SYSC 5108 - Deep Learning
Project Presentation

Deep Learning-Based
Credit card fraud transaction detection

Group #10
Jean-Gabriel Gaudreault
Sandeep Reddy Venna

Outline

1. Project Context
2. Exploration of the Data & Preprocessing
3. Methodology
4. Models Comparisons
5. Architecture Optimization
6. AutoEncoders
7. Conclusions

Introduction

Objectives:

- Find out and prevent fraudulent credit card transactions in order to protect customers.
- Test our knowledge of multiple Deep Neural Network (DNN) algorithms

Dataset:

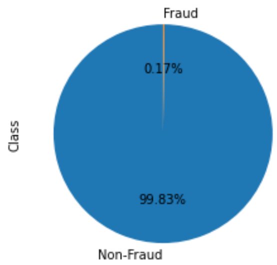
- Contains transactions made by credit cards in September 2013 by European cardholders.
- Presents transactions that occurred in two days with 492 frauds out of a total of 284,807 transactions.
- Highly unbalanced, positive class (frauds) accounts for 0.172% of all transactions.

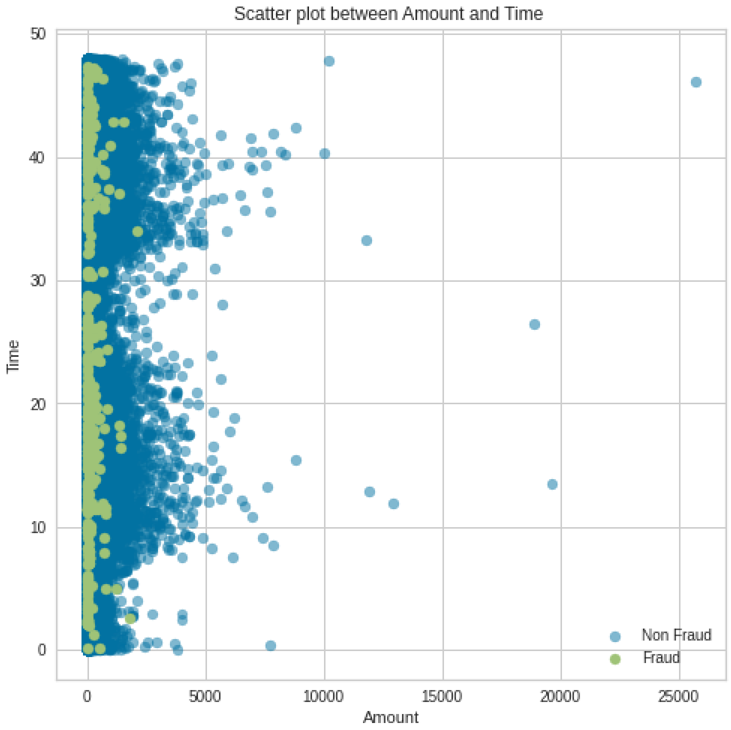
[Source: https://www.kaggle.com/datasets/mlg-ulb/creditcardfraud]

Exploratory Data Analysis

- Total number of features: **30**
- 28 anonymized variables, 1 "amount" variable, 1 "time" variable and 1 target variable..

	Target	Count	percent
0	0	284315	99.83
1	1	492	0.17





Data Preprocessing

1. Skewness Reduction (*PowerTransformer*)
   - **Goal**: Make the distributions of the features that present a high skew more Gaussian-like.
   - Performed on features with skewness higher than **[-1,1]**.
   - Performed on **19 Features**.

1. Standardization (*StandardScaler*)
   - **Goal**: Scale all data to be in a common range which can help improve model's performance.
   - Using "**Standardization Scaling**", i.e. Removing the mean and scaling to the variance.

Performance Evaluation

Problem: Since data is highly imbalanced, metrics such as accuracy can't be used effectively.

- Manually compare the confusion matrices
- Use **Precision**, **Recall** and **F1-Score** for performance evaluation.
- **F1-Score** combines both precision and recall (harmonic mean) for ease of comparison.

$$F_1 = 2 \cdot \frac{precision \cdot recall}{precision + recall}$$

Methodology

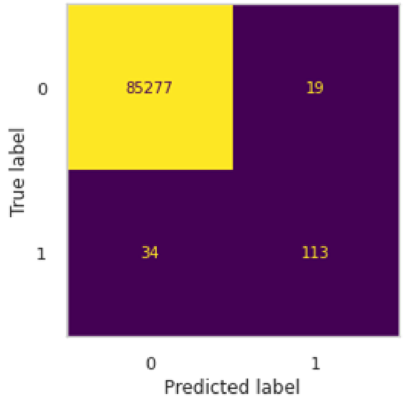
1. **Baseline Model on imbalanced data**
   - Loss Optimization: *Binary Cross Entropy*
   - Input Layer (*29 features input, 16 neurons*)
   - Dense Layer (*24 neurons, ReLU activation*)
   - Dropout Layer (*Frequency of 0.5*)
   - Dense Layer (*20 neurons, ReLU activation*)
   - Dense Layer (*24 neurons, ReLU activation*)
   - Output Layer (*Sigmoid activation*)
2. **Model on balanced data using SMOTE**
   - Oversampling the minority class (Fraudulent Transaction) to reach an equal amount of samples in both classes.
   - Using the same architecture as the baseline model.
3. **AutoML to improve model's architecture**
4. **AutoEncoders + Logistic Regression**

