Guilherme Ramon Santos Camargo - 10734218

PROJETO DE DATA SCIENCE: HR ATTRITION PREDICTION PIPELINE

Trabalho de final de disciplina apresentado à Universidade Presbiteriana Mackenzie.

Orientador(a): Matheus Pavani

São Paulo

2025

RESUMO EXECUTIVO

A rotatividade de funcionários na TechCorp Brasil virou um problema sério nos últimos tempos, com impacto direto tanto nos custos quanto no clima da empresa. Pensando nisso, o projeto teve como objetivo desenvolver um sistema que ajudasse o RH a identificar, com antecedência, quais funcionários têm maior chance de pedir demissão, permitindo que ações preventivas fossem tomadas a tempo.

Durante o desenvolvimento, foi montado um pipeline completo em Python, passando por todas as etapas: análise exploratória, criação de variáveis, tratamento da base, modelagem, avaliação e geração de relatórios. Algumas etapas se destacaram, como a criação de novas features (derivadas e polinomiais), o balanceamento da base com SMOTE e o ajuste de thresholds para melhorar o desempenho do modelo. A análise dos dados revelou padrões claros, como maior risco de saída entre pessoas mais jovens, com pouco tempo de casa e que fazem muitas horas extras.

Depois de testar diferentes modelos, o LightGBM se mostrou o mais eficiente. Apesar de ter um recall um pouco menor, ele teve a melhor combinação entre precisão e F1-Score, além de um tempo de treinamento muito mais rápido do que modelos como o Random Forest. Ele também mostrou boa capacidade de interpretação das variáveis, o que ajuda o RH a entender o que pode estar por trás da saída de um colaborador. Mesmo não pegando todos os casos, os alertas gerados são mais confiáveis — o que é melhor do que ter muitos falsos positivos.

A ideia é que esse modelo seja implantado dentro de uma estrutura MLOps, com automação, reavaliação constante e integração com ferramentas internas. Assim, o RH pode agir com mais estratégia, focando nos casos de maior risco e propondo ações mais direcionadas, como políticas de retenção, melhoria no equilíbrio entre vida pessoal e profissional ou revisões salariais.

Mais do que um exercício técnico, esse projeto mostrou como a ciência de dados pode ser aplicada de forma prática e trazer valor real para o negócio. Foi um processo que permitiu aprender bastante sobre dados, modelos e, principalmente, sobre como gerar impacto com aquilo que construímos.

Palavras-chaves: 1. Rotatividade. 2. Ações preventivas 3. Modelos. 4. MLOps. 5. Retenção

SUMÁRIO

1. INTRODUÇÃO

O trabalho se baseia na seguinte situação:

> "A TechCorp Brasil, uma das maiores empresas de tecnologia do país, com mais de 50.000 funcionários, está enfrentando um problema crítico: sua taxa de attrition (rotatividade de funcionários) aumentou 35% no último ano, gerando custos estimados em R$ 45 milhões.
>
> Cada funcionário que deixa a empresa representa não apenas custos de demissão e contratação (estimados em 1,5x o salário anual), mas também:
>
> - Perda de conhecimento institucional
> - Impacto na produtividade das equipes
> - Diminuição da moral dos colaboradores
> - Atrasos em projetos críticos
>
> Você foi contratado como Cientista de Dados para desenvolver um sistema preditivo que identifique funcionários com alto risco de deixar a empresa, permitindo que o RH tome ações preventivas."

A rotatividade de funcionários é algo estudado a tempos e se mostra prejudicial não somente por custos do processo de desligamento e contratação de funcionário, mas também pela perda intelectual, como apresentado na problematização do trabalho. Em a Teoria do Capital Humano, desenvolvida por Becker (1994), já era abordado o impacto positivo que a educação do profissional tem sobre a capacidade produtiva tanto do trabalhador como das empresas. (FERREIRA; ALMEIDA, 2015, p. 29).

Para resolução do problema foi criado um modelo estatístico preditivo. O termo modelo vem do italiano módello, por sua vez, derivado do latim vulgar modellus, alteração feita ao latim modulus, o qual é diminutivo de modus, ou seja, medida. (Japiassu e Marcondes,1989). Segundo Segundo Gouveia Jr. (2003), modelo é a forma ideal, o paradigma, tendo por função a criação de outros como ele.

Assim, através da linguagem Python, foi possível definir um modelo que tem como objetivo prever quais funcionários têm alto risco de deixar a empresa, permitindo que o RH identifique casos com antecedência e possa tomar ações com propósito de reter o funcionário.

