FACULDADE DE MEDICINA DA USP
PICM

inteli
instituto
de tecnologia
e liderança

Autores:

Giovane Goffi Andreussi

Jonas Viana Sales

Kaique Ramon Nogueira Dantas

Renato Silva Machado

Theo Albero Tosto

Vitor Augusto Menten de Barros

Data de criação: 12 de agosto de 2022

Versão: 5..1

Controle do Documento

Histórico de revisões

Data	Autor	Versão	Resumo da atividade
12/08/2022	Vitor, Jonas, Giovane, Renato, Theo, Kaique	1.1	Criação da parte de negócios (análise do mercado, SWOT, Matriz de Riscos, Canvas Value Proposition), persona e jornada do usuário, descrição da solução e descrição dos dados.
26/08/2022	Vitor, Giovane	2.1	Preparação dos Dados
09/09/2022	Vitor	3.1	Definição dos modelos e avaliação
23/09/2022	Vitor	4.1	Avaliação dos modelos com hiperparâmetros
06/10/2022	Vitor	5.1	Metodologia e Conclusão

Sumário

1. Introdução

Inaugurado em 2008, o Instituto do Câncer do Estado de São Paulo (ICESP), unidade do Hospital das Clínicas da Faculdade de Medicina da USP (HCFMUSP), é um dos maiores centros de oncologia da América Latina, com pilares em assistência, ensino e pesquisa, e atendimento 100% pela rede pública de saúde. Ao longo dos anos consolidou um padrão de gestão acolhedora ao prestar uma assistência com atendimento humanizado e de qualidade. Possui índices de 96,2% de satisfação dos usuários e foi eleito duas vezes como melhor hospital do estado pelos pacientes (2010 e 2014). Em pouco mais de uma década, já atendeu mais de 110 mil pessoas, foram realizados mais de 24 milhões de exames de análises clínicas e 1,8 milhão de exames de imagem. Diariamente circulam pelas instalações cerca de 10 mil pessoas. Com alta densidade tecnológica, o Instituto conta com um dos maiores parques radioterápicos da América Latina com aceleradores lineares para radioterapia, equipamento de braquiterapia e tomógrafo para simulação de procedimentos.

O problema a ser resolvido é a falta de padrões e a grande variabilidade da evolução do câncer de mama e sua resposta a tratamentos convencionais. Pacientes que possuem o mesmo subtipo de câncer de mama ou estão em uma mesma faixa de risco apresentam respostas diferentes a tratamentos iguais, alguns vivem mais do que o esperado e recebem alta e outros vão a óbito precocemente.

2. Objetivos e Justificativa

2.1. Objetivos

O objetivo principal é a criação de um modelo preditivo a partir de coortes de pacientes mulheres acompanhadas em projetos de pesquisa do Instituto do Câncer do Estado de São Paulo/Faculdade de Medicina da Universidade de São Paulo. O modelo deve mostrar um score de risco (esquema de cores) que ajudará na decisão da abordagem e do tratamento mais adequados, além de aumentar a objetividade no acompanhamento com dados mais precisos.

2.2. Proposta de Solução

A solução proposta do projeto - Modelo preditivo a partir de variáveis clínico-laboratoriais de pacientes com câncer de mama - ajudará na avaliação de dois pontos : a) Tempo de sobrevida do paciente e b) Resposta ao tratamento do paciente.

A evolução do câncer de mama e sua resposta a tratamentos convencionais é muito variável. O ICESP quer identificar padrões preditivos dessa variabilidade a partir de dados clínicos e do seguimento desses pacientes, dessa maneira temos o objetivo da criação de modelo preditivo a partir de dados de pacientes mulheres acompanhadas em projetos de pesquisa do Instituto do Câncer do Estado de São Paulo/Faculdade de Medicina da Universidade de São Paulo.

O projeto vai fornecer uma *classificação* de acordo com o *grau de prioridade e urgência* da paciente com câncer de mama. Por exemplo, um sistema de cores sendo verde para pacientes com pouco risco e que não precisam de um acompanhamento tão recorrente e vermelho para pacientes com alto risco que devem ser acompanhados de perto.

2.3. Justificativa

Nossa proposta de solução trará uma ferramenta a mais para auxiliar médicos a tomar decisões, tornando o acompanhamento e tratamento mais eficientes e precisos ao encontrar padrões e tendências ocultas. Potencialmente, será uma inteligência na qual aprende ao longo do tempo, quanto mais fazer uso da I.A mais ela se torna eficiente e gera resultados mais sólidos e isso torna uma exponencial de aprendizado e eficiência em relação ao tempo de uso. Além disso, é um produto o qual terá escalabilidade para outros tratamentos e tipos de câncer. Desse modo, nosso produto vai gerar grande valor para a sociedade melhorando a área de saúde e tecnologia, trazendo valores diretos para médicos, hospitais e pacientes. Em relação ao

diferencial podemos citar: Interfaces e estrutura de interpretação de dados de modo moderno, mais eficiente, simples e menos confuso. A maneira atual na qual as telas expõem os dados retidos e trabalhos ainda é confusa, chata e trabalhosa. No ano de 1840 teve o surgimento de gráficos de linhas, colunas e de pizza, e até os dias de hoje fazemos uso dessa estrutura para interpretação de dados devido ao fato de que nos acostumamos com isso, mas podemos mudar e fazer diferente.

3. Metodologia

3.1. CRISP-DM

CRISP-DM, que significa Cross-Industry Standard Process for Data Mining, é uma forma comprovada pela indústria para orientar os esforços de mineração de dados. Como metodologia, ela inclui descrições das fases típicas de um projeto, as tarefas envolvidas em cada fase, e uma explicação das relações entre essas tarefas. Como modelo de processo, o CRISP-DM fornece uma visão geral do ciclo de vida da mineração de dados.

Ela é dividida em seis fases:

- **Entendimento do Negócio:** se concentra na compreensão dos objetivos e exigências do projeto a partir de uma perspectiva do negócio, convertendo esse conhecimento em uma definição do problema de mineração de dados e em um plano preliminar projetado para atingir os objetivos.
- **Entendimento dos Dados:** começa com uma coleta inicial de dados e prossegue com atividades para se familiarizar com os dados, para identificar problemas de qualidade dos dados, para descobrir os primeiros insights sobre os dados ou detectar subconjuntos interessantes para formar hipóteses sobre informações ocultas.
- **Preparação dos Dados:** cobre todas as atividades para construir o conjunto de dados final (dados que serão usados na(s) ferramenta(s) de modelagem a partir dos dados brutos iniciais). As tarefas de preparação de dados tendem a ser realizadas várias vezes e sem uma ordem prescrita. As tarefas incluem seleção de tabelas, registros e atributos, assim como transformação e limpeza de dados para ferramentas de modelagem.
- **Modelagem**: várias técnicas de modelagem são selecionadas e aplicadas e seus parâmetros são calibrados para valores ideais. Tipicamente, existem várias técnicas para o mesmo tipo de problema de mineração de dados. Algumas delas têm requisitos específicos na forma dos dados. Portanto, muitas vezes é necessário voltar para a fase de preparação dos dados.
- **Avaliação:** um modelo (ou modelos) que parece ter alta qualidade do ponto de vista da análise de dados foi construído. Antes de proceder à implantação final do modelo, é importante avaliá-lo mais profundamente e revisar as etapas executadas para construí-lo para ter certeza de que ele atinge os objetivos comerciais. Um objetivo-chave é determinar se existe alguma questão comercial importante que não tenha sido suficientemente considerada. No final desta fase, uma decisão sobre o uso dos resultados da mineração de dados deve ser alcançada.
- **Deploy:** implantar uma representação do código do modelo em um sistema operacional. Isto também inclui mecanismos para pontuar ou categorizar novos dados não vistos à medida que eles surgem. O mecanismo deve utilizar as novas informações na solução do

problema comercial original. É importante destacar que a representação do código também deve incluir todas as etapas de preparação dos dados que levam à modelagem. Isto assegura que o modelo tratará os novos dados brutos da mesma maneira que durante o desenvolvimento do modelo.

3.2. Ferramentas

As ferramentas utilizadas foram:

- Google Colaboratory: ferramenta utilizada para o desenvolvimento dos códigos do projeto. Assim, foi utilizado para tratar os dados e treinar e comparar os modelos.
- GitHub: ferramenta utilizada para armazenar e publicar todo o conteúdo do projeto, desde os códigos até a documentação.
- Google Drive: Compartilhamento e manuseio do banco de dados pela equipe

3.3. Principais técnicas empregadas

Os algoritmos utilizados para treinar o modelo foram:

- **Regressão Logística**: é análogo ao de regressão linear para problemas de classificação. Em vez de acharmos a reta que melhor se ajusta aos dados, vamos achar uma curva em formato de 'S' que melhor se ajusta aos dados.

- **Random Forest**: cria várias árvores de decisão, uma estrutura similar a um fluxograma, com "nós" onde uma condição é verificada, e se atendida o fluxo segue por um ramo, caso contrário, por outro, sempre levando ao próximo nó, até a finalização da árvore. No fim há uma votação de um resultado final.

- **Boosting**: No Boosting os modelos são treinados com os mesmos datasets, porém os pesos das instâncias são ajustados de acordo com o erro das predições anteriores.

- Bagging: Todos os modelos deste tipo de ensemble são do mesmo algoritmo, porém os dados de entrada de cada um são amostras do dado original, com a mesma quantidade de dados do dataset original, selecionadas usando o método bootstrap (aleatória com repetição).

- KNN: leva em consideração a classificação dos pontos mais próximos para determinar a classificação do novo ponto. Ele executa um cálculo matemático para medir a distância

entre os dados para fazer sua classificação, como por exemplo: Euclidiana, Manhattan, Minkowski, Ponderada e etc.

- Bayes: É um classificador probabilístico que prevê com base na probabilidade de um objeto. O teorema de Bayes é um dos conceitos mais populares de aprendizagem de máquina que ajuda a calcular a probabilidade de ocorrência de um evento com conhecimento incerto enquanto outro já ocorreu.

- Árvore de decisão: funciona dividindo o conjunto de dados em subconjuntos cada vez menores com base em seus recursos. A ideia é que as árvores de decisão dividam os dados repetidamente até que reste apenas uma classe. Por exemplo, a árvore pode fazer uma série de perguntas do tipo “sim ou não” e dividir os dados em categorias a cada etapa.

- SVM: plota-se cada item de dados como um ponto no espaço n-dimensional (onde n é o número de recursos que se tem), com o valor de cada recurso sendo o valor de uma determinada coordenada. Então, executa-se a classificação encontrando o hiperplano que melhor diferencia as duas classes.

- Stacking: Nesse modelo as predições dos modelos anteriores são combinadas por um outro modelo para obter a saída final. Podem ser criadas várias camadas com modelos diferentes.

*OBS: os modelos **em negrito** foram escolhidos para uma avaliação final.

4. Desenvolvimento e Resultados

4.1. Compreensão do Problema

4.1.1. Contexto da indústria

As 5 Forças de Porter são um framework de análise setorial que permite entender o nível de competitividade de um mercado. O modelo apresenta os atores envolvidos (concorrentes, fornecedores, compradores, novos entrantes e substitutos), como eles se relacionam e como influenciam o sucesso dos negócios.

Rivalidade entre os concorrentes:

Principais players:

1) Fleury

   O Grupo Fleury é uma empresa de saúde brasileira fundada em 1926, cuja principal atividade é a prestação de serviços médicos e medicina diagnóstica.

