Anti-Money Laundering Prediction

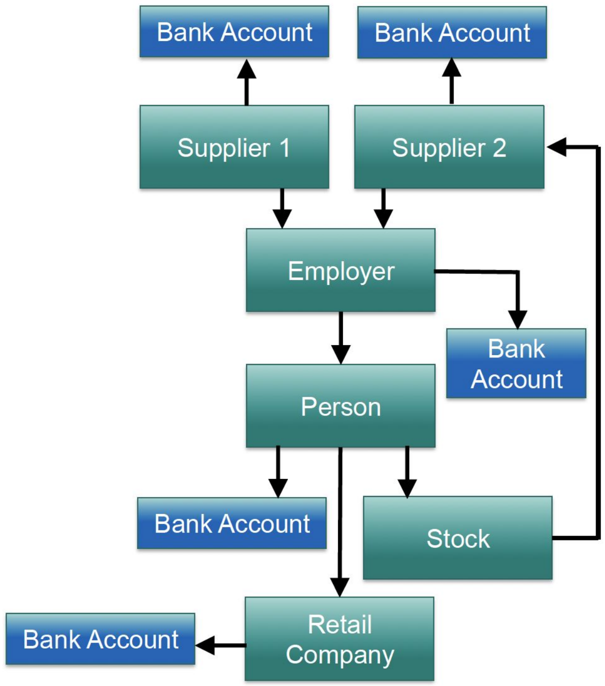
Nish Patel (nsp124)
December 12, 2023

Mentor: Dr. James Abello
TA: Haoyang Zhang

Introduction

Introduction

- Money laundering detection is a critical issue in the financial sector
- Real financial transaction data is highly confidential and difficult to obtain for privacy reasons
- IBM has developed a simulation model to generate synthetic transactions
- The simulation includes simulated money laundering and these transactions are labeled in the dataset



Dataset

Dataset

- Posted by IBM on Kaggle
- ~17GB file size
- ~180 million records (~100 bytes per record)
- ~2.1M distinct bank accounts
- 15 distinct currencies
- 7 distinct payment formats
- Data was generated via IBM simulation from August 1, 2022 to November 5, 2022

Timestamp	From Bank	Account	To Bank	Account	Amount Receive	Receiving Currency	Amount Paid	Payment Currency	Payment Format	Is Laundering
9/1/22 0:22	800319940	8004ED620	808519790	872ABC810	120.92	US Dollar	120.92	US Dollar	Credit Card	0
9/1/22 0:05	8021ADE00	80238F220	9A7F59FA0	A23691240	33.97	US Dollar	33.97	US Dollar	Credit Card	1
9/1/22 0:14	801946100	8023F0980	83585F5A0	948893910	79.20	US Dollar	79.20	US Dollar	Credit Card	0
9/1/22 0:05	80010C840	800122AA0	80010C840	800122AA0	8,834.09	Euro	10351.64	US Dollar	ACH	0
9/1/22 0:05	80010C840	800122AA0	80010CF20	80012DA00	8,834.09	Euro	8834.09	Euro	ACH	0
9/1/22 0:08	80010CF20	80012DA00	80010CF20	80012DA00	9,682.16	US Dollar	8262.75	Euro	ACH	0
9/1/22 0:08	80010CF20	80012DA00	80010BD60	80011E460	9,682.16	US Dollar	9682.16	US Dollar	ACH	0
9/1/22 0:03	800319940	800466670	80029A010	8002F6F20	9,125.22	US Dollar	9125.22	US Dollar	ACH	0

Fundamental Questions

Fundamental Question

- **Can deep learning be utilized to predict the accounts that are involved in a money laundering transaction?**
  - Recall from project 1 that the literature review led to graphs being used as the data structure to model this problem
  - With the advances in convolutional neural networks, graph convolutional neural network models are suggested in literature review
  - This boils down to a very complex binary classification problem
  - An accurate prediction could help the target end users (fraud analysts) identify which accounts need to be manually reviewed

Feature Engineering

Feature Engineering

- The only node features included in the dataset include account_id and bank
- In phase 1, “is_laundering” feature was created at the node level
- Additional node features needed to be created to include in the model

  - transactions_from
  - transactions_to
  - total_usd_from
  - total_usd_to
  - avg_usd_from
  - avg_usd_to
  - total_currencies_from
  - total_currencies_to

Model Architecture

Model Architecture - Graph Convolutional Network (GCN)

- Graph convolutional neural networks apply convolution principles to graph data types
- pytorch has a geometric extension that allows for graphs to be input as the training set
- The convolution step in a GCN involves a neighborhood aggregation process
    - At each node, information from neighboring nodes is aggregated based on the edge connections and associated weights
    - The aggregation weights are what is being “learned” and they capture the importance of each neighbor’s contribution to the target node
- This was chosen as the first modeling approach as the problem of anti-money laundering detection requires the ability to analyze not only a particular account but the accounts they transact with and a GCN gives the ability to do that

Model Architecture - Graph Attention Network (GAT)

- A graph attention network is an extension on the GCN with the addition of attention mechanism
    - In a GCN, the weights assigned to neighboring nodes is the same for all neighbors
    - By adding attention, the weights for neighboring nodes can be individually assigned and the network can selectively attend to the neighboring nodes that are more important
- This was selected as an alternative to GCN because of the class imbalance that is present in the data.
    - Attention provides the ability to give weightage to neighboring nodes that are involved in money laundering activity even if they are a much smaller portion of the total nodes

