IBM Transactions for Anti-Money Laundering (AML)

Lekha Ajit Kumar, Sushmitha Dandu, Dauren Omarov, Navyasree Sriramoju
Department of Information Systems, California State University
Los Angeles
e-mail: lajitku@calstatela.edu, sdandu3@calstatela.edu, domarov@calstatela.edu, nsriram@calstatela.edu

Abstract: This project focuses on the application of machine learning algorithms for classification in the field of anti-money laundering (AML) using the IBM Transactions dataset. The primary objective is to develop a predictive model that can accurately classify transactions as either suspicious or legitimate based on various features. Six different machine learning algorithms, including Logistic Regression, Gradient Boost Tree, Decision Tree, Random Forest, Factorization Machine, and Support Vector Machine, are evaluated for their performance. The evaluation of the models includes measuring precision, recall, and feature importance. The results indicate that the timing of the transaction, originating bank, and destination bank account exhibit the highest feature importance, suggesting that these factors play a crucial role in distinguishing suspicious transactions from legitimate ones.

The findings of this project demonstrate the efficacy of machine learning algorithms in the context of AML and emphasize the significance of specific features in identifying suspicious transactions.

Keywords: Regression, Classification, Hadoop, Precision, Recall, Train Split Validator, Cross Validator, Logistic Regression, GBT, Decision Tree, Random Forest, Support Vector Machine, Feature Importance, Data Bricks.

1. Introduction

Money laundering poses significant challenges to the financial industry, necessitating advanced techniques for detection and prevention. Machine learning algorithms applied to large-scale datasets have emerged as a promising approach in this context. This project focuses on leveraging machine learning algorithms to classify anti-money laundering (AML) transactions using the IBM Transactions dataset. The primary objective is to develop a predictive model that accurately classifies transactions as suspicious or legitimate, aiding in the fight against money laundering. Six machine learning algorithms.

By leveraging the insights gained from this project, financial institutions can enhance their anti-money laundering efforts by utilizing machine learning algorithms to automate the detection of potential money laundering activities. This research contributes to the existing body of knowledge in AML and provides valuable insights for practitioners in the field.

2. Related Work

[1] Guike Zhang explains in his Paper, “Machine learning approaches for Constructing the national anti-money laundering index” that they proposed a methodology for constructing an anti-money laundering (AML) index based on mutual evaluation reports and machine learning models. They employed LASSO Algorithm which is a regression analysis model and random forests five-factor (RF-FF) model. In that, the RF-FF model achieves high accuracy of 86.31%. Relating this to our IBM Transactions dataset for AML, we explored the application of machine learning algorithms such as the random forests model, FM, GBT, LR, SVM, and Decision Tree to identify important factors and predict the likelihood of money laundering in the transactions. Comparatively, our RF achieved an accuracy of 81.9% using TVS. This approach can help optimize the allocation of AML resources and provide insights into the effectiveness of AML systems in detecting money laundering activities.

[2] Kumar Pankaj explains in his Paper"The fight against money laundering: Machine learning is a game changer" highlights the challenges of money laundering and the potential of machine learning (ML) in enhancing anti-money laundering (AML) efforts. In line with this paper, our project focuses on leveraging the IBM Transactions dataset to employ various ML algorithms, such as logistic regression, gradient boost tree, decision tree, random forest, factorization machine, and support vector machine, to identify suspicious activities related to money laundering. By aligning our work with the paper's insights on the importance of AML expertise, data science talent, and reliable data sources, we contribute to the ongoing advancements in combating financial crimes and strengthening defenses against money laundering.

3. Data Set Used

The dataset used in this paper is taken from an open-source platform which is Kaggle.com, We chose the "LI_Medium_Trans.csv," file from the Dataset which is 2.98 GB and has 11 Columns.

Total Data Set Size: 8GB, Format: CSV

The different columns present in the dataset are listed below:

•Timestamp: The date and time when the transaction occurred or was recorded.

•From Bank: The name or identifier of the bank from which the funds were sent.

•Account: The account number or identifier associated with the sender's bank account.

•To Bank: The name or identifier of the bank to which the funds were received.

•Account: The account number or identifier associated with the recipient's bank account.

•Amount Received: The amount of money received in the recipient's bank account.

