ISAG - Instituto Superior Administração e Gestão

Pós-Graduação em Data Science e Business Intelligence

Projeto Final

Modelo de Previsão de Pagamentos em Atraso:

Identificação Proativa de Clientes em Risco

Porto, maio de 2024

Realizado por: Sara Helena M. Oliveira Silva

Orientado por: Ivo Nogueira e Rita Faria

Índice

1.) Compreensão do Negócio *(Business Understanding)*

O atraso no pagamento de faturas é um desafio comum e transversal a muitas empresas que, por consequência, pode causar problemas sérios de liquidez e impactar negativamente o desempenho financeiro das organizações. Perante este cenário, a capacidade de identificar proactivamente os clientes mais propensos a atrasarem o pagamento das suas faturas torna-se num desafio que, em caso de correta previsão, torna-se fundamental para a saúde financeira de uma empresa e para um aumento na satisfação dos seus clientes.

1.1) Motivação do Projeto

Tendo em consideração este enquadramento, este projeto foi motivado pela vontade de desenvolver um modelo de previsão de pagamentos em atraso que ajude as empresas a identificar antecipadamente os clientes que apresentam maior probabilidade de não efetuarem o pagamento das suas faturas dentro do prazo estabelecido.

Considero que este modelo, que será desenvolvido com técnicas de análise de dados e machine learning, tem um enorme potencial de se tornar uma ferramenta imprescindível e crucial para colmatar os riscos associados ao não pagamento de faturas para além de ser uma ferramenta que pode ajudar no trabalho aborrecido e cansativo das equipas que fazem gestão de cobranças nas organizações.

A metodologia adotada para o desenvolvimento deste projeto segue as orientações do processo *CRISP-DM* (*Cross-Industry Standard Process for Data Mining*) que engloba etapas de compreensão do negócio, compreensão e preparação dos dados, modelação e avaliação do modelo bem como proposta de como implementar o modelo em produção nas empresas.

1.2) Objetivos

O principal objetivo deste projeto é identificar clientes que são propensos a não efetuarem ou a atrasarem o pagamento das faturas dos produtos ou serviços que usufruíram. Desta maneira, as empresas podem tomar medidas preventivas e proactivamente ajudar os clientes com maior probabilidade de atrasarem ou não efetuarem o pagamento.

Todas as etapas necessárias para desenvolver este projeto serão implementadas utilizando a linguagem de programação *Python* e bibliotecas como Pandas, Numpy, Scikit-Learn, Scipy, seaborn, matplotlib entre outras

1.3) Desafios, Restrições e Riscos Associados

Implementar um modelo de previsão de pagamentos apresenta desafios, restrições e existem riscos associados. Em algumas empresas, a falta de histórico de pagamentos: ou porque

estão a iniciar a empresa há pouco tempo e ainda não têm um volume de dados suficiente ou porque têm sistemas desatualizados ou ainda porque ainda têm um sistema de cobrança manual. Além disso, a variabilidade nos prazos de pagamento dos clientes, influenciada por fatores individuais ou sazonais, torna difícil prever quando os pagamentos serão recebidos.

E por fim, a integração do modelo nos sistemas de faturação e cobrança das empresas pode ser complexa e cara.

Relativamente aos custos associados, destaco a contínua manutenção e atualização do modelo que exigirá custos tanto financeiros como de pessoal qualificado. E, o risco mais previsível está associado ao facto de existir uma margem de erro associada à previsão do modelo o que pode impactar os fluxos de caixa e levar a decisões financeiras erradas.

1.4) Benchmarking

No mercado já existem alguns softwares e ferramentas que oferecem previsão de pagamentos em atraso. Os sistemas de gestão financeira integrada oferecem módulos de previsão tendo em conta dados históricos. Por outro lado, os sistemas de análise de dados e *business inteligence* como *Tableau* e *Power BI*, entre outros, já têm também disponíveis recursos avançados de análise para previsão de pagamentos em atraso. Para além disso, existe também no mercado softwares de gestão de risco financeiro que foram desenvolvidos tanto para gerir como para prever risco financeiro, o que pode incluir previsão de pagamentos em atraso.

Apesar disto, o desenvolvimento deste projeto destaca-se em relação às alternativas já existentes no mercado uma vez que se consegue adaptar às necessidades específicas de cada empresa, sendo assim uma opção totalmente personalizada e flexível. Para além disso, o desenvolvimento deste modelo oferece um excelente custo benefício, apesar de ser necessário a constante atualização dos dados e manutenção que exigirá custos tanto financeiros como de pessoal qualificado, é uma solução mais em conta em comparação com as restantes ferramentas já disponíveis no mercado.

