P.E.D

Case Fraude

Daniele Santiago

Modelo de Fraude

TRANSAÇÃO	CLIENTE	•••	FEATURE K	SCORE
transação 1	cliente 1		a	0.5820
transação 2	cliente 2		b	0.6802
transação 3	cliente 3		c	0.2234
...	...		...	...
transação n	cliente n		z	0.9870

Como decidimos quais transações devem ser bloqueadas por fraude?

Métricas de Business

Valor da Transação: $100

Ganhamos $100 * 0.1

Perdemos $100

	Não fraude	Fraude
Não bloquear	Verdadeiros Negativos	Falsos Negativos
Bloquear	Falsos Positivos	Verdadeiros Positivos

= Deixamos de receber $100 * 0.1, mas economizamos o valor da transação

||

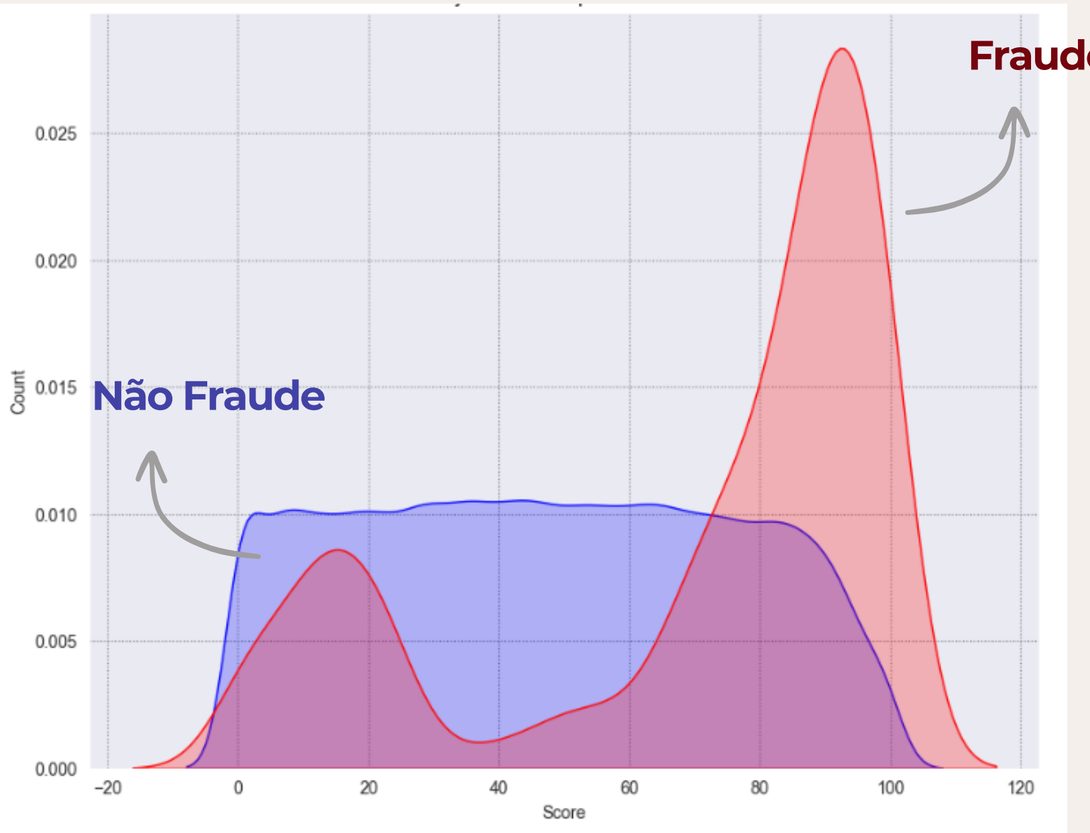
Perdemos * 100 * 0.1

Métricas de Business