•Receiving Currency: The currency in which the funds were received.

•Amount Paid: The amount of money paid or transferred from the sender's bank account.
•Payment Currency: The currency in which the payment was made or transferred.
•Payment Format: The method or format used for the payment, such as wire transfer, online payment, etc.
•Is Laundering: This column indicates whether the transaction is suspected or flagged.

4. Technical Specifications

HDFS ORACLE SPECIFICATION

HADOOP VERSION	3.1.2
No.of CPU'S	8
Py Spark version	3.0.2
Nodes	3
Total Storage	390.7 GB
CPU Speed	1995.3 MHz
Core	4 Core CPU

DATABRICKS SPECIFICATION

DB Community Version	10.4LTS (includes Apache Spark 3.1.1, Scala 2.12)
File System	DBFS (Data Bricks File System)
Nodes	5
Python Version	3.10.4

5. Background/Existing work

In our project, We used Six different machine learning algorithms, including Logistic Regression, Gradient Boost Tree, Decision Tree, Random Forest, Factorization Machine, and Support Vector Machine. We ran the sample data in Databricks and Full data in spark-submit.

5.1 Classification

The Classification algorithm is a Supervised Learning technique that is used to identify the category of new observations based on training data.

In Classification, a program learns from the given dataset or observations and then classifies new observations into a number of classes or groups. Such as, Yes or No, 0 or 1

5.2 Prediction System Flowchart

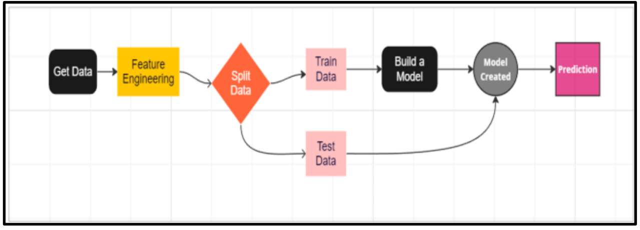


Figure 5.2 Flow Chart

Data Input: Gather the input data, which can be either structured or unstructured.

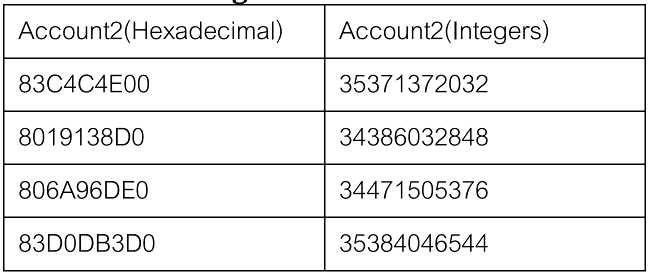
Feature Engineering: Formulate the machine learning problem by defining the predictions we want to make and identifying the relevant observational data for making those predictions.

Dataset Splitting: Divide the dataset into two subsets, namely the training data and the test data.

Model Building: Select an appropriate model to implement and predict the desired output. Train the model using the training dataset, and once it is trained, apply it to the test dataset.

Prediction: Generate predictions using the trained model. Evaluate and compare the results obtained from the predictions.

5.3 Data Cleaning



Account2(Hexadecimal)	Account2(Integers)
83C4C4E00	35371372032
8019138D0	34386032848
806A96DE0	34471505376
83D0DB3D0	35384046544

One particular cleaning step we implemented as shown in the above table was the conversion of hexadecimal values to integers for the column Account2 from the dataset. This step allowed us to transform the data into a more suitable format for analysis, ensuring consistency and compatibility across different features. By converting hexadecimal values to integers, we eliminated any potential inconsistencies or errors that could arise from working with mixed data types. This data-cleaning process was crucial in preparing our dataset for further analysis and modeling, ultimately enhancing the reliability and effectiveness of our machine-learning algorithms in detecting suspicious transactions and combating money laundering activities.

6. Our Work

Big Data refers to vast collections of data characterized by their extensive volume, velocity, and variety. Managing and processing such massive amounts of data can be challenging. To tackle this challenge, various methods such as Hadoop, Data Mining, and Machine Learning are employed. Machine Learning offers diverse and exciting approaches to handle Big Data. It involves transforming data into programs and automating systems. We divided the dataset into a training set (70% of the data) and a test set (30% of the data) to evaluate the performance of all the Algorithms implemented in our Project. Below are a few Machine Learning algorithms we implemented in Data Bricks:

6.1 Random Forest (RF)

Random Forest works by creating a multitude of decision trees, where each tree is built using a different subset of the features and a random sample of the training data. During the training process, the algorithm considers different split points for each feature and selects the best split based on metrics such as information gain.

