INTERNATIONALE
HOCHSCHULE
FERNSTUDIUM

Project: Data Science Use Case

(DLMDSPDSUC01)

A Use Case on

Chargeback Fraud Detection with Machine Learning at Airbnb

Author: Kaushik Puttaswamy

Registration number: 321150196

Customer ID: 10549633

Date: 25/04/2023

Place: Mysore, Karnataka, India

.

Table of contents

List of figures

List of Tables

Abbreviations

ML	Machine Learning
IP	Internet Protocol
FPR	False Positive Rate
TPR	True Positive Rate
TN	True Negative
TP	True Positive
FP	False Positive
FN	False Negative
DSUC	Data Science Use Case
AUC	Area Under the Curve
ROC	Receiver Operator Curve
KPI	Key Performance Indicator

1. Different subject areas of the Machine Learning Canvas

1.1 Value Propositions

What are we trying to do for the end-user(s) of the predictive system? What objectives are we serving?

Airbnb is an online platform that connects people who rent out their homes with those who need a place to stay. Therefore, during payment via online mode, Airbnb must detect chargeback fraud since roughly two million guests stay in Airbnb-listed houses in 191 countries at any given time. This means that the rapid expansion of their worldwide network is based on customer trust(*Fighting Financial Fraud with Machine Learning at Airbnb*, n.d.).

Chargebacks, which are widespread in online businesses, are transactions that are charged by unauthorised individuals using stolen credit cards. When the cardholder realises their card has been stolen and detects illegal transactions on their statement, the credit card company asks the merchant (Airbnb) for a refund, and the merchant restores the money. Unlike other organisations, Airbnb takes the whole cost of chargebacks and does not pass on the financial risk to the hosts(*Fighting Financial Fraud with Machine Learning at Airbnb*, n.d.). As a result, it is important to develop a machine learning canvas as a necessary step to fight against financial fraud.

Furthermore, the objectives of this use case analysis are as listed below:

- Determining the data sources (internal and external) that will be used to train an ML model to accurately detect chargeback fraud and the criteria for classifying transactions as fraudulent or not.
- Explaining the amount of new data that needs to be collected to ensure that our model can detect current chargeback patterns.
- Based on the prediction, describing a set of decisions to provide desired value to the end user (Airbnb).
- In order to keep the model up to date, stating an approach to regularly analyse and retrain a model.
- Representation of different input data fields (features) that are extracted from the selected raw data source.
- Explaining the evaluation metrics and methods for measuring the ML model's performance before and after deployment.

1.2 ML Task

Input, output to predict, type of problem

It is absolutely essential for online businesses to be able to identify fraudulent online bookings by fraudsters using stolen credit cards for online transactions so that customers do not request chargebacks for services they did not purchase. The ML's job in this case is to determine if the transaction is real or fraudulent. Furthermore, we know that fraud is relatively uncommon; this is an *unbalanced classification task* with few positive (fraud) classifications(*Fighting Financial Fraud with Machine Learning at Airbnb*, n.d.). As a result, when it comes to fraud detection, the more data there is, the more accurate the model and It can learn and improve its fraud detection abilities(Dima, 2022). Furthermore, transaction details such as *Payment platform, Currency, Payment method, Payment country, Payment amount, and so on are input for the model*, and because the ML model is trained on categorical and numerical datasets, the machine learning model is able to learn from patterns of normal behaviour. ML model learn quickly to changes in typical behaviour and can immediately recognise patterns of fraud transactions(*Machine Learning for Fraud Detection*, n.d.), allowing them to *predict whether a transaction is fraudulent or not as a model output*.

1.3 Data sources

Which data sources can we use (internal and external)?

To detect chargeback financial fraud transactions using a machine-learning model, the model must be trained on previous examples (datasets) of confirmed good and confirmed chargeback fraudulent behaviour patterns because we must deal with multiple scenarios such as false positives, false negatives, and true positives, as with any other ML model(*Fighting Financial Fraud with Machine Learning at Airbnb*, n.d.).

In case of internal data sources, that could be useful for chargeback fraud detection at Airbnb:

Transaction data: Transaction data includes information about the user, the booking, the payment method used, the amount paid, and the transaction date. This information can be used to identify unusual patterns or behaviours that may indicate fraud.

User data: User data includes details such as the user's location, device type, login time, behaviour, and history. This information can be used to spot anomalies and suspicious behaviour.

Chargeback history: Airbnb's internal chargeback history can provide useful datasets for detecting fraudulent chargebacks. We can identify potential fraudulent activity by analysing the patterns and trends of chargeback requests.