2) DASA

   A Dasa é a maior rede de saúde integrada do Brasil e tem a posição de líder em medicina diagnóstica no Brasil e na América Latina e é a 5ª maior do setor no mundo

3) Centro de Oncologia e Hematologia do Hospital Israelita Albert Einstein

   É um centro de excelência no atendimento ao paciente oncológico, reunindo toda cadeia de atendimento: da prevenção e diagnóstico às diversas modalidades de tratamento, além de oferecer uma ampla gama de serviços para ajudar no enfrentamento de doenças, com o objetivo de minimizar seus impactos e maximizar a qualidade de vida, aliado com um atendimento integral e humanizado.

4) Centro de Oncologia e Hematologia da Beneficência Portuguesa

   O Centro de Oncologia e Hematologia da BP oferece cuidado integrado de saúde para pacientes com câncer e doenças hematológicas, permeando todos os serviços oferecidos pela instituição, desde prevenção, diagnóstico e tratamentos.

5) A.C. Camargo Cancer Center

   Referência internacional, o A.C.Camargo Cancer Center é um centro integrado de diagnóstico, tratamento, ensino e pesquisa do câncer, além de ser modelo sustentável de atuação social. Há 65 anos, é a principal instituição de ensino em oncologia no país,

responsável pela formação de mais de mil especialistas e residentes, além de mais de 600 mestres e doutores.

Conclusão: Esse setor possui diversos players que rivalizam na concorrência. Em um setor concorrido, o diferencial será o investimento em novas tecnologias que atraiam mais pacientes em busca de diagnósticos e prognósticos mais precisos e de melhores atendimentos.

Poder de barganha dos fornecedores:
Pode-se inferir que o Instituto do Câncer do Estado de São Paulo, por ser um hospital público, depende da verba do governo. Dessa forma, o poder de barganha dos fornecedores é alto, já que o hospital depende quase que 100% de uma única fonte de verba, não tendo escolha de quanto será investido nem por qual instituição.

Poder de barganha dos clientes:
Por se tratar de um serviço público o poder de barganha dos clientes pode ser considerado pequeno, já que muitos pacientes não têm condições de irem a um hospital particular e acabam dependendo de hospitais públicos. Dessa forma, elas dependem do que é oferecido pelos hospitais, mesmo que não sejam de qualidade.

Ameaça de novos entrantes:
Agentes que trazem ameaças : Hospitais particulares , públicos e Startups.

Hoje, a tecnologia de Inteligência Artificial utilizada para fins de diagnóstico, detecção de câncer e classificação de grau de prioridade de tratamento está em alta e muitos falam sobre isso, a tendência do mercado é que cada vez mais Startups e Hospitais privados e públicos invistam nessa tecnologia, se tornando ameaças para o setor e possíveis futuros concorrentes.

Barreiras de entrada: falta de profissionais técnicos na área de tecnologia e banco de dados robustos(imagens e informações) nos quais poucos têm acesso.

Características de empresas que podem trazer ameaça: Boa imagem da marca (autoridade e reconhecimento), forte Know-how e acesso a matérias primas (dados, profissionais...)

Ameaça de produtos substitutos:

Linda lifetech

função : Detectar o câncer de mama

Proposta : consiste em um sensor infravermelho que captura a imagem da mama da paciente e envia para um servidor na nuvem. De lá, com ajuda de inteligência artificial, a imagem é comparada com um banco de dados, que conta com mais de 5 milhões de informações, em busca de indícios de lesões cancerígenas.Quanto mais a ferramenta for utilizada, mais dados ela terá e o diagnóstico será cada vez mais aprimorado

Um potencial serviço substituto seria a Linda Lifetech, uma startup que oferece serviços de diagnóstico de câncer de mama utilizando Inteligência Artificial de baixo custo, que poderia substituir uma grande parte dos pacientes do Hospital Universitário, pois os diagnósticos são mais precisos e eficientes, pois são compatíveis com telemedicina, com o custo sendo um grande ponto que ainda não o torna como solução número 1º sobre diagnóstico de câncer de mama.

Modelo de negócios:

Por ser uma instituição pública, o ICESP recebe verba do Estado de São Paulo, proveniente de dinheiro público, sendo pago por ciclos de atendimentos, já que por existir diferentes tipos de neoplasias e cada tumor não possuir um medicamento específico, o SUS aloca valores definidos para cada tratamento de cada paciente, e cabe aos médicos escolherem os medicamentos e protocolos que utilizarão no tratamento com base em dados científicos e padrões da instituição em que atendem.

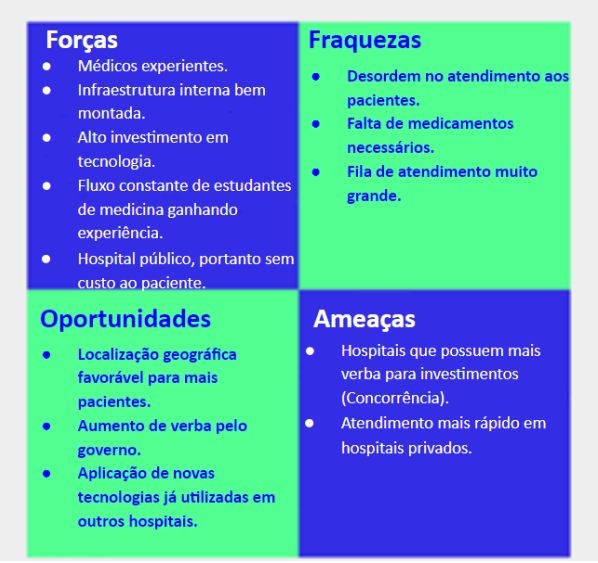
Tendências do setor de saúde:

1. Interoperabilidade em nuvem: O armazenamento de informações coletadas dos pacientes em plataformas nativas em nuvem ajudam o médico que está prestando o atendimento a fazê-lo de uma forma mais eficiente, humanizada e assertiva, já que ele terá disponível todo prontuário médico da pessoa mesmo numa primeira consulta. Além disso, diminuem custos com a criação e manutenção de bancos de dados locais e dificultam vazamento de dados.
2. Inteligência artificial cada vez mais presente: A inteligência artificial (IA) possibilita novos estudos e pesquisas, o aprimoramento de tratamentos e ajuda na detecção de diagnósticos. A análise preditiva, aliada a IA, traz a possibilidade de identificação de um problema de forma mais rápida a partir da antecipação dos riscos pela identificação de dados históricos e padrões.
3. Saúde Mental: Por conta da pandemia muitas pessoas sofreram com problemas relacionados com a saúde mental e emocional. Dessa forma, os tratamentos e tecnologias especializadas nesse tipo de tratamento devem ganhar um destaque ainda maior.

4. Medicina robótica, o uso de robôs para auxiliar nas cirurgias: Com um crescimento estimado em 30% ao ano, segundo a Sociedade Brasileira de Cirurgia Minimamente Invasiva e Robótica (SOBRACIL), o procedimento garante uma abordagem menos invasiva e consideravelmente mais precisa em comparação a outras técnicas. Por isso, é esperado que esse tipo de cirurgia ganhe mais espaço no Brasil e que o avanço dessa tecnologia possa trazer novas melhorias para o setor.
5. Atendimento remoto: O distanciamento social imposto pela pandemia fez com que a telemedicina ganhasse um enorme espaço no setor, possibilitando atendimentos à distância sem que o médico esteja presencialmente com o paciente. Tecnologias como o 5G e Internet das Coisas devem fortalecer o atendimento remoto ainda mais. Vale ressaltar ainda a sua importância social ao garantir ainda um avanço democrático da saúde no país ao possibilitar que pacientes e médicos fora dos grandes centros possam usufruir deste meio.
6. Crescimento das Health Techs: Com tantas transformações digitais demandadas pelas mudanças atuais, a força do mercado de health tech contribui significativamente para otimizar estratégias, por meio de soluções focadas no atendimento mais acessível e seguro das redes de saúde. Esse crescimento está atrelado à procura de hospitais, clínicas e laboratórios que recorrem às empresas de tecnologia para criar soluções que atuam na prevenção, detecção e tratamento de doenças e otimizar suas estratégias de gestão.

4.1.2. Análise SWOT

A análise SWOT tem como objetivo ter uma visão externa e interna do negócio. Desse modo, a matriz é organizada em quatro quadrantes que levam em conta o ambiente externo e interno, que são divididos em fatores internos controláveis e fatores externos incontroláveis. Essa forma de abordagem contribui para o fortalecimento dos pontos fortes e no amadurecimento dos pontos fracos, além de prevenir possíveis danos.



4.1.3. Planejamento Geral da Solução

a) quais os dados disponíveis (fonte e conteúdo - exemplo: dados da área de Compras da empresa descrevendo seus fornecedores)

Dados dos prontuários de pacientes com câncer de mama do ICESP. Esses dados descrevem informações pessoais das pacientes e informações a respeito do câncer.

b) Qual a solução proposta (pode ser um resumo do texto da seção 2.2)

O objetivo principal é a criação de um modelo preditivo a partir de coortes de pacientes acompanhadas em projetos de pesquisa do Instituto do Câncer do Estado de São Paulo/Faculdade de Medicina da Universidade de São Paulo.

A solução proposta do projeto refere-se a passar o risco do paciente de acordo com o tempo de sobrevida. O modelo irá fornecer uma *classificação* de acordo com o *grau de prioridade e urgência* da paciente com câncer de mama a partir de um sistema de cores que identificam risco alto ou baixo.

c) qual o tipo de tarefa (regressão ou classificação)

O tipo de tarefa será de classificação, ou seja, prever qual o risco do paciente e colocá-lo em uma classificação por grupos (risco alto ou baixo).

d) como a solução proposta deverá ser utilizada

A solução deverá ser utilizada como uma ferramenta de pesquisa para auxiliar o médico a tomar a melhor decisão de tratamento e acompanhamento. A partir da cor, o médico saberá a melhor abordagem para cada paciente.

e) quais os benefícios trazidos pela solução proposta

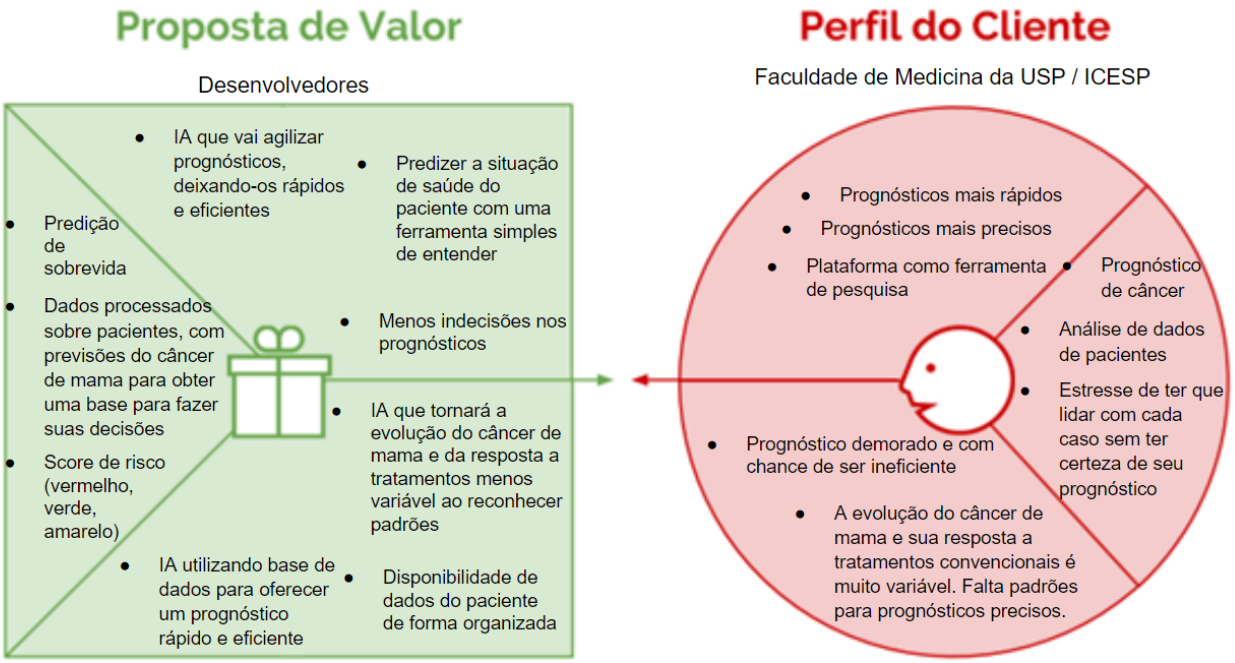
O benefício principal será a construção de uma plataforma de análise de coortes de pacientes assistidos no Instituto do Câncer como ferramenta para pesquisa. A solução fará com que os prognósticos sejam mais objetivos e assertivos de acordo com o risco de cada paciente.

f) qual será o critério de sucesso e qual medida será utilizada para o avaliar

O critério de sucesso será uma assertividade de 80% e para avaliá-la serão utilizadas pacientes já do banco de dados, já que sabemos o resultado previamente e poderá ser comparado com o resultado do modelo preditivo.