Below are the results:

Cross Validation: Precision - 0.047, Recall - 0.495

Train Validation: Precision - 0.065, Recall - 0.489

These metrics indicate the model's ability to accurately classify suspicious transactions.

Furthermore, the area under the curve (AUC) metric was used to assess the overall performance of the Random Forest model.

Cross Validation: AUC – 0.883

Train Validation: AUC – 0.914, 0.918

Overall, our analysis demonstrates the potential of Random Forest as a robust algorithm for classifying AML transactions in the IBM Transactions dataset. The high AUC values and satisfactory precision and recall metrics indicate the model's ability to accurately detect and flag suspicious activities, thereby contributing to the prevention of money laundering.

6.2 Gradient Boost Tree (GBT)

The Gradient Boosting Tree (GBT) algorithm is a powerful machine learning technique that combines the concepts of boosting and decision trees. It is an ensemble method that builds multiple decision trees in a sequential manner, where each subsequent tree focuses on improving the performance of the previous tree.

Below are the results:

Cross Validation: Precision - 0.0936, Recall - 0.979

Train Validation: Precision - 0.936, Recall – 0.979

These scores indicate that the model achieved a high precision, meaning that it accurately classified a significant portion of the suspicious transactions, and a high recall, indicating that it successfully identified a large proportion of the actual suspicious transactions.

Furthermore, the area under the curve (AUC) metric was used to assess the overall performance of the GBT model.

Cross Validation: AUC – 0.9077

Train Validation: AUC – 0.9077

Overall, the results of the GBT model indicate its effectiveness in classifying AML transactions. The high precision and recall scores demonstrate the model's capability to accurately identify suspicious transactions, which is crucial in combating money laundering. The AUC scores further support the model's ability to make reliable predictions and its potential for real-world application in financial institutions' anti-money laundering efforts.

6.3 Factorization Machine (FM)

Factorization Machine (FM) is a machine learning algorithm that is widely used for solving various prediction tasks, including classification and regression. FM excels in capturing interactions between features, especially in situations where there are sparse or high-dimensional data. It combines the advantages of linear models and factorization methods to model both linear and non-linear relationships between variables

Below are the results:

Cross Validation: Precision - 0.0422, Recall – 0.7868

Train Validation: Precision - 0.0422, Recall – 0.7868

The measures indicate that the model's ability to correctly identify positive instances is limited.

Cross Validation: AUC – 0.7461

Train Validation: AUC – 0.7461

The AUC provides an indication of the model's ability to discriminate between positive and negative instances. For both train validation and cross-validation, the AUC was 0.7461, suggesting that the model's discriminatory power is moderate.

Based on these results, it is evident that the FM algorithm did not perform optimally on the AML classification task. Further analysis and potential improvements are needed to enhance the precision and recall rates, as well as to increase the model's discriminatory power.

6.4 Support Vector Machines (SVM)

Support Vector Machine (SVM) is a powerful and widely used machine learning algorithm for both classification and regression tasks. It is particularly effective in scenarios where the data is not linearly separable and requires nonlinear decision boundaries. SVM aims to find an optimal hyperplane that maximally separates the different classes in the data.

Below are the results:

Cross Validation: Precision - 0.0285, Recall - 0.5

Train Validation: Precision - 0.0285, Recall – 0.5

Cross Validation: AUC – 0.5

Train Validation: AUC – 0.5

Based on the obtained results, the FM model exhibited limited performance with low precision and recall values. The AUC scores were also relatively low, suggesting that the model struggled to distinguish between suspicious and legitimate transactions effectively. These findings indicate that FM might not be the most suitable algorithm for this specific AML classification task on the given dataset.

6.5 Logistic Regression (LR)

Logistic Regression is a widely used machine learning algorithm for binary classification tasks. It is particularly suitable when the dependent variable or target variable is categorical and has only two possible outcomes. The goal of Logistic Regression is to estimate the probability of the binary outcome based on a set of input features.

