Fraud and Anomaly Detection in Banking

A Step-by-Step Guide to Incorporating Machine Learning Into Models

A GUIDEBOOK BY DATAIKU

www.dataiku.com

INTRODUCTION

The business value of anomaly detection use cases within financial services is obvious. From credit card or check fraud to money laundering and cybersecurity, accurate, fast anomaly detection is necessary in order to conduct business and protect clients (not to mention the company) from potentially devastating losses.

The clearest use case for machine learning- or AI-based anomaly detection in banking is fraud detection. The U.S. Federal Trade Commission[1] has logged millions of fraud-related complaints in the last five years[2]; the scale of the problem is incredible. Yet it is very difficult to build an accurate system that works in real-time, as the number of fraudulent card charges is so much lower than the total volume (VISA alone processes over 2,000 transactions every second) that establishing a training set is difficult.

Traditional rules based systems are insufficient, often resulting in false positive rates that exceed 90%[3]. This creates a massive number of false positive alerts which then need to be cleared through human intervention. Repetitive analyst action can lead to a "numbness to false positives," raising operational risk and subsequently regulatory risk in the process.

Additionally, traditional systems are by definition reactive, since they can only generate alerts based on previous rules. Alternatively, AI-based systems are proactive and can help augment regulatory alert systems and improve analysts' workflow by reducing noise without discarding alerts.

Establishing a firm understanding of what data science, machine learning, and AI technologies can bring to fraud detection and other anomaly detection use cases in banking today is a first step to making headway in these areas. Combining this knowledge with more overarching AI project best practices, including operationalization and data democratization, will ensure that banks stay ahead of the curve.

[1] https://www.ftc.gov/news-events/media-resources/identity-theft
[2] https://www.creditcards.com/credit-card-news/credit-card-security-id-theft-fraud-statistics-1276.php
[3] https://www.forbes.com/sites/baininsights/2018/02/14/cutting-through-complexity-in-financial-crimes-compliance/#41a91853588d

ABOUT THE GUIDEBOOK

This guidebook is intended to provide a high-level overview of what fraud detection and anomaly detection are, how they work and when banks can apply them. By the end, readers should have an understanding of:

- What fraud and anomaly detection are, in technical terms.
- The three types of anomalies that can be detected.
- Use cases and examples of how banks use fraud and anomaly detection.
- How to build a basic, machine learning-based fraud detection system.
- Two use cases from financial institutions that have improved their ability to detect fraud by employing machine learning-based systems.

For those completely unfamiliar with data science in the context of fraud or anomaly detection in banks, this guide will provide a short introduction to the topic and walk through the core aspects. But on top of that, for those already familiar, the guide includes some code and practical examples for execution.

WHAT

Anomaly detection is all about finding patterns of interest (outliers, exceptions, peculiarities, etc.) that deviate from expected behavior within dataset(s). Given this definition, it's worth noting that anomaly detection is, therefore, very similar to noise removal and novelty detection. Though patterns detected with anomaly detection are actually of interest, noise detection can be slightly different because the sole purpose of detection is removing those anomalies - or noise - from data.

As an application domain within anomaly detection, fraud detection dominates the banking industry. Fraud detection uses anomaly detection to uncover behavior intended to mislead or misrepresent an actor. Common examples include check and credit card fraud, but fraud detection also occurs in other financial spheres, including insurance.

As with most data science projects, the ultimate end goal or output of anomaly detection is not just an algorithm or working model. Instead, it's about the business outcome and kicking off the necessary processes that follow. For example, it wouldn't be good enough to simply identify bad actors, fraudsters, fraudulent transactions, or network intrusions; the full AI system should also take actions based on these identifications, like escalate cases to a fraud investigation team, block accounts, or alert the proper teams to nefarious actions.

In addition, anomaly detection requires a system that is agile and constantly learning because:

- The very nature of the use cases for anomaly detection means fraudsters or other bad actors are specifically and deliberately trying to produce inputs that don't look like outliers. Adapting to and learning from this reality is critical.
- As spending trends and an increasingly global world continue to change the face of banking, datasets will shift over time, so a system needs to evolve along with its users. Anomalies, by their nature, are unexpected, so it's important that any methods used are adaptive to the underlying data and the natural drift that will occur.
- Financial use cases are extremely time sensitive; businesses and customers can't afford to wait, and any delay in transaction speed or stock trade can have huge consequences. Attempting to predict patterns, thus anticipating anomalies before they occur is risky, but can help guide timely decision-making.