Taxa de Fraude = transações fraudulentas aprovadas / transações totais aprovadas

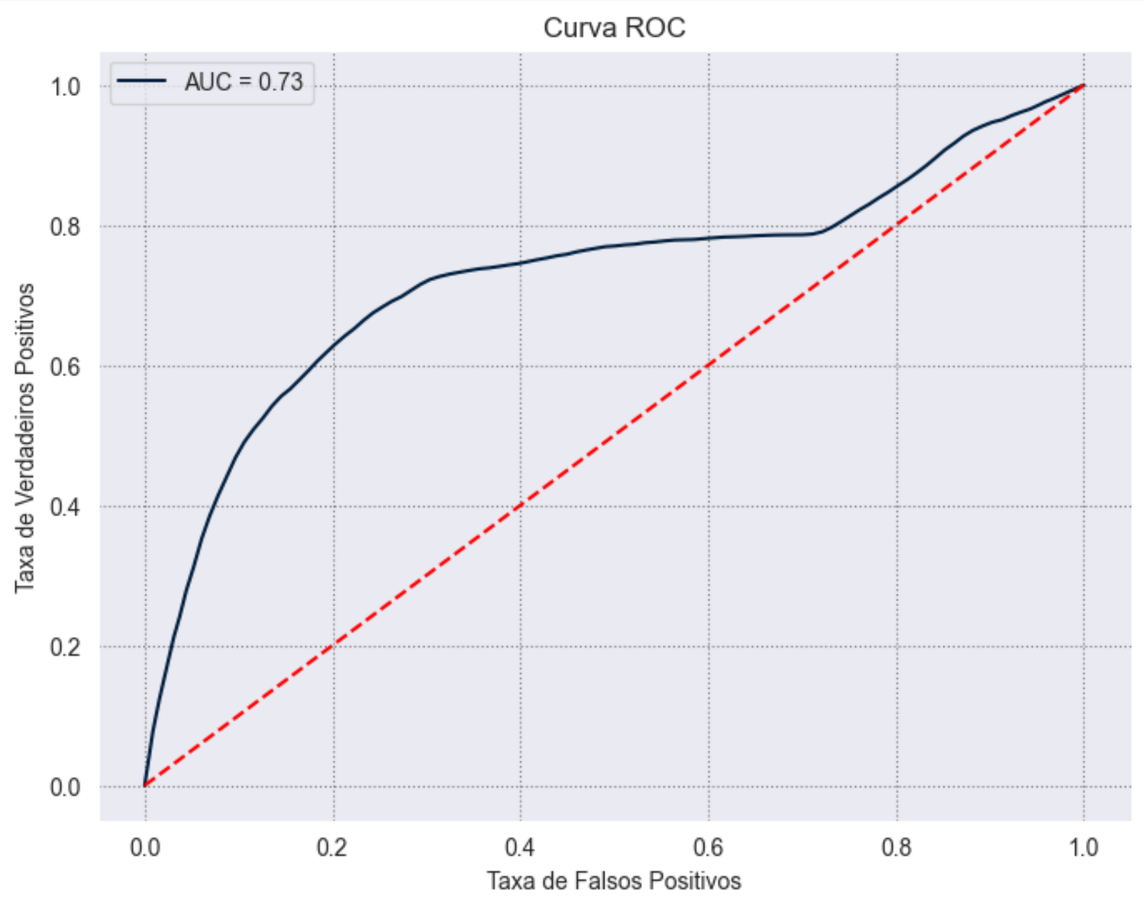
Taxa de aprovação = transações totais aprovadas / transações recebidas

Modelo Atual



Para as transações não fraudulentas, a distribuição é ampla, indicando uma variedade de scores atribuídos. Contudo, para as transações fraudulentas, embora muitas recebam scores altos - um bom indicativo de detecção correta - há uma tendência notável de algumas serem classificadas com scores mais baixos. Esse comportamento sugere que, apesar dos acertos, ainda há espaço para melhorias no modelo, principalmente na identificação precisa de todas as atividades fraudulentas.