Para criação de um modelo é foi seguido alguns passos por algumas etapas, sendo:

- Definição do Problema
- Definição de "Variável" Alvo

- Definição de fonte de dados (base com dados que serão utilizados – base gerada IBM)
- Tratamento dos dados (base gerada IBM)

Propriedade	Valor
Nome do Dataset	IBM HR Analytics Employee Attrition & Performance
Número de Registros	1,000,000
Número de Features	35
Variável Alvo	Attrition (Yes/No)
Taxa de Attrition	~16%
Tipo de Problema	Classificação Binária

- Análise exploratória de dados (EDA)
    - Compreensão das variáveis (informação de variáveis – tipo, nome; descrição – quantidade, desvio padrão, média, entre outros; validação de dados nulos; distribuição de variável alvo, correlação de variáveis, entre outros)
    - Compreensão de variáveis descritivas (gráficos com variáveis categóricas relavantes em relação a variável alvo)
    - Compreensão de variáveis numéricas (gráficos com variáveis númericas relavantes em relação a variável alvo)
- Feature Engineering
    - Definição de variáveis utilizadas no modelo (variáveis da base e novas)
- Modelagem
    - Definição dos modelos utilizados
    - Desbalanceamento (variável alvo é desbalanceada)
    - Otimização de hiperparâmetros
    - Gráficos com resultados
    - Relatório de resultados
    - Salvamento de modelo

Seguindo esse processo e através dos resultados apresentados, foi possível definir

o modelo que melhor se adequa a situação e que teria maior eficácia em prever quais funcionários têm alto risco de deixar a empresa.

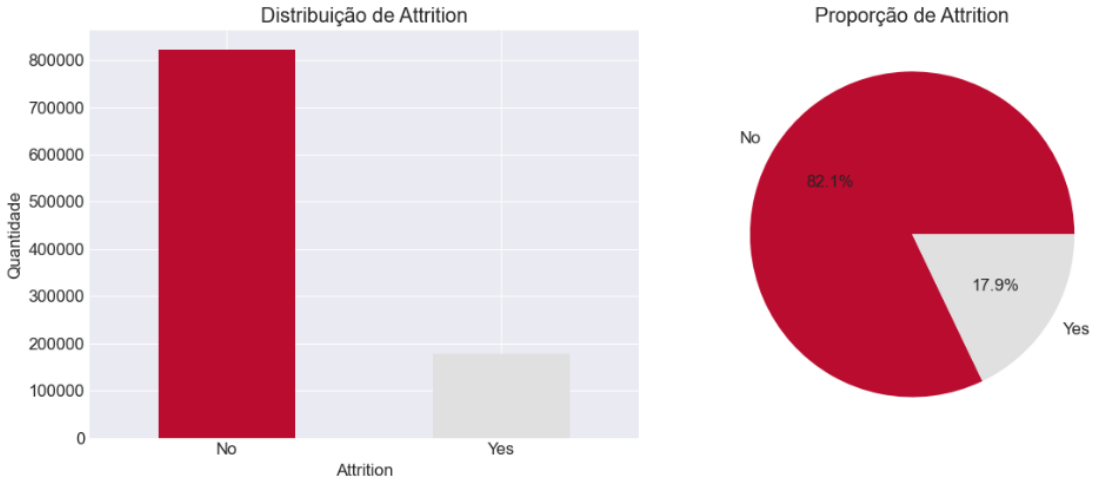
2. ANÁLISE EXPLORATÓRIA DE DADOS

Na etapa de análise explorátoria de dados foi analisado, a priori, uma visão geral acerca das variáveis buscando distinguir sua tipagem (categórica e numérica) e também entender algumas informações descritivas iniciais, como média e mediana de idade, salário, satisfação do cliente, entre outros:

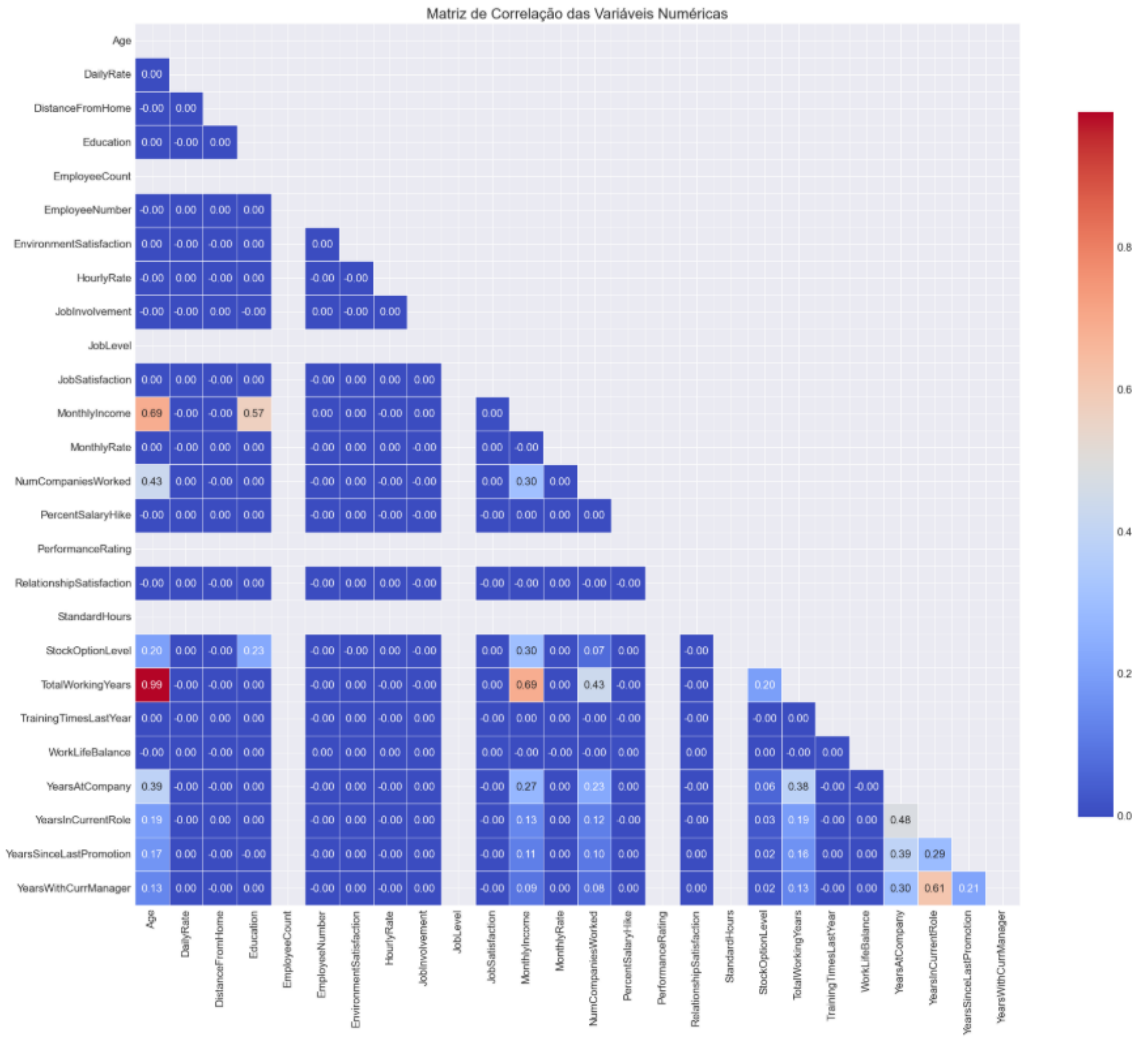
	count	mean	std	min	25%	50%	75%	max
Age	1000000.0	37.573109	9.752131	18.0	31.00	37.0	44.00	65.0
DailyRate	1000000.0	800.110722	404.135932	100.0	451.00	800.0	1150.00	1499.0
DistanceFromHome	1000000.0	7.389395	6.487020	1.0	3.00	5.0	10.00	29.0
Education	1000000.0	3.250821	0.993399	1.0	3.00	3.0	4.00	5.0
EmployeeCount	1000000.0	1.000000	0.000000	1.0	1.00	1.0	1.00	1.0
EmployeeNumber	1000000.0	500000.500000	288675.278933	1.0	250000.75	500000.5	750000.25	1000000.0
EnvironmentSatisfaction	1000000.0	2.898258	0.944161	1.0	2.00	3.0	4.00	4.0
HourlyRate	1000000.0	64.528856	20.204914	30.0	47.00	65.0	82.00	99.0
JobInvolvement	1000000.0	3.050033	0.803532	1.0	3.00	3.0	4.00	4.0
JobLevel	1000000.0	2.960664	0.700052	1.0	3.00	3.0	3.00	5.0
JobSatisfaction	1000000.0	2.900421	0.943386	1.0	2.00	3.0	4.00	4.0
MonthlyIncome	1000000.0	11311.698693	2585.337246	1258.0	9612.00	11336.0	12929.00	20000.0
MonthlyRate	1000000.0	14504.885874	7210.287533	2000.0	8266.00	14516.0	20745.00	26999.0
NumCompaniesWorked	1000000.0	3.437956	2.749722	0.0	1.00	3.0	6.00	9.0
PercentSalaryHike	1000000.0	14.960437	3.132714	11.0	13.00	15.0	17.00	25.0
PerformanceRating	1000000.0	3.160585	0.367148	3.0	3.00	3.0	3.00	4.0
RelationshipSatisfaction	1000000.0	2.900015	0.942661	1.0	2.00	3.0	4.00	4.0
StandardHours	1000000.0	80.000000	0.000000	80.0	80.00	80.0	80.00	80.0
StockOptionLevel	1000000.0	1.104926	0.943522	0.0	0.00	1.0	2.00	3.0
TotalWorkingYears	1000000.0	17.646932	9.711037	0.0	11.00	17.0	24.00	47.0
TrainingTimesLastYear	1000000.0	2.818794	1.328194	0.0	2.00	3.0	4.00	6.0
WorkLifeBalance	1000000.0	2.750494	0.886850	1.0	2.00	3.0	3.00	4.0
YearsAtCompany	1000000.0	8.291746	5.723518	0.0	3.00	8.0	13.00	20.0
YearsInCurrentRole	1000000.0	3.712092	2.980426	0.0	1.00	3.0	6.00	10.0
YearsSinceLastPromotion	1000000.0	2.792037	2.236920	0.0	1.00	2.0	5.00	7.0
YearsWithCurrManager	1000000.0	2.084859	1.952559	0.0	0.00	2.0	3.00	7.0

Também nesse processo foi verificado a distribuição da variável target, onde foi possível verificar que possivelmente seria necessário um balanceamento para a a criação

do modelo.



