision point, the algorithm selects the feature that best splits the data according to a criterion such as entropy or Gini index. The process continues until the leaf nodes are reached, at which point the algorithm makes a prediction based on the majority class of the data points in that leaf node.

We trained two different decision trees, where one of them uses undersampled data and another uses oversampled data. The decision tree classifier is from Scikit-learn’s DecisionTreeClassifier class with default parameters.

**Decision tree with Under-sampling**: For undersampling the dataset, we used cluster generation method. The majority class in the dataset is clustered using KMeans clustering algorithm to generate the centroids. Then the centroids are used as a new dataset representing the majority class.

The new dataset is then split into a train and test set and the decision tree is validated using repeated K-fold cross validation.

**Decision tree with Over-sampling**: To oversample the minority dataset we used SMOTE (Synthetic Minority Oversampling Technique). New synthesized data is generated by selecting the existing minority class samples and finding its k nearest neighbors in the feature space. Then, it creates a new sample by randomly selecting one of the k nearest neighbors and combining it with the original sample, using a random amount of interpolation between the two. The new dataset containing the synthesized minority and the original majority class is used to train the classifier.

* 1. **Random Forest**

Random forest classifiers are a type of ensemble learning method that can be used for classification and regression tasks. An ensemble method is a machine learning technique that combines the predictions of multiple models to make more accurate and stable predictions. A random forest classifier consists of a collection of decision tree classifiers, each of which is trained on a different subset of the training data. The final prediction of a random forest classifier is made by aggregating the predictions of all the individual decision trees.

Random forest classifiers are popular because they are relatively easy to implement, can handle a large number of features, and are robust to overfitting. They are also highly accurate and have been used in a wide range of applications, including credit fraud detection, customer churn prediction, and medical diagnosis.

We used Scikit-learn’s RandomForestClassifer to build the classifier model and it is trained on both undersampled and oversampled data. For undersampling, we used the similar clustering technique that is used for decision tree classifiers. For over-sampling the minority class, again the SMOTE method is used to generate the synthetic minority samples.

Along with the normal random forest classifier, we also built another set of standalone random forest classifiers with XGBoost. Here we train three different random forests-

1. Random Forest with XGBoost on unbalanced data
2. Random Forest with XGBoost on undersampled data
3. Random Forest with XGBoost on oversampled data  
   1. **Support Vector Machines**

Support Vector Machines (SVMs) are a type of supervised learning algorithm that can be used for classification or regression tasks. The main idea behind SVMs is to find the hyperplane in a high-dimensional space that maximally separates the different classes. The data points closest to the hyperplane are called support vectors and have the most influence on the position of the hyperplane.

SVMs are particularly useful when the data is not linearly separable, as they can use the kernel trick to transform the data into a higher-dimensional space where it becomes linearly separable. There are different types of kernels that can be used, such as the linear kernel, polynomial kernel, and radial basis function (RBF) kernel.

For our current use case to determine the best possible kernel and hyperparameters, we have used GridSearchCV as a cross-validation strategy. GridSearchCV is a sci-kit library which helps us determine the best possible hyper-parameters for the SVM. The cross validation is done on the scoring parameter of the auc-roc value. The results of the tuning which provides the best score and the corresponding parameters for the model are then used to train the model and classify the instances.

The results from the GridSearchCV are as follows:

1. Linear SVM with L1 regularization:   
   Auc: 0.97, C: 0.3,
2. Linear SVM with L2 regularization:  
   Auc: 0.97, C: 6.93, dual: True (indicating we are using solving the dual problem)
3. SVM with RBF kernel:  
   Auc: 0.97, C: 4.64, gamma: 0.002
4. SVM with Polynomial kernel  
   Auc: 0.97, C: 1 , degree: 2

Since we are classifying the fraud transactions based on the rebalanced dataset, our main focus is in achieving better recall rather than precision. The recall is a better measure when one class is considerably smaller than the other and thus we focus on getting better AUC values in the ROC curve.