Modelo Atual



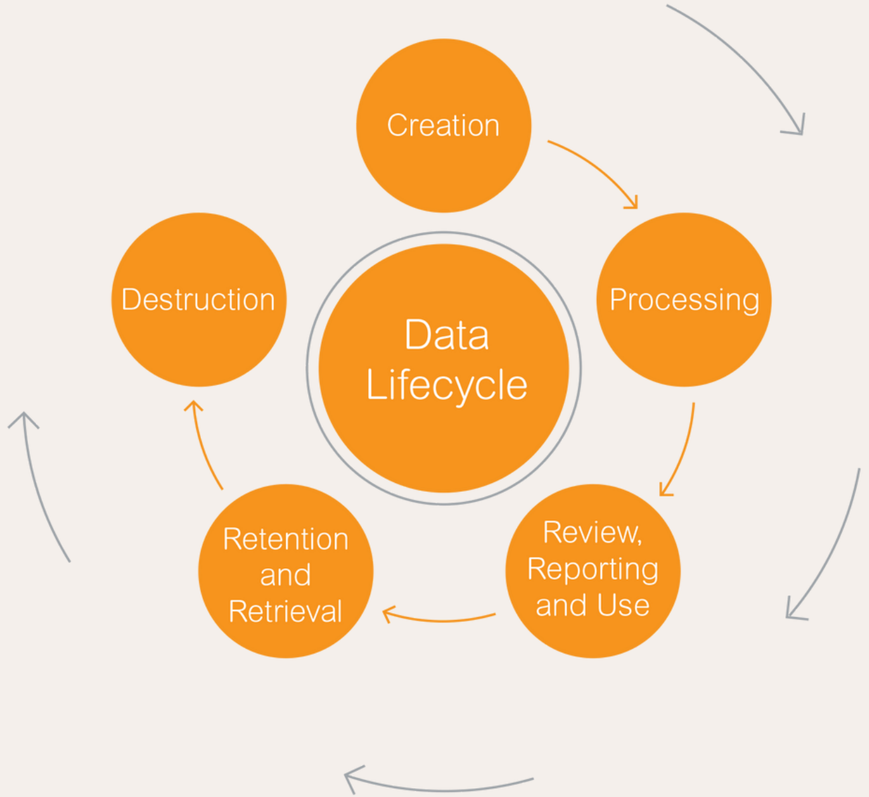
Para cada threshold do modelo anterior, calculou-se a receita e a perda em fraude. Para o dataset de treino, o maior lucro calculado foi para o threshold de 73.

No conjunto de teste, obtivemos:

Lucro: 54976.675
Perda em fraude: 25353.320
Receita: 80329.995
Taxa de fraude: 0.02
Taxa de aprovação: 0.74

Como melhorar o modelo atual?

Em linhas gerais, a criação de um modelo primeiro envolve a coleta, entendimento e limpeza dos dados, treino e teste do modelo e, por fim, ajustes finos.



Preparação e limpeza dos dados

Antes de iniciar a etapa de modelagem, é preciso entender como os dados estão distribuídos e tratá-los caso necessário. Alguns tratamentos feitos e as respectivas colunas nos quais foram aplicados:

Missing: Muitos algoritmos têm dificuldade em lidar adequadamente com valores ausentes, o que pode comprometer a performance. Para lidar com isso, existem várias técnicas, desde simples preenchimentos com média ou mediana até métodos avançados de imputação.

No dataset em questão, a maioria das colunas apresenta uma quantidade mínima de valores ausentes. No entanto, destaca-se a coluna “**entrega_doc_2**” com mais de **72% dos dados não preenchidos**. É crucial notar que, em certos contextos, um valor nulo pode ser informativo. Por exemplo, a ausência de um documento pode ser indicativo de comportamentos suspeitos.

Optei por criar uma **feature booleana** para “entrega_doc_2”, onde "1" indica documento entregue e "0" indica ausência. Adicionalmente, **valores nulos** foram **substituídos por "0"**, denotando a não entrega do documento.