Furthermore, since it is an unbalanced classification problem, the datasets may provide a barrier for certain classic machine learning approaches; nonetheless, there are some cases where the natural distribution of the data across the classes is not equal. This is common in fraud detection issues(Ravaglia, 2022). As a result, it is important to use an *external open data source* (https://www.neuraldesigner.com/files/datasets/creditcard-fraud.csv) as this dataset includes many legitimate transactions, and integrating these databases into the chargeback fraud detection system can aid in the identification of potential fraudsters. Moreover, this dataset contains 11 features about 3075 payments(*Credit Card Fraud Detection Using Machine Learning*, n.d.). In addition, it is recommended to use a publicly available *external data source* containing one month of raw credit card transactions (https://www.kaggle.com/datasets/dmirandaalves/predict-chargeback-frauds-payment). It also indicates whether the transaction was detected as a chargeback. Using external industry datasets on the latest fraud trends and techniques, Airbnb can stay ahead of fraudsters and better protect its platform.

1.4 Collecting Data

How do we get new data to learn from (inputs and outputs)?

The new data needs to be collected from Airbnb's transaction data database. To be meaningful in this use case analysis, the data must contain a representative number of chargebacks. We wanted to use as much new data as possible to ensure that our model could detect current chargeback patterns. However, because banks report chargebacks with a few months' delay, we had to select data from the beginning of the year. Furthermore, Airbnb handles a massive volume of transactions. A year's worth of data would provide about a hundred million transactions (Fick & Gunther, n.d.). However, in order to use the most recent transaction data to retrain the ML model, we must define new data on payment transaction details every couple of weeks, so that every time a customer makes a purchase, we will collect transactional data via the customer's transaction history, which is usually recorded automatically through the point-of-sale system or the platform that company (Airbnb) uses to manage their website.

Furthermore, the new data collection includes categorical and numerical data. where the data sets include both generic transaction data like amount, currency, and product category as well as additional user-specific information like payment nation, IP address, fraud score, etc (Fick & Gunther, n.d.).

1.5 Decisions

How are predictions used to make decisions that provide the proposed value to the end-user?

The task of the ML system is to predict that a booking is fraudulent, which led to the chargebacks. However, this prediction may be difficult to get right, and getting an accurate prediction of values that would be observed after a long time can be impressive, but in the end, predictions are just information, and they do not do anything useful on their own. Therefore, we need to turn prediction into a decision that delivers the value that we proposed. Furthermore, to make things concrete in terms of what to do with prediction, it is important to consider when a decision should be made. This could be dictated by how frequently end users will use the ML system. The latter depends on how we can or how we choose to integrate the system into their workflows.

In this case of chargeback prediction, it sounds reasonable to work with the end-user to agree on a new workflow that uses an ML system and to decide how often the system will make its decisions (or recommendations) available (e.g., on a weekly or monthly basis). This will then have an impact on how far into the future we want to predict things, hence the ML task. For instance, if we build a chargeback prevention system that will be used on a weekly basis based on the time gap since the previous prediction, it makes sense to predict chargebacks on a weekly or monthly basis. There are several reasons why we should concentrate on predicting whether a transaction will result in a chargeback for the duration of week or months gap from the old prediction, as follows:

a) Timeliness: By predicting whether a transaction will result in a chargeback as soon as possible, we can take immediate action to avoid or reduce the chargeback. This enables us to address the issue before it escalates.
b) Resource allocation: We can allocate resources more efficiently by focusing on transactions that are likely to result in a chargeback in the near future. For example, rather than wasting time and resources on transactions that are unlikely to result in a chargeback, we can prioritise reviewing or investigating these transactions.
c) Accuracy: Predicting future chargeback status may be more difficult, necessitating additional data and modelling resources. We can potentially achieve higher accuracy and reduce false positives and negatives by focusing on predicting chargebacks in the coming weeks of transaction data.

However, in addition to predicting transaction that result chargebacks, it may be useful to rate each incoming transaction on its eventual chargeback risk. This data can provide a more complete picture of the transaction risk and help to inform longer-term strategies and decisions.

In some ML use cases, our input could be made up of features which are given and fixed, and others that we could control. For instance, in chargeback prediction, we might be able to change payment features such as payment method, amount and duration. We can think of the controllable features as levers that can be pulled to influence an outcome. The decisions to make could be to adjust these feature values in order to maximize a certain quantity (e.g., predicted sales), or the probability of observing a desired outcome (genuine transaction).

In addition, decisions are often based on the model’s confidence in its predictions. In this case of chargeback prediction case. If the model is very confident that the transaction is real/fake, we can let the system automatically decide to accept/reject that transaction. Otherwise, we may leave the decision to a human. A confidence value (usually between 0 and 1) is given by the model along with each of its predictions, and it can be used to automate decisions when above certain thresholds.

For chargeback prediction, we propose to do the following:

- Filter out customers who are not predicted to fraudulent, and anomalous customers
- Sort customers by descending chargeback probability times monthly revenue loss
- Target the first K customers in the list

To target the first K chargeback fraud customers on the list, we must examine the transaction amount and purchase history for patterns that are unusual or inconsistent with their previous behaviour, as well as the billing address, email address, and IP address associated with each transaction to see if they match the customer's known information.