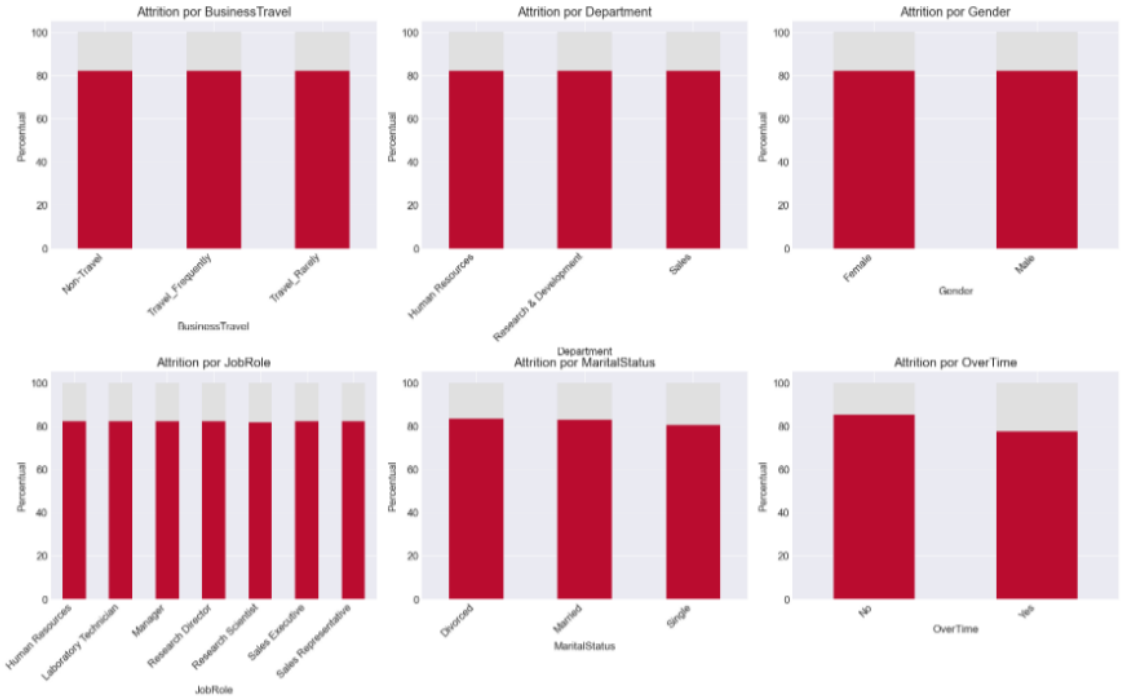
Após essa etapa, foi verificado a correlação entre as variáveis visando entender variáveis que pudessem ter efeitos semelhantes e também um primeiro contato para definição de features:



Nesse gráfico, quanto mais próximo do vermelho, maior a correlação da variável, como por exemplo: “age” e “TotalWorkingYears” que apresentam forte correlação, ou

“JobLevel” e “MonthlyIncome”.

Após isso, foi verificado as variáveis categóricas mais relevantes em comparação à variável target (“Atrittion”):



Aqui conseguimos visualizar a relação de algumas variáveis com a taxa de saída de funcionários (Attrition). Um ponto que chama atenção é o caso das horas extras (OverTime), onde quem faz tem uma taxa de saída visivelmente maior do que quem não faz, o que pode indicar um impacto direto da carga de trabalho no desligamento. Algo parecido acontece com quem viaja com frequência a trabalho, que também apresenta um attrition um pouco mais alto.

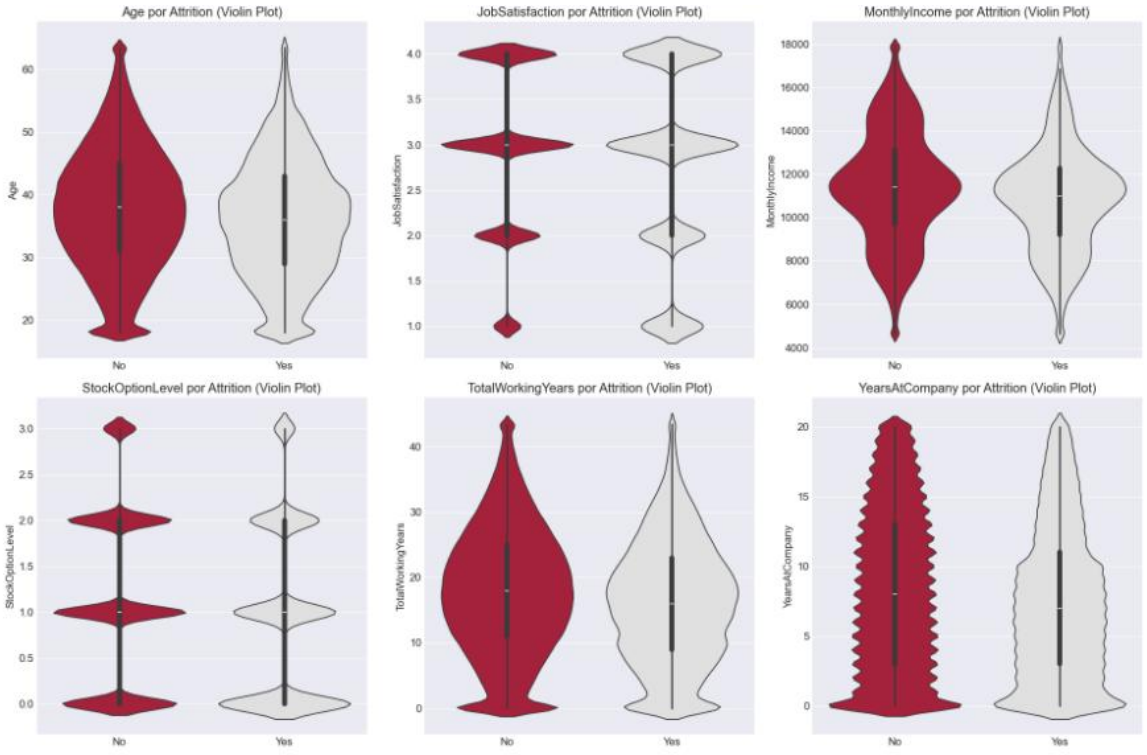
Apesar de a maioria das categorias não mostrar diferenças tão grandes, vale destacar que funcionários solteiros têm uma taxa de saída um pouco maior, o que pode indicar maior flexibilidade para trocar de emprego. Esses padrões ajudam a reforçar alguns perfis que podem demandar atenção especial do RH.