4.1.4. Value Proposition Canvas

O Canvas da proposta de valor serve para ajudar a criar e organizar produtos/serviços que se alinhem com o que seu cliente realmente valoriza e precisa.



4.1.5. Matriz de Riscos

A matriz de risco é uma forma usual de se avaliar o risco. Uma matriz de risco é uma representação da combinação da probabilidade de ocorrer um evento associando a esta probabilidade a consequência caso o evento ocorra.

Ameaças

Probabilidade	Muito Baixo	Baixo	Moderado	Alto	Muito Alto
90%					
70%					Pouco tempo de desenvolvimento
50%				Falta de conhecimento técnico	
30%				Desalinhamento/ desentendimento da equipe	IA possuir viés negativo
10%				Recusa dos médicos em utilizarem a IA	Dados errados/escasso

Impacto

Oportunidades

Probabilidade	Muito Alto	Alto	Moderado	Baixo	Muito Baixo
90%		Oportunidade de aprender novas tecnologias como I.A			
70%		Conhecer uma nova área para ter interesse em seguir como carreira			
50%	IA ser espalhada para outros hospitais públicos	Diagnósticos rápidos			
30%	Referência para criação de outras IA			Criar empregos de manutenção da AI	
10%	Diminuir a taxa de mortalidade do câncer de mama				

Impacto

4.1.6. Personas

NOME: João Dias Pinto Filho

IDADE: 43 anos

GÊNERO: Masculino

OCUPAÇÃO: Oncologista

“O melhor médico é aquele que mais esperança inspira”

João é um oncologista do Instituto do Câncer do Estado de São Paulo especializado em câncer de mama. Calmo e atencioso, quer sempre ajudar o máximo de pacientes possível. Contudo, em todos os seus anos como oncologista presenciou muitas de suas pacientes vindo a óbito por causa do câncer, já que sua evolução e respostas a tratamentos são muito variáveis. Assim, João precisa de um modelo preditivo que seja fácil de utilizar e que de alguma forma encontre padrões nessa variabilidade para ajudá-lo a decidir a melhor abordagem no tratamento de suas pacientes de acordo com o risco de cada uma. No fim, ele terá seu trabalho facilitado e poderá atender ainda mais pacientes de forma precisa e eficaz.

NOME: Ana Maria Gabriela de Jesus

IDADE: 33 anos

GÊNERO: Feminino

OCUPAÇÃO: Marceneira

"Não deixe que as pessoas te façam desistir daquilo que você mais quer na vida. Acredite. Lute. Conquiste. E acima de tudo, seja feliz!"

Ana Maria vive na cidade de São Paulo e trabalha como marceneira. Sua renda não é tão elevada, então ela acaba dependendo do atendimento público quando tem algum problema de saúde. Recentemente, ela descobriu que tinha câncer de mama, mesmo mantendo hábitos saudáveis, e por não confiar 100% no sistema público ela tem grande receio em como será seu tratamento e em relação ao verdadeiro risco de seu câncer. Para tranquilizar Ana Maria, o modelo preditivo irá, através do médico, garantir um prognóstico e tratamento eficientes além de mostrar o possível risco do câncer dela. Assim, ela terá um atendimento de qualidade e personalizado, e saberá explicitamente o risco da doença.

4.1.7. Jornadas do Usuário

*OBS: O score de risco será para definir um **prognóstico** mais preciso. Na jornada de usuário o médico fará um prognóstico a partir do modelo, e não um diagnóstico.

4.2. Compreensão dos Dados

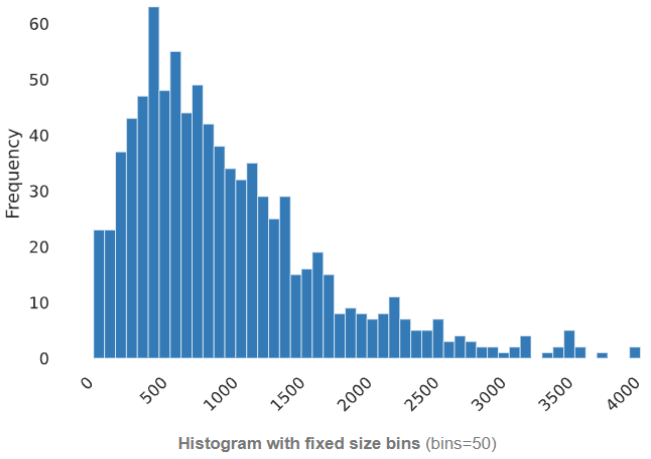
1. **Descreva os dados a serem utilizados (disponibilizados pelo cliente e outros se tiverem sido incluídos), detalhando a fonte, o formato (CSV, XLSX, banco de dados, etc.), o conteúdo e o tamanho.**

Os dados, no formato CSV, são dos prontuários de pacientes com câncer de mama do ICESP. Esses dados descrevem informações pessoais das pacientes (como escolaridade, IMC, raça, câncer na família, etc.) e informações a respeito do câncer (como subtipo, recidiva, regime de tratamento, etc.). A tabela possui aproximadamente 63500 linhas, com 4132 pacientes (linhas únicas). As pacientes são identificadas pela primeira coluna (record_id), que nesse caso foi alterada para proteger a identificação. Contudo, muitos registros estão vazios, assim, mesmo tendo uma grande diversidade de informações, muitas acabam se tornando inutilizáveis pela falta de informação. Dessa forma, uma análise aprofundada dos dados é necessária para saber quais estão completos e quais são relevantes para, então, dividi-los em subconjuntos e retirar dados nulos.

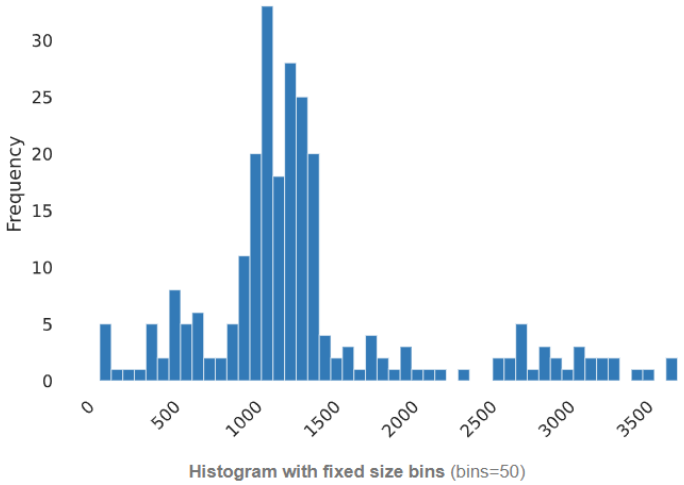
2. **Descrição estatística básica dos dados, principalmente dos atributos de interesse, com inclusão de visualizações gráficas e como essas análises embasam suas hipóteses.**

No primeiro momento, foram escolhidos três atributos para análise: o tempo decorrido entre o diagnóstico e a última vez que houve contato com o paciente (follow_up_days), tipos de tratamento (treatment) e última informação que se tem em relação a óbito ou alta (ultinfo). A primeira análise foi feita com a relação entre follow_up_days e utlinfo, abaixo seguem os gráficos:

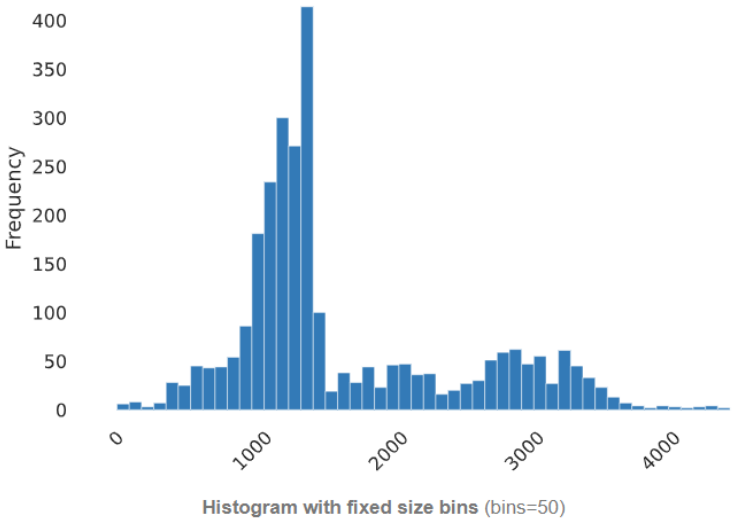
follow_up_days de pessoas que vieram a óbito, por câncer (Média: 1045 dias)



follow_up_days de pessoas que vivas, com câncer (Média: 1387 dias)



follow_up_days de pessoas vivas, sem outras especificações (Média: 1668 dias)



Essa análise embasa a hipótese de que algumas pacientes que vieram a óbito não terminaram ou começaram um tratamento e, por isso, a média de dias é menor do que pacientes que tiveram alta (1045 dias contra 1387 dias para vivas com câncer e 1668 dias para vivas sem câncer).

