Aplicação e comparação das Técnicas de Regressão Logística, Random Forest e SVM para a Detecção de Doenças Hepáticas.

Luis Felipe Santos Chagas – luis.fschagas@gmail.com
MBA Executivo em Business Analytics
Instituto de Pós-Graduação - IPOG
Salvador, BA, 10 de janeiro de 2024

Resumo
Este artigo apresenta uma análise sobre a aplicação e comparação de técnicas de machine learning para a detecção de doenças hepáticas. Essas doenças representam um desafio substancial para a saúde pública, sendo necessário métodos precisos e ágeis para a identificação delas, visando a melhora dos resultados clínicos dos pacientes. Além da aplicação das técnicas de Regressão Logística, Random Forest e SVM, foram abordadas estratégias de pré-processamento de dados, como a padronização, balanceamento e codificação dos dados, além de métricas de avaliação de performance, como a acurácia e a matriz de confusão. Foi utilizado um conjunto de dados com 583 observações, sendo 416 pacientes sem doenças hepáticas e 167 pacientes com algum tipo de doença no fígado. Ao fim do estudo, constatou-se que a baixa quantidade de dados foi um fator crucial, para este caso, no desempenho dos modelos de machine learning. Mesmo após a aplicação de técnicas de otimização, os modelos não atingiram desempenhos aceitáveis para a utilização médica.

***Palavras-chave**: Detecção de Doenças Hepáticas. Regressão Logística. Random Forest. SVM. Pré-processamento de Dados.*

1. Introdução

As doenças hepáticas representam um grande desafio para sistemas de saúde ao redor do mundo. De acordo com a Organização Mundial da Saúde (OMS), as hepatites são a segunda maior causa de morte entre doenças infecciosas, com cerca de 1,4 milhão de óbitos ao ano, no mundo, ficando atrás apenas da tuberculose (OMS, 2023). Além das hepatites, cirrose, câncer e esteatose hepática são outras condições que podem afetar esse órgão. A detecção precoce dessas doenças é crucial na melhora de resultados clínicos dos pacientes.

Nesse contexto, os avanços em aprendizado de máquina têm proporcionado novas oportunidades para diagnósticos mais precisos e ágeis na detecção de doenças hepáticas. A capacidade de lidar com grandes conjuntos de dados e encontrar padrões complexos são características que chamam atenção da área da saúde.

O objetivo geral deste trabalho é aplicar e comparar diferentes tipos de algoritmos de aprendizado de máquina para a realização de um diagnóstico médico de doenças hepáticas. Serão abordados, além da aplicação dos algoritmos de Machine Learning, toda a etapa de pré-processamento de dados e possíveis maneiras de se realizar otimizações nas performances dos modelos.

2. Fundamentação teórica

2.1. Aprendizado de Máquina

O aprendizado de máquina (do inglês machine learning) é uma área da inteligência artificial (IA) e da ciência de dados, que tem o objetivo de modelar matematicamente a forma como o ser humano aprende, por meio de dados e algoritmos. A utilização do aprendizado de máquina permite uma maior precisão nas análises dos dados e, consequentemente, maior assertividade nas tomadas de decisão.

Através de dados históricos, os algoritmos de machine learning utilizam padrões e informações para realizar previsões. Por exemplo, utilizando dados de pacientes anteriores, o médico pode identificar a probabilidade de um futuro paciente ter ou não a determinada doença.

Os modelos de aprendizado de máquina podem ser classificados em 3 categorias:

- Aprendizado supervisionado: quando os dados são rotulados. Caso o objetivo seja prever um valor numérico, como o preço de uma casa, por exemplo, será um problema de regressão. Já, se a finalidade do algoritmo for prever uma classe, será um problema de classificação. No exemplo anterior, o algoritmo recebe a informação de que os pacientes prévios possuem ou não a doença, portanto, são rotulados. O novo paciente será classificado em ter ou não essa doença.
- Aprendizado não supervisionado: quando o algoritmo não recebe uma saída específica. Ele é treinado para realizar agrupamentos de acordo com os padrões e as características comuns encontradas nos dados. Por exemplo, clientes de uma loja online podem ser agrupados de acordo com seus padrões de compra. Com isso, o departamento de marketing poderá trabalhar estratégias diferentes para cada um desses grupos.
- Aprendizado por reforço: quando o algoritmo aprende pelo método da tentativa e erro, constantemente interagindo com o ambiente, ao invés de depender de dados históricos. Normalmente é utilizado na área da robótica.

2.2. Aprendizado de Máquina

Os algoritmos de machine learning são conjuntos de regras matemáticas e estatísticas que permitem que a máquina aprenda os padrões nos dados. No aprendizado supervisionado, o objetivo final é a previsão de um valor (seja classificação ou regressão). Portanto, esse conjunto de regras será responsável por tornar a previsão o mais próximo do valor real. Para isso, o algoritmo criará um loop, com as etapas de previsão, comparação com o valor real e reajuste de parâmetros, até que sejam alcançados critérios específicos.