1.5) Identificação do melhor método de desenvolvimento

Tendo em consideração que o objetivo deste projeto é prever os pagamentos em atraso (*target*), é importante destacar que se trata do desenvolvimento de um modelo supervisionado.

Durante o processo de desenvolvimento, foram exploradas diversas abordagens alinhadas com as práticas atuais na área de ciência de dados e análise preditiva, nomeadamente regressão linear, regressão *Ridge* e *Lasso*, modelos de árvore de decisão bem como *Random Forests* e *Gradient Boosting.*

No decorrer deste relatório, irei detalhar e apresentar os resultados de cada abordagem testada, contudo, é importante ressaltar que o modelo que se destacou e apresentou melhores resultados foi o *Gradient Boosting.*

Desta maneira, o *Gradient Boosting Regressor*, algoritmo disponível na biblioteca do *Scikit-Learn,* foi o que apresentou melhores resultados. Esta observação não foi surpreendente, uma vez

que este algoritmo é capaz de lidar com dados complexos e não lineares, comuns em problemas financeiros. Para além disso, é uma ótima opção, uma vez que é conhecido pela sua flexibilidade e precisão, já que é capaz de capturar correlações complexas e fazer previsões confiáveis.

2.) Compreensão dos dados (Data Understanding)

Nesta primeira fase do projeto, compreender os dados disponíveis é muito importante uma vez que servirão de base para o desenvolvimento do modelo proposto.

Durante esta etapa de compreensão de dados, é crucial estudar a estrutura do conjunto de dados (*dataset)* escolhido: que variáveis estão disponíveis, de que tipo são, o seu significado, bem como proceder ao cálculo de dados estatístico descritivos básicos e entender a distribuição dos dados.

Nesta etapa irei igualmente fazer deteção de valores omissos (*missing values*), *outliers* bem como analisar possíveis tendências e relações entre as variáveis.

2.1) Identificação das fontes de dados

O *dataset* selecionado foi extraído da plataforma *Kaggle*. Trata-se de uma plataforma amplamente reconhecida por ter à disposição conjuntos de dados com elevada qualidade para diversos projetos na área de *data science* e *business inteligence*.

Relativamente ao tipo de ficheiro, o dataset está no formato CSV *(Comma-separated values)*, um formato de arquivo de texto muito utilizado para armazenar dados tabulares.

2.2) Descrição dos dados

Este *dataset* contém informações sobre o comportamento de clientes relativamente ao pagamento da fatura de um serviço que adquiriu. Cada entrada (*data point)* representa uma fatura emitida com datas de emissão compreendidas entre 01-01-2023 e 09-09-2023, identificada pelo número da fatura *(invoiceNumber)* e respetivo cliente através do seu *customerID.*

O *dataset* é composto por 12 variáveis e 2466 visualizações *(data points).*

As variáveis que compõem o *dataset*, bem como o seu tipo e o seu significado podem ser consultados na tabela abaixo.

Nome da variável	**Tipo**	**Significado**
countryCode	object	Código do país do cliente.
customerID	object	Identificação Única do Cliente.
PaperlessDate	datetime	Data em que cliente consentiu fatura digital.
invoiceNumber	object	Número da fatura.
InvoiceDate	datetime	Data da fatura.
DueDate	datetime	Data Vencimento da fatura.
InvoiceAmount	float64	Valor total da fatura.

Disputed	object	Indica se fatura foi contestada pelo cliente. Valores Possíveis: ('Yes' (fatura contestada) ou 'No' (fatura não contestada)
SettledDate	datetime	Data em que fatura foi paga ou contestação resolvida.
PaperlessBill	object	Indica se cliente recebe fatura digital. Valores Possíveis: ('Paper' (fatura em papel) ou 'Electronic' (fatura eletrónica).
DaysToSettle	int64	Número de dias que cliente demorou para efetuar pagamento.
DaysLate	int64	Número de dias que fatura ficou em atraso em relação à data de vencimento.

Estes dados servirão de base para desenvolver este projeto. Para além disso, são importantes para passarmos à fase da análise exploratória dos dados onde se vai analisar a distribuição das variáveis, o comportamento de pagamento dos clientes, perceber se existem padrões de atraso que possam indicar um maior risco de atraso nos pagamentos, entender se existe diferença entre pagamento com fatura eletrónica ou em papel entre outras análises.

2.3) Exploratory Data Analysis

2.3.1) Missing Values: Não existem nem valores omissos nem observações idênticas ou duplicadas.