A outra análise feita foi a relação entre o utlinfo e o treatment. Abaixo estão os gráficos e uma legenda para os valores:

Legenda:

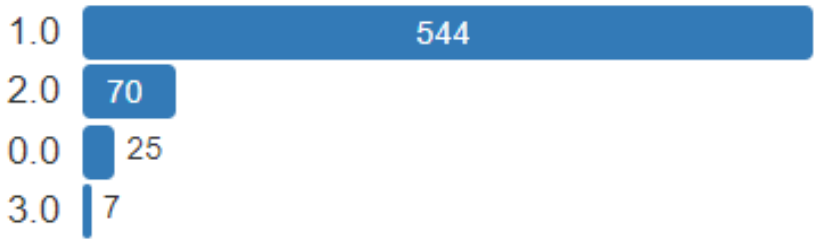
0, Não fez quimioterapia

1, Terapia Adjuvante (realizada depois de uma cirurgia ou radioterapia)

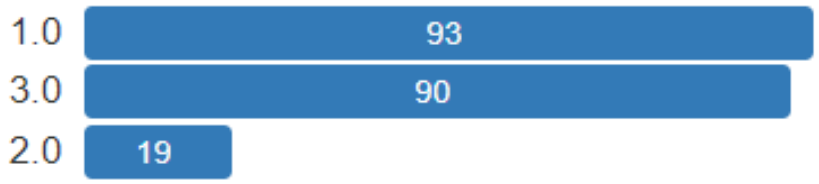
2, Terapia Neoadjuvante (realizada antes de uma cirurgia ou radioterapia)

3, Paliativo (realizada em pacientes em um estado mais grave)

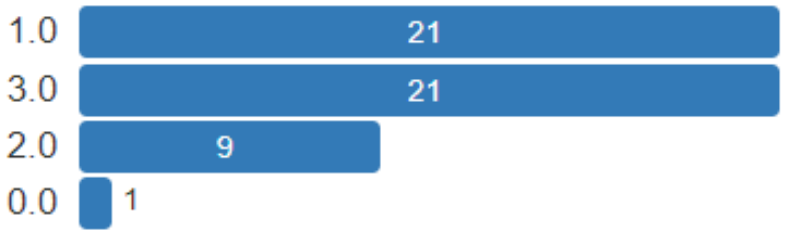
treatment de pessoas vivas, sem outras especificações (23.3% respondido)



treatment de pessoas que vieram a óbito, por câncer (23.2% respondido)



treatment de pessoas vivas, com câncer (20.7% respondido)



Pelos gráficos é possível notar que pacientes que vivem sem câncer acabam realizando mais a terapia adjuvante enquanto que os outros grupos possuem duas terapias com praticamente o mesmo número de casos.

3. Descrição da predição desejada (“target”), identificando sua natureza (binária, contínua, etc.)

Deseja-se uma predição binária (dois tipos de classificação) em que a saída do modelo apresente corretamente uma das duas classificações (baixa e alta) a partir de um esquema de cores. Se o risco da nova paciente for baixo, mostrar como output a cor verde e se o risco for alto, mostrar a cor vermelha.

4.3. Preparação dos Dados

4.3.1 Escolha das Features

1) Remoção de features com poucos dados

Foi decidido remover as features com poucos dados, pois supomos que aquelas com poucas informações não terão um impacto significativo para o modelo. Inicialmente, foi escolhido o número 100 como número de dados mínimo para cada feature. Contudo, a feature “meta04” foi removida e as features “meta01”, “meta02” e “meta03” foram mantidas. Assim, o número foi diminuído para 68 para manter todas as features referentes a metástase.

```
# Remoção das colunas com menos de 68 dados preenchidos
min_number = 68
less_data = data.drop(data.columns[data.notnull().sum() < min_number], axis=1)
```

2) Remoção de features com pouca influência

Após uma análise do significado de cada feature, pesquisando tanto no dicionário do banco de dados quanto na internet, e das respostas de cada paciente, decidimos quais features poderiam não ter um impacto nos resultados e que, portanto, não seriam utilizadas no modelo preditivo.

Aquelas que possuíam registros para uma única resposta (ou a grande maioria dos registros) foram retiradas pois supomos que um único valor não teria impacto no modelo. As demais, supomos que não teriam grande impacto na evolução e gravidade do câncer de mama e, por isso, não seriam importantes para o projeto.

Ao fim, ficamos com **32** features, as quais estão abaixo :

```
record_id                 er_percentage
dob                       follow_up_days
pregnancy_history         ultinfo
menarche                  follow_up_days_recidive
period                    desctopo
bmi                       descmorfo
hormone_therapy           tumor_stage
antec_fam_cancer_mama     meta01
tobaco                    meta02
alcohol                   meta03
birads                    meta04
primary_diganosis         treatment
histological_grade        hormone_therapy_yn
grau_hist                 data_2
tumor_subtype             weight
progesteron_perct         height
```

4.3.2 Manipulação dos Dados

1) Criação de Features

A partir dos registros de data de nascimento e da data da primeira consulta decidimos derivar um novo atributo, *"diag_age"* (idade da paciente no diagnóstico), pois acreditamos que esse dado possa ter um impacto, pois supomos que, em geral, quanto maior a idade mais frágil é a saúde da paciente. Por isso, deveria ser uma coluna própria. Abaixo está o código que cria esse atributo:

```
consulta = final_data.loc[:,['record_id', 'data_2']]
consulta_sorted = consulta.sort_values(['record_id', 'data_2'], ascending = [True, True])

#data de nascimento e de consulta de string para formato datetime
manipulated_data['dob'] = pd.to_datetime(manipulated_data['dob'])
consulta_sorted['data_2'] = pd.to_datetime(consulta_sorted['data_2'])

total_ids = manipulated_data.record_id
actual_id = 54

diag_age = []

#criação de uma coluna "ano de nascimento" a partir da data de nascimento
manipulated_data['year_birth'] = pd.DatetimeIndex(manipulated_data['dob']).year

#criação de uma coluna "ano de consultas" a partir das datas de consulta
manipulated_data['year_consul'] = pd.DatetimeIndex(consulta_sorted['data_2']).year

for i in range(0, len(total_ids)):
    data_birth_year = manipulated_data['year_birth'][i]
    data_consul_year = manipulated_data['year_consul'][i]

    #cálculo da idade a partir das colunas ano de nascimento e de consultas
    age = data_consul_year - data_birth_year

    diag_age.append(age)

#adição da coluna na 5ª posição
manipulated_data.insert(4, 'diag_age', diag_age)
```

2) Transformação de Features categóricas em Features numéricas

A feature "antec_fam_cancer_mama", que representa se a paciente teve antecedente de câncer de mama na família, era uma feature categórica. Como o modelo não aceita string (texto) decidimos transformar os valores "Sim" e "Não" para números, respectivamente 2 e 1.

```
manipulated_data['antec_fam_cancer_mama'] = manipulated_data['antec_fam_cancer_mama'].replace(["Não", "Sim"], [1, 2])
```

```
63332    1.0
63333    NaN
63334    NaN
63335    NaN
63336    NaN
63337    NaN
63338    NaN
63339    NaN
63340    NaN
63341    NaN
63342    1.0
63343    NaN
63344    NaN
63345    NaN
63346    NaN
63347    NaN
63348    2.0
```

3) Normalização de Features (Divisão dos valores em grupos)

Algumas features possuíam valores muito acima das outras e isso poderia fazer o modelo descartar features com valores mais baixos. Assim, decidimos agrupar os dados em classificações para aproximar o intervalo de valores com as outras features.

Acreditamos que o bmi (imc) tem uma influência maior por causa da sua classificação, já que o que afetará a saúde da paciente é o grupo em que se encaixa. Por isso, decidimos classificar os dados da feature "bmi" em:

bmi menor que 18,5 → **Peso 1** (Abaixo do peso)

bmi maior ou igual a 18,5 e menor que 24,9 → **Peso 0** (Normal)

bmi maior ou igual a 25 e menor que 29,9 → **Peso 2** (Acima do peso)

bmi maior ou igual a 30 e menor que 34,9 → **Peso 3** (Obeso)

bmi maior ou igual a 35 → **Peso 4** (Extremamente Obeso)

Os **pesos** representam, a partir da nossa suposição, uma influência maior na evolução do câncer. Isso acontece devido às consequências que a obesidade gera para o corpo como por exemplo, o aumento de gordura que estimulam hormônios prejudiciais ao tratamento do câncer de mama.

Para tornar o resultado mais eficaz, decidimos agrupar as idades da primeira menstruação em grupos, já que pessoas na mesma faixa etária provavelmente terão comportamentos parecidos, tendo mais impacto para o modelo:

Infância - até os 9 anos → **Peso 3**

Pré-adolescência – dos 10 aos 14 anos → **Peso 2**

Adolescência – dos 15 aos 19 anos completos → **Peso 1**

Os **pesos** para cada grupo estão na ordem decrescente (infância com maior peso), pois, a partir de pesquisas na internet, descobrimos que quanto mais cedo a menstruação maior o risco de desenvolver câncer. Então podemos supor que o impacto na evolução do câncer de mama aumenta quanto mais cedo ocorre a menstruação.

```
#transformação dos valores
cond_list = [manipulated_data['menarche'] >= 15, manipulated_data['menarche'] >= 10, manipulated_data['menarche'] >= 1]
choice_list = [1, 2, 3]

aux = np.select(cond_list, choice_list, manipulated_data['menarche'])
manipulated_data['menarche'] = pd.Series(aux)
manipulated_data['menarche'] = manipulated_data['menarche'].replace(np.nan, 0)
```

bmi	menarche
2.0	0.0
0.0	0.0
0.0	0.0
NaN	2.0
NaN	0.0
NaN	0.0
2.0	0.0

4) Agrupando os dados importantes na primeira linha

Esse código agrupa “recidive” e “follow_up_days_recidive” no primeiro registro do paciente, que ficam juntos com “follow_up_days” e “ultinfo”.

```
store_id = 1338

def fudr(store_id):
  if store_id == manipulated_data.record_id[i]:
    return

  store_id = manipulated_data.record_id[i]
  manipulated_data.follow_up_days_recidive[i-1] = manipulated_data.follow_up_days_recidive[i]

for i in range(1, len(manipulated_data.record_id)):
  fudr(store_id)
```

5) Remoção de Valores em Branco

Ao analisarmos detalhadamente cada variável, encontramos que as variáveis relacionadas a última resposta que obteve de como anda a situação de saúde do paciente (“ultinfo”) e quantos dias se passam desde o diagnóstico do câncer de mama até a última resposta (“follow_up_days”) sempre vinham preenchidas como o primeiro registro do paciente.

Porém, “recidive” e “follow_up_days_recidive” são as únicas variáveis que apresentam dados nulos, pois quando um paciente possui uma recidiva, o campo “recidive” é preenchido com 1.0 e ao “follow_up_days_recidive” é atribuído os dias desde o diagnóstico do câncer de mama até a recidiva, e caso o paciente não tenha recidiva, os dois campos ficam nulos.

Então, decidimos preencher “follow_up_days_recidive” com 0 para representar nulo, pois utilizaremos no final do tratamento uma função que cria uma tabela o primeiro registro de cada ID, tornando o tratamento das outras duas variáveis redundante.

```
manipulated_data['follow_up_days_recidive'] = manipulated_data['follow_up_days_recidive'].fillna(0.0)
```

Ao analisarmos as features descobrimos que muitos dados estão preenchidos em apenas uma linha do id, já que ele não muda ao longo do tempo. Dessa forma, copiamos esses dados constantes para todas as linhas do id para eliminar dados em branco.

Para as features "pregnancy_history" e "period" decidimos mudar os dados que tinham valores 1 para 2 e de 0 para 1. Assim, dados em branco poderiam ser substituídos por 0 e não teriam impacto no resultado.

```
manipulated_data.loc[manipulated_data.pregnancy_history == 1, 'pregnancy_history'] = 2
manipulated_data.loc[manipulated_data.pregnancy_history == 0, 'pregnancy_history'] = 1
manipulated_data['pregnancy_history'] = manipulated_data['pregnancy_history'].replace(np.nan, 0)
```

Para as features "hormone_therapy", "antec_fam_cancer_mama", "tobaco", "alcohol" e "birads" decidimos transformar todos os dados em branco para 0.