O loop será finalizado quando o erro entre o valor real e o previsto não puder mais ser diminuído, chegando à equação final, que é o modelo de machine learning.

Normalmente, para se obter o modelo final, utiliza-se uma separação do conjunto de dados utilizado, em dados de treino e teste. Os dados de treino serão utilizados para o treinamento do modelo. Em outras palavras, são os dados de treino que passarão pelo loop mencionado anteriormente, com o objetivo de se chegar à equação final. Já, os dados de teste são utilizados para a avaliação do modelo. Essa etapa será abordada posteriormente. Aqui, vale a menção para os conceitos de overfitting e

underfitting.

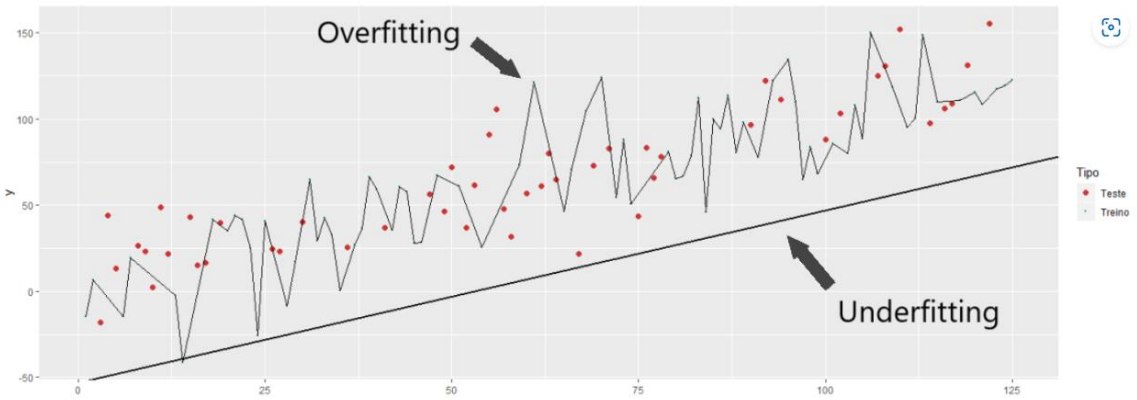

Figura 1 – Overfitting e Underfitting
Fonte: website Didática Tech

- Overfitting: quando o modelo aprende demais sobre os dados de treino. Nesse caso, o modelo criado explica exatamente as variações nesses dados. Nessa situação, o modelo apresentará uma ótima precisão nos dados de treino, porém não terá a mesma performance nos dados de teste. Esse tipo de modelo não possui a capacidade de generalização.
- Underfitting: quando o modelo não consegue encontrar padrões nos dados de treino. Nessa situação, o modelo apresentará um baixo desempenho ainda no treinamento, não sendo adequado para previsões.

Um modelo de aprendizado de máquina aceitável será um meio termo entre esses dois conceitos. Esse modelo aprenderá padrões nos dados de treino, porém sem decorá-los. Isso o torna generalizável, sendo apto a realizar previsões com boa precisão para novos dados.

2.2.1. Regressão Logística

Apesar do nome regressão, esse algoritmo é utilizado para problemas de classificação, em que sua saída (previsão) será valores entre 0 e 1. Matematicamente, a regressão logística é representada pela função sigmoide.

$$f(x) = \frac{1}{1 + e^{-y}}$$

Essa função retorna valores entre 0 e 1, podendo ser entendido, portanto, como uma probabilidade. O termo y da função anterior será uma regressão linear.

$$y = a_1 x_1 + a_2 x_2 + \cdots + a_n x_n + B$$

Nessa equação, os termos representados pela letra “x” são os dados de entrada, também chamados de atributos previsores. Os termos representados pelas letras “a” e “B” são os coeficientes de regressão. O aprendizado do algoritmo de Regressão Logística é feito por meio do ajuste desses coeficientes. A cada loop do treinamento, novos coeficientes são calculados com o objetivo de se ter o menor erro possível entre os valores previstos e os valores reais.

- Vantagens: modelos de regressão logística apresentam boa interpretação, além de uma fácil implementação. Eles demonstram bons desempenhos, principalmente para dados linearmente separáveis.
- Desvantagens: quanto maior a dimensionalidade do problema (quantidade de atributos previsores), mais susceptível a overfitting será o modelo.

2.2.2. Árvore de Decisão

É um algoritmo de estrutura simples, baseado em pontos de decisão. De acordo com o conjunto de dados, esses pontos, ou nós, serão criados e, a partir dele, haverá um ou mais caminhos (ramos). Esses ramos serão percorridos de acordo com a decisão escolhida, até chegar ao valor previsto. Diversas arvores de decisão podem ser construídas, para um mesmo conjunto de dados.

