Financial Fraud Detection using Azure ML and Spark ML

Bhagyashree Bhagwat, Niklas Melcher, Priyanka Purushu, Jongwook Woo
Department of Information Systems, CaliforniaStateUniversity
Los Angeles

E-mail: bbhagwa@caltstatela.edu, nmelche@calstatela.edu, ppurush@calstatela.edu, jwoo5@calstatela.edu

Abstract:
This research paper aims at providing insights on Financial Fraud Detection on a mobile money transactional activity. We classified the transaction as normal or fraud using Data analysis tools like Spark ML and Azure ML. Experimenting with sample dataset in Azure, we found that the Decision Forest model is most suitable to proceed in IBM/Databricks because of the recall value. Further, with Spark we found that the Random Forest classifier algorithm of the classification model proves to be the best algorithm in terms of our analysis. Finally, we reached a good recall score with 0.73 which implies a satisfying prediction quality in predicting fraudulent transactions.

1. Introduction

Financial frauds can be a devastating issue with extensive ramifications on any business, finance industry, corporate and government segments and also for individual consumers [1]. With technological advancements, these transaction frauds are becoming more intricated. Today in the data-driven world, we can track down the fradulent transactions with the use of big data tools and data mining approaches.

While carrying research on this topic, we encountered challenges in finding a dataset on financial fraud detection, as these kind of financial datasets are not publicly available due to the nature of the information. A synthetic transactional data was developed by PaySim [2] simulator which incorporated both: normal customer behabiour and fradulent behaviour.

We aim at doing predective analysis on the target value which is column "isFraud" which detects if a money transaction is a fraud or not. The dataset size is approximately 470MB and it has eleven features. We want to predict if a money transaction is a fraud or not using classification models. In this paper we have analyzed the data with the help of two machine learning platforms: Microsoft Azure ML and Apache Spark ML.

2. Related Work

While working on this project, we underwent extensive research on lot of similar papers on financial fraudulents. The most common problem we noticed was that many researchers were having hard time finding an appropriate dataset for analysis. Also, PaySim simulator fixed the problem for a majority of researchers. We could relate our work on the same lines just like other researchers. For us, finding the dataset was not much difficult due to the availability of PaySim's synthetic dataset. Where others research paper was more focussed on creating a synthetic financial dataset [2], ours was primarily targetted on detecting the fradulent transactions from the synthetic dataset.

3.Background

Here is the background on which we started our research paper. Starting from determing the type of problem, understanding the importance of machine learning and determining the algorithms.

Two main problems machine learning is trying to solve: Classification & Regression problems. Mathematically speaking, regression is a combination of multi-dimensional feeding and function interpolation. With a regression problem you are trying to find a function approximation with a minimal error deviation or cost function. In other words, regression is simply trying to predict numeric dependency – a function value, for example, price of a house – from a set of input parameters like square footage, age, number of bedrooms and so on.

Classification is a different type of problem which identifies group membership. That means that if you have multiple events characterized by input parameters, which can be labelled differently, and you want your system to predict which label should be used, this is the classification problem. Take spam filters, for example. Emails in your inbox are processed by the machine learning spending algorithm. And if some criteria is met, emails are labelled as spam.

Machine learning is a fascinating topic as it incorporates substantial parts of different fields – statistic, artificial intelligence theory, data analytics and numerical methods. Simply put, machine learning is an application that is capable of improving its prediction results with sucessive iterations. Or you could say that is improves with experience.

Decision tree is a analytical tool which supports decision making by including event outcomes or their possible consequences.

Random forest can be expressed as a set of decorrelated decision trees. The example of random forest can be a data set which contains different random values and their class. Then we divide the data set into lot of subsets with random values and random classes. After the division, the algorithm decides and allocates different classes to each of the independent forest. This can be used for predictive analysis as the algorithm assigns classes to each forest, and predicts the class which is repeated the most in the classification.

4. Data

For this experiment we use a synthetic dataset generated using the simulator PaySim [2] as an approach to such a problem. PaySim uses aggregated data from the private dataset to generate a synthetic dataset that resembles the normal operation of transactions and injects malicious behavior to later evaluate the performance of fraud detection methods. All in all, PaySim simulates mobile money transactions extracted from one month of financial logs from a mobile money service implemented in an African country. The original logs were provided by a multinational company, who is the provider of the mobile financial service which is currently running in more than 14 countries all around the world.