Where K is a parameter, whose value will need to be fixed, and so are the thresholds. Changing them, or in general, any parameter of the decision-making system, will have an impact on the performance of the whole system. We’ll need to discuss performance evaluation and define metrics of interest before we can start looking for optimal parameter values.

The ultimate vision when building intelligent systems can be to automate decisions completely. In this domain, decisions made by a machine alone are better than those made with a human in the loop. In others, involving humans is hardly feasible, so automation is just a necessity.

Nevertheless, it is recommended to avoid full automation when we are just starting out with ML. Instead, we could list all possible decisions to be made, have the machine sort them based on predictions, and have a human review. The machine would be providing its result, and the final decisions would be ours(Dorard, 2021).

1.6 Making Prediction

When do we make predictions on new inputs? How long do we have to featurize a new input and make a prediction?

Making predictions on new inputs when the trained model is deployed in a production environment or used for inference on a dataset not previously seen during training requires the new input dataset to be obtained from available data sources (databases, web scraping, etc.) and must go through pre-processing and be transformed into a format that the model can understand before being input into the ML model, which means the representative size of the dataset must be lowered and misleading information eliminated. A dataset will comprise variables (i.e., features) with numerical, category, and/or textual values. Certain aspects may not be relevant to or compatible with our targeted use cases.

As a result, features should be carefully chosen so they can be leveraged to define output value propositions. Overall, this process is known as featurization. This data must now be fed into the trained ML model, which must have already been run numerous times until it achieves a relatively high degree of accuracy in comparison to the testing set. In addition, because it is a classification problem, the output of a prediction model is probability. Moreover, depending on a predefined threshold, for example, transactions with a probability greater than 80% are reported as fraudulent. Further, it is absolutely essential to predict fraud on a weekly basis because this enables Airbnb to analyse enormous volumes of online transaction data to find prospective events and opportunities before they occur, allowing for better decisions to be made. Furthermore, based on the predicted value, Airbnb will take the appropriate action and make the appropriate decision so that overall company objectives and project goals are aligned.

However, identifying the best feature, featurizing new input data, and making a prediction for forthcoming events can vary in chargeback detection depending on various factors, such as the size and complexity of the dataset. Furthermore, featurizing is instantaneous since feature development doesn't take too long in our use case because new online bookings and transactions are occurring simultaneously. Finally, making predictions for forthcoming events can also be relatively quick because the dataset size is relatively small since we are collecting a certain couple of weeks of transaction data and because the data is comparatively less complex. As a result, new transaction and online booking data must be filtered and entered into the trained ML model to provide the most up-to-date predictions for chargeback fraud detection.

1.7 Building Model

When do we create/update models with new training data?

In our use case, whenever there is a significant change in the data distribution or fraud patterns over time, the chargeback fraud detection ML models should be updated with new training data. This is because if ML models are not trained on new data that reflects the most recent fraud trends and patterns, their performance may deteriorate over time.

ML models for chargeback fraud detection may need to be updated with new training data in a variety of cases, including:

- New fraud patterns: The chargeback fraud detection ML model may need to be updated to recognize such patterns if new fraud types or fraudsters' methods for committing fraud are discovered.

- Changes in the data distribution: The ML model might need to be updated to account for changes if the distribution of chargeback data significantly changes, for instance, if customer behaviour or the types of transactions being processed change.
- New data sources: Our ML model might need to be updated to include new data if new sources of data become available, such as if a merchant (Airbnb) starts gathering more information about chargeback transactions.
- Performance degradation: It may be time to update the model with new training data if the performance of our ML model begins to deteriorate over time, as in the case of an increase in the false-positive rate or a decrease in accuracy.

Furthermore, in order to maintain our ML model's effectiveness at identifying and preventing fraud, chargeback fraud detection ML models should typically be updated with new training data on a regular basis, for instance, every couple of weeks to six months.

In addition, while creating a machine learning model, it is critical to understand how our data will evolve over time. A well-designed system should take this into consideration, and we should formulate a plan for keeping our models up-to-date.

Furthermore, keeping a chargeback fraud detection ML model up-to-date is important to ensure its effectiveness in detecting and preventing fraudulent activities. Here are some strategies to consider:

- Collect and analyze new data: Data should be gathered and analyzed as new transactions are made in order to look for any patterns or anomalies that might point to fraud. This will enable us to incorporate the most recent data into our model.
- Monitor model performance: Monitoring the performance of our ML model on a regular basis to spot any adjustments or problems that might affect its accuracy. This will give us the flexibility to modify it as needed to increase its efficacy.
- Use feedback from users: On the other hand, users should be advised to report any transactions they believe to be fraudulent by leaving feedback. This will enable us to spot newer forms of fraud that our model might not have previously picked up on and help us improve our ML model accordingly.
- Stay up-to-date with industry trends: Following the trend of the most recent developments and trends in chargeback fraud. This will enable us to understand new types of fraud and modify our model accordingly.
- Use multiple detection methods: Aside from our ML model, we might also use rule-based systems, behavioural analysis, and manual reviews as additional detection techniques. Our fraud detection will be more accurate as a result, and the number of false positives will decrease.