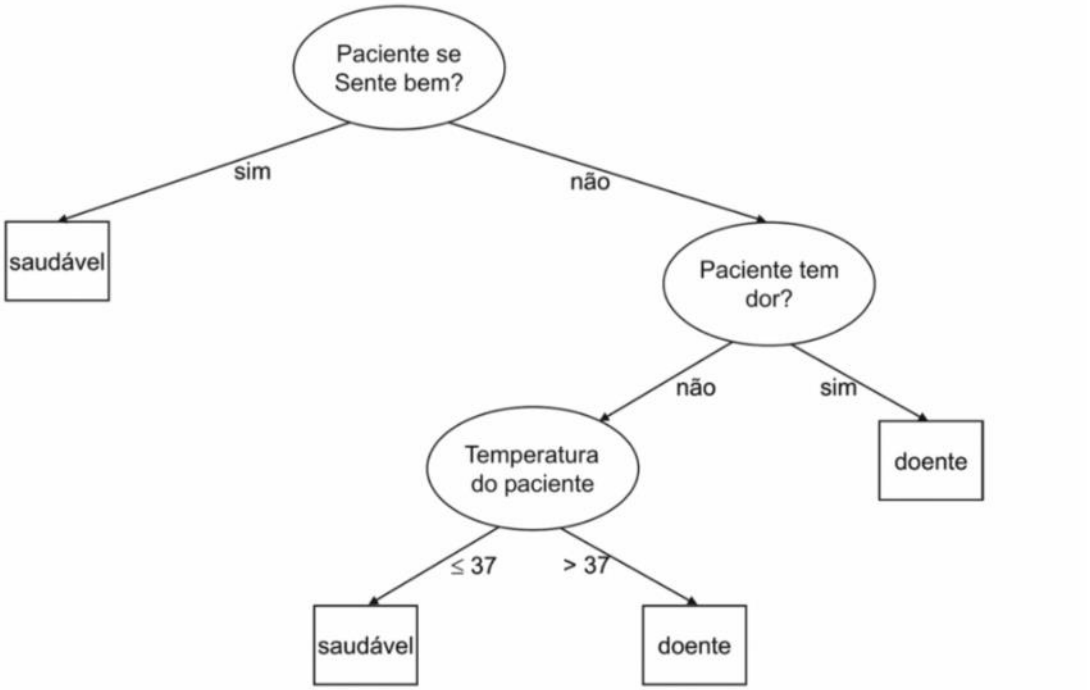

Figura 2: Árvore de Decisão
Fonte: Monard e Baranauskas (2003:60)

A figura 2 ilustra uma árvore de decisão para a classificação de pacientes em saudável e doente. O modelo de machine learning, diferente da equação retornada pela regressão logística, é a própria árvore.

2.2.3. Floresta Aleatória (Random Forest)

É um tipo de método ensemble. Esses métodos combinam o resultado de diferentes modelos, com o intuito de produzir um modelo preditivo mais preciso. A floresta aleatória, como o nome sugere, cria de forma aleatória, diversas árvores de decisão, para um mesmo conjunto de dados, combinando seus resultados.

2.2.4. Support Vector Machine (SVM)

A máquina de vetores de suporte (do inglês support vector machine) é um algoritmo potente, que tem o objetivo de alterar a dimensionalidade dos dados, para que seja possível, por meio de um hiperplano ótimo, separar as classes de interesse. Os conceitos matemáticos do algoritmo SVM são complexos e não serão abordados neste trabalho.

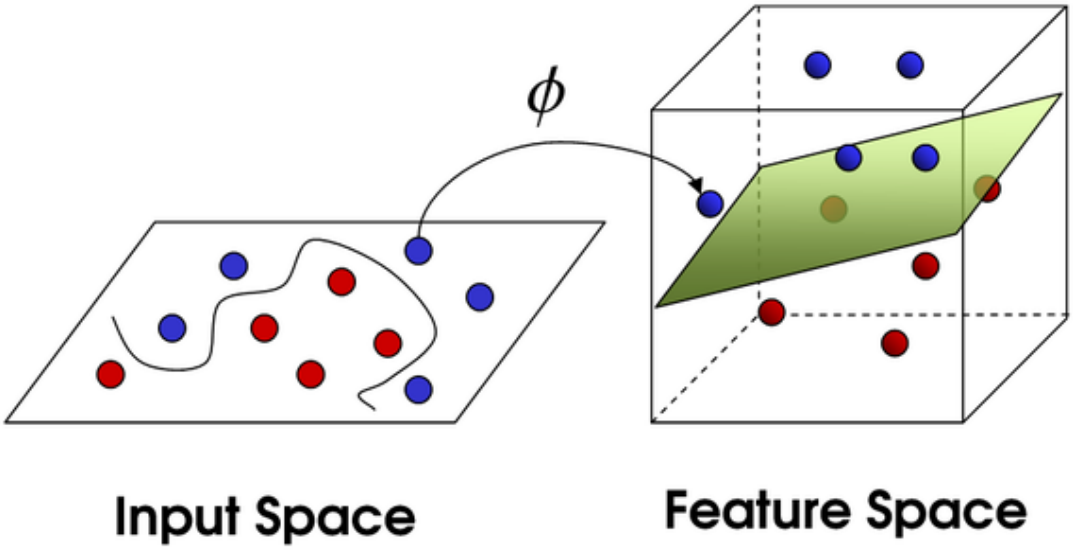

Figura 3: representação gráfica da alteração de dimensionalidade
Fonte: website lapix ufsc

Na figura 3, os dados originais (à direita) estão representadas no em um plano 2-D. Nesse plano, não há a possibilidade de separar os pontos linearmente. Após a alteração de dimensionalidade (agora no plano 3-D) é possível separar esses dados utilizando um plano (não mais uma linha).