2.3.2) *Outliers:* Existem 259 *outliers* no *dataset* (*boxplot* das variáveis onde se pode verificar a presença de *outliers* nos anexos). Estes *outliers* foram detetados recorrendo ao método do intervalo interquartil (IQR) que permitiu identificar observações que se encontram significativamente distantes dos quartis das respetivas variáveis.

Embora tenha sido identificado presença de *outliers* no *dataset,* devido ao contexto específico do modelo que vou desenvolver, decidi manter estes valores atípicos uma vez que considero importante existir uma variedade de situações que podem influenciar o comportamento dos clientes. Desta forma, acredito que o modelo vai ser capaz de lidar com estes casos extremos e prever o atraso dos pagamentos numa grande variedade de cenários.

Apesar disto, considero fazer testes com o *dataset* com e sem os *outliers* para perceber se a performance melhora com a exclusão destes.

2.3.3) Estatísticas Descritivas: A análise das estatísticas descritivas é fundamental para entender a natureza e distribuição dos dados. Desta forma, nas linhas seguintes, faremos uma análise das mesmas para cada variável, quer numérica, quer categórica.

2.3.3.1) Variáveis numéricas: Nas estatísticas descritivas das variáveis numéricas do *dataset*, consegue-se extrair as seguintes informações:

• 'InvoiceAmount': o valor médio das faturas é de aproximadamente 59,90 unidades monetárias com um desvio padrão de 20,44 unidades monetárias. Este desvio padrão indica alguma dispersão dos valores em relação à média. O valor menor de uma fatura é de 5,26 unidades

monetárias e o maior valor é de 128,28 unidades monetárias. Analisando os valores das faturas nos quartis 25% e 75% podemos concluir que os valores das faturas estão bem distribuídos.

• DaysToSettle: o tempo médio para liquidar uma fatura é de aproximadamente 26 dias com um desvio padrão de 12 dias. O menor tempo registado no *dataset* para liquidar a fatura é de 0 dias o que indica que algumas faturas foram liquidadas no mesmo dia em que foram faturadas. Por outro lado, a fatura que demorou mais tempo a ser liquidada, demorou 75 dias. De realçar que a mediana do tempo de liquidação são 26 dias, o que sugere uma distribuição simétrica dos dados em torno da mediana.

• 'DaysLate': A média de dias em atraso é de aproximadamente 3 dias com um desvio padrão de 6 dias. A maioria das faturas não entra em atraso, ou seja, são liquidadas dentro do tempo previsto, por outro lado, o maior atraso registado é de 45 dias o que sugere presença de outliers conforme já tinha informado nas alíneas anteriores.
Nos anexos, pode-se encontrar suportes gráficos que corroboram com esta análise (figura 2 e 3).

2.3.3.2) Variáveis categóricas: Ao analisar as frequências das variáveis categóricas, pode-se concluir o seguinte:

• 'Disputed': Existem 1905 registos onde esta variável assume o valor 0 (fatura não contestada) e 561 em que assume o valor 1 (fatura contestada). Desta forma, pode-se concluir que a maioria das transações não é contestada.
• 'Country_Califórnia, US': 387 registos são da Califórnia.
• 'Country_Georgia, US': 506 registos são da Califórnia.
• 'Country_Germany, US': 616 registos são da Alemanha.
• 'Country_Kansas, US': 396 registos são do Kansas.
• 'PaperlessBill_dummy, US': Existem 1263 registos com valor 1 (faturas eletrónicas) e 1203 registos com valor 0 (faturas em papel). Isso sugere uma distribuição quase uniforme entre as duas opções.

2.4) Análise de Correlação

A análise de correlação desempenha um papel importante na compreensão das relações entre as variáveis num conjunto de dados. Desta maneira, a última etapa do EDA deste projeto passa por compreender como as variáveis se relacionam entre si o que nos permite perceber a direção e a força dessas relações. Ao aplicarmos um *threshold* de 0,95 na análise de correlação, estamos a estabelecer um critério para identificar correlações significativas entre as variáveis. Este limite indica que se tem interesse em identificar correlações significativas entre as variáveis já que, quando isto acontece, quando as variáveis estão altamente relacionadas entre si, significa que uma variável pode ser prevista a partir da outra. Ao se manter estas correlações tão altas no *dataset*,

pode-se causar problemas de multicolinearidade levando a estimativas imprecisas e instáveis nos modelos.