Preparação e limpeza dos dados

Cardinalidade de Features Categóricas: A presença de alta cardinalidade em características categóricas pode afetar negativamente a eficácia do modelo. Features com muitas categorias distintas podem falhar em destacar padrões úteis e, ao invés disso, tornar os dados esparsos, comprometendo a performance do modelo. No nosso conjunto de dados, as colunas que demonstram essa alta cardinalidade incluem:

- **País**: Esta coluna indica a origem geográfica das transações. Dado que mais de 90% das fraudes se concentram no Brasil e na Argentina, vamos segmentar a coluna em três categorias: Brasil, Argentina e Outros.
- **Produto**: Refere-se ao item adquirido. Enquanto certos produtos podem ser mais visados por fraudadores, a subcategoria pertinente já está contida em outra coluna. Por causa da sua elevada cardinalidade, esta coluna será excluída.
- **Categoria do Produto**: Denota a categoria do item comprado. Interessantemente, 80% das fraudes estão associadas a apenas 12% das categorias (considerando um total de 1000 categorias). Para reduzir a cardinalidade, reteremos as 1000 categorias mais proeminentes. As categorias remanescentes serão agrupadas sob o rótulo "outras".

Preparação e limpeza dos dados

Correlações Entre Features: A análise de correlação é vital para garantir que não incluamos no modelo características com significados redundantes. Ao usar a correlação de Pearson, verificou-se que nenhuma das colunas apresentou alta correlação linear.

Distribuição das Features Numéricas: Compreender a natureza das distribuições numéricas é essencial. Notou-se que a maioria das features não segue uma distribuição normal. Considerando que outliers podem diferenciar eficazmente entre fraudadores e não fraudadores, optou-se por não tratá-los. Para valores faltantes, utilizou-se a mediana como método de preenchimento.

Relação entre o Target e as Features: Antes da modelagem, buscou-se entender a relação das colunas com o target através de testes estatísticos. As colunas retidas apresentaram variações nas suas distribuições quando comparadas ao target, indicando que as features se mantêm consistentes independentemente da classificação do target.

Encoding: Para a feature "categoria_produto", devido à sua alta cardinalidade, empregou-se o target encoding com kfold, minimizando possíveis colisões. Para as demais colunas, aplicou-se o método one hot encoder.

Proporção entre as Classes: O conjunto de dados apresenta um desbalanceamento significativo: 95% representam não fraudadores e apenas 5% correspondem a fraudadores.

Modelos Escolhidos

Considerando a distribuição não normal e o desbalanceamento dos dados, e tendo um modelo baseline para referência, exploraremos ensembles eficazes que, com ajustes de parâmetros, gerenciam bem dados desbalanceados.