- Vantagens: tem a capacidade de lidar com dados não linearmente separáveis, levando os dados para um espaço de alta dimensão, onde há uma maior probabilidade de eles serem linearmente separáveis.
- Desvantagens: devido à sua alta complexidade matemática, o algoritmo perde capacidade de interpretação.

2.3. Pré-processamento dos Dados

Existem algumas transformações que são aplicadas ao conjunto de dados antes do treinamento do modelo de aprendizado de máquina. Algumas delas são necessárias para a aplicação dos algoritmos, outras são aplicadas com o objetivo de tornar o modelo mais simples e eficiente. A seguir, serão descritas algumas etapas mais comuns de pré-processamento de dados.

2.3.1. Padronização dos Dados

Uma das etapas mais comuns de pré-processamento de dados é a aplicação de algum

tipo de padronização nos dados numéricos. Essa técnica é necessária quando existem atributos com diferentes grandezas no conjunto de dados. Durante o treinamento do modelo de machine learning, o algoritmo irá entender que os dados de maior grandeza são mais importantes, gerando assim um viés.

Para evitar esse problema, aplica-se a padronização, com o objetivo de manter todas as variáveis na mesma escala. A seguir, serão descritas algumas técnicas.

- Normalização: a normalização pelo método mínimo-máximo, tem o objetivo de colocar os dados em valores entre 0 e 1 (ou -1 e 1, caso existam valores negativos). Segue a equação para a normalização:

$$X_{norm} = \frac{X - X_{min}}{X_{max} - X_{min}}$$

Sendo:

X: valor original do dado;

X_min: valor mínimo desse dado no conjunto de dados;

X_max: valor máximo desse dado no conjunto de dados.

- Padronização (Standard Scaler): esse método tem o objetivo de padronizar os valores de média e desvio padrão para 0 e 1, respectivamente. O valor padronizado é calculado da seguinte forma:

$$Z = \frac{X - \mu}{S}$$

Sendo:

X: valor original do dado;

μ: média aritmética dos dados;

S: desvio padrão dos dados.

De maneira geral, a normalização é uma boa técnica para quando não se sabe qual a distribuição da variável, ou ainda, quando se sabe que ela não possui uma distribuição normal. Portanto, essa técnica é mais utilizada para algoritmos que não fazem suposição dos dados com relação à sua distribuição. Já a padronização é uma técnica mais efetiva quando a distribuição dos dados é normal.

2.3.2. Codificação da Variáveis

Dados que representam categorias são denominados de variáveis categóricas. Eles podem ser nominais, quando não há uma ordem natural. Por exemplo, feminino e masculino. Já, quando há uma ordem, são denominados de ordinais. Um exemplo desse tipo de dado é a representação de níveis de satisfação de algum serviço.

Uma outra etapa do pré-processamento de dados é a manipulação das variáveis categóricas do conjunto de dados. Para evitar que esses dados cheguem ao algoritmo de machine learning com o tipo de texto, as técnicas de codificação fazem o papel de transformá-los em número, mantendo a ordem das informações, caso exista. A seguir, dois tipos de codificação serão abordados.

- Label Encoding: quando os dados categóricos são transformados em dados numéricos, por exemplo 0 para feminino e 1 para masculino. É possível realizar essa técnica para dados com mais categorias também.
- One Hot Encoding: técnica em que são adicionadas novas colunas de acordo com o número de categorias do dado. O exemplo a seguir ilustra essa técnica:

Nome	Profissão
Francisco Silva	Empresário(a)
Aline Rocha	Médico(a)
Rogério Fernandes	Engenheiro(a)
Larissa Silva	Engenheiro(a)
João Rodrigues	Dentista

Figura 4 – Conjunto de dados sem codificação
Fonte: Tabela produzida pelo autor (2024)

Nome	Empresário(a)	Médico(a)	Engenheiro(a)	Dentista
Francisco Silva	1	0	0	0
Aline Rocha	0	1	0	0
Rogério Fernandes	0	0	1	0
Larissa Silva	0	0	1	0
João Rodrigues	0	0	0	1

Figura 5 – Conjunto de dados com One Hot Encoding
Fonte: Tabela produzida pelo autor (2024)

Nesse exemplo, existem 4 tipos diferentes de profissão, portanto, são criadas 4 colunas, com valores 0 ou 1.

De modo geral, o label encoding é mais utilizado para variáveis categóricas ordinais, pois é natural a representação de ordem por números. Já o one hot encoding é mais utilizado quando não existe essa ordem. Em contrapartida, a aplicação dessa codificação pode ter um custo computacional alto, já que irá aumentar a dimensionalidade do conjunto de dados e, quanto mais categorias, mais novas colunas serão criadas.