Tendo isto em consideração, para fazer a análise de correlação entre as variáveis deste conjunto de dados, defini um *threshold* de 0,95. Com esta análise concluída, é possível perceber que as variáveis *'SettledDate_year*' e *'DueDate_year'*, têm um coeficiente de correlação de 0,96. Por este motivo, decidi excluir a variável '*SettledDate_year'* do *dataset.*

3.) Preparação dos Dados *(Data Preparation)*

A próxima etapa deste projeto passa pela preparação dos dados. Esta etapa assume um papel muito importante em modelos preditivos uma vez que é nesta fase que se garante que os dados têm qualidade, estão estruturados de forma adequada e são capazes de generalizar para diversos cenários. Durante a fase do EDA, já foram tomadas algumas medidas de preparação de dados para garantir a qualidade dos mesmos e facilitar a análise, nomeadamente a deteção de *outliers* e *data points* duplicados, identificação e remoção de variáveis redundantes e ainda a transformação de variáveis categóricas em variáveis *dummy*.

A deteção de outliers e data points duplicados foram importantes para garantir a precisão das estatísticas descritivas e evitar *bias.*

A eliminação de variáveis redundantes simplificou o dataset o que facilitou a interpretação dos resultados.

Por fim, a transformação das variáveis categóricas em variáveis *dummy* permitiu que as mesmas fossem consideradas na análise da estatística descritiva das variáveis categóricas que não era possível se não fossem transformadas em *dummies.*

3.1) Tratamento de variáveis categóricas

Apesar deste passo já ter sido realizado na etapa anterior, não foi explicado o procedimento. Desta forma, as próximas linhas são dedicadas à explicação deste passo importante.

No dataset original, tão como foi dito no início deste relatório tinha variáveis categóricas nomeadamente o '*CountryCode'*, '*Disputed'* e '*PaperlessBill'*.

A variável '*CountryCode'* que era composta pelo código do país a que o cliente vivia foi transformada numa variável *dummy*, ou seja, foi transformada e, variáveis binárias para cada país presente nos dados. No que diz respeito à variável '*PaperlessBill'*, foi feita mesma transformação. Já na variável '*Disputed'*, fez-se apenas uma substituição do '*Yes'* que corresponde a uma fatura contestada por '1' e do '*No'* para '0' que corresponde a fatura não contestada.

Ao ter-se realizado o tratamento adequado das variáveis categóricas, garantiu-se que podem contribuir de forma significativa para o modelo de previsão que se vai desenvolver já que contêm informações importantes e pertinentes que, com certeza, irão melhorar a qualidade das previsões.

3.2) Feature enginnering

Tal como todas as outras etapas, esta é muito importante pois permite-nos explorar o máximo potencial dos dados que já temos criando variáveis novas.

No contexto deste projeto, foi criada uma variável '*IsLate*' que permite identificar quais clientes estão atrasados. Por limitação de tempo, não foi considerada esta abordagem, contudo, numa próxima *release* deste modelo, podia ser considerada a hipótese de considerar esta variável como target e desenvolver um modelo de classificação onde seria possível prever que clientes teriam maior ou menor probabilidade de ser atrasados a efetuar o pagamento das faturas.

3.3) Normalização

A normalização é essencial para garantir que todas as variáveis têm uma escala comparável. Neste projeto, utilizei o *StandardScaler,* uma técnica comum de normalização que transforma as variáveis de forma que todas tenham uma média 0 e desvio padrão 1. Faz sentido aplicar esta técnica na maioria dos projetos de *data science* contudo, é particularmente útil quando existem variáveis com magnitudes diferentes, o que é muito comum em dados reais.

Tomando como exemplo as variáveis do dataset utilizado para o desenvolvimento deste modelo: as variáveis '*InvoiceAmount*' e *'DaysToSettle'* têm escalas completamente diferentes já que uma representa o valor em dinheiro e outra o tempo em dias. Se não fosse aplicada a normalização, a variável '*InvoiceAmount*' sendo a variável com escala maior, iria dominar na contribuição da previsão do modelo o que iria enviesar os resultados do mesmo.

Desta forma, aplicando esta técnica de normalização, garanti que todos as variáveis deste projeto terão o mesmo impacto no modelo, garantindo uma distribuição mais equilibrada e justa dos pesos das *features.*

4.) Modelação *(Modelling)*

Nesta fase, já se encontram reunidas todas as condições para avançar com a construção de modelos que ajudaram a cumprir o objetivo proposto, ou seja, prever os pagamentos em atraso. O objetivo é utilizar os dados que foram entendidos e preparados nas etapas anteriores para desenvolver modelos capazes de prever comportamentos futuros dos clientes tendo em consideração os dados históricos disponíveis.

Desta forma, irei explorar diversos algoritmos e mediante os resultados preliminares, fazer otimização dos hiperparâmetros dos mesmos para chegar ao melhor resultado possível, aumentando a precisão e diminuindo o erro gerado. De uma forma lógica, comecei por abordar modelos básicos e evoluir para modelos mais complexos.