- Continuously improve and refine the model: Monitoring and evaluating our model's performance over time, then making improvements based on the results. This will make it easier to maintain our model's effectiveness in identifying and preventing chargeback fraud as it evolves.

1.8 Features

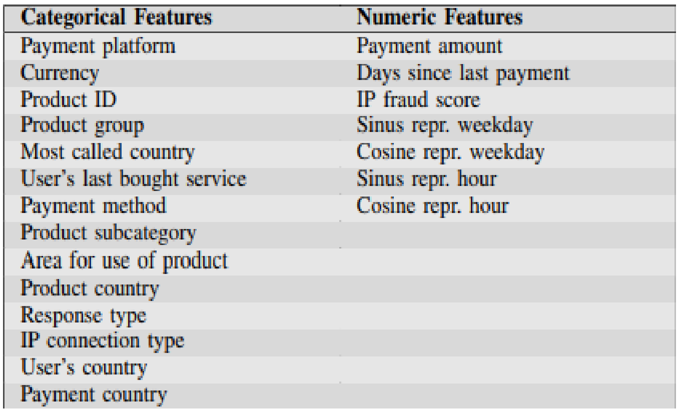
Input representations extracted from rawdata sources.

According to the research (Ucar, 2020) feature selection is an essential aspect of data pre-processing in machine learning. Feature selection algorithms can be used to identify significant features from a large set of options. Aside from that, feature selection techniques may be employed to select features that enhance model performance. They can also assist in accelerating model learning. because datasets with numerous features slow down learning processes. The research in (Pant & Srivastava, 2015) further reveals that difficulties of unbalanced data or category imbalance are frequently encountered in actual applications of machine learning or data mining. Assuming that datasets contain two categories of outcomes and the number of results in one of the categories in the training sample is much higher than the number of results in the other category, the datasets will be imbalanced. In these cases, the model's performance will be less accurate than expected because the model will be biased towards the category with the highest number of results and will incorrectly classify the targeted category with the lowest number of results as the category with the highest number of results, lowering its accuracy. Feature selection can also be used to address unbalanced dataset issues. The goal of feature selection is to locate features that are significantly associated with results among all features, and these selected features allow a model to perform better(Wei et al., 2022).

The external dataset file, which is data source file openly available mentioned in the *data source section* (https://www.neuraldesigner.com/files/datasets/creditcard-fraud.csv), which contains 11 features about 3075 payments. Furthermore, Input representations extracted from available raw data sources includes the following variable.

- merchant_id: id of the merchant is used to identify and track the transactions processed by the merchant.
- avg_amount_day: average of amount per transaction per day.
- transaction_amount: the amount of the transaction that the customer had made.
- is_declined: yes = the credit card is declined, no = the credit card is not declined.
- number_declines_day: total number of declines per day.
- foreign_transaction: yes = it is a foreign transaction, no = it is not a foreign transaction.
- high_risk_country: yes = it is a high-risk country, no = it is not a high-risk country.
- daily_chbk_avg_amt: daily average of chargeback.

- 6m_avg_chbk_amt: 6-months average of chargeback.
- 6m_chbk_freq: frequency of the 6-months chargeback.
- is_fradulent: fraudulent = the payment is fraudulent, not-fraudulent = the payment is not fraudulent (target variable).



Categorical Features	Numeric Features
Payment platform	Payment amount
Currency	Days since last payment
Product ID	IP fraud score
Product group	Sinus repr. weekday
Most called country	Cosine repr. weekday
User's last bought service	Sinus repr. hour
Payment method	Cosine repr. hour
Product subcategory	
Area for use of product	
Product country	
Response type	
IP connection type	
User's country	
Payment country	

Table1: Features information

Source: https://www.diva-portal.org/smash/get/diva2:1706610/FULLTEXT01.pdf

In addition, as mentioned in the above table 1, the data source contains 23 required fields (features) that are essential to be utilised in training our model. The features were chosen based on important attributes that occur in online transactions. When selecting features, our objective was to collect as many distinct features as possible that have a significant impact upon training the model, and all of these features are accessible at the time of transaction acceptance, which was made by the customer at the time of booking. This assures that the model can be employed in a real-world scenario. In this way, we don't know what features will be useful in the chargeback fraud classification task. Furthermore, by only examining features accessible at the time of purchase, we avoid selecting features that are a result of a chargeback occurring. Such features run the risk of destroying the model by effectively training on the feature we're attempting to predict.