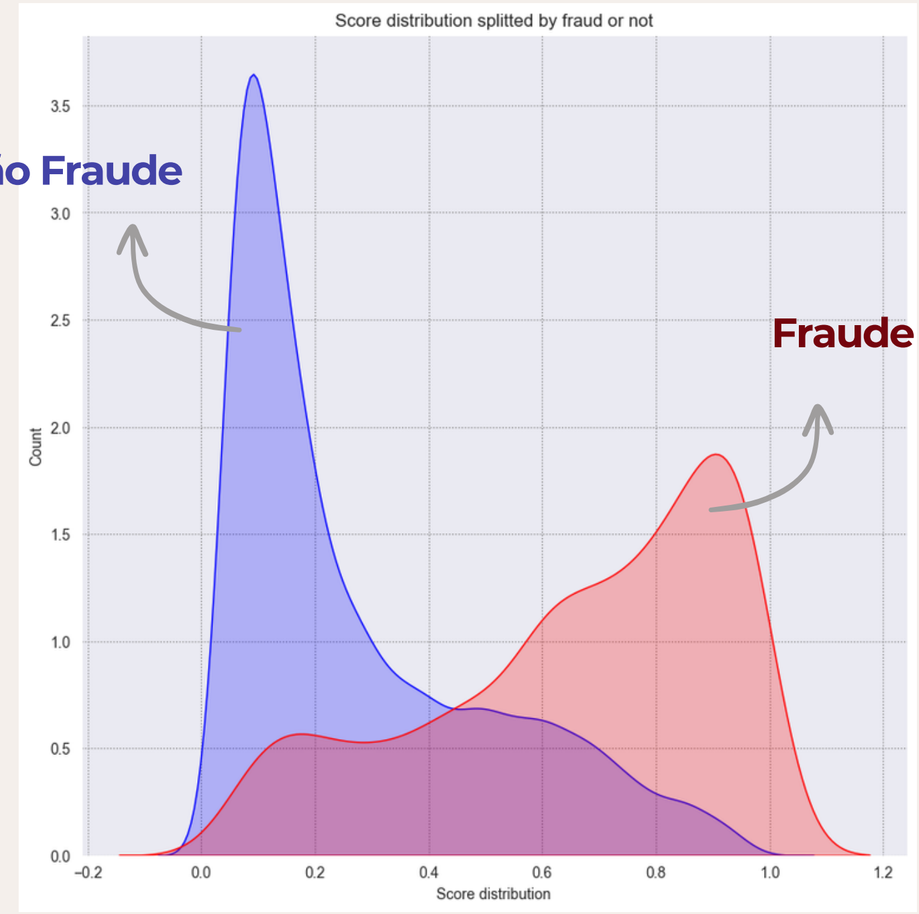
Balanced Random Forest : Partindo da biblioteca imblearn, o BRF é uma adaptação do Random Forest. Enquanto o Random Forest ajusta uma série de classificadores de árvore de decisão em várias subamostras do conjunto de dados, utilizando a média para melhorar a precisão preditiva, o BRC vai além. Ele se dedica a balancear os dados aleatoriamente em cada subamostra, garantindo uma representação justa de todas as classes.

XGBoost : O XGBoost é uma implementação otimizada do algoritmo boosting. O boosting combina os resultados de vários classificadores "fracos", como árvores de decisão, para criar um modelo mais robusto. A cada iteração, o modelo "fraco" treinado considera também os aprendizados do modelo anterior. O XGBoost em particular é conhecido pela sua capacidade de calcular a melhor divisão usando algoritmos pré-classificados e baseados em histograma.

LightGBM: O LightGBM é uma implementação avançada do algoritmo boosting que, assim como o XGBoost, também utiliza árvores de decisão como classificador base. A principal característica do LightGBM é sua técnica de amostragem GOSS (Gradient-based One-Side Sampling), que filtra instâncias de dados para encontrar o melhor valor de divisão.