4.1) **Explicar modelo**

Nesta etapa, vou desenvolver um modelo de regressão para prever o target, que neste caso será a variável '*DaysLate*' que representa quantos dias o cliente se vai atrasar a efetuar o pagamento. Visto estarmos a falar de uma variável contínua, aplicar algoritmos de regressão é uma escolha adequada para este projeto.

Para avaliar o desempenho do modelo irei utilizar o RMSE *(root mean squared error)* que fornece uma medida de dispersão de erros com a mesma unidade do target, sendo desta forma, facilmente interpretável. E utilizarei também o R^2 que expressa a quantidade de variância dos dados explicado pelo modelo. Para além disto, leva em consideração o número de variáveis do modelo e o número total de data points penalizando os modelos com muitas variáveis. O R^2 ajustado está varia entre 0 e 1 sendo que, quanto mais próximo de 1 estiver, melhor é a performance do modelo.

4.2) Divisão dos dados entre treino e teste

Antes de avançar para a modelação, os dados foram divididos em conjuntos de treino e teste utilizando a função '*train_test_split*' da biblioteca *scikit-learn.* Esta divisão é fundamental para treinar e avaliar os vários modelos que se vão testar.

Para esta divisão, considerei uma proporção de 80% para o conjunto de treino e 20% para o conjunto de teste o que significa que 80% dos dados serão usados para treinar o modelo e os restantes 20% ficam guardados para avaliar o desempenho do modelo. Com esta divisão feita, reúne-se todas as condições necessárias para avançarmos para a fase de modelação.

4.3) Treino dos modelos

Antes de avançar com a informação acerca dos modelos utilizados, vale a pena relembrar nesta etapa do relatório que, nesta *release* do projeto, estamos focados no desenvolvimento de um algoritmo de regressão. O objetivo é prever uma variável contínua, nomeadamente os dias de atraso nos pagamentos ('*DaysLate*').

Desta maneira, o *target* (y) definido será a variável '*DaysLate*' que representa o número de dias que um pagamento está atrasado em relação à data de vencimento da fatura, é esta variável que se pretende prever. As restantes variáveis disponíveis (X) são as *features* que serão utilizadas para prever o y definido.

Ao longo desta etapa, foram explorados diferentes algoritmos de regressão para perceber qual melhor se ajusta aos dados disponíveis e nos fornece as previsões mais precisas.

Existem uma margem grande de possibilidade de otimização do projeto em que, em próximas releases se pode considerar expandir esta análise testando algoritmos de classificação cujo objetivo seria ajudar na classificação de clientes com baixo/médio/elevado risco de atraso nos pagamentos.

Julgo ser pertinente, ainda nesta etapa, mencionar que métricas de avaliação de performance do modelo decidi utilizar.

- MSE (*Mean Squared Error*): é uma medida que calcula a média dos quadrados dos erros entre os valores reais e os valores previstos pelo modelo. Visto se tratar de uma métrica que calcula o erro, o objetivo é ter o menor valor possível.
- RMSE *(Root Mean Squared Error*): tal como o nome indica, é a raiz quadrada do MSE e fornece uma medida de dispersão dos erros. Apresenta a mesma unidade que a variável target considerada o que se torna de mais fácil interpretação. Tal como MSE, visto se tratar de uma métrica que calcula o erro, o objetivo é ter o menor valor possível.
- R^2: é uma medida de proporção da variabilidade dos dados que é explicada pelo modelo, contudo. Esta métrica varia entre 0 e 1, onde 0 indica que o modelo não explica nenhuma variabilidade nos dados e 1 indica que o modelo explica toda a variabilidade. Desta forma, quanto mais próximo de 1 estiver, melhor.

São métricas comuns utilizadas para avaliar a performance do modelo, o objetivo no decorrer das tentativas para encontrar o melhor modelo é minimizar o MSE e o RMSE e aumentar o R^2.

Finalmente, irei então abordar os modelos utilizados em cada tentativa e os resultados obtidos:

4.1) 1.ª Tentativa - Regressão Linear: Na primeira tentativa, optei por utilizar um modelo de regressão linear simples da biblioteca *scikit-learn*. Os resultados obtidos foram os seguintes:

MSE	11,1788
RMSE	3,3435
R^2	0,7056

Estes resultados indicam que o modelo conseguiu capturar 70,56% da variabilidade observada. O RMSE sendo 3,3435 indica que, em média, as previsões estão a 3 valores de distância dos valores reais.