Generating false positives by flagging valid transactions has a serious cost to the user experience and generates risk as well, so it's imperative to find the appropriate middle ground to balance these needs.

It is important to note that, despite the most common use cases being the detection of fraud or system failure, anomalies are not always bad - that is, they don't always have to indicate that something is wrong. Anomaly detection can also be used, for example, to detect or predict slight changes in customer or user behavior that may then result in a shift in selling, development, or marketing strategy, allowing more accurate market predictions and the ability to stay a step ahead of new trends.

THERE ARE THREE BASIC TYPES OF ANOMALIES THAT MAY BE DETECTED:

1. Point anomalies: Point anomalies are simply single, anomalous instances within a single larger dataset. For example, a transaction representing $1 trillion would be a point anomaly, as that would be more money than even the richest conglomerates make in a year. Anomaly detection systems often start by identifying point anomalies, which can be used to detect more subtle contextual or collective anomalies.

2. Contextual (or conditional) anomalies: These are points that are only considered to be anomalous in a certain context. A good example is a transaction again; while $10,000 is considered to be within the range of possible transaction amounts, if it is outside a credit limit then it is clearly anomalous. With spatial data, latitude and longitude are the contexts, while with time-series data, time is the context.

3. Collective anomalies: When multiple related datasets or parts of the same dataset taken together are anomalous with respect to the entire data set (even when individual datasets don't contain anomalies). For example, say that there is data from a credit card making a purchase in the US, but also a dataset showing money being taken out of ATMs in France at the same time. A collective anomaly may occur if no single anomaly happens in any one dataset, but all datasets measuring various components taken together signal an issue.

WHY

Anomaly detection is an approach that can be useful in a variety of use cases. But the underlying, unifying factor is the ability to detect small changes or differences in a system that might otherwise go unnoticed. Uncovering anomalies using machine learning allows humans (or other systems) to take action based on these outliers.

Specifically, a non-exhaustive look at use cases for anomaly detection systems in financial institutions include:

Use Cases	Challenges
Fraud detection (credit cards, insurance, etc.)	Financial anomaly detection is high risk so it must be done truly in real-time so that it can be stopped as soon as it happens. Also, it's more important perhaps than other use cases to be careful with false positives that may disrupt the user experience.
Stock market analysis	Data can (and should) be factored in from a variety of different sources, which creates highly varied data. This can require serious fuzzy matching and merge adjustments to drive accuracy.
Early detection of insider trading	Relatively standard ML models, so long as they possess good feature engineering regarding time-based features, should be sufficient when analyzing tabular transactional data. However, similarly to fraud detection, insider trading has high liability risks associated, so truly real-time detection is critical.

Because of its wide array of applications within the banking industry alone, mastering anomaly detection from a data scientist's perspective - and even for quants, actuaries, and other statistics-minded employees in the banking world - is an incredibly applicable use case.

HOW

If you're familiar with the seven fundamental steps to building a data project, then you already know the basics for how to get started using anomaly detection to benefit your organization. But there are also several particularities to bear in mind when working with anomaly detection:

1. CHOOSE AND UNDERSTAND THE USE CASE

The first step in successful anomaly detection is to really understand what kind of a system the line of business needs and to lay out a framework for the requirements and goals before diving in. These are important preliminary discussions because not all anomaly or fraud detection work is the same; exactly what qualifies as an anomaly and the subsequent processes kicked off by anomaly detection vary vastly by (and even among) use cases.

Notably, the nature of the data, of the problem at hand, and the goals of the project necessarily dictate the techniques employed for anomaly detection.

Even within the finance industry, different projects will have different definitions of what makes a datapoint an anomaly. Very small fluctuations in a system tracking stock prices, for example, could be considered anomalies, while other systems like card charge location could tolerate a much larger range of inputs. So it's not as easy to universally apply a single approach as it is for other types of data projects.

Fraud detection is a particularly good use case for a proof-of-concept in a bank as it does not examine as many social factors as stock trading to make recommendations and does not attempt to drive transactions itself. This is a useful "gateway model" to implement and increase familiarity with AI project processes. However, for organizations already integrating AI-driven systems, a more advanced quantitative stock trading use case might be an appropriate place to try out anomaly detection.