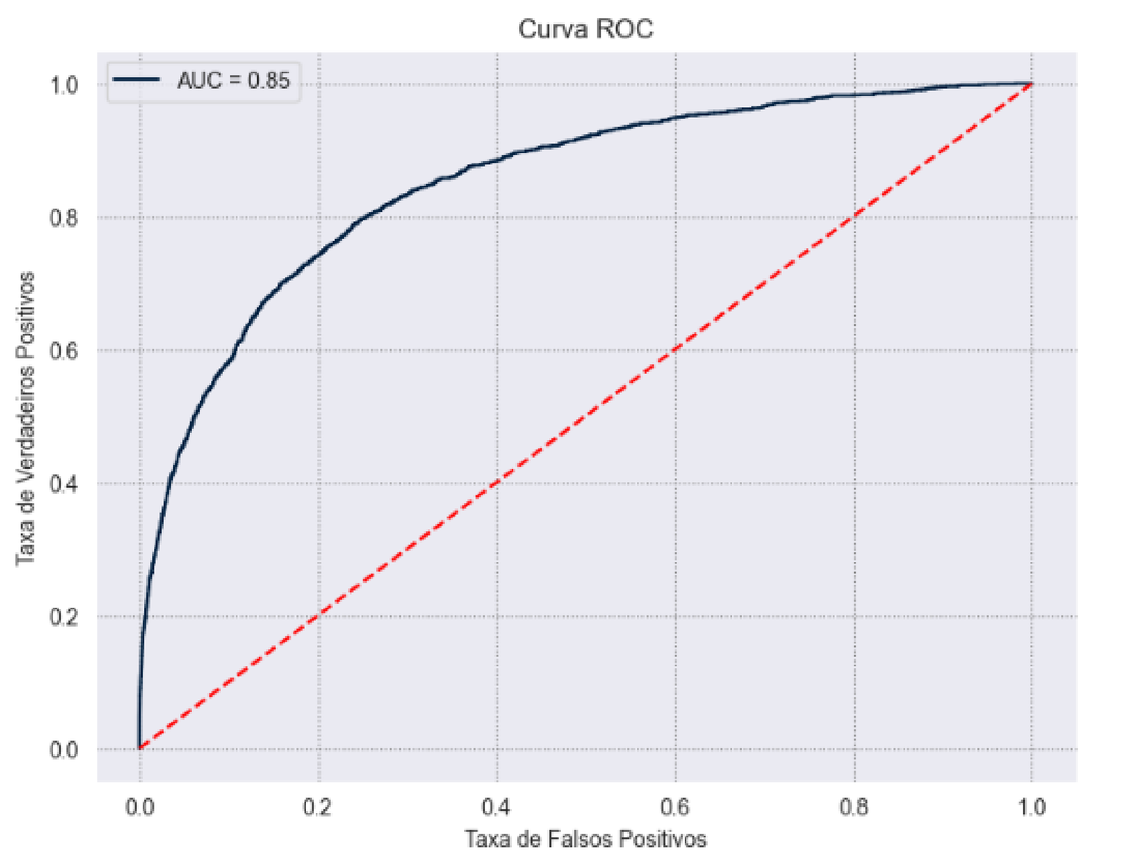
*Dos algoritmos testados, o Balanced RF e o Light GBM tiveram as melhores métricas. O XGBoost também se mostrou competitivo. Dada a eficiência e rapidez no processamento, optei por prosseguir com o **Light GBM**. A métrica escolhida para maximizar o resultado foi a **AUC**, para que se maximize a taxa de verdadeiros positivos (bloquear transações fraudulenta) e minimize a taxa de falso positivos (não bloquear transações boas para o negócio).*

Modelo Treinado



O gráfico ilustra a distribuição de scores do modelo, com clara distinção entre fraudadores (vermelho) e não fraudadores (azul). Contudo, a interseção considerável entre as curvas pode ter implicações financeiras significativas para a empresa.

Modelo Treinado



Para cada threshold do modelo anterior, calculou-se a receita e a perda em fraude. Para o dataset de treino, o maior lucro calculado foi para o threshold de 61.
No conjunto de teste, obtivemos:

Lucro: 68191.768
Perda em fraude: 27007.200
Receita: 95198.968
Taxa de fraude: 0.02
Taxa de aprovação: 0.85

Modelo Atual vs Modelo Treinado

No conjunto de teste

Modelo Atual:
Lucro: $54.976
Perda em fraude: $25.353
Receita: $80.329

Modelo Treinado:
Lucro: $68.191
Perda em fraude: $27.007
Receita: $95.198

68%
Lucro/Receita

72%
Lucro/Receita

Após 1 mês*

$ 197.3 M
Lucro/Mês

$ 208.9 M
Lucro/Mês

$ 11.6 M

Aumento de lucro/mês, além de aumento de AUC e taxa de aprovação.

** Considerando 20 M de pagamentos por trimestre e que o valor médio de transação é de $ 43.5*

Obrigada!

Quer resolver esse case? Ele está disponível no P.E.D
(Preparatório para Entrevistas em Dados).
Acesse em: www.renatabiaggi.com/ped