Na parte das variáveis numéricas, primeiro foram tratados valores nulos e outliers, e depois selecionadas as variáveis que mais faziam sentido para análise. Dá para notar que funcionários mais jovens, com menos tempo de casa e menos experiência no geral tendem a sair mais. Isso reforça a importância de olhar com mais atenção para esse grupo, seja com ações de retenção ou acompanhamento mais próximo.

Outros pontos relevantes observados nos gráficos incluem menor satisfação no trabalho e salários mais baixos entre os que saem, o que também pode ser um fator de

desmotivação. Já o tempo total de carreira (“TotalWorkingYears”) e os anos na empresa seguem o mesmo padrão: quanto menor, maior a chance de desligamento.

Abaixo estão os gráficos de violino que ajudam a visualizar melhor essas diferenças. Eles trazem uma ideia parecida com os boxplots, mas mostram de forma mais clara onde está concentrado o maior volume de funcionários em cada situação.



3. DESENVOLVIMENTO DA SOLUÇÃO

Após uma análise descritiva das variáveis, foi possível ter maior noção acerca do público trabalhado e assim ter noção referente a variáveis presentes na base. Com isso, foi iniciado o pipeline de modelagem.

3.1. Feature Engineering (*load_and_prepare_data, create_features* e *create_polynomial_features*)

O primeiro passo é a seleção de variáveis que devem ser usadas, que podemos chamar de feature engineering. Para isso, inicialmente foi criado algumas variáveis derivadas das oferecidas na base que poderiam ser relevantes e utéis para o modelo:

- Features Numéricas Derivadas:
    - Renda mensal dividida pela idade: Proxy de progressão da carreira.

    - Total de anos de experiência dividido pelo número de empresas: Mede a estabilidade ou rotatividade da carreira.
    - Renda ajustada pelo nível educacional: verifica retorno financeiro por nível de formação (possível insatisfação por subvalorização).
    - Distância ponderada pela satisfação ambiental: alto desgaste no trajeto pode impactar no desejo de sair.
    - Medição de quanto tempo a pessoa está sem ser promovida proporcionalmente ao tempo de empresa: útil para entender estagnação no mercado.
- Features Binárias:
    - Flag de distância: indica se a distância de casa é elevada (mais de 20km), o que pode contribuir para o desejo de desligamento.
    - Flag de satisfação: identifica pessoas com satisfação muito baixa em qualquer um dos critérios (risco elevado de saída).
    - Flag de promoção: identifica funcionários há mais de 5 anos sem promoção que pode ser fator possível para desmotivação ou estagnação.
    - Flag de nota de desempenho: marca os funcionários com nota máxima de desempenho, oq ue pode levar a uma concorrência pelo funcionário no mercado externo.
    - Flag de idade por cargo: indica jovens (<30) em cargos de alto nível (JobLevel ≥ 3) que podem buscar maiores desafios no mercado
- Features Compostas:
    - Score médio de satisfação em três dimensões: visão holística da satisfação no trabalho (JobSatisfaction, EnvironmentSatisfaction e RelationshipSatisfaction).
    - Composição da intensidade de trabalho: soma efeito de horas extras e frequência de viagens
- Features Categóricas:
    - Faixa de idade: útil para capturar efeitos geracionais e aplicação de fairness
    - Faixa de renda: permite entender padrões entre classes salariais

Para finalizar essa etapa foi criado uma função que selecionaria outras 10 variáveis polinomiais relevantes que poderiam melhorar a capacidade do modelo ao capturar relações não lineares, por exemplo.

3.2. Pré processamento de dados (*create_preprocessor*)

Após essa etapa temos a etapa pré modelagem, onde é idetinficado os features e padroniza as variáveis catégoricas e numéricas que serão utilizadas.

3.3. Definição de hiperparâmetros (*optimize_hyperparameters* e *objective*)

Seguindo adiante, chegamos numa das estapas mais importantes do processo de modelagem. A definição de hiperparâmetros, que afeta diretamente o desempenho do modelo (a partir dessa etapa, a função pedirá ao chamar para definir o % de treino e teste e random_state que garante que a mesma divisão seja reproduzida sempre que o código rodar, sem ele será sempre aleátorio o público selecionado).