The data has a size of 470 MB with 6.362.620 rows. The dataset contains 11 attributes and the target column is 'isFraud'. A transaction can either be non-fraudulent, indicated by a 0, or fraudulent, indicated by a 1, which makes this to a binary classification problem.

4.1 Attributes

Here is a sample row of the dataset and with headers explanation:

1,PAYMENT,1060.31,C429214117,1089.0,28.69,M1591654462,0.0,0.0,0,0

step: maps a unit of time in the real world. In this case 1 step is 1 hour of time. Total steps 744 (30 days simulation).

type: CASH-IN, CASH-OUT, DEBIT, PAYMENT and TRANSFER.

amount: amount of the transaction in local currency.

nameOrig: customer who started the transaction.

oldbalanceOrg: initial balance before the transaction.

newbalanceOrig: new balance after the transaction.

nameDest: customer who is the recipient of the transaction.

oldbalanceDest: initial balance recipient before the transaction. Note that there is not information for customers that start with M (Merchants).

newbalanceDest: new balance recipient after the transaction. Note that there is not information for customers that start with M (Merchants).

isFraud: This is the transactions made by the fraudulent agents inside the simulation. In this specific dataset the fraudulent behavior of the agents aims to profit by taking control or customers' accounts and try to empty the funds by transferring to another account and then cashing out of the system. The attribute is binary, either 0 or 1.

isFlaggedFraud: The business model aims to control massive transfers from one account to another and flags illegal attempts. An illegal attempt in this dataset is an attempt to transfer more than 200.000 in a single transaction.

4.2 Data Understanding

The dataset provides 5 numeric attributes (amount, oldbalanceOrg, newbalanceOrg, oldbalanceDest, newbalanceDest), 4 categorical attributes (step, type, isFraud, isFlaggedFraud) and two string attributes (nameOrig, nameDest).

The dataset contains 98.87% non-fraud transactions and 0.12% fraud transactions which implies a big imbalance in the data. In the next step, we need to understand the dataset and try to recognize certain patterns that would be helpful for our experiment. Figure 1 shows the amount of transactions grouped by the type. We see that most of the transactions are made with CASH_OUT and PAYMENT.

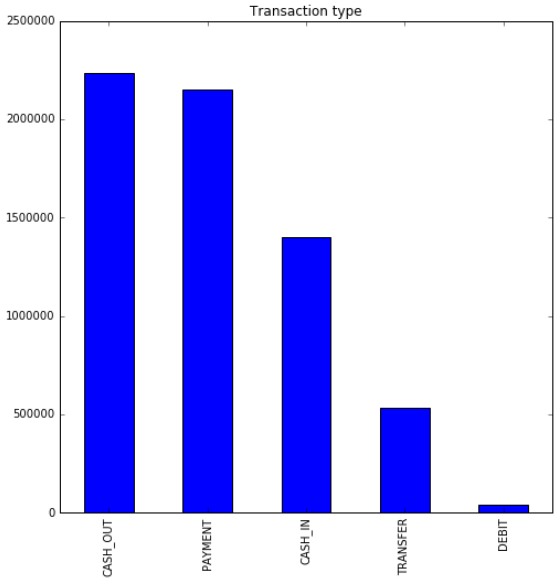

Figure 1: Amount of transactions grouped by type

Next, we want explore which transaction types are vulnerable to fraud. Figure 2 shows clearly that fraud transactions are only made with the type CASH_OUT and TRANSFER. This is an interesting fact since the type TRANSFER is in the fourth place when it comes to the number of transactions.

Furthermore, there is an interesting attribute in the dataset called **isFlaggedFraud**. This attribute is supposed to flag suspicious transactions as a fraud to help the system detecting them. Unfortunately, out of the 6.362.620 rows there are only 16 transactions flagged as fraud which makes this attribute useless for our data model.

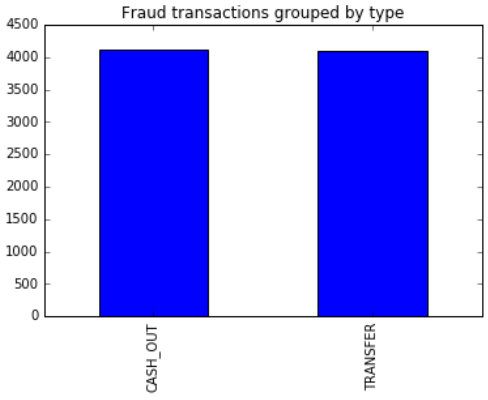

Figure 2: Fraud transactions grouped by type

Next, we want to explore correlations between attributes which would be useful for our model. Figure 3 and 4 show relationships between **newbalanceDest** and **oldbalanceDest**, and **newbalanceOrig** and **oldbalanceOrg**, because there are strong positive correlations.