Por fim, decidimos juntar todas as linhas de um id em uma única linha mantendo o último dado registrado. Esse último dado se refere aos dados da última consulta, ou seja, o estado mais recente das pacientes. Para obter esses valores, ordenamos as linhas por id e data da consulta, confirmando que o último valor fosse o último registro.

```
#mantém apenas uma linha por id
df_new = manipulated_data_sorted.ffill().drop_duplicates('record_id', keep = 'last', ignore_index = True)
df_new = df_new.drop(['data_2', 'dob'], axis=1)
df_new
```

Tabela manipulada final:

record_id	registro_de_tumores	dados_histopatologicos_mama	dados_antropometricos	age	pregnancy_history	menarche
54	3	3	18	74	2	0
302	1	1	16	66	0	0
710	1	1	28	73	0	0
752	1	1	4	71	0	0
1589	1	1	5	56	0	0
1705	1	1	8	59	0	0
1843	1	1	24	66	0	0
1873	1	1	12	54	0	0
1898	1	1	1	75	0	0
1960	1	1	12	43	0	0
1968	1	1	19	50	0	0
1985	1	1	12	89	0	0
2016	1	1	4	56	0	0
2058	1	1	3	60	0	0
2076	1	1	15	65	0	0
2157	1	1	17	66	0	0
2168	2	1	10	46	0	0
2170	1	1	4	66	0	0
2348	1	1	8	45	0	0
2350	1	1	2	86	0	0
2370	1	1	1	64	0	0
2466	2	1	41	78	0	0
2556	1	1	9	60	0	0
2580	1	1	7	64	0	0
2630	1	1	11	57	0	0
2714	1	1	25	75	0	0

period	bmi	hormone_therapy	antec_fam_cancer_mama	tobaco	alcohol	birads
0	2	0	0	2	2	5
0	4	0	0	0	0	0
0	2	0	0	0	0	0
0	4	0	0	0	0	0
0	0	0	0	0	0	0
0	0	0	0	0	0	0
0	0	0	0	0	0	0
0	2	0	0	0	0	0
0	0	0	0	0	0	0
0	4	0	0	0	0	0
0	0	0	0	0	0	0
0	2	0	0	0	0	0
0	2	0	0	0	0	0
0	0	0	0	0	0	0
0	2	0	0	1	3	0
0	0	0	0	0	0	0
0	0	0	0	0	0	0
0	3	0	0	0	0	0
0	2	0	0	2	2	0
0	2	0	0	0	0	0
0	0	0	0	0	0	0
0	0	0	0	0	0	0
0	2	0	0	0	0	0
0	3	0	0	0	0	0
0	3	0	0	1	2	0
0	4	0	0	0	0	0

Manipulação feita no Google Colab:
https://colab.research.google.com/drive/1Z1BAhA74cEhuBmdwAbGAPUO7BBejBjOH#scrollTo=O9d73slYC3aX&uniqifier=1

4.4. Modelagem

A avaliação dos modelos foi feita no seguinte colab:
https://colab.research.google.com/drive/1qzKLgQs7kjLuCxd7L23xlF-NOccvrBvX

Decidimos testar o modelo para pessoas que estão vivas sem câncer e para pessoas que vieram a óbito por câncer, além de todos juntos, a fim de analisar se os resultados são semelhantes ou se há comportamentos distintos.

Como dados de saída utilizamos a feature “output_os”, que representa o risco da gravidade da paciente de acordo com o número de dias do diagnóstico até o último contato com ela (“follow_up_days”), dividida em High OS (alto risco) e Low OS (baixo risco).

Modelos sem hiperparâmetros

Todos os dados

Os modelos foram escolhidos a partir da sua acurácia e do número de falsos Low OS (falsos positivos), ou seja, o número de erros em predizer High OS como Low OS. Além disso, foram escolhidos modelos de classificação, pois o objetivo é classificar as pacientes a partir de classes e não valores contínuos (regressão). Essas métricas foram escolhidas pois supomos que é pior predizer que o risco da paciente é baixo mas na realidade é alto, já que ela achará

estar saudável, aumentando a chance de óbito. Além disso, a acurácia engloba todos os tipos de acertos e erros, representando a assertividade do modelo de forma completa.

Para analisar os falsos positivos utilizamos uma matriz de confusão para cada modelo e o gráfico abaixo (Gráfico 1.0). A matriz de confusão é uma tabela que mostra as frequências de classificação para cada classe do modelo.

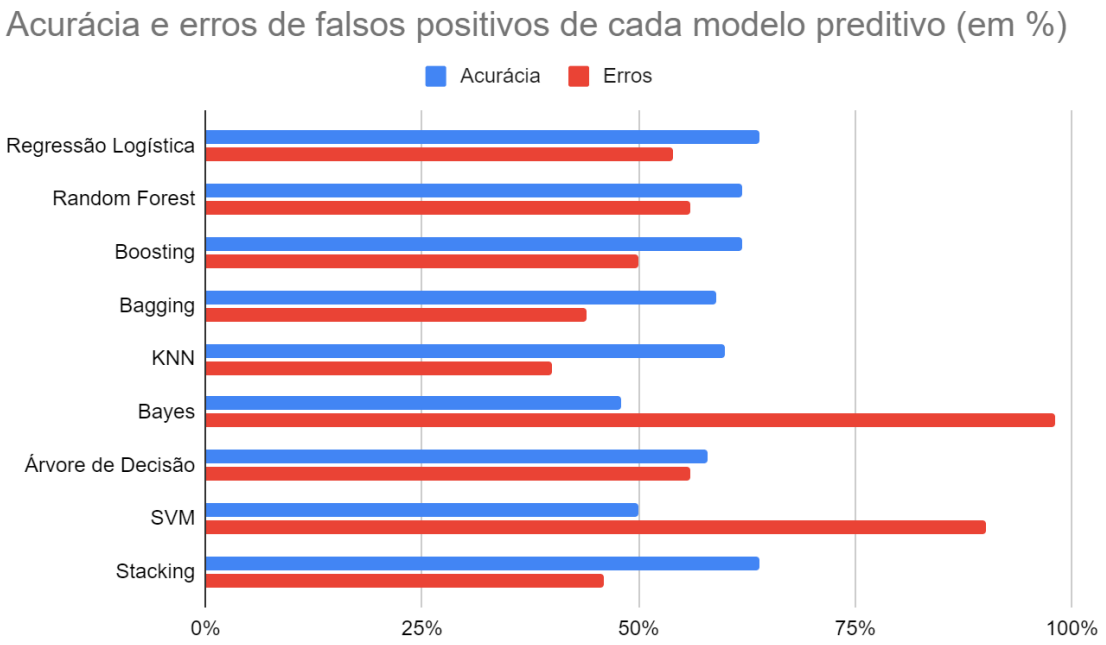


Gráfico 1.0

A partir dessas métricas de avaliação escolhemos os três melhores modelos: Regressão Logística, Random Forest e Boosting.

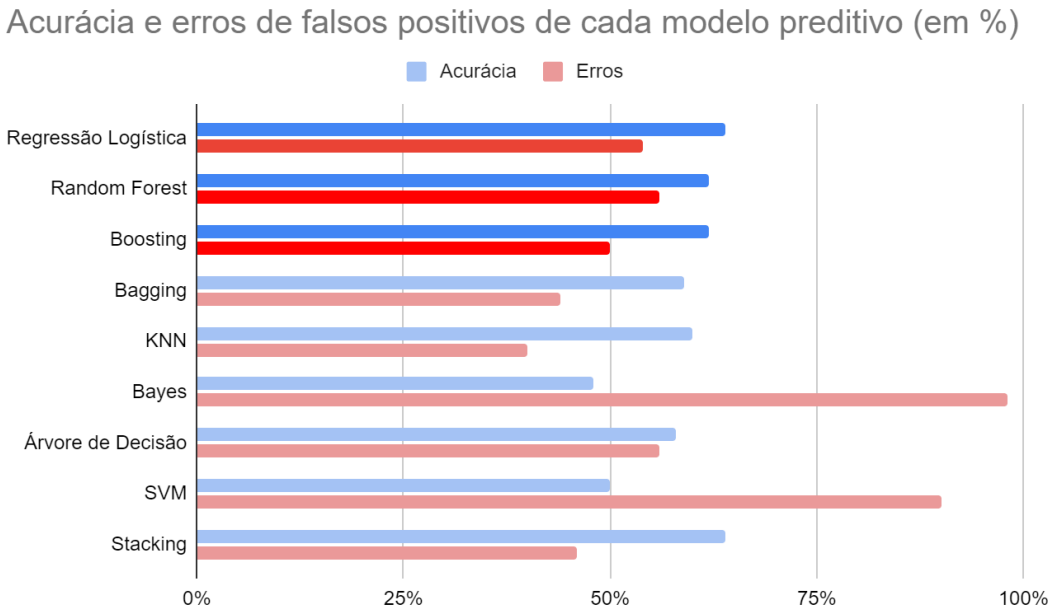
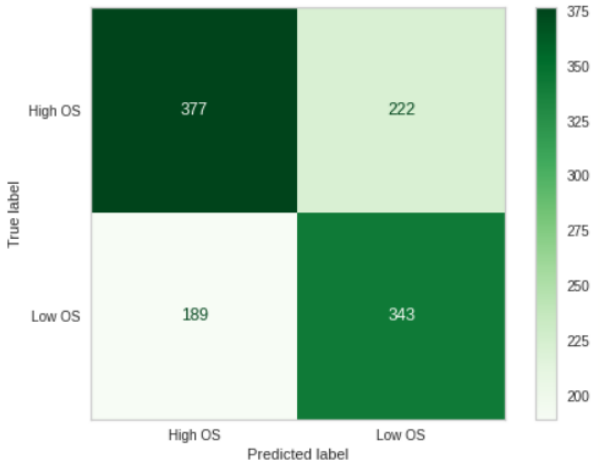


Gráfico 1.1

Regressão Logística (64%)

Resultados:

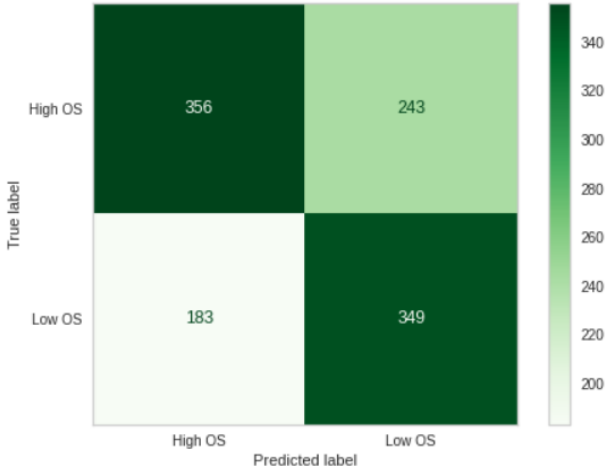
	precision	recall	f1-score	support
High OS	0.67	0.63	0.65	599
Low OS	0.61	0.64	0.63	532
accuracy			0.64	1131
macro avg	0.64	0.64	0.64	1131
weighted avg	0.64	0.64	0.64	1131



Random Forest (62%)

Resultados:

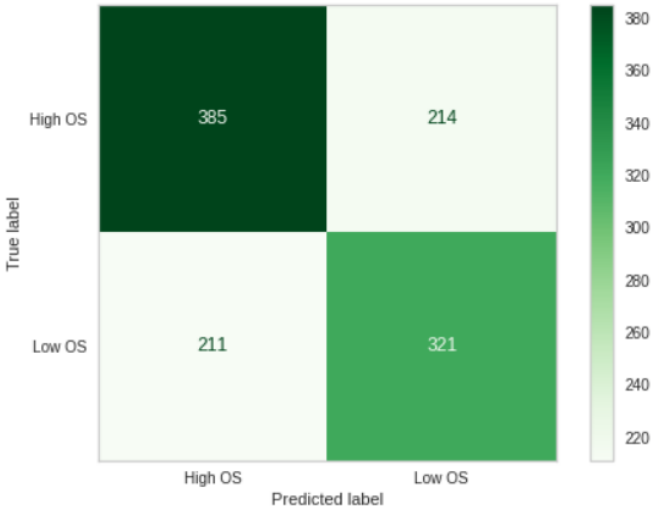
	precision	recall	f1-score	support
High OS	0.66	0.59	0.63	599
Low OS	0.59	0.66	0.62	532
accuracy			0.62	1131
macro avg	0.63	0.63	0.62	1131
weighted avg	0.63	0.62	0.62	1131



