Report Fraud detection

ECAM

BRUSSELS ENGINEERING SCHOOL

Name of the student

- Bouhnine Salaheddine (195159)

Name of the associate professor

- HASSELMANN Ken (HSL)

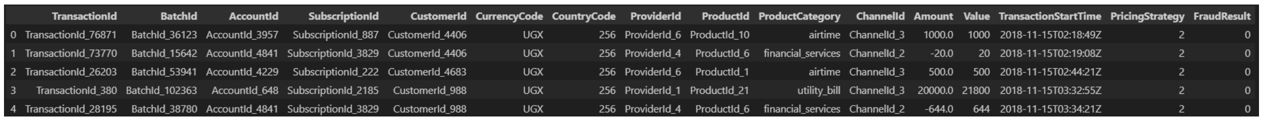
Contents

1 Introduction

For the laboratory of the cursus of Artificial Intelligence, we had to participate in a machine learning challenge in which we had to make an AI capable of detecting fraudulent transactions, specifically, the objective of this project is to develop a model that accurately predicts fraudulent transactions based on a dataset of historical transactions.

2 Data Analysis

First of all, the dataset contains imbalanced data: this is a situation in which the distribution of classes in a dataset is not equal or even. In other words, one class has significantly more examples than the other classes. This can lead to biased models, as they are more likely to predict the majority class, especially in classification tasks like fraud detection, indeed, in credit card transactions, the number of fraudulent transactions is significantly lower than the number of legitimate ones.



	TransactionId	BatchId	AccountId	SubscriptionId	CustomerId	CurrencyCode	CountryCode	ProviderId	ProductId	ProductCategory	ChannelId	Amount	Value	TransactionStartTime	PricingStrategy	FraudResult
0	TransactionId_76871	BatchId_36123	AccountId_3957	SubscriptionId_887	CustomerId_4406	UGX	256	ProviderId_6	ProductId_10	airtime	ChannelId_3	1000.0	1000	2018-11-15T02:18:49Z	2	0
1	TransactionId_73770	BatchId_15642	AccountId_4841	SubscriptionId_3829	CustomerId_4406	UGX	256	ProviderId_4	ProductId_6	financial_services	ChannelId_2	-20.0	20	2018-11-15T02:19:08Z	2	0
2	TransactionId_26203	BatchId_53941	AccountId_4229	SubscriptionId_222	CustomerId_4683	UGX	256	ProviderId_6	ProductId_1	airtime	ChannelId_3	500.0	500	2018-11-15T02:44:21Z	2	0
3	TransactionId_380	BatchId_102363	AccountId_648	SubscriptionId_2185	CustomerId_988	UGX	256	ProviderId_1	ProductId_21	utility_bill	ChannelId_3	20000.0	21800	2018-11-15T03:32:55Z	2	0
4	TransactionId_28195	BatchId_38780	AccountId_4841	SubscriptionId_3829	CustomerId_988	UGX	256	ProviderId_4	ProductId_6	financial_services	ChannelId_2	-644.0	644	2018-11-15T03:34:21Z	2	0

Figure 1: Visualization of the head of the dataset

3 Data Preprocessing and feature engineering

Data preprocessing is an important step in "cleaning" the dataset. In the project, the data preprocessing step involves handling missing values and encoding categorical variables using label encoding. The feature engineering part was performed by converting the transaction start times into separate features for hour, day, weekday and month, after that I extracted them and dropped irrelevant columns and using only the relevant features for model training

4 Handling Imbalanced Data

Since the data in the dataset are unbalanced, processing them is an important step to avoid overfitting problems. Several techniques can be employed to address this issue, including oversampling the minority class, undersampling the majority class, or using a combination of both. In this project, I chose SMOTE (Synthetic Minority Over-sampling Technique) to handle the imbalanced data. SMOTE works by generating synthetic samples for the minority class based on the interpolation of existing minority samples, thus creating a more diverse set of training instances without the risk of overfitting. The following code demonstrates the application of SMOTE in the project:

```
# Oversampling using SMOTE
smote = SMOTE(random_state=42)
X_train_resampled, y_train_resampled = smote.fit_resample(X_train, y_train)
```

5 Model Selection and Training

The Scikit-learn library was used to train and test the machine learning models. The dataset was split into training and testing sets, and model performance was assessed using metrics such as precision, recall and F1 score. In addition, we were given access to the challenge website, which allowed us to submit our submission file for an other performance score. The best model will be the one that will have the higher score and will be used to generate the submission file.