To ensure the success of a fraud detection or other anomaly detection project, it will be critically important to bring together technical profiles carrying out the work (whether data scientists, quants, or actuaries) with the business side (risk team, analysts) to:

- **Define and continually refine what constitutes an anomaly.** It might constantly change, which means continual re-evaluation.
- **Define the goals and parameters for the project overall.** For example, the end goal is probably not just to detect anomalies, but something larger that impacts the business, like blocking fraudulent charges. Having larger goals will allow you to better define the scope of the project and the expected output, and will be critical to get user and stakeholder buy-in. If you're interested in learning more broadly about communicating the importance of ML tech in the banking space, download our white paper communication guide.
- **Determine, once an anomaly is detected, what the system will do next.** For example, send anomalies to another team for further analysis and review.
- **Develop a plan to monitor and evaluate** the success of the system going forward.
- **Identify what anomaly detection frequency (real-time vs. batch) is appropriate** for the use case at hand.

2. GET THE DATA

Having as much data for anomaly detection as possible will allow for more accurate models because one never knows which features might be indicative of an anomaly. Using multiple types and sources of data is what allows banks to move beyond point anomalies into identifying more sophisticated contextual or collective anomalies. In other words, variety is key.

For example, looking at fraud detection, it's possible that transaction data isn't anomalous because the fraudster has stayed within the "normal" range of the actual user's habits. But data from ATM use or account weblogs may reveal anomalies.

Go Further

For the sake of simplicity, this guidebook will walk through a simple fraud detection example where the goal is to predict whether or not a mobile payment transaction is fraudulent.

The theoretical dataset contains several fields of information regarding the transactions themselves, the recipients, and the clients that made the payments. The schema would look like this:

```
transaction_id | transaction_date | transaction_type | transaction_amount |
recipient_id | recipient_country | client_ip_address | client_card_mask |
client_card_country | client_email_address | is_fraudulent
```

In a supervised context, the is_fraudulent column represents the target variable, namely the actual status of the transaction (fraudulent or legit).

3. EXPLORE, CLEAN, AND ENRICH DATA

When doing anomaly detection, this stage is even more important than usual, because often the data contains noise (usually errors, either human or not) which tends to be similar to the actual anomalies. Hence, it is critical to distinguish between the two and remove any problematic data that could produce false positives.

In an ideal world, there would be a sufficient amount of labeled data from which to begin; that is, analysts or data scientists would be able to enrich the datasets from the bank with information on which records represent anomalies and which are normal. If possible, starting with data known to be either anomalous or normal is the preferred way to begin building an anomaly detection system because it will be the simplest path forward, allowing for supervised methods with classification (as opposed to unsupervised anomaly detection methods).

For some of the use cases detailed above, this is likely very attainable. Specifically in fraud detection cases there is a clear mechanism for feedback that create anomalous cases (e.g. customer relationship manager data detailing fraud complaints).

Go Further

In most data science use cases, and especially in anomaly detection, the data preparation portion can have a significant impact on the result of the model. Applying the right processing steps and engineering relevant features are indeed very good ways to better characterize potential outliers.

In the case of fraud detection and given this particular dataset, several useful operations can be performed on the initial dataset to create additional information on each transaction, for example:

- Parse `transaction_date` and extract date features (e.g., day of week, week of year).
- Derive the client's country from `client_ip_address` via IP geolocation.
- Extract the email domain from `client_email_address`

More advanced feature engineering operations will also prove useful here. In the case of fraud detection, it is common to compute rank features by leveraging window functions (performing a calculation across a set of table rows that are somehow related to the current row).

For example, in this fraud detection case, it's important to know how many different IP addresses were used by a given client (identified by his email address) for his purchases. The corresponding PostgreSQL query is:

```
SELECT “transaction_id”,
              SUM[CASE WHEN “first_seen” THEN 1 ELSE 0 END]
              OVER [PARTITION BY “client_email_address” ORDER BY
“transaction_date”] AS “distinct_ip”
FROM[
  SELECT “transaction_id”,
                “client_email_address”,
                “transaction_date”,
                “transaction_date” = MIN[“transac_date”] OVER [PARTITION
BY “client_email_address”, “client_ip_addr”] AS “first_seen” FROM
“transactions_dataset”
```

Other examples of typical rank features for fraud detection include:

- How many times did a client make a transaction from a given country?
- How many times were a given user and a given beneficiary involved in a common transaction?

Finally, it's important to verify that there are no duplicates in the dataset, which would by definition alter the status of the actual outliers.