Esse processo ocorre antes do treino do modelo e é o momento onde se define configurações do modelo. Para essa etapa foi utilizado o Optuna e seguimos algumas etapas:

- Validação dos dados
- Validação cruzada estratificada
- Definição da função objetivo do Optuna
    - Definição do número de tentativas de otimização
    - Definição do modelo (Random Forest, Logistic Regression, Light GBM e Ridge Classifier) – O modelo poderia ser qualquer outro que a biblioteca possua, contando que tenha os hiperparâmetros definidos corretamente.
    - Divide os dados com K-Fold
    - Treina o modelo
    - Avalia com roc_auc_score usando predict_proba ou decision_function
    - Cria um estudo onde o objetivo é maximizar o score.
    - Usa o TPESampler, que é um algoritmo eficiente de busca de hiperparâmetros.
    - Por fim a função deve retornar os melhores parâmetros

3.4. Treino de modelo (*train_models)*

Na etapa de treino de modelo é quando núcleo do treinamento dos modelos**,** onde todo o pipeline entra em ação. Até o momento só deifinimos variáveis e

parâmetros/configurações, e tudo isso é para essa etapa em que isso será colocado em prática.

Para esse processo também seguimos algumas etapas:

- Remoção de colunas duplicadas – Segunda validação apesar de ser feito um na etapa de feature engineering.
- Roda a etapa de pré processamento de dados que padroniza as variáveis.
- Balanceamento e conversão: Na etapa de EDA, verificamos que a variável target estava desbalanceada e para lidar com ela, foi considerado a função SMOTE que poderia ser útil para essa situação, diante do fato que ela cria amostras sintéticas da classe minoritária na base teste.
- Guarda nome das features para situações futuras, como gráfico de importÂncia de feature no modelo.
- Treinamento dos modelos definidos, executando a busca de hiperparâmetros com Optuna e selecionando os melhores parâmetros
- Realiza as previsões
- Calcula métricas importantes: accuracy, precision, recall, f1, auc_roc, e o tempo gasto
- Salva as informações no dicionário
- Ensemble: Voting Classifier: Faz as previsões combinadas (modelos de classificação – Random Forest e Light GBM) e calcula as métricas do ensemble.

3.5. Threshold (*recommend_threshold*)

Essa etapa foi adicionada após o treinamento: é nela que é ajustado o threshold de decisão do modelo — o ponto de corte que determina a partir de qual valor de probabilidade uma amostra será classificada como 0 ou 1. O modelo costuma ter um threshold de 50%/50%, mas sabemos que nosso target não é assim pelo EDA e nessa etapa realizamos esse ajuste.

3.6. Gera visões e relatório (*plot_results*, *generate_report* e *analyze_errors_and_fairness*)

Aqui é feito alguns alguns gráficos a partir do treino de modelo, como: importância do feature de cada modelo, tempo de treinamento por modelo, curvas ROC, compartivos de métricas de cada modelo, curvas precision-recall, distribuiçaõ de classe e

outros.

Também é feito um relatório em que apresenta algumas informações como o comparativo de métricas de cada modelo como acurácia, precisão, recall, F1-Score, AUC-ROC e tempo de treinamento, e a partir desses dados retorna uma recomendação de melhor modelo. Também retorna o hiperparâmetros otimizados e recomendações e insights relevantes.

Por fim e não menos importante, temos informação sobre o volume de falsos positivos, falsos negativos atra´ves da análise de erros e fairness.

3.7. Salva modelos (*save_models*)

Após todas essas etapas, finaliza-se o pipeline com o salmento dos modelos na pasta definida

3.8. Comparação de modelos

Foi desenvolvido um código que permite chamar o modelo tanto sem os *features* criados quanto com eles, para fins de comparação. No entanto, devido ao alto tempo de processamento, optou-se por desconsiderar a versão sem os *features* e manter apenas a versão com os atributos gerados

4. RESULTADOS E AVALIAÇÃO

Após rodar o pipeline, a partir dos resultados apresnetados foi possível definir o melhor modelo a partir de métricas como:

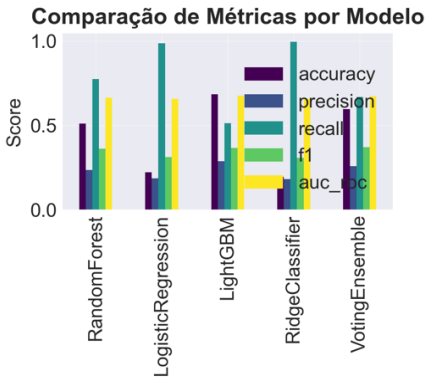
- Acurácia
- Precisão
- Recall
- F1
- AUC-ROC
- Tempo de treinamento do modelo

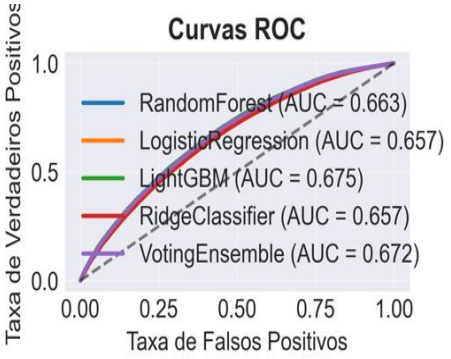
Abaixo temos os resultados comparativos onde foi possível definir o LightGBM como modelo mais apto:

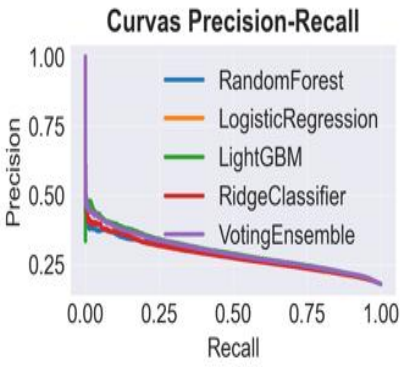
```
MÉTRICAS DE DESEMPENHO:
              Modelo Acurácia Precisão Recall F1-Score AUC-ROC Tempo (s)
        RandomForest   0.5102   0.2352 0.7744   0.3608  0.6629  18117.16
  LogisticRegression   0.2218   0.1849 0.9860   0.3114  0.6570   3162.88
            LightGBM   0.6848   0.2860 0.5117   0.3669  0.6751   1906.61
     RidgeClassifier   0.1955   0.1810 0.9951   0.3063  0.6571    475.86
      VotingEnsemble   0.5955   0.2569 0.6691   0.3712  0.6719   2420.94

✨ MELHOR MODELO: LightGBM (AUC-ROC: 0.6751)
```

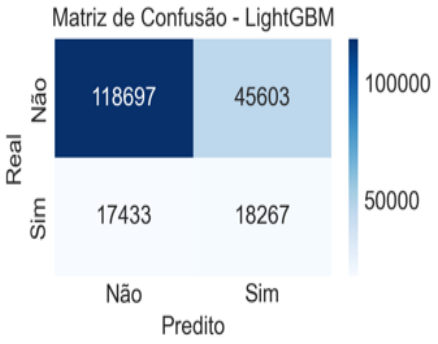
Apesar de todos os modelos terem tido resultados semelhantes, o LightGBM se mostrou o melhor, pois, apesar de um percentual de recall menor, apresentou uma precisão maior, além de um maior equilíbrio entre precisão e recall (F1-Score), o que leva a um menor número de falsos positivos. Além disso, apresentou um AUC-ROC moderadamente maior que os demais modelos, com um tempo de treinamento muito inferior a modelos como o Random Forest, que demorou 9x mais. O que impacta negativamente nesse modelo é seu recall, que pode levar a não informar todos que realmente irão sair, mas sua precisão compensa esse detalhe.

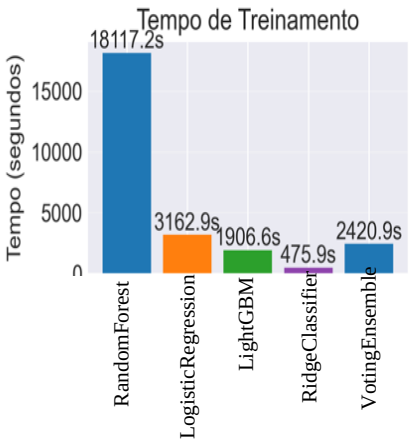
A partir dos gráficos, é possível ter uma visão acerca dos resultados apresentados acima (Os gráficos ficaram desajustados em tamanho):