6 Model Evaluation and Comparison

In this section, we present the results of three classification models: Logistic Regression, Random Forest and XGBoost. The performance indicators include Precision, Recall and F1-Score. Precision indicates the proportion of true positives among the positive predictions for each class, while recall indicates the proportion of true positives among the actual samples for each class. The F1-score is the harmonic mean of precision and recall, which is used to assess the overall quality of the model by taking into account both precision and recall. The following tables show the results for each model :

Table 1: Logistic Regression Results

	Precision	Recall	F1-Score
Class 0	1.00	0.99	0.99
Class 1	0.14	1.00	0.25
Macro Avg	0.57	0.99	0.62
Weighted Avg	1.00	0.99	0.99

Table 2: Random Forest Results

	Precision	Recall	F1-Score
Class 0	1.00	1.00	1.00
Class 1	0.90	1.00	0.95
Macro Avg	0.95	1.00	0.97
Weighted Avg	1.00	1.00	1.00

Table 3: XGBoost Results

	Precision	Recall	F1-Score
Class 0	1.00	1.00	1.00
Class 1	0.82	1.00	0.90
Macro Avg	0.91	1.00	0.95
Weighted Avg	1.00	1.00	1.00

XGBoost shows a lower performance compared to the Random Forest model, despite the fact that this model is known for being a powerful and high-performing algorithm. Several factors may contribute to this result, including data characteristics, model hyperparameters, and training conditions. To enhance the performance of the XGBoost model, we can address these factors accordingly.

1. **Hyperparameter Tuning:** XGBoost generally requires careful hyperparameter tuning to achieve optimal performance. Experiment with different combinations of hyperparameters, such as the number of boosting rounds, learning rate, maximum tree depth, and regularization parameters. Techniques such as grid search, random search, or Bayesian optimization can be employed to systematically search for the optimal hyperparameter set.

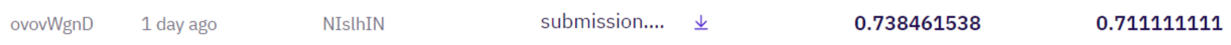
2. **Training Conditions:** Investigate the training conditions, such as the number of boosting rounds and learning rate. Increasing the number of boosting rounds or adjusting the learning rate can help the model converge to a better solution. Be cautious

not to overfit the model by using too many rounds or an overly aggressive learning rate.

3. **Cross-Validation:** Use cross-validation techniques to better estimate the model's performance on unseen data. K-fold cross-validation or stratified k-fold cross-validation can provide a more reliable evaluation of the model's performance, allowing you to compare the performance of XGBoost and Random Forest with higher confidence.

By addressing the mentioned factors and making necessary adjustments, the performance of the XGBoost model can potentially be improved, bringing it closer to or surpassing the performance of the Random Forest model. It is essential to iteratively experiment with these strategies and continually evaluate the model's performance on the validation dataset to find the best possible solution. However, I tried to find the hyperparameters that would give the best score for the XGBoost model but it took a lot of time and computing power so I abandoned this idea.

Finally, I obtained a score of 71.1% on the challenge website with the model that had obtained the best score during my tests, namely the Random Forest model.



ovovWgnD	1 day ago	NIslhIN	submission....	0.738461538	0.711111111

Figure 2: Score of the Random Forest model on the web site of the challenge

7 Conclusion

To conclude, this project was my first real experience in the field of machine learning, it was very useful because I was able to learn and put into practice the theoretical notions seen during the laboratories, in particular the use of the scikit-learn and pandas: I was able to explore and preprocess the data, select the most relevant features, balance the data, and choose a suitable model for our situation. After training the models, I evaluated them on the test set and measured their performance in terms of precision, recall and F-1-Score. Our results showed that detecting fraud in financial transactions is a complex task that requires careful exploration of the data and a good selection of the most relevant features. I also learned that it's important to look at different ways to measure performance and to test the model with various data for a complete and reliable evaluation.