4. GET PREDICTIVE

There are two primary architectures for building anomaly detection systems:

- Supervised anomaly detection, which you can use if you have a labeled dataset where you know whether or not each datapoint is normal or not.
- Unsupervised anomaly detection, where the dataset is unlabeled (i.e., whether or not each datapoint is an anomaly is unreliable or unknown).

When using a supervised approach, apply a binary classification algorithm. Exactly which algorithm is less important than making sure to take the appropriate measures regarding class imbalance (i.e., the fact that for anomaly detection, it's highly likely that you have far more "normal" cases than anomalous ones).

When using an unsupervised approach, there are two ways of training algorithms:

- Novelty detection: The training set is made exclusively of inliers so that the algorithm learns the concept of "normality" (hence the prefix "one-class" found in some methods). At test time, the data may also contain outliers. This is also referred to as semi-supervised detection.
- Outlier detection: The training set is already polluted by outliers. The assumption is made that the proportion of outliers is small enough, so that novelty detection algorithms can be used. Consequently, those algorithms are expected to be robust enough at training time to ignore the outliers and fit only on the inliers.

Go Further

In a supervised context, the `is_fraudulent` label is available, and the prediction problem can thus be seen as a binary classification task that takes a matrix (X) containing all the features in numeric form and a target vector (y) representing the labels.

However, some extra steps have to be taken because this case deals with a highly class-imbalanced problem (i.e., the outliers are vastly underrepresented):

1. During the cross-validation phase, ensure that both train and test sets have the same proportion of outliers. This can be done using a stratified split - here's an example in Python using the scikit-learn library:

```
from sklearn.model_selection import train_test_split
X_train, X_test,y_train, y_test = train_test_split[X, y,
test_size=0.35, stratify=y, random_state=42]
```

2.An additional option is to use resampling techniques, i.e., only taking a subset of inliers while keeping the full population of outliers.
In a fully unsupervised case, there is no access to the is_fraudulent label. Consequently, it's necessary to resort to special outlier detection algorithms that are trained only on the feature matrix (X) and return an anomaly score for each data point at evaluation time.

Following the example for an unsupervised case, first train the Isolation Forest algorithm on the transaction data:

```
from sklearn.ensemble import IsolationForest
dtr = IsolationForest[].fit[X]
df["y_scored"] =  -dtr.decision_function[X]
```

Then compute the anomaly rank of each transaction and append it to the original dataset (in a pandas DataFrame format). By doing so, it will then be possible to sort the transactions by decreasing anomaly level for direct access to the most suspicious ones (as defined by the model) at the top of the list:

```
df["anomaly_rank"] = df["y_scored"].rank[method="dense", ascending=0]
df.sort_values[by=["anomaly_rank"], ascending=True, inplace=True]
```

In some cases, such a ranking method can be improved by adding a projected damage column, i.e., the amount of money that would be lost in a given transaction if it were to be fraudulent.

5. VISUALIZE

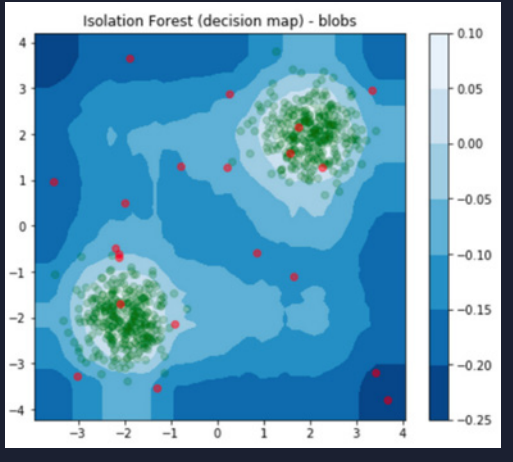
Visualizations are especially useful in the process of building and testing anomaly detection models because sometimes they are the clearest way to see outliers, especially in very large datasets.

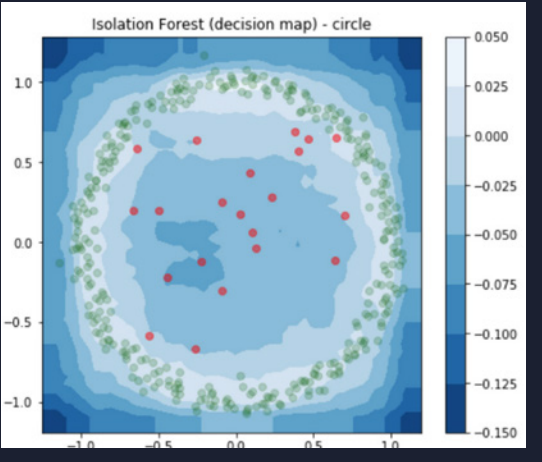
Go Further

In an unsupervised context, it is possible to build a decision map where the anomaly score is computed at each point of a grid that spans over the feature space. In practice, this allows the observation of the zones where inliers are likely to be regrouped according to the model. Any point lying outside of those areas has a higher probability of being an anomaly.