Boosting (62%)

Resultados:

	precision	recall	f1-score	support
High OS	0.65	0.64	0.64	599
Low OS	0.60	0.60	0.60	532
accuracy			0.62	1131
macro avg	0.62	0.62	0.62	1131
weighted avg	0.62	0.62	0.62	1131

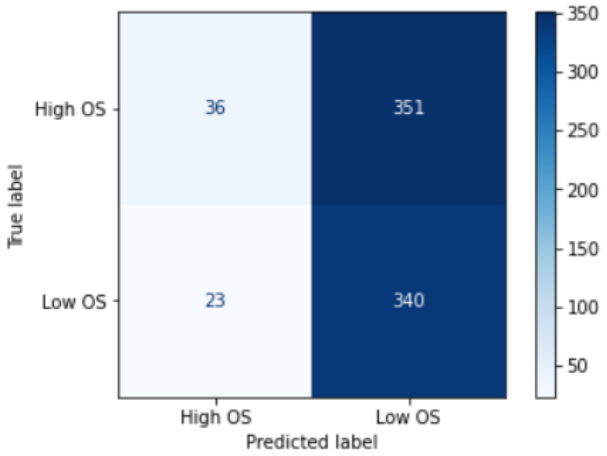


Os outros modelos foram descartados por possuírem acurácias mais baixas ou por terem um erro maior em predizer High OS como Low OS.

Pessoas vivas sem câncer

A partir do treinamento apenas com pessoas vivas foi possível identificar certos padrões. Os primeiros quatro modelos apresentaram maior taxa de erro nos falsos positivos (High OS como Low OS) enquanto os outros cinco apresentaram número de erros mais próximos entre falsos positivos e negativos.

Os quatro modelos tiveram comportamentos parecidos com o visto abaixo



E os outros cinco tiveram comportamentos parecidos com o visto abaixo

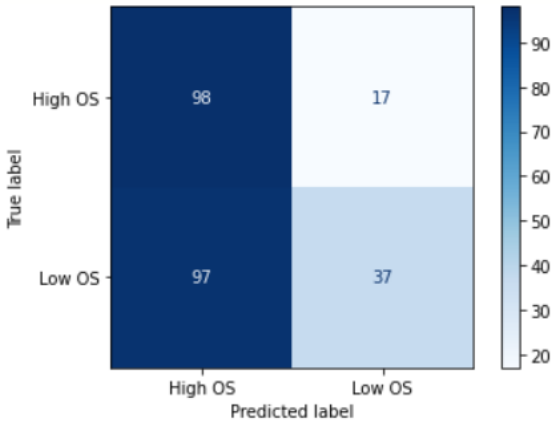
Ao percebermos isso, criamos uma hipótese que o modelo acertou sim várias vezes, porém como ”output os” era relativo se a pessoa estava acima ou abaixo do “follow_up_days”, ele misturava várias categorias de expectativa de vida de várias pessoas.

Um bom exemplo visível é que em “life_data” faz sentido muitos High OS (isto é, que possuíam risco gravíssimo de vida) serem classificados como Low OS, já que várias pessoas sobreviveram a luta contra o câncer, então fica mais fácil de predizer seu risco de vida.

Pessoas que vieram a óbito por câncer

A partir do treinamento apenas com pessoas que vieram a óbito foi possível identificar um padrão. Oito dos nove modelos apresentaram maior taxa de erro nos falsos negativos (Low OS como High OS).

Esses oito modelos tiveram comportamentos parecidos com o visto abaixo



Ao percebermos isso, criamos uma hipótese que o modelo acertou sim várias vezes, porém como "output_os" era relativo se a pessoa estava acima ou abaixo do "follow_up_days", ele misturava várias categorias de expectativa de vida de várias pessoas.

Um bom exemplo visível é que em "death_data" faz sentido vários Low OS serem preditos como High OS, pois está levando em consideração somente dados de pessoas que vieram a óbito, então fica mais fácil de predizer seu risco de vida.

Modelos com hiperparâmetros

Para chegar nos valores para cada hiperparâmetro dos modelos escolhidos anteriormente, já que estes apresentaram melhor acurácia e menor número de erros em falsos positivos, utilizamos o método "tune_model" do PyCaret. O PyCaret é uma biblioteca de aprendizagem de máquina projetada para facilitar a execução de tarefas padrão em um projeto de aprendizagem de máquina, permitindo que os modelos sejam avaliados, comparados e sintonizados em um dado conjunto de dados com apenas algumas linhas de código. O método "tune_model" é responsável por analisar os modelos e descobrir a melhor combinação de hiperparâmetros para cada um.

Regressão Logística (65%)

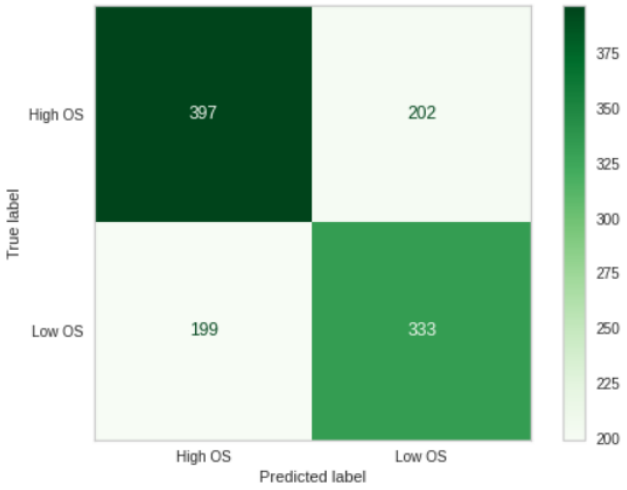
Obtivemos os seguintes hiperparâmetros:

- C=4.359 → Inverso da força de regularização; deve ser um decimal positivo. Como nas máquinas vetoriais de suporte (SVM), valores menores especificam uma regularização mais forte.
- class_weight='balanced' → Pesos associados às classes. Se não forem dadas, todas as classes terão peso 1.

- dual=False → Formulação dupla ou primária. A formulação dupla só é implementada para 'penalty: l2' e 'liblinear solver'. É preferível dual=False quando n_amostras > n_features.
- fit_intercept=True → Especifica se uma constante (conhecida como viés ou interceptação) deve ser adicionada à função de decisão.
- intercept_scaling=1 → x torna-se [x, self.intercept_scaling], ou seja, uma característica "sintética" com valor constante igual a 'intercept_scaling' é anexada ao vetor da instância.
- l1_ratio=None → O parâmetro de mistura Elastic-Net, com 0 < = l1_ratio < = 1.
- max_iter=100 → Número máximo de iterações tomadas para que os solvers converjam.
- multi_class='auto' → Se a opção escolhida for 'ovr', então um problema binário é adequado para cada rótulo. Para 'multinomial', a perda minimizada é a perda multinomial que se encaixa em toda a distribuição de probabilidade, mesmo quando os dados são binários. A opção 'auto' seleciona 'ovr' se os dados forem binários, ou se solver='liblinear', e caso contrário seleciona 'multinomial'.
- n_jobs=None → Número de núcleos de CPU usados na paralelização sobre classes se multi_classe='ovr'.
- penalty='l2' → adicionar um termo de penalidade L2 e é a escolha padrão.
- random_state=None → Usado para embaralhar os dados.
- solver='lbfgs' → Algoritmo a ser usado no problema de otimização.
- tol=0.0001 → Tolerância para critérios de parada.
- verbose=0 → Controla a verbosidade ao treinar e predizer.
- warm_start=False → Quando ajustado para True, reutilizar a solução da chamada anterior para usar como inicialização, caso contrário, apenas apaga a solução anterior.

Resultado:

	precision	recall	f1-score	support
High OS	0.67	0.66	0.66	599
Low OS	0.62	0.63	0.62	532
accuracy			0.65	1131
macro avg	0.64	0.64	0.64	1131
weighted avg	0.65	0.65	0.65	1131



Random Forest (66%)

Obtivemos os seguintes hiperparâmetros:

- bootstrap=False → Se as amostras de bootstrap são usadas na construção de árvores. Se falso, todo o conjunto de dados é usado para construir cada árvore.
- ccp_alpha=0.0 → Parâmetro de complexidade utilizado para a poda de mínimo custo-complexidade. A sub-árvore com a maior complexidade de custo que é menor que 'ccp_alpha' será escolhida.
- class_weight='balanced' → Pesos associados às classes. Se não forem dadas, todas as classes terão peso 1.
- criterion='entropy' → A função para medir a qualidade de uma divisão.
- max_depth=10 → A profundidade máxima da árvore. Se 'None', então os nós são expandidos até que todas as folhas sejam puras ou até que todas as folhas contenham menos do que amostras de 'min_samples_split'.
- max_features='sqrt' → O número de features a considerar quando se procura a melhor divisão. Nesse caso, 'sqrt' será a raiz quadrada do número de features.
- max_leaf_nodes=None → Cresce árvores com max_leaf_nodes de melhor forma. Os melhores nós são definidos como a redução relativa da impureza. Se 'None', então número ilimitado de 'leaf nodes'.
- max_samples=None → Se o bootstrap for True, o número de amostras a serem retiradas de X para treinar cada estimador de base. Se 'None' (padrão), então desenhe amostras X.shape[0].
- min_impurity_decrease=0.001 →Um nó será dividido se esta divisão induzir uma diminuição da impureza maior ou igual a este valor.

- min_samples_leaf=3 → O número mínimo de amostras necessárias para estar em um nó de folha.
- min_weight_fraction_leaf=0.0 → A fração mínima ponderada da soma total dos pesos (de todas as amostras de entrada) necessária para estar em um nó de folha. As amostras têm peso igual quando o 'sample_weight' não é fornecido.
- n_estimators=50 → O número de árvores na floresta.
- n_jobs=None → O número de jobs a serem executados em paralelo.
- oob_score=False → Se irá usar amostras fora do saco para estimar a pontuação de generalização.
- random_state=20 →Controla tanto a aleatoriedade do bootstrapping das amostras utilizadas na construção de árvores quanto a amostragem das características a serem consideradas na busca da melhor divisão em cada nó.
- verbose=0 → Controla a verbosidade ao treinar e predizer.
- warm_start=False → Quando ajustado para True, reutilizar a solução da chamada anterior para usar como inicialização, caso contrário, apenas apaga a solução anterior.

Resultado:

	precision	recall	f1-score	support
High OS	0.65	0.77	0.71	599
Low OS	0.68	0.54	0.60	532
accuracy			0.66	1131
macro avg	0.67	0.66	0.65	1131
weighted avg	0.66	0.66	0.66	1131

Apresentando uma taxa de erro de 34% e apenas 135 erros em falsos positivos, o modelo de Random Forest foi escolhido como o de melhor desempenho.