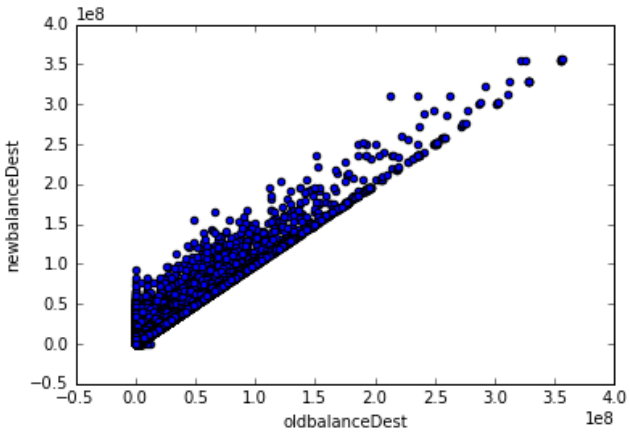

Figure 3: Scatterplot between newbalanceDest and oldbalanceFest

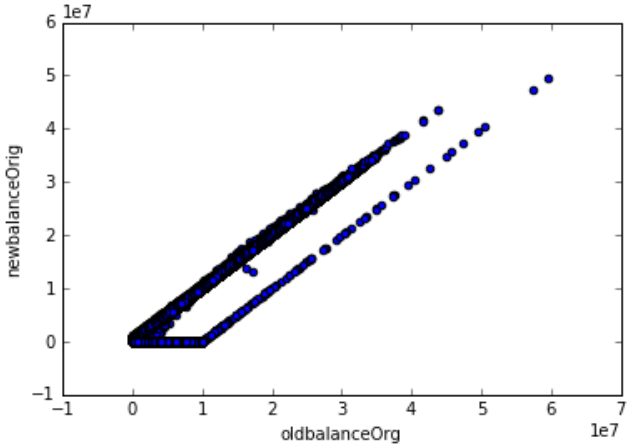

Figure 4: Scatterplot between newbalanceOrig and oldbalanceOrg

The next step is to drop useless attributes for the model. Table 1 shows the attributes we drop and the attributes we keep.

Columns dropped	Columns kept
step	amount
nameDest	oldbalanceOrg
nameOrig	newbalanceOrig
isFlaggedFraud	newbalanceDest
	oldbalanceDest

Table 1: Columns to drop and columns to kep from the dataset

We drop **step** because there is no correlation between the time for the simulation and the transactions. Furthermore, we drop the two string attributes **nameDest** and **nameOrig** because they are unique values which have no relationship to any other attributes and is thus not helpful. As already explained, the attribute **isFlaggedFraud** makes no impact to our model.

5. Experiment

In the first part, we build the model in Azure ML[1]. Mainly to try different classification models with a subset of the original data. In the second part, we build our model with Spark ML in Databricks[2]. Here, we run our model with the best algorithms from the first part and try to improve our result taking the whole dataset into consideration.

The main workflow for both systems is illustrated by figure 5. First, we need to prepare our data. As already mentioned in section 4.2, we now which columns are useful for our model and which we can drop. This step also contains the process of make attributes categorical and numeric, as we need to do this procedure with the attribute **type**. Additionally, the dataset has to be normalized and split into a train and test set, respectively 70% and 30%. Afterwards, we build the model with different algorithms, tune the hyperparameters and cross validate each model. Finally, we evaluate the model and interpret the results. The whole procedure is an iterative process and can be done several times before finding the best model.

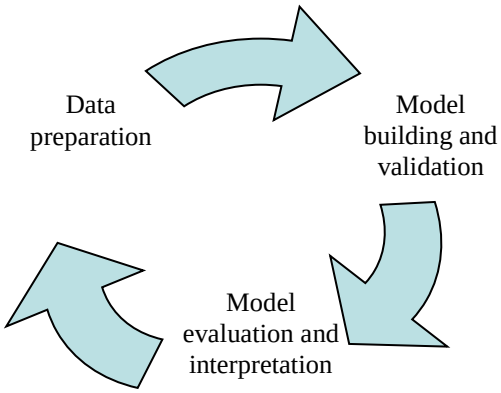

Figure 5: Raw description for the workflow

There are three main metrics which are important in terms of evaluating the model. The accuracy, precision and recall. In our case, the recall is the most important metric because the aim of this classification problem is to detect fraudulent transactions. A good recall implies that the model is good in predicting a transaction as a fraud when it is actually a fraud.