2.3.3. Seleção de atributos

Não há nenhum tipo de obrigação no aprendizado de máquina em se utilizar todas as informações existentes no conjunto de dados. A seleção de atributos (do inglês, feature selection) é a etapa de pré-processamento responsável pela escolha dos atributos que irão fazer parte do treinamento do modelo de machine learning.

A seleção de atributos tem como objetivo a melhora da performance e a simplificação dos modelos, reduzindo o custo computacional para o seu treinamento. Além disso,

essa técnica visa a eliminação de:

- atributos redundantes: variáveis altamente correlacionadas entre si, que não agregam valor ao modelo e ainda podem causar viés.
- Atributos irrelevantes: são atributos que não carregam informação útil para o treinamento do modelo. Em geral, colunas de identificação (ID) são atributos irrelevantes e devem ser descartados do modelo.

Uma das técnicas de seleção de atributos utilizado neste trabalho será a Correlation-based feature selection (CFS). Ela leva em consideração a correlação entre as variáveis. Segundo Mark Hall (1999:04), “bons subconjuntos de recursos contêm recursos altamente correlacionados com a classe, mas não correlacionados entre si”. Ele ainda cita que a “seleção de atributos para atividades de classificação em aprendizado de máquina pode ser realizada baseado na correlação entre os atributos”. Em outras palavras, as variáveis preditivas utilizadas no algoritmo de machine learning devem ter alta correlação com a variável alvo, porém baixa correlação entre elas.

A alta correlação com a variável alvo parte do princípio de que, caso essa variável preditiva não tenha uma boa relação com o alvo, a primeira não consegue explicar variações da última, não havendo, portanto, um padrão nos dados.

Já, a baixa correlação entre as variáveis refere-se à atributos redundantes, como já mencionado anteriormente. Uma alta correlação entre duas variáveis preditivas implica dizer que elas carregam a mesma informação, ou, pelo menos, informações muito similares. Caso ambas as variáveis permaneçam no conjunto de dados, o modelo de machine learning será enviesado, devido ao excesso de importância para essa informação.

2.3.4. Balanceamento dos dados

Quando se trabalha em um problema de classificação, uma situação que pode acontecer é o desbalanceamento dos dados. Isso acontece quando a variável alvo apresenta frequências muito diferentes entre as categorias. Esse tipo de situação é comum, principalmente em conjunto de dados que naturalmente, uma classe é mais dominante do que a outra.

Como consequência desse desbalanceamento, o modelo aprenderá mais sobre a classe dominante, sendo menos preciso na definição de padrões da classe com menos frequência. Além disso, esse modelo provavelmente terá uma alta precisão, conseguindo prever as observações da classe dominante, porém, não atingindo o mesmo desempenho para as da classe com menos frequência, mascarando, portanto, o problema anterior. Para contornar essa situação, existem duas possíveis estratégias, o undersampling e o oversampling.

O undersampling, é uma técnica que consiste em diminuir a quantidade de observações da classe dominante, para equilibrar com a quantidade da classe menos dominante. Apesar dessa técnica resolver o problema de desbalanceamento, ela é pouco recomendada para conjuntos que possuam poucos dados.

Já, o oversampling, é uma técnica que irá aumentar a quantidade de observações da classe menos dominante, até atingir o equilíbrio entre elas. Essa técnica tem um custo

computacional maior, porém não haverá perdas de dados, como acontece no undersampling.

Uma forma de se realizar o oversampling, sem duplicação de observação já existentes é utilizando o SMOTE (Synthetic Minority Over-sampling Technique). De acordo com Chawla (2002:328), o SMOTE é uma técnica de oversampling que cria exemplos “sintéticos”, relativamente próximos aos dados originais.

2.4. Avaliação dos modelos

Algumas métricas são utilizadas para a avaliação do desempenho de um modelo de machine learning. Elas são importantes para que seja possível comparar diferentes algoritmos utilizados, além de diferentes modelos de um mesmo algoritmo.

Para a avaliação do modelo, utilizam-se os dados de teste. O modelo já treinado receberá as variáveis preditoras dos dados de teste e tentará prever a variável alvo. As métricas de avaliação irão trabalhar em cima da comparação entre esses valores previstos e os valores reais do atributo alvo dos dados de teste.

2.3.5. Acurácia

A acurácia mede a quantidade de previsões corretas. Para um problema de classificação, se, por exemplo, for utilizado um conjunto de treino com 100 observações e o modelo criado acerte 80 desses casos, então a acurácia desse modelo é de 80%.

2.3.6. Matriz de Confusão

		Valor Predito	
		Sim	**Não**
Real	**Sim**	Verdadeiro Positivo (TP)	Falso Negativo (FN)
	Não	Falso Positivo (FP)	Verdadeiro Negativo (TN)

Figura 6 – Matriz de Confusão Binária
Fonte: website Diego Nogare

A matriz de confusão (do inglês, confusion matrix) é uma tabela que oferece a relação entre os acertos e os erros do modelo. Ela apresenta quatro possibilidades:

- **Verdadeiro positivo:** quando a observação for positiva e a previsão também;
- **Falso positivo:** quando a observação for negativa, porém a previsão for positiva;
- **Falso negativo:** quando a observação for positiva, porém a previsão for negativa;

- **Verdadeiro negativo:** quando a observação for negativa e a previsão também.