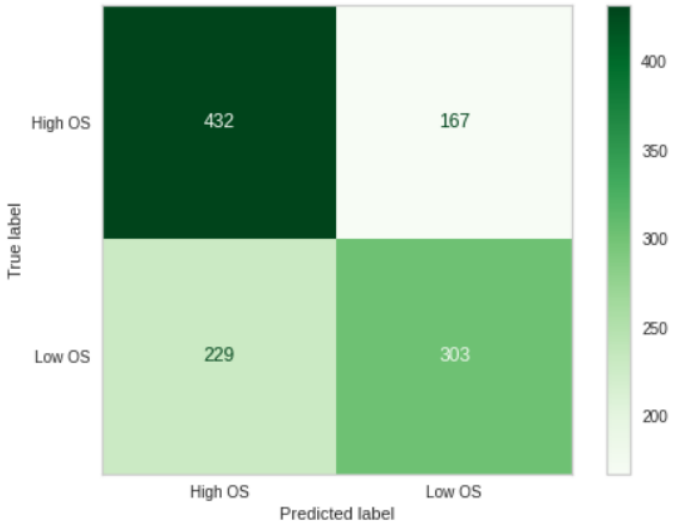
Boosting (65%)

Obtivemos os seguintes hiperparâmetros:

- bagging_fraction=0.9 → seleciona aleatoriamente parte dos dados sem fazer uma nova amostragem.
- bagging_freq=3 → frequência do bagging.
- boosting_type='gbdt' → Tradicional Árvore de Decisão de Gradient Boosting.
- class_weight=None → Pesos associados às classes. Se não forem dadas, todas as classes terão peso 1.
- colsample_bytree=1.0 → Proporção de subamostra de colunas na construção de cada árvore.
- feature_fraction=0.8 → O LightGBM selecionará aleatoriamente um subconjunto de features em cada iteração (árvore), se a fração de características for menor que 1,0. Por exemplo, se você definir para 0,8, o LightGBM selecionará 80% das características antes de treinar cada árvore.
- importance_type='split' → O tipo de importância da feature a ser preenchida em 'feature_importances_'.
- learning_rate=0.15 → Taxa de aprendizagem do Boosting.
- max_depth=-1 → Máxima profundidade de árvore.
- min_child_samples=76 → Número mínimo de dados necessários em uma criança (folha).
- min_child_weight=0.001 → Soma mínima de peso de instância (Hessian) necessária em uma criança (folha).
- min_split_gain=0 → Redução mínima das perdas necessárias para fazer uma nova partição em um nó de folha da árvore.
- n_estimators=10 → Verdadeiro número de iterações boosting realizadas.
- n_jobs=-1 → O número de jobs a serem executados em paralelo.
- num_leaves=10 → Máximo de folhas de árvores para aprendizes de base.
- objective=None → Especificar a tarefa de aprendizagem e o objetivo de aprendizagem correspondente ou uma função objetivo personalizada a ser usada.
- random_state=None → Seed de número aleatório.
- reg_alpha=1 → Termo de regularização L1 sobre pesos.
- reg_lambda=0.15 → Termo de regularização L2 sobre pesos.
- silent='warn'
- subsample=1.0 → Sub-amostra da instância de treinamento.
- subsample_for_bin=200000 → Número de amostras para a construção de silos.
- subsample_freq=0 → Frequência da sub-amostra.

Resultado:

	precision	recall	f1-score	support
High OS	0.65	0.72	0.69	599
Low OS	0.64	0.57	0.60	532
accuracy			0.65	1131
macro avg	0.65	0.65	0.65	1131
weighted avg	0.65	0.65	0.65	1131



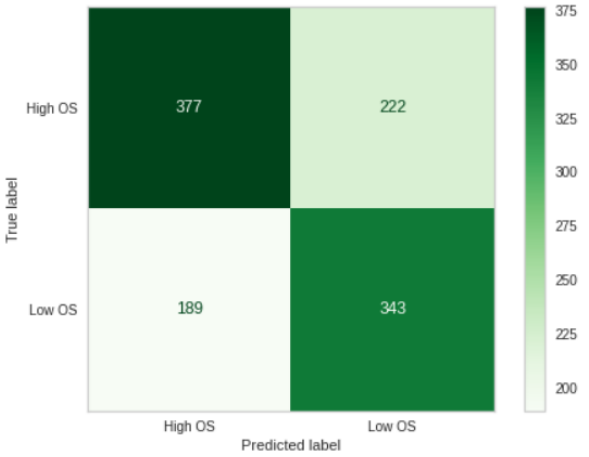
4.5. Avaliação

Modelos sem hiperparâmetros

Para uma avaliação preliminar os resultados podem ser considerados satisfatórios e os modelos escolhidos os mais adequados, já que possuem uma acurácia maior do que 60%, ou seja, acertam a classificação de mais de 60% das pacientes de acordo com o risco de ter maior ou menor sobrevida, e erros de falsos positivos menores do que 250.

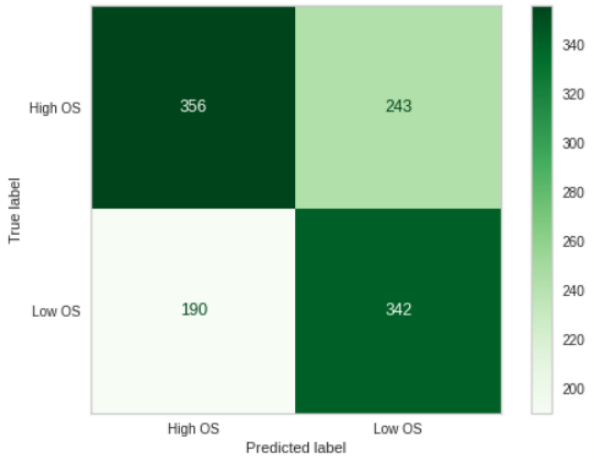
Alguns modelos, como Bayes e Árvore de Decisão por exemplo, tiveram entre 45 - 60% de acurácia e muitos erros em falsos positivos. Assim, eles não eram adequados.

Regressão Logística (64%)



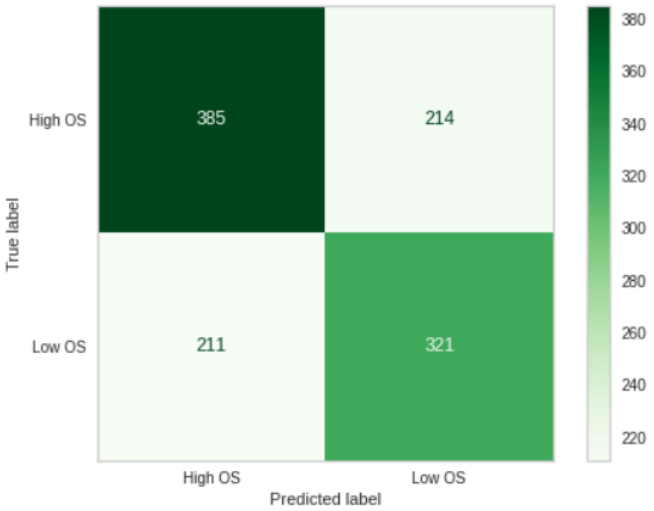
O modelo de Regressão Logística apresentou a terceira menor quantidade de erros em falsos positivos (222) e a melhor acurácia quando comparado aos outros modelos, apresentando uma taxa de erro de 36%.

Random Forest (62%)



O modelo Random Forest apresentou a quarta menor quantidade de erros em falsos positivos (242) e uma boa acurácia quando comparado aos outros modelos, apresentando uma taxa de erro de 38%.

Boosting (62%)



O modelo Boosting apresentou a segunda menor quantidade de erros em falsos positivos (214) e uma boa acurácia quando comparado aos outros modelos, apresentando uma taxa de erro de 38%.

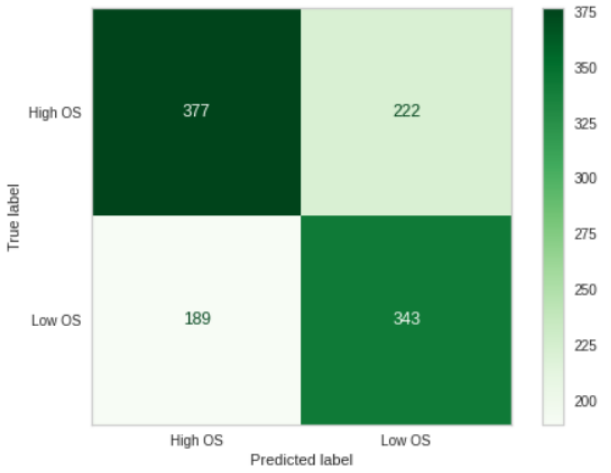
Modelos com hiperparâmetros

Os resultados utilizando hiperparâmetros podem ser considerados satisfatórios e os modelos escolhidos os mais adequados, já que possuem uma acurácia maior ou igual a 64%, ou seja,

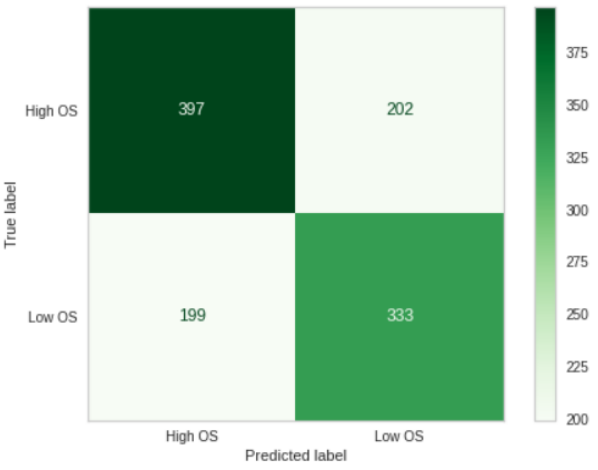
acertam a classificação de mais de 64% das pacientes de acordo com o risco de ter maior ou menor sobrevida, e erros de falsos positivos menores do que 205.

Regressão Logística (65%)

Antes



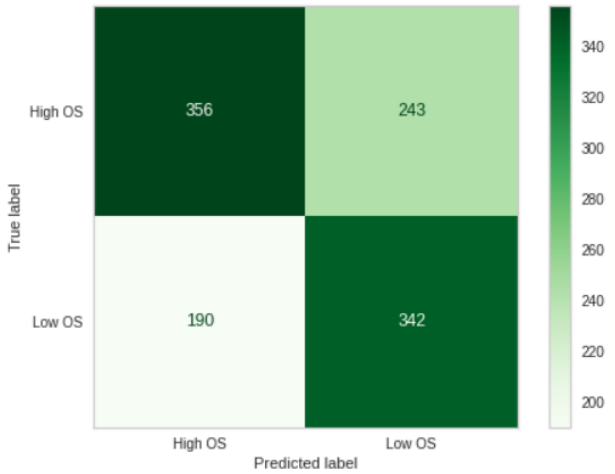
Depois



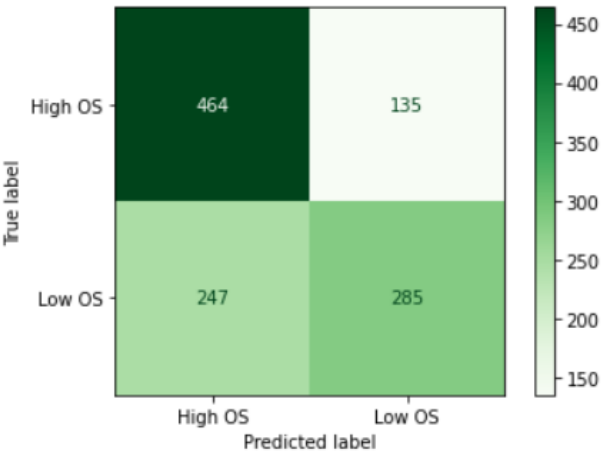
Utilizando os hiperparâmetros conseguimos aumentar a acurácia em 1%, de 64% para 65% (a taxa de erro caiu para 35%), se tornando o segundo melhor nessa métrica, e diminuir os erros de falsos positivos de 222 para 202.