Model Architecture - Loss Function

- Since this is a binary classification problem, the natural loss function to use is binary cross-entropy loss
- The data has a large class imbalance (only ~0.1% of the transactions and ~2% of accounts are flagged as money laundering)
- Pytorch offers an option to provide weighting to account for the class imbalance in the loss function if you use BCEWithLogitsLoss so the models were updated to use this loss function
- The weighting was calculated based on the counts
  - Weighting for laundering transactions was given as the total number of accounts/number of laundering transactions

Model Architecture - Activation Function

- Many different activation functions were tried to provide the nonlinearity required by the neural network
- For GAT, ELU was used as the activation function
- For GCN, ReLU was used as the activation function
  - The main difference between ELU and ReLU is that ReLU replaces any negative values in hidden layers with 0 while ELU provides the ability for a negative value to be activated
- For the output layer in both models, sigmoid is used as the activation function to convert the outputs into probabilities

Model Architecture - Hyperparameters

- Hyperparameters were tuned:
  - Learning rate – a final value of 0.0001 is used
  - Number of epochs – a final value of 15 epochs is used
  - Number of convolutional layers – 2 convolutional layers are used in both GAT and GCN
  - Number of dense layers – 2 dense layers are used in GCN and 1 dense layer is used in GAT
  - Dropout – a value of 0.3 is used in both GAT and GCN

Results

Results

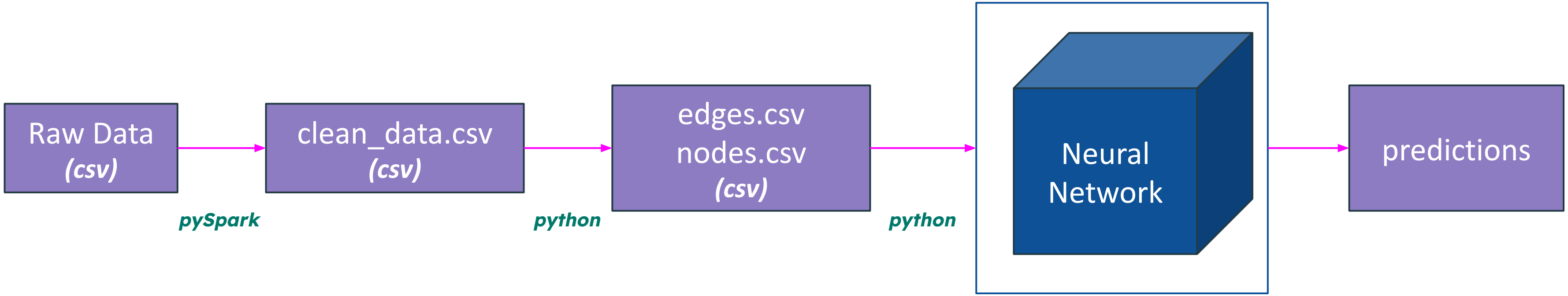
#	Feature Engineering?	Dataset	Model	Loss Function	Conv Layers	Dense Layers	Dropout	LR	Accuracy
1	no features	small	GCN	BCE	2	2	0.3	0.0001	0.4941
2	features	small	GCN	BCE	2	2	0.5	0.0001	0.5013
3	features	small	GAT	BCE	2	1	0.3	0.0001	0.5019
4	features	small	GCN	BCELogit(weight)	2	2	0.5	0.0001	0.5052
5	features	small	GAT	BCELogit(weight)	2	1	0.3	0.0001	0.5232
6	features	small	GAT	BCELogit(weight)	3	1	0.3	0.0001	0.5083
7	features	small	GAT	BCELogit(weight)	2	2	0.3	0.0001	0.4988
8	features	small	GAT	BCELogit(weight)	2	2	0.5	0.0001	0.5062
9	features	small	GCN	BCELogit(weight)	2	2	0.3	0.0001	0.5077
10	features	small	GCN	BCELogit(weight)	3	3	0.3	0.001	0.5084
11	features	small	GAT	BCELogit(weight)	2	2	0.3	0.0005	0.52
12	features	large	GAT	BCELogit(weight)	2	1	0.3	0.0001	0.5204

Results

- It is peculiar that the results are almost the same throughout the modeling process
- The similarity of the results was why GAT was introduced as an alternative modeling methodology
- One reason could be that there not enough distinguishing features to accurately predict accounts that are involved in money laundering
  - In a real-life scenario, other information about the owner of the account would be available and provide context of multiple accounts being owned by the same person.
  - In this dataset, there is no information on individuals and their assets, which makes the task more difficult
- It takes about 6 hours to train the neural network with 50 epochs (170000000*50/6/60/60 = ~400k records processed for training per second)
- A smaller dataset was used for the iterative approach of model selection and hyperparameter tuning. The best performing model from this dataset was used on the full large dataset.

Project Architecture

Project Architecture



Project Architecture

- Node attributes:
  - accountID
  - is_laundering
  - bank
  - transactions_from
  - transactions_to
  - total_usd_from
  - total_usd_to
  - avg_usd_from
  - avg_usd_to
  - total_currencies_from
  - total_currencies_to
- Transaction attributes:
  - amount_usd
  - bank_from
  - bank_to
  - currency_from
  - currency_to
  - is_laundering
  - payment_format
  - year, month, day
  - hour, minute

Gantt Chart Of Project Timeline

Project Timeline

Future Work

Future Work

- Try other types of models (evolveGCN has been suggested in literature review for temporal aspect of the problem)
- Integrate the predictions into the visualization application from Phase 1
- Incorporate comments provided on Phase 1 project

References

References

- "Graph Attention Networks"
- "Anti-Money Laundering in Bitcoin"
- "EvolveGCN"
- Kaggle Dataset
- IBM Dataset Generation Documentation