Aqui vale uma observação. Os significados de positivo e negativo, para a classificação dos dados, não seguem a literalidade das palavras. Isso quer dizer que não necessariamente a classe positiva representa algo bom ou a classe negativa representa algo ruim, mas sim uma diferença de classes. Essa classificação binária poderia também ser representado pelos valores 0 e 1, por exemplo.

Os verdadeiros, positivo e negativo, são acertos do modelo, enquanto os falsos, positivo e negativo, são os erros. O ideal para um modelo é que sua matriz de confusão seja composta pelo máximo de verdadeiros e o mínimo de falsos.

3. Ferramentas e dados utilizados

Neste trabalho, foi utilizado o software Python e suas principais bibliotecas de manipulação de dados e aprendizado de máquina. A seguir, será feita uma breve descrição das duas bibliotecas mais utilizadas durante o desenvolvimento do trabalho.

- Pandas: biblioteca que apresenta ferramentas para análise e manipulação de dados para a linguagem de programação Python;
- Scikit-learn: biblioteca de aprendizado de máquina para a linguagem Python. Possui ferramentas para modelos de classificação, regressão, clusterização, além de ferramentas para o pré-processamento dos dados.

O conjunto de dados utilizado foi o Indian Liver Patient Dataset (ILPD), doado ao repositório de machine learning da Universidade da Califórnia Irvine (UC Irvine) em 20/05/2012, criado por Ramana, Bendi e Surendra. Foram coletados marcadores bioquímicos de 583 pacientes da Índia, com o objetivo de detectar se o paciente sofre ou não com alguma doença de fígado.

3.1. Conceituação do conjunto de dados

A seguir, seguem alguns conceitos sobre os marcadores bioquímicos utilizados no trabalho.

3.1.1. Colunas TB (Total Bilirubin) e DB (Direct Bilirubin)

A bilirrubina é produzida pela quebra natural de células vermelhas do sangue no corpo. A quantidade bilirrubina é importante para a detecção de doenças do fígado e da vesícula biliar. Pessoas que ingerem bebidas alcóolicas em excesso podem ter altas taxas de bilirrubina no sangue e, por isso, têm maior risco de ter doença no fígado.

Os valores normais de bilirrubina total para adultos são entre 0,2 mg/dL e 1,20 mg/dL. Já, os valores normais de bilirrubina direta para adultos são de até 1,0 mg/dL.

3.1.2. Coluna Alkphos (Alkaline Phosphotase)

A fosfatase alcalina é uma enzima encontrada em maior quantidade nas células dos ductos biliares, que são os canais que conduzem a bile do interior do fígado para o intestino, e nos ossos. Ela está envolvida no metabolismo dos ossos e na regulação de certos compostos no fígado.

O exame da fosfatase alcalina é geralmente utilizado para investigar doenças no fígado ou nos ossos, quando estão presentes sinais e sintomas como dor no abdômen, urina escura, icterícia ou deformações e dor ósseas, por exemplo. Os valores de referência para adultos são de 46 a 120 U/L.

3.1.3. Coluna Sgpt (Alamine Aminotransferase)

A alanina aminotransferase também é uma enzima, encontrada principalmente no fígado, desempenhando um papel fundamental no metabolismo dos aminoácidos. Essa enzima, quando presente na corrente sanguínea pode significar lesões ou danos nas células do fígado. Os valores de referência para esse indicador são de 7 a 56 U/L.

3.1.4. Coluna Sgot (Aspartate Aminotransferase)

O aspartato aminotransferase é uma enzima presente no fígado e que normalmente se encontra elevada quando há lesão do fígado. Porém, por estar localizada mais internamente na célula do fígado, normalmente indica lesões mais crônicas. No entanto, essa enzima também pode estar presente no coração, podendo também indicar infarto ou isquemia.

Os valores acima de 150 U/L geralmente indicam alguma lesão no fígado e acima de 1000 U/L pode indicar hepatite causada pelo uso de medicamentos ou hepatite isquêmica, por exemplo. Por outro lado, os valores diminuídos de AST podem indicar deficiência de vitamina B6 no caso de pessoas que precisam fazer diálise. Os valores de referência são de 5 a 40 U/L, podendo variar de acordo com o laboratório.

3.1.5. Colunas TP (Total Protiens) e ALB (Albumin)

A medida das proteínas totais no sangue reflete o estado nutricional da pessoa. Se os níveis de proteínas totais estiverem alterados, outros testes devem ser feitos para identificar qual a proteína específica que está alterada, para que possa ser feito o diagnóstico correto. A albumina, por sua vez, é um tipo específico de proteína.

Os valores de referência para pessoas com idade igual ou acima de 3 anos são de 6 a 8 g/dL para as proteínas totais e de 3 a 5 g/dL para a albumina.