1.9 Offline evaluation

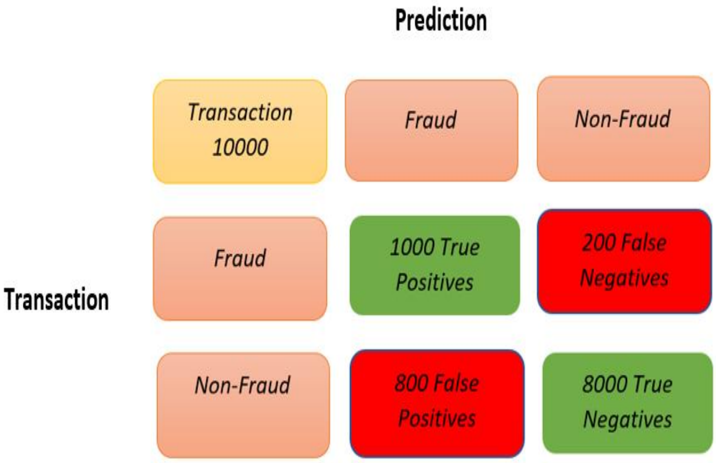
Methods and metrics to evaluate the system before deployment?

Machine learning tasks are aligned to evaluation measures. There are several evaluation metrics for chargeback fraud detection (classification tasks). Some measures, such as precision recall, are beneficial for a variety of applications. as an example of supervised learning, which accounts for the vast majority of machine learning applications. Using various performance measures, we should be able to increase our model's overall prediction power before deploying it in production on unknown data. Without properly evaluating the machine learning model using several assessment metrics and relying just on accuracy, there can be issues when the model is deployed on unknown data, causing inaccurate predictions(Agrawal, 2021).

There are many ways to measure our classification ML model's performance. Confusion matrix, precision, recall, F1 score, and AUC-ROC are the most popular metrics that we have used, and moreover, those that address the imbalanced data classification issue are described below.

a) Confusion Matrix:

The confusion matrix is a key component that can be used to assess the performance of our unbalanced ML classification model, but it is not a metric. It is, by nature, a two-dimensional table that displays actual and predicted values("Machine Learning Metrics," n.d.). In our case, we need to create a classifier that can predict chargebacks.



		Prediction	
	Transaction 10000	*Fraud*	*Non-Fraud*
Transaction	*Fraud*	*1000 True Positives*	*200 False Negatives*
	Non-Fraud	*800 False Positives*	*8000 True Negatives*

Figure1: Confusion Matrix example

Furthermore, both dimensions include class instances, such as:

- True Positive (TP) — A class is predicted to be true and is really true (Transaction that are fraud and predicted as chargeback);
- True Negative (TN) — A class is predicted to be false and is found to be false (Transaction that are not fraud and predicted as genuine transaction);
- False Positive (FP) — A class is predicted to be true but is really false (Transaction that are genuine but predicted as fraud); and
- False Negative (FN) — A class is predicted to be false but is really true (Transaction that are fraud but predicted as genuine).

b) Precision:

$$\text{Precision}=\frac{Number\ of\ correct\ positive\ result}{Total\ number\ of\ positive\ result}$$

Precision indicates what proportion of all positive predictions were accurate. To compute it, divide the number of correct positive outcomes (TP) by the total number of positive results predicted by the classifier (TP + FP).

In our example, how many of the transactions identified as fraudulent by the model were accurately classified? We divide the entire number of fraud cases and predicted fraud transactions (1,000) by the total number of fraud cases and predicted fraud transactions (1,000) and those transactions that are genuine but predicted as fraud (800). The precision is 55.7 percent.

Precision performs effectively in our cases where we need to or are able to prevent false negatives but cannot disregard false positives.

When dealing with unbalanced data, precision is our go-to assessment metric. However, it is not an ultimate solution because there are instances when both false and true negatives should be considered. For example, it is critical to know how many fraudulent transactions were misclassified as real and then ignored(“Machine Learning Metrics,” n.d.). However, since it ignores the trade-offs between recall and precision, precision as a stand-alone metric may not be adequate for assessing the effectiveness of a chargeback fraud detection system (i.e., the ratio of true positives to all actual fraudulent transactions). A system with high precision may have a low recall, which means it can accurately identify fraudulent transactions but may miss many actual fraudulent transactions. Therefore, we should aim for a balance between recall and precision. In addition to addressing the imbalance issue, the technique is to use ensemble methods, such as bagging or boosting, to combine multiple models trained on balanced data(Kalirane, 2023).

c) Recall:

$$\text{Recall}=\frac{Number\ of\ correct\ positives}{Number\ of\ all\ positives}$$

Recall is the proportion of correct positive predictions produced out of all possible positive predictions made by a model. To compute it, divide the total number of true positives by the total number of true positives and false negatives in the dataset. In this manner, recall indicates missed positive predictions.

In our case, it answers the question, "How many transactions did the model correctly predict as fraud out of all truly fraudulent transactions?" So, if we follow the formula, we'll obtain 83.3 percent of the model's correct predictions for all positives. The closer our model is to one, the better it is since it does not miss any true positives. The recall measure, like precision, is not a one-size-fits-all answer(“Machine Learning Metrics,” n.d.).

d) F1 Score:

F1 Score=Harmonic mean of Precision and Recall=$2 * \frac{Precision*Recall}{Precision+Recall}$=$\frac{2TP}{2TP+FP+FN}$

The F1 Score attempts to strike a balance between precision and recall by calculating their harmonic mean. It is a test of accuracy where the maximum value is 1. This indicates perfect precision and recall.