Models Comparison

Imbalanced Data vs. SMOTE Resampled Data

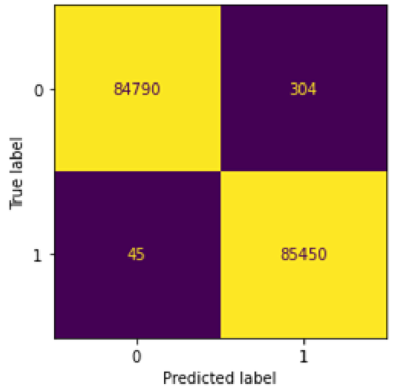
Imbalanced Data (***Baseline***)

- Precision: **0.856**
- Recall: **0.769**
- F-1 Score: **0.810**



SMOTE Resampled Data

- Precision: **0.996**
- Recall: **0.999**
- F-1 Score: **0.998**

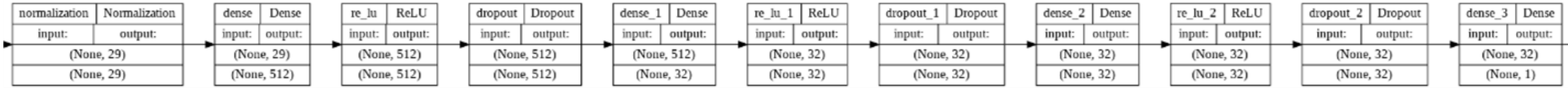


Architecture Optimization

Baseline Architecture vs. AutoML Optimized Architecture

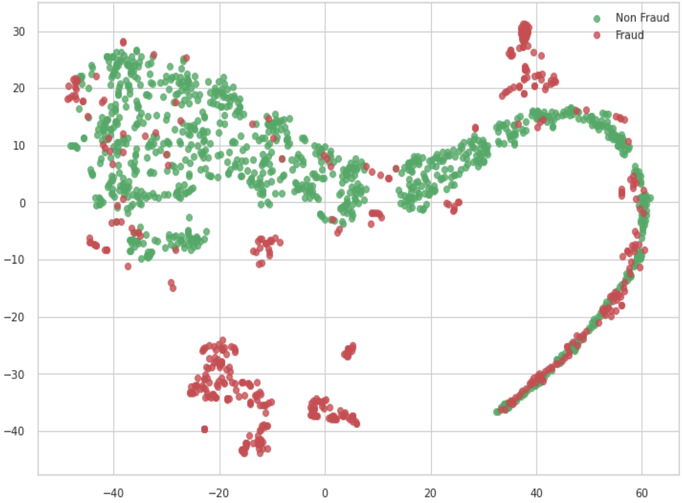
	Imbalanced Data			SMOTE Resampled Data		
	Precision	***Recall***	***F-1 Score***	***Precision***	***Recall***	***F-1 Score***
Baseline Architecture	0.856	0.769	0.810	0.996	0.999	0.998
AutoML Optimized Architecture	0.903	0.762	0.827	0.998	1.0	0.999

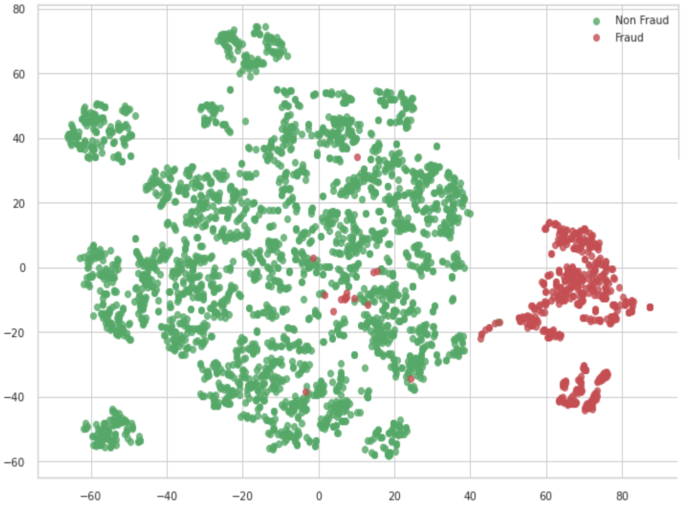
Optimized Architecture *(Resampled Data)*:



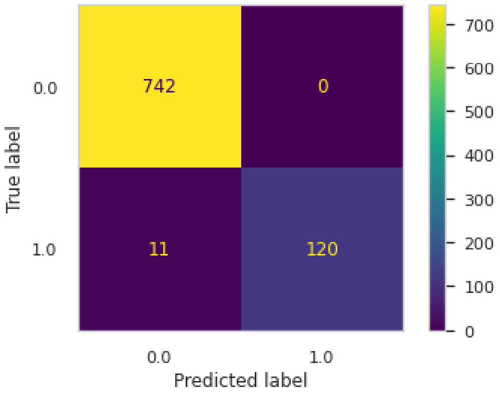
Semi-Supervised Learning

AutoEncoder + Logistic regression





- Precision: **1.0**
- Recall: **0.916**
- F1-score: **0.956**



Conclusions

Class Imbalance:

- Class Imbalance has a really big impact on model's performance.
- Without changing the model itself, we managed to substantially improve the model's performance by simply equalizing the class.
- Common problem (Anomaly Detection, Fraud Detection, etc.)

AutoEncoders:

- Too many samples of data are not required
- No need of any complex model to classify

Model Architecture:

- Model's precision improved substantially on imbalanced data, small improvements on balanced data.
- Computationally expensive, trials were limited to 5 in our case to stay at a manageable level, model architecture could potentially be improved much more.

General Conclusions:

- Once the dataset was balanced, a model could easily be fitted on the data. Performing the same experiments on a more complex dataset (multiclass, etc.) could be the next step forward and lead to interesting results.

Thanks!

Any questions?

You can find us at:

- jgaud042@uottawa.ca
- svenn075@uottawa.ca