3.1.6. Coluna A_G_Ratio (Albumin and Globulin Ratio)

A globulina, assim como a albumina também é um tipo específico de proteína. Ambas proteínas são produzidas principalmente no fígado, embora alguns tipos de globulina sejam produzidos pelos glóbulos brancos. Muitas doenças podem prejudicar o equilíbrio entre a albumina e a globulina no sangue. A razão albumina/globulina (razão A / G) é um teste que compara as concentrações destas duas proteínas no sangue.

Os valores de referência são entre 1.1 e 2.5, embora possa variar dependendo do laboratório.

4. Atividades Realizadas

4.1. Análise Descritiva

A seguir, será feita uma breve análise descritiva das variáveis numéricas do conjunto de dados. A tabela a seguir apresenta a coluna “Age”, representando a idade dos pacientes. Vale destacar que as outras colunas foram conceituadas no capítulo anterior. As linhas representam, respectivamente a média, o desvio padrão, o valor mínimo, o primeiro, segundo e terceiro quartil e o valor máximo de cada coluna.

	Age	TB	DB	Alkphos	Sgpt	Sgot	TP	ALB	A_G_Ratio
mean	44.8	3.3	1.5	291.4	81.1	110.4	6.5	3.1	0.9
std	16.2	6.2	2.8	243.6	183.2	289.9	1.1	0.8	0.3
min	4.0	0.4	0.1	63.0	10.0	10.0	2.7	0.9	0.3
25%	33.0	0.8	0.2	175.5	23.0	25.0	5.8	2.6	0.7
50%	45.0	1.0	0.3	208.0	35.0	42.0	6.6	3.1	0.9
75%	58.0	2.6	1.3	298.0	61.0	87.0	7.2	3.8	1.1
max	90.0	75.0	19.7	2110.0	2000.0	4929.0	9.6	5.5	2.8

Figura 7 – Análise descritiva das colunas numéricas
Fonte: tabela produzida pelo autor (2024)

Dessa análise inicial, é possível observar alguns pontos:

- A idade segue aproximadamente uma distribuição normal, pois apresenta a média muito próxima do segundo quartil (coluna “50%”). Além dela, as duas últimas colunas também possuem a mesma característica;
- As colunas “TB” até “Sgot” possuem valores atípicos (outliers). É possível observar isso pela linha de valor máximo, em que essas colunas apresentam valores distantes da média;
- As colunas numéricas apresentam diferentes escalas e, por isso devem ser padronizadas, para a aplicação de algoritmos de machine learning.

Além das colunas numéricas, o conjunto de dados apresenta também duas variáveis categóricas. A primeira, com a informação do gênero do paciente e a última, a variável alvo do problema, se o paciente possui ou não algum tipo de doença hepática.

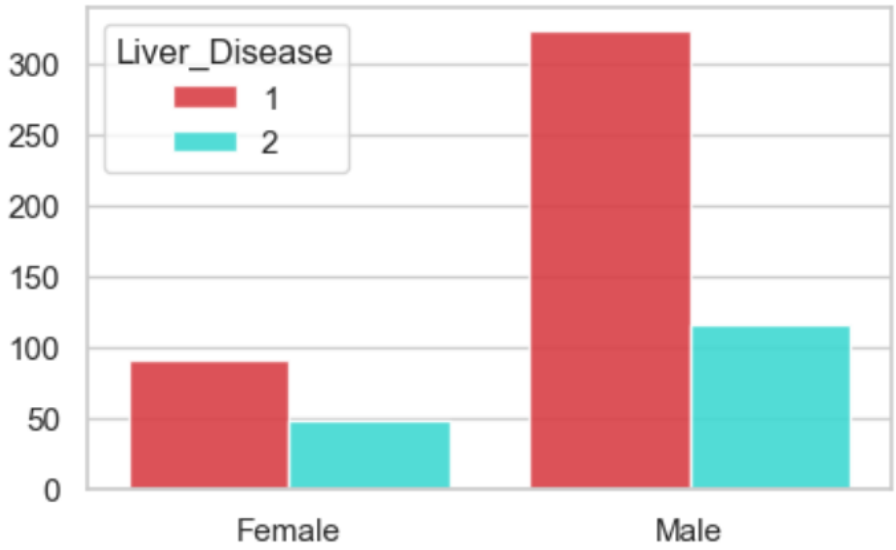

Figura 8 – Quantidade de pacientes por gênero e condição hepática
Fonte: gráfico produzido pelo autor (2024)

Para a coluna de gênero, temos um total de 439 pacientes homens e 140 mulheres. Já a variável alvo apresenta 416 pacientes sem a doença e 167 com doenças hepáticas. Ambas as colunas possuem um desbalanceamento entre as classes.

O gráfico anterior demonstra a contagem, por gênero e condição hepática dos pacientes do conjunto de dados. Observando a legenda do gráfico, o valor 1 representa a não presença da doença, enquanto o valor 2 representa a presença da doença hepática.