The F1 Score is a more sophisticated statistic that helps us achieve more accurate results in unbalanced classification situations. In our model, for example, the average of precision and recall is 69.5 percent, whereas the F1 score is 66.76 percent.

In comparison to a high F1 score, a low one is less informative. It simply provides information on performance at a certain level. We won't know if it's a recall or a precision problem with it (“Machine Learning Metrics,” n.d.). Furthermore, since F1 score is a harmonic mean of precision and recall, and it balances the trade-off between them by taking into account of both false positives and false negatives. Moreover, F1 score becomes especially valuable when working on classification models in which our data set is imbalanced to address this class imbalance, it's important to use a weighted version of the F1 score, where the contribution of each class to the score is proportional to the number of samples in that class. This ensures that the model is evaluated equally across all classes, regardless of their size. Furthermore, the weighted average of the F1 scores for each class is computed to obtain the weighted F1 score. The following is the formula for the weighted F1 score(Shmueli, 2022):

weighted F1 score = (w_1 * F1 score for class 1 + w_2 * F1 score for class 2 + ... + w_n * F1 score for class n) / (w_1 + w_2 + ... + w_n)

where the weights for each class are w_1, w_2,..., w_n.

In our case, where there is a class imbalance in the dataset, the weighted F1 score offers a more accurate indication of the overall performance of the ML model.

e) AUC-ROC:

The Receiver Operator Characteristic (ROC) is a probability curve that shows the TPR (True Positive Rate) vs the FPR (False Positive Rate) at various threshold levels, separating the "signal" from the "noise".

The Area Under the Curve (AUC) is a measure of a classifier's ability to discriminate between classes. For example, we can simply express the area of the curve ABDE and the X and Y axes from the graph.

According to the graph below, the greater the AUC, the better the model's performance at different threshold points between positive and negative classifications. This basically indicates that when AUC is equal to 1, the classifier can perfectly discriminate between all positive and negative class points. When AUC is equal to 0, the classifier predicts all negatives as positives and vice versa. When AUC is 0.5, the classifier is unable to discriminate between the positive and negative classes.

High AUC-ROC scores in the context of our chargeback fraud detection ML model show that the model is capable of accurately identifying a large portion of fraudulent transactions while minimising false positives.

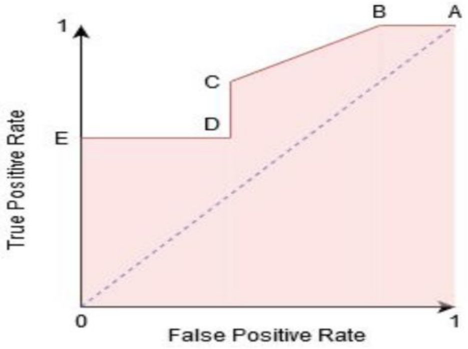


Figure2: Example ROC curve of a ML model predicting

Image source: https://www.analyticsvidhya.com/blog/2020/06/auc-roc-curve-machine-learning/

Working of AUC —In a ROC curve, the X-axis value represents the False Positive Rate (FPR), while the Y-axis value represents the True Positive Rate (TPR). A higher X-axis value implies a greater number of False Positives (FP) than True Negatives (TN), whereas a higher Y-axis value suggests a greater number of TP than FN. As a result, the threshold chosen is determined by the capacity to balance between FP and FN(Agrawal, 2021).

1.9.1 The impact of the *Precision-Recall* trade-off in fraud detection

The ideal prediction model should be able to identify every fraud in the dataset and confirm that each fraud that is predicted is actually a fraud. Using our metrics, it translates to having high Recall (detection of all fraud) and high Precision (as little error as possible on the transactions predicted as fraud). A trade-off exists between precision and recall; as we strive to increase precision, the value of recall will decline, and vice versa.

I. *Precision optimization:* If we fine-tune the parameters of our classifier so that it will only classify as fraud the transactions with a high probability of being real frauds, we will be very confident that a predicted transaction marked as fraud is actually a real fraud. The disadvantage is that not all fraudulent transactions will be caught, including those that most closely resemble legitimate transactions. By classifying fraudulent transactions as normal ones, the system will lose some of them, which could be a problem.

II. *Recall optimization:* We'll be very confident that we are detecting all the frauds if we tune the parameters of our classifier in a way that it will detect as much fraud as possible. The drawback is that in trying to detect all the fraud, even the normal transactions that are similar to fraud will be classified as such.