4.2) 2.ª Tentativa - Regressão de *Ridge* e *Lasso*: A segunda tentativa concentrou-se numa extensão da regressão linear. A regressão de Ridge e Lasso são métodos de regressão linear que penalizam coeficientes grandes e no caso da Lasso, essa penalização é mais pesada uma vez que leva em consideração a seleção automática de variáveis transformando-as com coeficiente igual a 0. Os resultados obtidos foram os seguintes:

	Ridge	Lasso
MSE	11,122	10,959
RMSE	3,335	3,310
R^2	0,707	0,711

Ambos os modelos mostram resultados semelhante, o que sugere não haver diferença significativa entre entres. Ambos conseguiram capturar cerca de 71% da variabilidade observada e

têm igualmente ambos um RMSE perto de 3 o que significa que em média, as previsões estão a 3 valores de distância dos valores reais observados.

4.3) 3.ª Tentativa - Árvore de Decisão: A 3.ª experiência focou-se na construção de uma árvore de decisão. Este algoritmo divide um conjunto de dados em subconjuntos menores com base em características que levam a outputs mais homogéneos. Os resultados obtidos com esta abordagem foram os seguintes:

MSE	0,356
RMSE	0,597
R^2	0,991

Os resultados desta tentativa são bastante melhores do que as abordagens tentadas anteriormente: o MSE e RMSE são bastante baixos o que indica que o modelo está a fazer previsões precisas. O R^2 próximo a 0,99 sugere que o modelo está a ajustar muito bem os dados de treino. Apesar destes resultados promissores, existe a preocupação do modelo estar a fazer *overfitting* visto o MSE ser muito baixo e o R^2 ser muito alto.

4.4) 4.ª Tentativa – *Random Forest:* a 4.ª tentativa concentrou-se numa extensão da árvore de decisão. A grande diferença entre estes dois métodos é que as árvores de decisão constroem uma única árvore usando todo o conjunto de dados e a *Random Forest* constrói várias árvores independentes, cada uma delas treinada numa amostra aleatória do conjunto de dados. As previsões finais são obtidas através da média das previsões individuais de cada árvore. Os resultados foram os seguintes:

MSE	0,416
RMSE	0,644
R^2	0,99

Os resultados obtidos foram ligeiramente inferiores aos da árvore de decisão o que pode indicar que o problema de overfitting que se tinha identificado no método anterior, nesta abordagem já não acontece. Existe uma margem grande para se melhorar este modelo otimizando os seus hiperparâmetros, contudo, antes de procedermos com esta sugestão, vou experimentar uma última abordagem.

4.5) 5.ª Tentativa – *Gradient Boosting Regressor:* este algoritmo é uma versão mais avançada e sofisticada da *Random Forest* utilizada na tentativa anterior. A *Random Forest* constrói múltiplas árvores de decisão independentes e utiliza a média de todas para apresentar o resultado enquanto o *Gradient Boosting* constrói árvores sequencialmente corrigindo os erros da árvore anterior. Os resultados obtidos são apresentados na tabela em baixo:

MSE	0,274
RMSE	0,524
R^2	0,994

O MSE e o RMSE apresentam valores bastante baixos, indicando uma boa precisão do modelo em prever os valores reais, tendo apenas uma diferença média de 0,524 valores entre os valores previstos e os valores reais. O R^2 próximo de 1 sugere que o modelo consegue explicar a maioria da variância dos dados de teste.

Estes resultados são muito promissores e fazem-me concluir que cheguei ao modelo mais eficaz para este problema superando significativamente as outras abordagens testadas.

4.4) Otimização

Como abordado ao longo de vários módulos desta pós-graduação, a otimização dos hiperparâmetros é uma etapa fundamental em projetos de data science para garantir que os modelos estão a atingir o seu potencial máximo. Tendo isto em consideração, esta fase do relatório irá focar-se na explicação do método de otimização utilizada para melhorar ainda mais os resultados do modelo escolhido*: Gradient Boosting Regressor*.

O algoritmo utilizado é um algoritmo altamente sensível aos seus hiperparâmetros como por exemplo a learning rate utilizada ou ainda a profundidade máxima da árvore.

Para encontrar então o melhor valor para os hiperparâmetros utilizados utilizei a função '*RandomizedSearchCV'* do pacote *Scikit-learn* para fazer uma procura aleatória entre intervalos de valor previamente definidos para cada hiperparâmetro. Esta função executa uma validação cruzada em várias combinações de hiperparâmetros, avaliando sempre o desempenho de cada combinação através de uma métrica de avaliação que neste caso, foi utilizado o MSE.