In the following examples, decision maps are built after training an Isolation Forest algorithm on simple two-dimensional datasets with various cluster shapes for the inliers (in green), and a randomly uniform repartition for outliers (in red). Data located in an area with darker shades of blue is more likely to be an anomaly.

Note that in practice, feature reduction needs to be applied to our transaction dataset to essentialize it down to a two-feature (and thus two-axis) mapping if decision map methodology is the goal. While other types of visualization are possible (as this recent Google patent demonstrates) the clarity and easy understanding of this method often leads to increased transparency and understanding, thus improving stakeholder trust in the system.





6. DEPLOY AND ITERATE

To have a real impact with a fraud or anomaly detection system, the model should be scoring data real-time in production. Fraud and anomaly detection in banks are generally extremely time-sensitive, so going to production to make predictions on live data rather than retroactively on test or stale data is more important than ever. This can be especially challenging with sensitive personal financial data, which ideally is available to a limited number of trusted users and systems.

In all financial anomaly detection use cases, data governance and privacy loom large, but they don't need to prevent model implementation. For more information on executing data-privacy compliant models, download the guidebook.

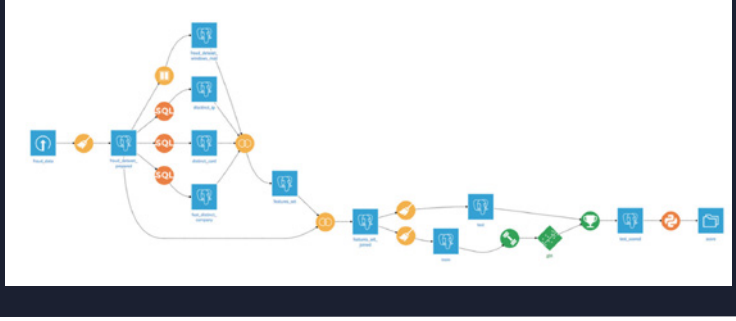
But putting a model in production isn't the end. Iteration and monitoring of fraud - and any other anomaly - detection systems is critical to ensuring that the model continues to learn and be agile enough to continue detecting anomalies even as environments and behaviors change. However, unlike other types of machine learning models, accuracy is not a viable metric for anomaly detection. Since the vast majority of data is not composed of anomalies (i.e., there could be hundreds of thousands of "normal" data points), the system could achieve a very high accuracy but still not actually be accurately identifying anomalies.

Go Further

Instead of accuracy, anomaly detection systems may rely on the following evaluation methods:

- **Recall:** The ratio of correctly detected anomalies to total anomalies.
- **False-positive rate:** The ratio of misclassified anomalies to total records.
- **Receiver Operator Characteristic (ROC) curve:** Tradeoff between detection and false alarm rate.

Regarding iteration, keep in mind that by far the most laborious step when it comes to anomaly detection is feature engineering. Continuing to iterate until false positives/negatives are reduced and the system is effective yet agile is a time consuming yet critical part of the process. What can be helpful here is having a visual representation of the entire process so that iteration is simpler and faster every time, even once the model is in production; for example, here's an overview of our credit card fraud detection example that we worked through in Dataiku DSS, but any visual representation can be effective:



WHAT ARE THE PITFALLS [+ SOLUTIONS]?

Too many false negatives/positives: Evaluating anomaly detection is a particularly fine balance. False negatives can be detrimental, of course, but on the other hand, a system that identifies too many false positives is of almost no use either. In a real-time system, there is no room for a second review of any potential anomalies if the system is producing too much noise. And in many use cases, false positives in anomaly detection could destroy the trust users place in an organization (think of the frustration, for example, if your bank was constantly blocking your funds due to false-positive fraud detection).
Solution: Spending the time upfront on feature engineering to make sure there aren't too many false negatives or positives and continuing to iterate, refine, and make improvements even after the model is in production: both are critical.