5.1 Azure ML

We took a small subset of our dataset with approximately 10000 rows and tried four different classification algorithm following the procedure displayed by figure 5. Overall, it was easy to realize because Azure ML is mostly a drag and drop tool with elements to configure.

Since we have a binary classification problem, we use two class classification algorithms: Two Class Logistic Regression (LR), Two Class Decision Forest (DF), Two

[1] https://azure.microsoft.com/en-us/services/machine-learning/

[2] https://databricks.com/

Class Decision Jungle (DJ) and Two Class Support Vector Machine (SVM). Table 2 summarize the results of the experiments.

Model	Accuracy	Precision	Recall
LR	0.991	1.000	0.100
DF	0.995	0.727.	0.800
DJ	0.997	1.000	0.700
SVM	0.993	1.000	0.300

Table 2: Results for the small subset with different classification algorithms using Azure ML

Clearly, the DF is the best algorithm for our model since it has the highest recall score. The DJ has a good performance as well but needed approximately five times longer than the DF to calculate. Based on this results we will continue building our model in Databricks mainly with the DF.

5.2 Databricks with Spark ML

We took the whole dataset and tried three different classification models following the procedure displayed by figure 5. This time we used a train validation split instead of cross validation for every model because it takes much less time to train the model with the train validation split. We used the Random Forest Classifier (RF), the Decision Tree Classifier (DT) and Logistic Regression (LR). Although the result of LR was very bad in the Azure ML experiment, we gave it another try in Databricks, because we wanted to examine the LR's performance using the whole dataset. Table 3 summarizes the results of the experiment. We added the Receiver Operating Characteristic (ROC) with the Are Under Curve (ROC AUC) as another metric to better visualize the performance of a binary classifier summarized in a single number.

Model	ROC AUC	Precision	Recall
RF	0.860	0.927	0.719
DT	0.829	0.967	0.679
LR	0.726	0.846	0.453

Table 3: Results for the whole dataset with different classification algorithms using Spark ML

Clearly, the RF has the best recall and ROC AUC score which indicates that this model is the best compared to the other two. To understand the results better, table 4 summarizes the confusion matrix.

Confusion matrix	Predicted: NO	Predicted: YES
Actual: NO	TN = 1905940	FP = 78
Actual: YES	FN = 644	TP = 1761

Table 4: Confusion matrix using the RF

The confusion matrix shows that our model is good in predicting non-fraudulent transaction when they are actual not a fraud, indicated by the high number of the true negative (TN) and small amount of false positive (FP) numbers (Specificity = 0.99). Nevertheless, when predecting fraudulent transactions we still have some errors because the number of false negative results (FN) is still high.

6. Conclusion

We investigated a dataset containing fraudulent and non-fraudulent transactions which made it to a binary classification problem. Since the dataset was about 470 MB we had to use big data technologies like Azure ML and Databricks with Spark ML to handle it. We showed the model bulding process, both in Azure ML and Spark ML. For Azure ML, we used a small subset to examine the results of several different classification algorithms. The Decision Forest Classifier scored the best recall score with 0.800.

In the next step, we used Databricks with Spark ML to train three different classification algorithms on the whole dataset. The Random Forest Classifier scored the best recall score with 0.719 and a specificity of 0.99. From this it can be concluded that our model is very good in predicting non-fraudulent transaction when they are actual non-fraudulent. But we still have some errors prediciting fraudulent transaction when they are actually fraudulent. This can be explained by the misbalanced data since 98.7% of our data contains non-fraudulent transactions which makes it hard to train a model properly. Nevertheless, our model is acceptable in predicting fraudulent transactions.

References

[1] https://www.acfe.com/financial-transactions-and-fraud-schemes.aspx

[2] E. A. Lopez-Rojas , A. Elmir, and S. Axelsson. "PaySim: A financial mobile money simulator for fraud detection". In: The 28th European Modeling and Simulation Symposium-EMSS, Larnaca, Cyprus. 2016

[3] https://www.kaggle.com/ntnu-testimon/paysim1

[4] https://github.com/nmelche/IntroductionToBigDataScience