Os melhores hiperparâmetros encontrados foram:

- *learning rate* = 0,4177
- *max_depth* = 3
- *max_features* = None
- *min_samples_leaf* = 3
- *min_samples_split* = 4
- *n_estimators* = 388
- *subsample* = 0,803

Este hiperparâmetros encontraram um menor MSE com um valor de 0,0104. Este resultado indica que, ao utilizar estes hiperparâmetros no modelo consegue-se uma capacidade superior de generalização e maior eficácia na previsão dos valores do *target*, ou seja, o número de dias que um pagamento está atrasado em relação à data de vencimento da fatura

Após a aplicação dos melhores hiperparâmetros, os resultados foram os seguintes:

MSE	0,058
RMSE	0,057
R^2	0,998

Estes valores indicam uma melhoria substancial em relação ao desempenho anterior do modelo, tendo melhorado a capacidade de prever o número de dias que um pagamento está atrasado em relação à data de vencimento da fatura.

Vale a pena mencionar que, este processo embora ajude bastante na procura dos melhores valores para os hiperparâmetros, trata-se de um processo custoso visto consumir bastante tempo e recursos computacionais.

5.) Implementação do modelo

Tendo já obtido um modelo capaz de fazer o que era proposto neste projeto, é muito importante perceber de que maneira se consegue transformar esta ferramenta em ações práticas do dia a dia de uma empresa bem como perceber o impacto real que terá nos negócios. Desta forma, a questão que se coloca é: como implementar este modelo em ambiente de produção?

5.1) Como empresas podem implementar o modelo em ambiente de produção

Na fase de implementação do modelo em ambiente de produção, o foco principal está na engenharia de dados, ou seja, não está no scope da especialidade de data science. Embora a implementação técnica seja mais orientada para equipas de engenharia de dados, é crucial para os profissionais de data science compreenderem como podem estes modelos serem implementados em ambiente de produção para colaborar e participar na integração e operacionalização dos modelos.

Por este motivo, julgo ser pertinente incluir neste relatório algumas etapas que as empresas podem seguir para implementar este modelo: Em primeiro lugar está a preparação dos dados em tempo real garantindo que os mesmos estejam disponíveis e formatados corretamente, isto envolve a construção de fluxos de trabalhos em tempo real para processar e limpar os dados à medida que chegam. Depois, é necessário fazer efetivamente a integração do modelo com os sistemas existentes da empresa, isto pode incluir a implementação de APIs que recebe os dados reais, aplicam o modelo e retornam as previsões. Para se garantir que o modelo não perde performance é muito importante fazer-se uma monitoração contínua para acompanhar o desempenho do modelo em produção: isto inclui desenvolver um sistema de controlo que monitorize as métricas de desempenho. Outra sugestão que julgo ser muito pertinente dar é para o retreinamento e atualização periódica do modelo com dados novos para garantir que continue relevante e preciso ao longo do tempo. Existem softwares orquestradores que têm a capacidade de fazer este re-treino e atualização do modelo que forma automática e sem intervenção humana.

E por fim, manter sempre a documentação do modelo atualizada contendo informação detalhada acerca do mesmo, incluindo a arquitetura, que algoritmos estão a ser utilizados, métricas utilizadas para avaliar o desempenho e limitações identificadas.

6.) Conclusão

Na conclusão deste relatório que documenta o que foi abordado no decorrer do desenvolvimento deste projeto, quero destacar que a sua realização foi uma jornada emocionante não só porque é o sinal que a pós-graduação que me propus fazer terminou como representa a solidificação da certeza que pretendo fazer carreira profissional na área de data science.

Durante o projeto, apliquei técnicas de exploração de análise de dados, criei variáveis novas, testei vários modelos de regressão para prever os pagamentos em atraso, apliquei algoritmos de otimização e ainda fiz sugestões de como as empresas podem implementar o modelo em ambiente de produção.

Para releases futuras deste projeto, consigo listar uma variedade de atualizações que é possível fazer nomeadamente o desenvolvimento de modelos de classificação, explorar técnicas avançadas de feature engineering, adicionar novas features como por exemplo emails e insights escritos dos clientes o que leva ao desenvolvimento de outro tipo de modelos como redes neuronais.

Estamos a viver um marco importante na nossa sociedade com a evolução da inteligência artificial. Considero importante terminar este projeto destacar a importância da ética em todas etapas do desenvolvimento e implementação de um modelo deste género: é crucial lembrar que por trás de números de de modelos, estão pessoas. Embora a adoção destes modelos de previsão seja ótimo para as empresas otimizarem recursos, não podemos jamais perder de vista a sensibilidade do tema: o não pagamento das faturas pode ser resultado de dificuldades financeiras, imprevistos ou outras circunstâncias pessoais dos clientes. É fundamental humanizar o contacto com o problema quando o objetivo é resolver problema de faturas em atraso, esta abordagem, para além de ser empática e tornar a imagem da empresa melhor fortalece o relacionamento com os clientes.