Missing or incomplete data: A system that is not robust enough is not the only cause of false positives. Another can be simply a lack of revealing data, which is, unfortunately, a persistent problem, as anomaly detection models are notoriously data-hungry.
Solution: Invest in automating data collection and analysis from a host of conventional (e.g. transaction data) and unconventional (e.g. social media) sources. Because no matter how good your model is, it will not perform well if it doesn't have access to enough data to ground the full picture.

Lack of agility to handle changes in the norm: In today's world, change is the only constant. Human beings change over time, so the idea of normal behavior in the context of anomaly detection will continue to shift. In addition, systems also change over time, but gradual change doesn't always equate to anomalous behavior.
Solution: Any anomaly detection system you build, no matter what the use case, needs to be agile enough to shift with changing norms. Similarly, there should be a plan in place to continually monitor and review the system to ensure it's still performing as expected over time. Systems should also take into account any seasonality or other patterns - for example, a customer at a bank may generally make larger-than-normal purchases around the holidays, but these should not be necessarily flagged as fraudulent.

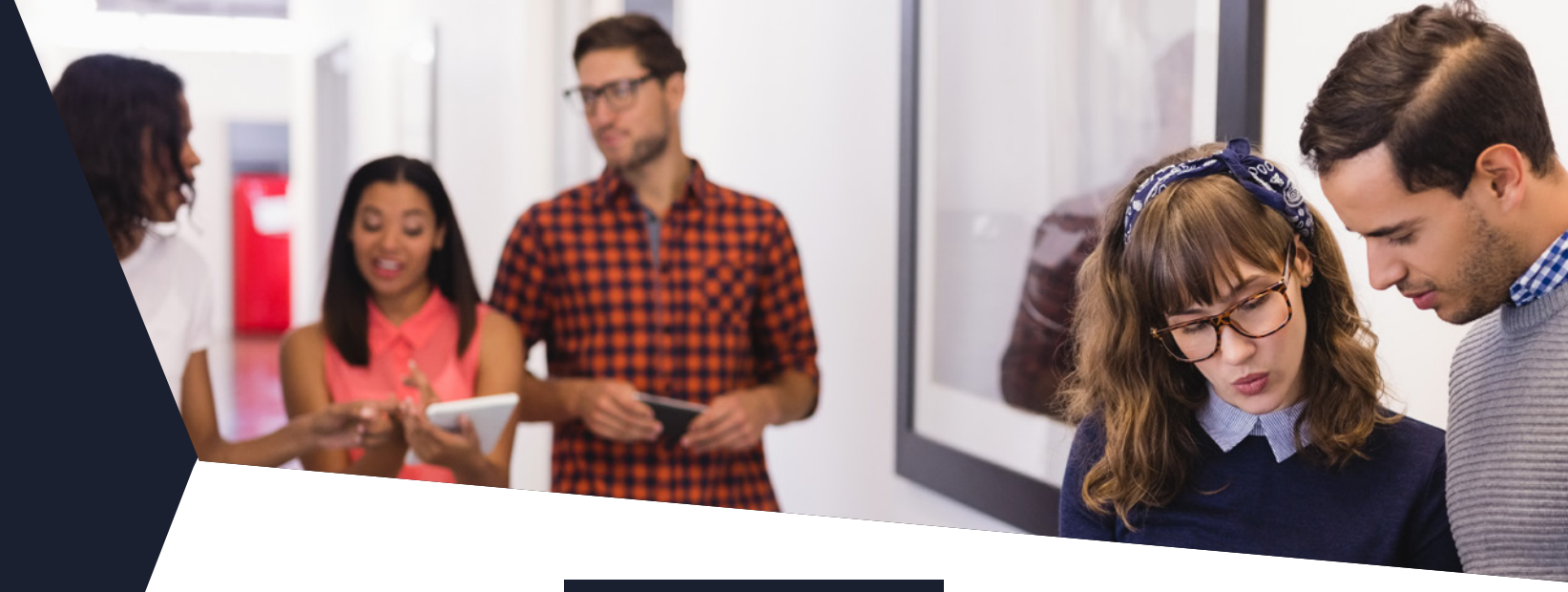
Failure to tie into business objectives: Like many AI systems, it's easy to get disconnected from the business side and develop a system that doesn't take the proper next steps or follow-up based on detected anomalies.
Solution: Fraud or anomaly detection systems in banks particularly need to actually take some action in the end to be effective. Building a system in a vacuum of a data science team won't do any good. Once detected, what is the next-step action? Manual review by an analyst or risk team in the case of fraud, or IT team in the case of system security? The system should take these next steps into account and not just identify anomalies, but kick off the chain.