* 1. **Logistic Regression**

Logistic regression is a supervised learning algorithm used for classification tasks. It is used to predict the probability that a given input belongs to a particular class. Logistic regression is named after the logistic function, which is used to model the probability of a binary outcome as a function of input features.

The logistic function, also known as the sigmoid function, maps any real-valued number to a value between 0 and 1, which can be interpreted as a probability. In logistic regression, the logistic function is used to model the probability of a binary outcome (such as "success" or "failure") as a function of input features. The input features are combined using weights, which are learned during the training process.

Similar to the approach in SVM’s, we first use GridSearchCV to tune the hyperparameters and identify the best set. Once the best set is available, we train our Logistic Regression model using this hyperparameter set to obtain the results.

We make use of the LogisticRegressionCV library from sci-kit to model the data with 10 cross validations and a liblinear solver.  
  
Result of hyperparameter tuning:

Auc: 0.93, regularization: L2, solver: liblinear cross-validation:10

1. **Results**

The Results from all the above models are illustrated and compared below.

| **Model** | **Precision** | **Recall** | **Accuracy** | **F-Score** | **AUC** |
| --- | --- | --- | --- | --- | --- |
| DT - Over Sampling | 0.997 | 0.998 | 0.998 | 0.998 | 0.998 |
| DT - Under Sampling | 0.981 | 0.981 | 0.979 | 0.981 | 0.983 |
| RF - Over Sampling | 0.999 | 1 | 0.999 | 0.999 | 0.998 |
| RF - Under Sampling | 0.987 | 1 | 0.993 | 0.994 | 0.992 |
| RF-XGB-Over  Sampling | 0.985 | 0.937 | 0.961 | 0.960 | 0.961 |
| RF-XGB-Under  Sampling | 0.986 | 0.986 | 0.986 | 0.986 | 0.986 |
| SVM Linear - C=1 | 0.95 | 0.885 | 0.95 | 0.95 | 0.94 |
| SVM Linear - L1 regularize | 0.95 | 0.87 | 0.94 | 0.94 | 0.94 |
| SVM Linear - L2 Regularize | 0.93 | 0.85 | 0.93 | 0.92 | 0.93 |
| SVM - RBF kernel | 0.95 | 0.88 | 0.94 | 0.94 | 0.92 |
| SVM Polynomial kernel | 1 | 0.79 | 0.9 | 0.88 | 0.90 |
| LR | 0.97 | 0.87 | 0.93 | 0.92 | 0.93 |

In conclusion we have compared the results of various models with the tabular results. The process of comparison of multiple models with different techniques and parameters with cross-validated results is one of the first and was not available beforehand. The scores in itself are representative of the accuracy of the model and we can see that out of the experiment conducted the Random Forest Classifier has a higher recall score which indicates that it has classified all instances as expected. However, for accuracy scores we see that the SVM with polynomial kernel has better scores although it has a lesser recall. The future work for this project includes analyzing the fit of data as this is an imbalance dataset and we have modified it to create a sense of balance, the models which are trained may be slightly overfit and this may be reflective of the scores obtained for each of the models. Additionally, we can also look at various other algorithms which may help us in either normalizing, standardizing, dimensionality reduction and better fit.   
  
Thus, in conclusion, we have achieved the result of classification of multiple models for the credit card data set with the comparison of the scores.

1. References
   1. <https://www.kaggle.com/datasets/mlg-ulb/creditcardfraud>
   2. <https://scikit-learn.org/stable/modules/classes.html#>
   3. <https://numpy.org/doc/stable/reference/index.html#reference>
   4. <https://pandas.pydata.org/docs/reference/index.html#api>
   5. <https://xgboost.readthedocs.io/en/stable/tutorials/rf.html>
   6. <https://imbalanced-learn.org/stable/references/index.html>