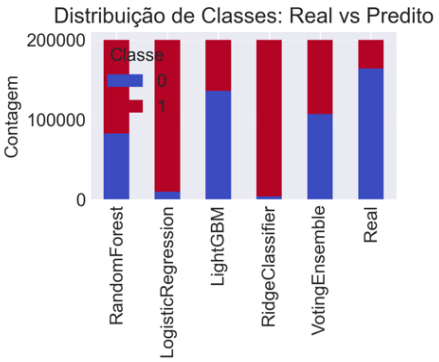










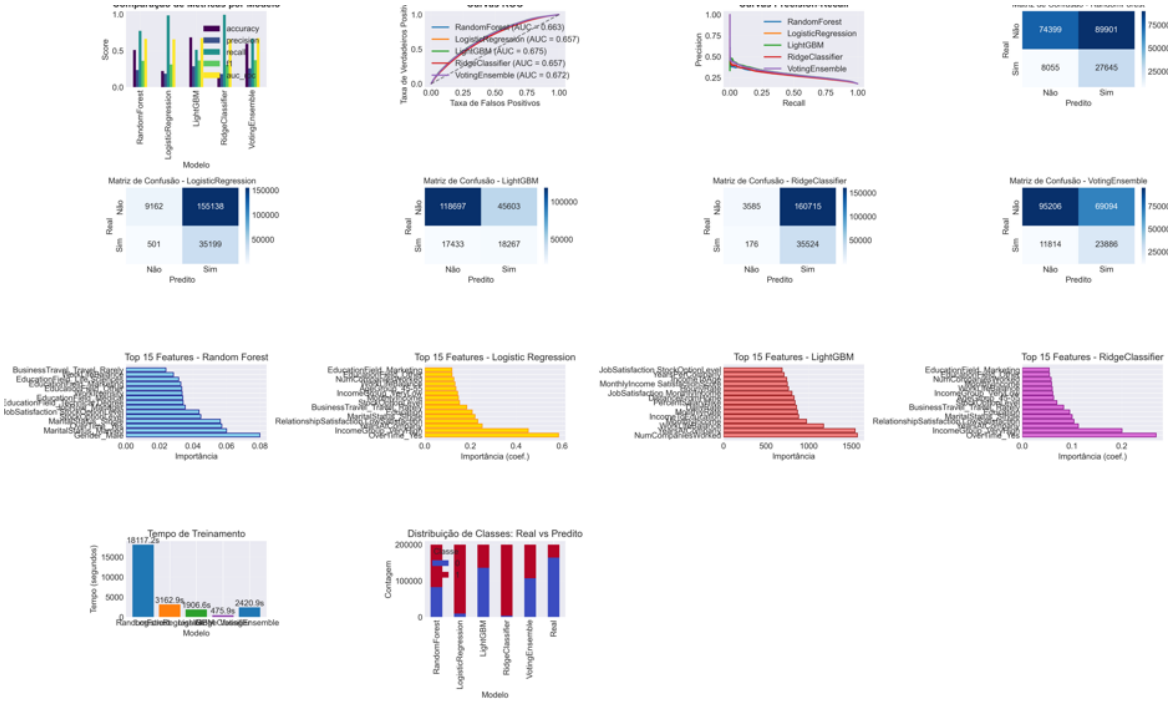


Os gráficos apresentados acima levam a crer que o LightGBM (um modelo baseado em árvores de decisão) foi o que apresentou o melhor desempenho. Um ponto interessante é que tanto o LightGBM quanto o Voting Ensemble (que utiliza o LightGBM como um dos modelos) foram os que mais se aproximaram da distribuição real, embora o Voting tenha ficado visivelmente abaixo.

Quanto aos features, temos o seguinte top 5:

- NumCompaniesWorked: Número de empresas em que já trabalhou
- YearsAtCompany: Anos na empresa atual
- WorkLifeBalance: Equilíbrio entre vida pessoal e trabalho
- MonthlyIncome: Renda Mensal
- IncomeToEducation: Renda em relação ao nível educacional

Segue abaixo comparativo completo de resultados:



5. IMPLEMENTAÇÃO E PRÓXIMOS PASSOS

Com o modelo definido, será possível ter um alerta muito mais assertivo sobre quais funcionários possuem tendência a sair da empresa. Mas, para que isso funcione na prática, é necessário que o modelo seja implantado seguindo o ciclo de vida do MLOps.

Segundo a Amazon, MLOps (Machine Learning Operations) é um conjunto de práticas que integra ciência de dados, engenharia e operações, com foco em automatizar e gerenciar todo o ciclo de vida de modelos de machine learning. Inspirado no DevOps, o MLOps garante automação, controle de versões, governança e atualizações contínuas, permitindo entregas mais ágeis, seguras e com mais valor para o negócio.

Com o modelo implantado, o RH poderia utilizá-lo para lidar antecipadamente com possíveis casos de demissão, priorizando situações mais críticas de acordo com os resultados do modelo e com as variáveis mais relevantes, como equilíbrio entre vida pessoal e profissional, tempo sem promoção ou questões salariais. A ideia é que o modelo ajude a direcionar ações mais assertivas de retenção, com base em dados.

O monitoramento será essencial, já que com o tempo o histórico de dados tende a crescer, o que permitirá um modelo mais completo e treinado com mais exemplos. Além disso, será possível identificar desbalanceamentos ou mudanças no padrão dos dados que possam exigir ajustes ou um novo treino do modelo.

6. CONCLUSÃO

O projeto e a disciplina ajudaram a ter uma visão mais clara de todas as etapas envolvidas na criação de um modelo preditivo. Ao seguir o processo completo, foi possível tirar insights relevantes de negócio e aplicar técnicas importantes, como o balanceamento com SMOTE, ajustes de threshold, tratamento de outliers, além de lidar com situações que impactam diretamente a performance do modelo.

Na parte de análise descritiva, por exemplo, foi possível perceber a relação entre horas extras e a saída de funcionários, além de padrões ligados ao tempo de casa, que se mostraram relevantes no gráfico de correlação. Isso ajudou a entender melhor o perfil de quem sai da empresa e direcionar a construção do modelo com foco no problema real. A criação de variáveis polinomiais, por exemplo, também foi uma forma de capturar relações que não eram lineares, e que melhoraram o desempenho do modelo.

Com base nisso, foi possível escolher o LightGBM como modelo final, principalmente pelo equilíbrio entre as métricas, pela precisão nas previsões e pela clareza nas importâncias das variáveis, o que facilita o entendimento do RH. A aplicação de técnicas de avaliação, como o ajuste de thresholds e análise de erros, também trouxe mais segurança sobre o uso do modelo em cenários reais.

Ao ser implantado, o modelo permite que a empresa identifique grupos com maior risco de saída e possa agir antes que isso aconteça. Isso pode incluir, por exemplo, melhorias em políticas de promoção para quem está há mais tempo na empresa, ações para melhorar o equilíbrio entre vida profissional e pessoal (como controle de horas extras), ou até mesmo ajustes salariais em casos onde o modelo apontar insatisfação com a remuneração.

No geral, o projeto mostrou na prática como a ciência de dados pode ajudar diretamente o negócio a evitar perdas, reduzir custos com demissões inesperadas e reter talentos por meio de decisões mais estratégicas e baseadas em dados.