LOOKING AHEAD

In the years to come as more and more use cases depend on anomaly detection, expect to see an overall streamlining of anomaly detection processes as organizations scale and improve the level of trust they are willing to put in AI-driven systems. This means more and more organizations investing in the right architecture to retrieve data critical for anomaly detection work, the means to process it quickly, and apply models for the biggest impact and business value.

This does not mean that trust is inherent, or that once stakeholders trust one model the rest will naturally follow; transparency and clear visualizations will always be critical to ensure adoption and integration. But it does mean that the seeming impossibility of predicting (and thus preventing) disastrous fraudulent charges (in addition to conserving organizational time and resources) is on the horizon.

However, anomaly detection isn't the only opportunity for AI in banking. There are numerous challenges facing banks that anomaly detection alone is poorly situated to solve.



Use Cases for Data Science, Machine Learning and AI in Banking

Asset Management

- Analyzing Money Flows
- Competitive & Benchmark Analysis

Insurance

- Optimal Pricing & Conversation
- Claims Forecasting
- Claims Triage
- Cash Management Product Risk
- Customer Retention and Churn Prediction

Commercial Banking

- Customer Retention and Churn Prediction
- Recommendation and Upselling
- Reduction of Unjustified Blocked Payments
- Revenue Attribution

Investment Banking

- Evidence-Based Research
- Alternative Dataset Validation
- Trade Failure Prediction

General (i.e., not specific to one sector)

- Credit Risk & Loss Reporting
- Regulatory Reporting Automation
- Fraud Detection & Prevention
- Cyber Security
- AML & KYC
- Improvement and Automation of Processes

- Increased Revenue
- Decreased Risk
- Decreased Costs
- Competitive Advantage

OTHER AREAS OF OPPORTUNITY

AML, KYC, and Wire Transfer Fraud: The value that AI can bring to anti-money laundering (AML), know your customer (KYC) and wire transfer fraud efforts is not only greater accuracy in detecting issues overall, but also - importantly - reducing the number of false positives and reducing overall the number of people doing manual work (particularly in investigating false positives). Because machine learning-based fraud detection systems are more agile and can learn patterns over time, they can continue detecting anomalies even as user behaviors change, which makes them more powerful and prevents costly (not only in terms of staff time, but also in terms of customer frustration and loyalty) manual review of false positives. This is especially important for US banks in the wake of Customer Due Diligence (CDD) Rule.

Trade Failure Prediction: When trades fail, it impacts client satisfaction and has regulatory implications, both of which hurt the overall business. This use case is a good one for predictive analytics because combining data from various sources and applying machine learning models can infer the likelihood of failures with new transactions. This would already be a good first step, but AI can also be introduced by operationalizing these insights and deploying them into a production environment to automatically flag and remediate potential failures.

Credit Risk & Loss Forecasting: McKinsey released a report in late 2018[4] detailing that increasing a credit model's predictive power by just one percent is in reach for most banks. And indeed, the way to do so is through machine learning, which can analyze more data from more sources, faster, to make credit decisions often better than a human analyst. Using a data science, machine learning, or AI platform to do so is even more ideal as it allows for the level of transparency required to ensure that models determining credit risk and loss are interpretable and not a black-box solution.

Cash Management Product Risk: Cash management is one of the most basic, fundamental, day-to-day activities of every bank, and that's exactly what makes it such a prime candidate for revolution by machine learning and AI. Most teams are already using predictive analytics to identify high-risk invoices, duplicate payments, etc. However, this process is extremely cumbersome (thus risky) as data is cleaned in one tool, given to data scientists who develop models in R or Python, and then communicate back to the business. So the key to this use case is optimization between internal stakeholders (like finance, internal operations, and other groups who are involved in payments and receivables) and data experts to increase the speed at which they accurately identify at-risk invoices, duplicate payments, etc., and address the potential issues.