Below are the results:

Cross Validation: Precision - 0.0285, Recall - 0.5

Train Validation: Precision - 0.0285, Recall – 0.5

Cross Validation: AUC – 0.5296

Train Validation: AUC – 0.53

The results showed relatively low precision and recall values, indicating that the model struggled to accurately identify suspicious transactions. In our case, the precision and recall values were consistent for both train validation and cross-validation. The AUC scores obtained were relatively low, suggesting that the Logistic Regression model had limited discriminatory power in separating suspicious transactions from legitimate ones. These findings indicate that Logistic Regression may not be the most effective algorithm for this particular AML classification task on the given dataset.

6.6 Decision Tree Classifier (DT)

The decision tree classifier is a popular machine learning algorithm used for classification tasks. It operates by creating a tree-like model of decisions and their possible consequences.
Below are the results:
Cross Validation: Precision - 0.0285, Recall - 0.5106
Train Validation: Precision - 0.0285, Recall – 0.5217
Cross Validation: AUC – 0.5
Train Validation: AUC – 0.6415
These measures suggest that the decision tree classifier achieved a higher recall rate when evaluated using cross-validation. These results highlight the performance of the decision tree classifier in classifying AML transactions. However, further optimization and fine-tuning of the model may be necessary to improve precision and overall accuracy. Additionally, it would be beneficial to explore other machine learning algorithms and feature engineering techniques to enhance the classification performance further.

7. Comparison Table

Algorithms Comparison

ALG	TVS				CV			
	AUC	CT	P	R	AUC	CT	P	R
GBT	0.79	14	0.93	0.97	0.80	14	0.09	0.97
RF	0.81	13	0.06	0.48	0.80	12	0.04	0.49
FM	0.45	18	0.04	0.78	0.45	19	0.04	0.78
DT	0.5	9	0.02	0.52	0.5	7.4	0.02	0.51
LR	0.52	54	0.02	0.5	0.52	70	0.02	0.5
SVM	0.5	26	0.02	0.5	0.5	16	0.02	0.5

Keywords: ALG-Algorithm, TVS- Train Validation Split, CV-Cross-Validation Split

AUC-Area Under the Curve, CT-Computation Time, P- Precision, R- Recall

8. Conclusion

Based on the analysis of the results, it can be inferred that Gradient Boost Tree and Random Forest are the most effective algorithms for detecting suspicious transactions in the IBM Transactions dataset for AML. They demonstrated high recall values and competitive AUC scores, indicating their potential in accurately identifying suspicious activities. However, it is important to consider further optimization, parameter tuning, and exploring other machine learning techniques to enhance the performance of the models. Additionally, future work could focus on incorporating additional features and conducting more extensive feature engineering to improve the accuracy of the classification models for AML detection.

9. References

[1]Author links open overlay panelGuike Zhang a, et al. “Machine Learning Approaches for Constructing the National Anti-Money Laundering Index.” Finance Research Letters, 5 Dec. 2022, www.sciencedirect.com/science/article/abs/pii/S1544612322007449.

[2]Kumar, Pankaj, et al. “The Fight against Money Laundering: Machine Learning Is a Game Changer.” McKinsey &amp; Company, 7 Oct. 2022, www.mckinsey.com/capabilities/risk-and-resilience/our-insights/the-fight-against-money-laundering-machine-learning-is-a-game-changer.

[3] Li, Susan. “Machine Learning with Pyspark and MLlib - Solving a Binary Classification Problem.” Medium, Towards Data Science, 7 May 2018, https://towardsdatascience.com/machine-learning-with-pyspark-and-mllib-solving-a-binary-classification-problem-96396065d2aa.

[4] “Regression Analysis.” Corporate Finance Institute, 3 May 2023, https://corporatefinanceinstitute.com/resources/data-science/regression-analysis/.

[5] Jagdeesh. “PySpark Decision Tree – How to Build and Evaluate Decision Tree Model for Classification Using PySpark MLlib.” Machine Learning Plus, 1 May 2023, https://www.machinelearningplus.com/pyspark/pyspark-decision-tree/.

DataSet URL:
https://www.kaggle.com/datasets/ealtman2019/ibm-transactions-for-anti-money-laundering-aml