Precision-recall is a trade-off; by preferring precision optimization, we are detecting less fraud and decreasing the recall. By preferring recall optimization, we are detecting more fraud, but it is possible that some of them will be misclassified, lowering the precision value. We can prioritize one metric over another, but it's crucial to always keep both in mind. A model that achieves 99% precision and 15% recall, for instance, is not a good predictor because, despite its extremely high precision rate, it is unable to identify 85% of the fraud!!(Ravaglia, 2022).

Overall, in our chargeback fraud detection ML model, it is more crucial to catch all fraud, regardless of whether there are false alarms (***recall optimization***), because missed fraudulent transactions can result in sizable losses for the company (Airbnb), and chargeback fraud can have serious financial and reputational repercussions. To reduce these losses, it is essential to find as many fraud cases as we can.

1.10 Live evaluation and monitoring

Methods and metrics to evaluate the system after deployment, and to quantify value creation.

After successfully evaluating the prediction model using the above-mentioned evaluation metrics, it is ready to be used to generate the DSUC (Chargeback fraud detection with machine learning at Airbnb) value for the associated chargeback detection online business transaction problem. Meanwhile, the end user (i.e., the decision-maker) should be able to assess if the DSUC value has been successfully implemented in their organisation (Airbnb). This evaluation is carried out using a set of key performance indicators (KPIs) to determine how many business objectives have been met. Most KPIs will be concerned with growing revenue, lowering chargeback costs, boosting efficiency, and/or improving customer satisfaction.

In our case, effective KPIs frequently applied to monitor DSUC performance post-ML model deployment includes the following:

i. Precision and recall: Recall is the percent of actual fraudulent cases that the ML model correctly identified, while precision is the percentage of detected fraud cases that are fraudulent. To make sure the model is correctly detecting chargeback fraud, it is essential to monitor precision and recall.

ii. False positive rate: This metric measures the proportion of non-fraudulent transactions that the ML model incorrectly flags as fraudulent. To minimize interference with authorized transactions, it is critical to maintain this rate as low as possible.

iii. False negative rate: This is the percentage of fraudulent transactions that the ML model incorrectly classifies as non-fraudulent. In order to prevent interference with legal transactions, it is crucial to keep this rate as low as possible.

iv. Average time to detect: This is the typical amount of time the ML model needs to identify a fraudulent transaction. This metric will be closely monitored to ensure that fraud is found as soon as possible.

v. Reduction in chargeback rate: The percentage of transactions that are charged back as a result of fraud is known as the "chargeback rate." By continuously enhancing the functionality of our ML model, it is necessary to eventually lower this rate.

vi. Detection rate: This is the proportion of fraudulent transactions that the ML model has identified. To reduce losses brought on by chargeback fraud, a high detection rate must be attained.

These KPIs aid in tracking the effectiveness of our ML model so that adjustments can be made as necessary to effectively identify and stop chargeback fraud.

2. Filled-in version of the Machine Learning Canvas

Decisions

How are predictions used to make decisionsthat provide the proposed value to the end-user?

Computing predictions for all payment transaction data that had occurred recently (a couple of weeks ago) and that are used to make decisions:

- Filter out customers who are not predicted to fraudulent, and anomalous customers.

- Sort customers by descending chargeback probability times monthly revenue loss.

- Target the first K customers in the list.

ML Task

Input, output to predict, type of problem

Input: Transaction details such as Payment platform, Currency, Payment method, Payment country, Payment amount, and so on

Output: Classification task- Predict whether a fraudulent online booking or not

Value Propositions

What are we trying to do for the end-user(s)of the predictive system? What objectives are we serving?

Developing a machine learning canvas using various machine learning techniques with the goal of reducing Airbnb's own exposure to chargeback fraud; moreover, the main aim of this task is to provide a reliable and effective ML canvas in which the final outcome of the predictive system should enable end-users to detect and prevent fraudulent chargeback requests, minimise revenue losses, and maintain a positive business reputation.

Objective:

•Determining the data sources (internal and external)

•Explaining the amount of new data that needs to be collected

•Describing a set of decisions to provide desired value to the end user

•Stating an approach to regularly analyse and retrain a model

•Representation of different input data fields (features)

•Explaining the evaluation metrics and methods for measuring ML model's performance

Data Sources

Which data sources can we use (internal and external)?

•Internal: Transaction data, User data, Chargeback history

•External: (https://www.neuraldesigner.com/files/datasets/creditcard-fraud.csv) This dataset includes many legitimate transactions and contains 11 features about 3075 payments.

(https://www.kaggle.com/datasets/dmirandaalves/predict-chargeback-frauds-payment) This data source containing one month of raw credit card transactions.

Collecting Data

How do we get new data to learn from (inputs and outputs)?

Collecting new transactional data via the customer's transaction history, which is usually recorded automatically through the point-of-sale system or the platform that company (Airbnb) uses to manage their website. Furthermore, redefining the data on new transactions and online bookings that should contain a payment, features, and a representative number of chargebacks is requested.

Making Predictions

When do we make predictions on new inputs? How long do we have to featurize a new input and make a prediction?