Combinar a eficiência dos modelos preditivos com a empatia pelos clientes é o caminho ideal para construir relações duradouras o que garantirá o sucesso a longo prazo da empresa.

Para terminar, quero ainda agradecer à Rita Faria e ao Ivo Nogueira, coordenadores desta pós-graduação toda a orientação dada no decorrer não só deste projeto final como de toda a pós graduação. Obrigada!

ANEXOS

Figura 1 – Boxplot das variáveis numéricas

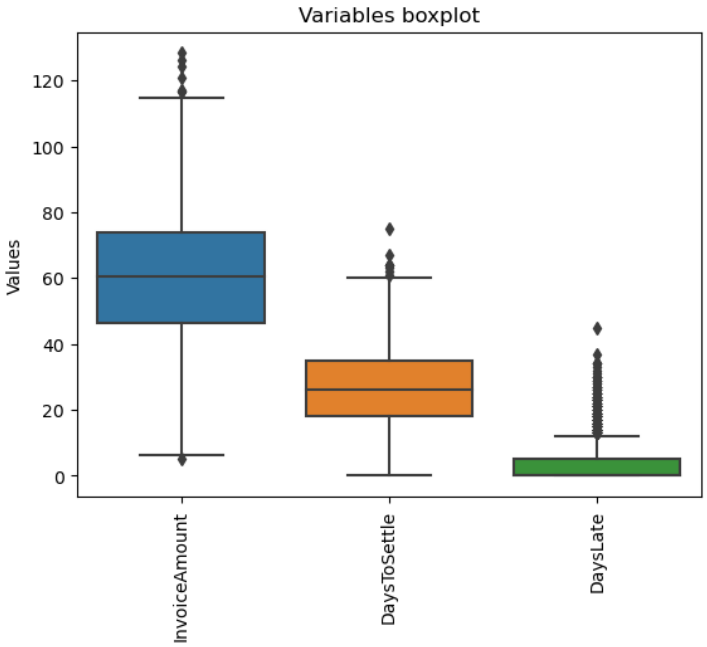

Figura 2 – Estatísticas Descritivas das variáveis numéricas

	InvoiceAmount	DaysToSettle	DaysLate
count	2466.000000	2466.00000	2466.000000
mean	59.895856	26.44485	3.442417
std	20.435838	12.33493	6.290607
min	5.260000	0.00000	0.000000
25%	46.400000	18.00000	0.000000
50%	60.560000	26.00000	0.000000
75%	73.765000	35.00000	5.000000
max	128.280000	75.00000	45.000000

Figura 3 – Histograma das variáveis numéricas

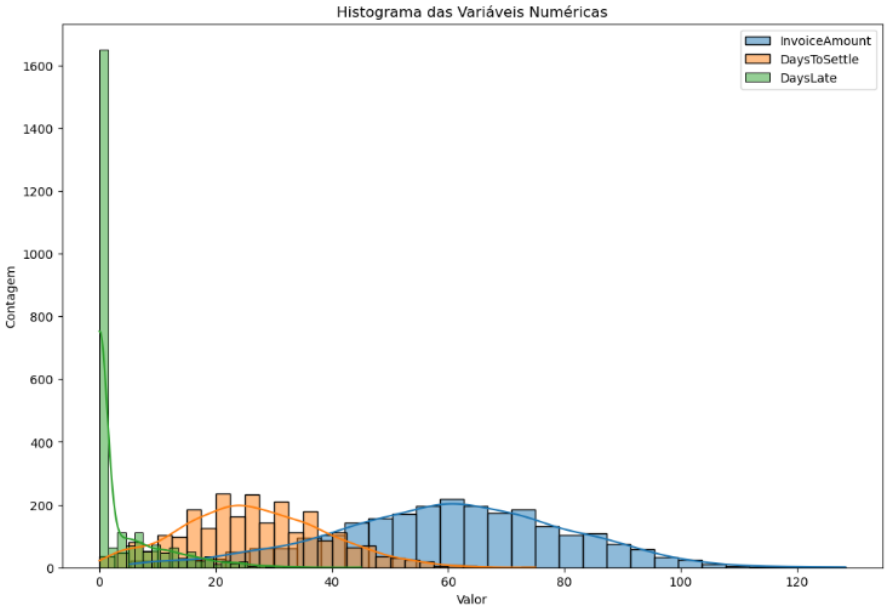

Figura 4 – Cálculo das frequências das variáveis categóricas