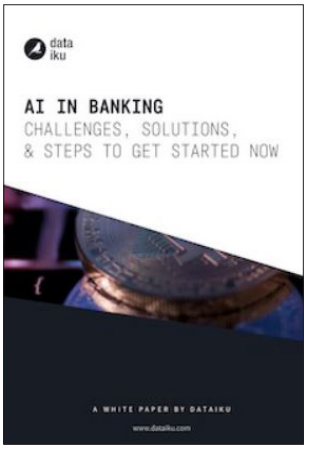
[4] https://www.mckinsey.com/business-functions/digital-mckinsey/our-insights/digital-blog/whats-the-value-of-a-better-model-more-than-you-may-think

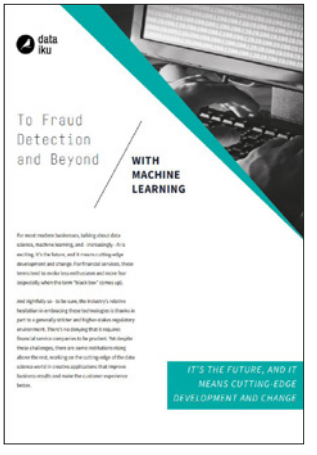
CONCLUSION

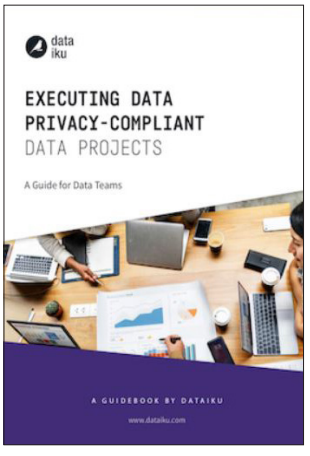
It's clear that minimizing fraud with anomaly detection is important to users and stakeholders in banks, but the question remains as to where to start. Beginning with concrete, incremental goals—such as a 5% increase in flagging fraudulent behavior, or a 10% increase in detection speed—demonstrates clear ROI and helps highlight future room for growth.

Additionally, by prioritizing integration into existing systems, rather than overriding workflows, ML-powered fraud detection is more likely to work within a particular organization, since users and stakeholders will be able to slowly develop trust and cross-check potential mistakes.

There's no way to integrate a complete fraud detection overhaul overnight, but by carefully selecting a use case and encouraging teams' adoption, banks are poised to offer enormous value and peace of mind to their customers and shareholders.







data iku

Your Path to **Enterprise AI**

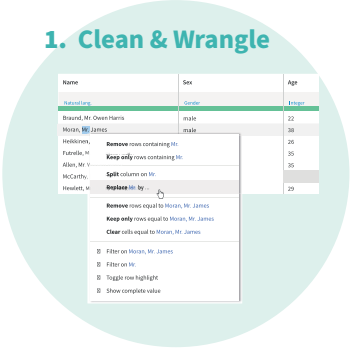
Dataiku is one of the world's leading AI and machine learning platforms, supporting agility in organizations' data efforts via collaborative, elastic, and responsible AI, all at enterprise scale. Hundreds of companies use Dataiku to underpin their essential business operations and ensure they stay relevant in a changing world.

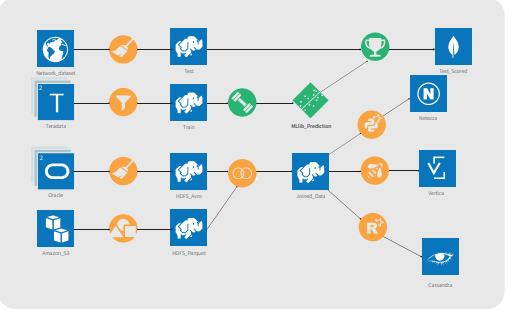
300+
CUSTOMERS

30,000+
ACTIVE USERS

*data scientists, analysts, engineers, & more

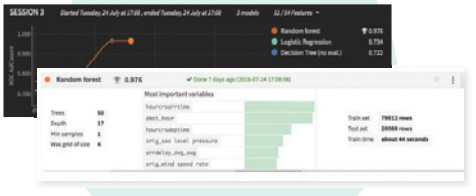
1. Clean & Wrangle



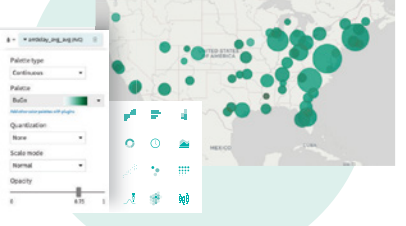


5. Monitor & Adjust

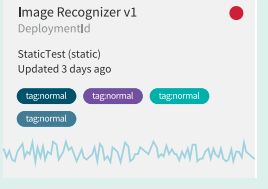
2. Build + Apply Machine Learning



3. Mining & Visualization



4. Deploy to production



GUIDEBOOK

www.dataiku.com