Making predictions on new inputs when the trained model is deployed in a production environment or used for inference on a dataset not previously seen during training requires the new input dataset to be obtained from available data sources (databases, web scraping, etc.).

Featurizing is instantaneous since feature development doesn't take too long in our use case because new online bookings and transactions are occurring simultaneously. Finally, making predictions for forthcoming events can also be relatively quick, because the dataset size is relatively small since we are collecting a certain week of transaction data and it is comparatively less complex.

Offline Evaluation

Methods and metrics to evaluate thesystem before deployment?

Metrics to measure classification performance offline are,

- Confusion matrix
- Precision
- Recall
- F1 score
- AUC-ROC

Features

Input representations extracted from rawdata sources.

Features that are significantly associated with results among all features, and these selected features allow a model to perform better,

Example: Payment platform, Currency, Payment Method, Product ID, Product Group, most called country, users last bought service, Payment amount, Days since last payment etc....

Building Model

When do we create/update models with new training data?

ML models for chargeback fraud detection may need to be updated with new training data in a variety of cases, including:

- New fraud patterns
- Changes in the data distribution
- New data sources

Performance degradation keeping a chargeback fraud detection ML model up-to-date is important to ensure its effectiveness in detecting and preventing fraudulent activities. Here are some strategies to consider:

- Collect and analyze new data
- Monitor model performance
- Use feedback from users
- Stay up-to-date with industry trends
- Use multiple detection methods
- Continuously improve and refine the model

Live Evaluation and Monitoring

Methods and metrics to evaluate the system after deployment, and to quantify value creation.

- Precision and recall

- False positive rate

- False negative rate

- Average time to detect

- Reduction in chargeback rate

- Detection rate

References

Agrawal, S. K. (2021, July 20). Evaluation Metrics For Classification Model | Classification Model Metrics. *Analytics Vidhya*. https://www.analyticsvidhya.com/blog/2021/07/metrics-to-evaluate-your-classification-model-to-take-the-right-decisions/

Credit card fraud detection using machine learning. (n.d.). Retrieved March 19, 2023, from https://www.neuraldesigner.com/learning/examples/credit-card-fraud#DataSet

Dima. (2022, January 20). *How to Use Machine Learning in Fraud Detection and Prevention*. Intellias. https://intellias.com/how-to-use-machine-learning-in-fraud-detection/

Dorard, L. (2021, March 9). From Data to AI with the Machine Learning Canvas (Part III). *Own Machine Learning*. https://medium.com/louis-dorard/from-data-to-ai-with-the-machine-learning-canvas-part-iii-868fe17b9be6

Fick, O., & Gunther, T. (n.d.). *Detecting Chargebacks in Transaction Data with Artificial Neural Networks*.

Fighting Financial Fraud with Machine Learning at Airbnb. (n.d.). InfoQ. Retrieved January 14, 2023, from https://www.infoq.com/news/2018/03/financial-fraud-ml-airbnb/

Kalirane, M. (2023, January 20). Ensemble Learning Methods: Bagging, Boosting and Stacking. *Analytics Vidhya*. https://www.analyticsvidhya.com/blog/2023/01/ensemble-learning-methods-bagging-boosting-and-stacking/

Machine learning for fraud detection. (n.d.). Ravelin. Retrieved January 14, 2023, from https://www.ravelin.com/insights/machine-learning-for-fraud-detection

Machine Learning Metrics: How to Measure the Performance of a Machine Learning Model. (n.d.). *AltexSoft*. Retrieved January 25, 2023, from https://www.altexsoft.com/blog/machine-learning-metrics/

Pant, H., & Srivastava, D. R. (2015). *A SURVEY ON FEATURE SELECTION METHODS FOR IMBALANCED DATASETS*.

Ravaglia, A. (2022, December 21). Imbalanced classification in Fraud Detection. *Data Reply IT | DataTech*. https://medium.com/data-reply-it-datatech/imbalanced-classification-in-fraud-detection-8f63474ff8c7

Shmueli, B. (2022, July 21). *Multi-Class Metrics Made Simple, Part II: The F1-score*. Medium. https://towardsdatascience.com/multi-class-metrics-made-simple-part-ii-the-f1-score-ebe8b2c2ca1

Ucar, M. (2020). Classification Performance-Based Feature Selection Algorithm for Machine Learning: P-Score. *IRBM*, *41*. https://doi.org/10.1016/j.irbm.2020.01.006

Watson, M. (2018, March 8). Keeping Your Machine Learning Models Up-To-Date. *Center for Open Source Data and AI Technologies*. https://medium.com/codait/keeping-your-machine-learning-models-up-to-date-f1ead546591b

Wei, Y.-C., Lai, Y.-X., & Wu, M.-E. (2022). An evaluation of deep learning models for chargeback Fraud detection in online games. *Cluster Computing*. https://doi.org/10.1007/s10586-022-03674-4