Random Forest (66%)

Antes



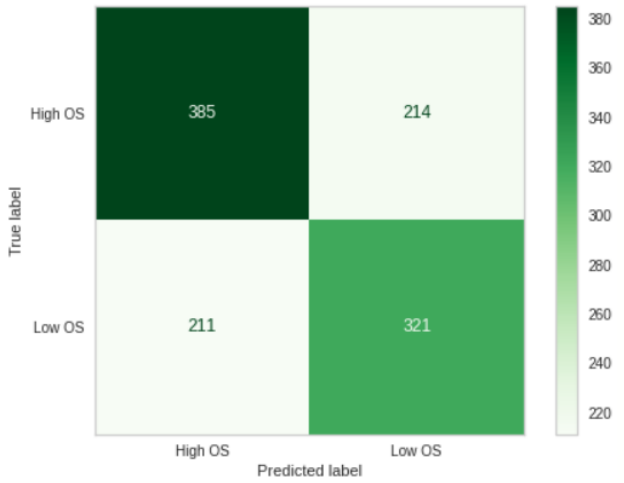
Depois



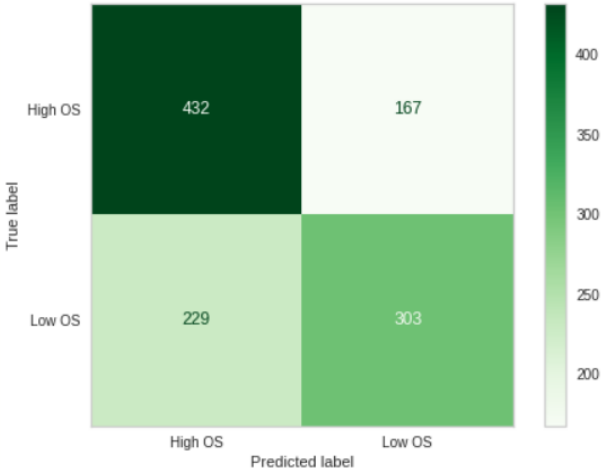
Utilizando os hiperparâmetros conseguimos aumentar a acurácia em 4%, de 62% para 66% (a taxa de erro caiu para 34%), se tornando o melhor nessa métrica, e diminuir os erros de falsos positivos de 243 para 135.

Boosting (65%)

Antes



Depois



Utilizando os hiperparâmetros conseguimos aumentar a acurácia em 3%, de 62% para 65% (a taxa de erro caiu para 36%), se tornando o terceiro melhor nessa métrica, e diminuir os erros de falsos positivos de 214 para 167.

Análise

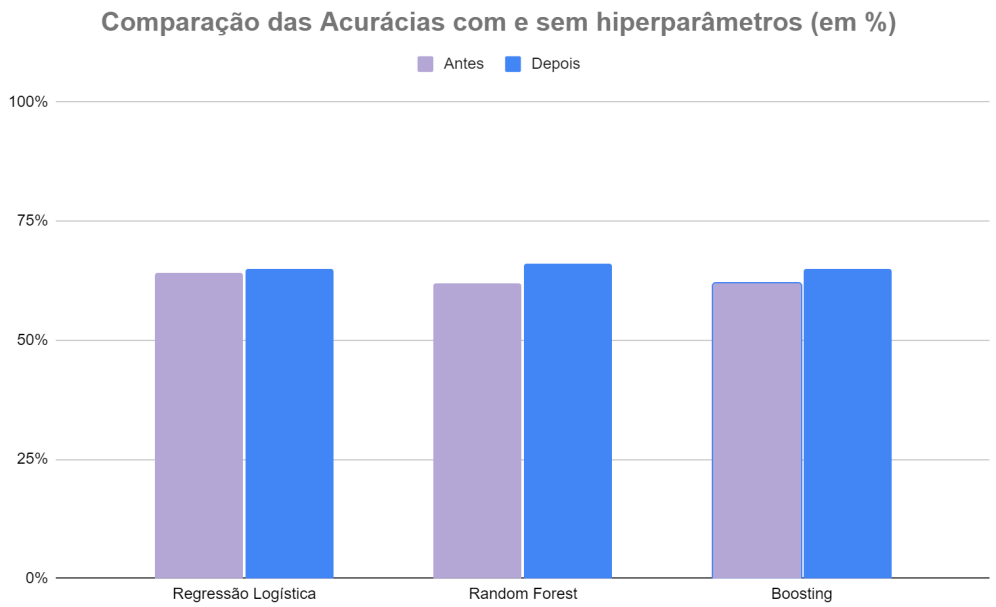


Gráfico 1.2

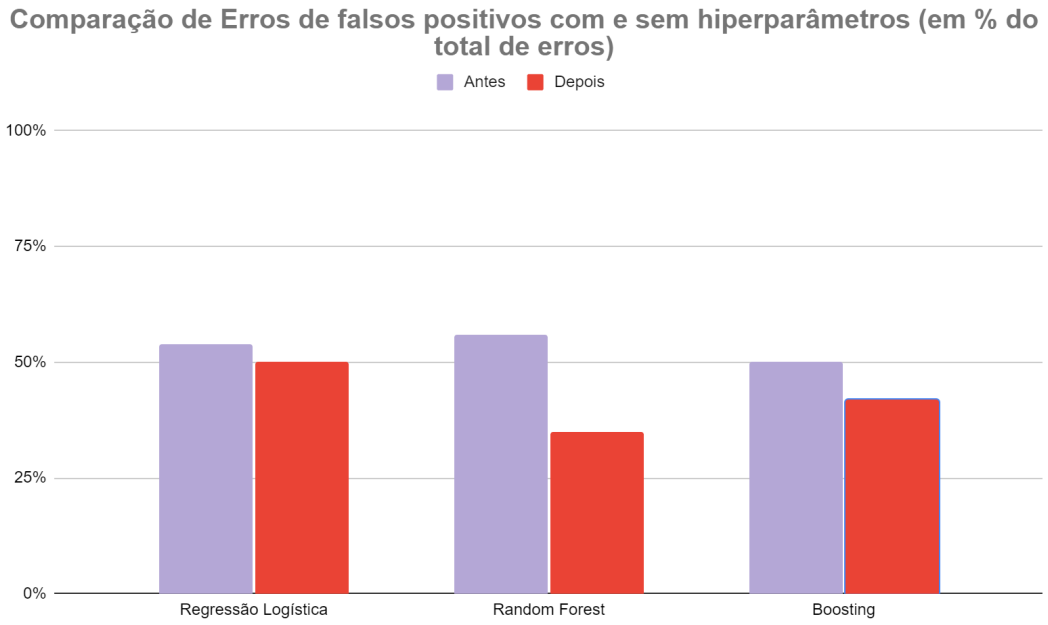


Gráfico 1.3

A partir dos gráficos 1.2 e 1.3 é possível perceber que todos os modelos melhoraram nas métricas escolhidas (acurácia e erros em falsos positivos) a partir do uso de hiperparâmetros. Ou seja, o uso dos mesmos foi fundamental para melhorar ainda mais os modelos.

Além disso, o Random Forest apresentou a maior queda de erros em falsos positivos (predizer High OS como Low OS) e o maior crescimento de acurácia (4%). Apesar desse crescimento, a acurácia final dos três modelos foram próximas, então a quantidade de erros foi o fator determinante para a escolha.

Dessa forma, o Random Forest foi escolhido como modelo final. Ele apresentou a maior acurácia e o menor número de erros em falsos positivos, sendo o melhor em ambas as métricas escolhidas para avaliação.

Falhas

Não foi possível identificar possíveis falhas.

5. Conclusões e Recomendações

Após todo o processo de tratamento dos dados e treinamento dos modelos chegamos ao modelo final, de Random Forest, que apresentou uma acurácia de 66% (taxa de erro de 34%), ou seja, de cada 100 pacientes 66 terão sua predição correta. Além disso, esse modelo apresentou a menor quantidade de erros ao predizer riscos altos como baixos (de todos os erros do modelo, estes somam apenas 35%), caso a ser evitado ao máximo. Apesar de não termos obtido uma acurácia tão elevada, o modelo foi considerado satisfatório levando em consideração a base de dados inicial e os desafios no processo de tratamento de dados, além da questão dos erros que para nós foi a métrica avaliada mais importante.

Por se tratar de um modelo preditivo sobre sobrevida de pacientes com câncer de mama, em que a vida destas está em risco, a decisão final deve ser tomada por alguém experiente no assunto. Nosso modelo serve apenas para auxiliar no processo de decisão do prognóstico e não representa o resultado verdadeiro e absoluto. Isso deve ser compartilhado de forma transparente para que as pacientes saibam como e porque o modelo está sendo usado e que sua assertividade não é 100%.

6. Referências

CHAPMAN, Pete et al. **CRISP-DM 1.0: Step-by-step data mining guide**. SPSS inc, v. 9, n. 13, p. 1-73, 2000.

JENSEN, Kenneth. **Ibm spss modeler crisp-dm guide**. Disponível em: IBM SPSS Modeler CRISP-DM Guide, 2016.

DA SILVA, Leandro Augusto; PERES, Sarajane M.; BOSCARIOLI, Clodis. **Introdução à Mineração de Dados - Com Aplicações em R**: Grupo GEN, 2016. E-book. ISBN 9788595155473. Disponível em: https://integrada.minhabiblioteca.com.br/#/books/9788595155473/. Acesso em: 16 ago. 2022.

FACELI, Katti; LORENA, Ana C.; GAMA, João; AL, et. **Inteligência Artificial - Uma Abordagem de Aprendizado de Máquina**: Grupo GEN, 2021. E-book. ISBN 9788521637509. Disponível em: https://integrada.minhabiblioteca.com.br/#/books/9788521637509/. Acesso em: 23 ago. 2022.

APRENDA como selecionar features para seu modelo de machine learning. 5 mar. 2020. 1 vídeo (31 min 30 s). Publicado pelo canal Stack. Disponível em: https://www.youtube.com/watch?v=4RGT2YRHERY. Acesso em: 23 ago. 2022.

EMANUEL G DE SOUZA. **Entendendo o que é matriz de confusão com python**. 27 mar. 2019. Disponível em: https://medium.com/data-hackers/entendendo-o-que-é-matriz-de-confusão-com-python-114e683ec509. Acesso em: 31 ago. 2022.

REGRESSÃO logística. Disponível em: https://matheusfacure.github.io/2017/02/25/regr-log/. Acesso em: 7 set. 2022.

ENSEMBLE learning - bagging, boosting, and stacking explained in 4 minutes! 29 mar. 2021. 1 vídeo (3 min 46 s). Publicado pelo canal ggnot2. Disponível em: https://www.youtube.com/watch?v=eLt4a8-316E. Acesso em: 7 set. 2022.

A GENTLE introduction to pycaret for machine learning. Disponível em: https://machinelearningmastery.com/pycaret-for-machine-learning/. Acesso em: 20 set. 2022.

PYCARET/BINARY Classification Tutorial Level Beginner - CLF101.ipynb at master · pycaret/pycaret. Disponível em: https://github.com/pycaret/pycaret/blob/master/tutorials/Binary%20Classification%20Tutorial%20Level%20Beginner%20-%20%20CLF101.ipynb. Acesso em: 20 set. 2022.

Anexos

Não há anexos para este documento.