4.2. Pré-processamento dos Dados

4.2.1. Seleção de Atributos

O mapa de calor (do inglês heatmap), a seguir, foi construído com o intuito de eliminar atributos redundantes para o treinamento dos modelos. Esse gráfico retorna a correlação entre as variáveis numéricas do conjunto de dados, portanto, nele, não existem as colunas categóricas mencionadas anteriormente. Da análise do mapa de calor, pode-se perceber um trio de altas correlações:

- DB e TB;
- Sgot e Sgpt;
- TP e ALB.

As altas correlações são esperadas nesse conjunto de dados, como observado na descrição dos marcadores bioquímicos, no capítulo anterior. Inicialmente, as colunas DB e ALB serão eliminadas, por elas representarem menos informação do que suas respectivas colunas correlatas. Já as colunas Sgot e Sgpt serão analisadas durante a aplicação de cada algoritmo.

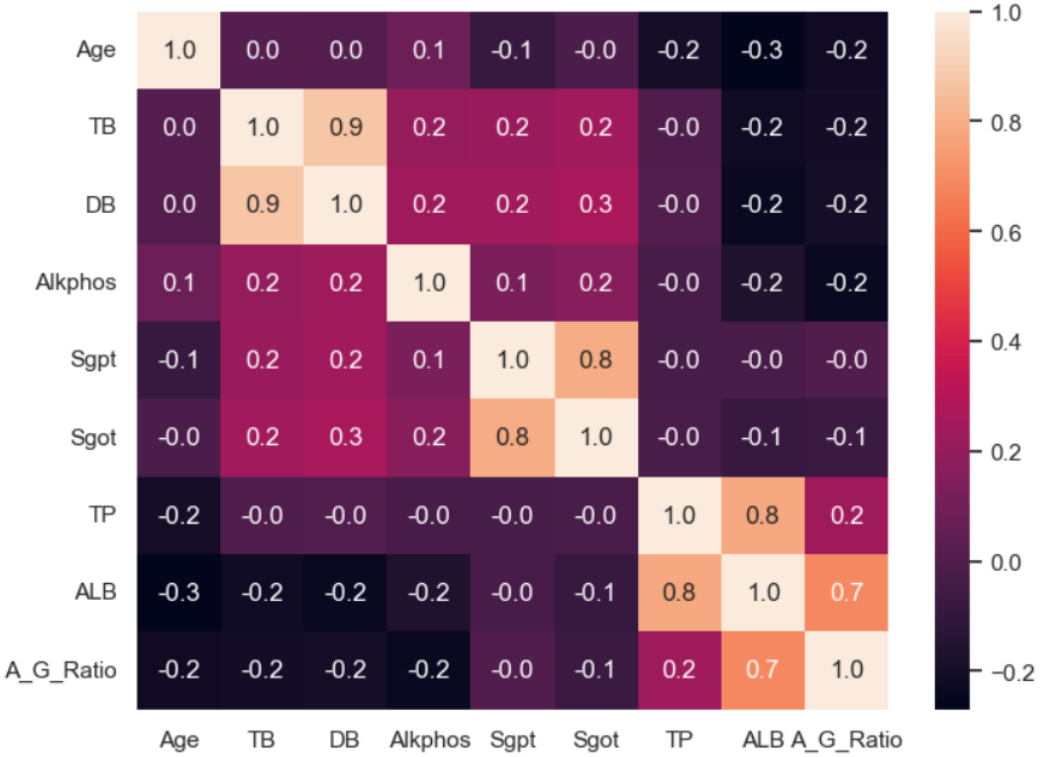

Figura 9 – Mapa de Calor
Fonte: gráfico produzido pelo autor (2024)

4.2.2. Tratamento de Outliers e Codificação

As colunas TB, DB, Alkphos, Sgpt e Sgot possuem alguns valores muito distantes de suas respectivas médias e medianas (rever item 4.1 – Análise Descritiva), ou seja, são prováveis outliers. Vale ressaltar que a coluna DB não faz mais parte do conjunto de dados.

Para essas colunas, os valores extremos aparecem poucas vezes, tendo, em alguns casos a ordem de grandeza muito maior em relação à média e mediana da informação. Ainda assim, realizando pesquisas na área, é possível existir valores bastante altos para essas variáveis. Portanto, a decisão tomada será realizar a separação delas em categorias. Serão elas:

- Abaixo do vr (0) - abaixo do valor de referência
- Dentro do vr (1) - dentro do valor de referência
- Valor Elevado (2) - até 10x o valor de referência
- Valor alto (3) - até 100x o valor de referência
- Valor muito alto (4) - acima de 100x o valor de referência

Com isso, não será perdido nenhum registro do conjunto de dados por exclusão de outliers. Além disso, dados futuros poderão facilmente se encaixar em uma dessas categorias. Os valores de referências utilizados, para cada um dos marcadores bioquímicos, estão descritos no capítulo anterior deste trabalho.

As variáveis categóricas serão, portanto, codificadas, de acordo com as seguintes estratégias:

- Gênero: será realizado o label encoding, por ser uma variável binária (0 para Mulher e 1 para Homem);
- Colunas com outliers: de acordo com a estratégia mencionada anteriormente. Nesse caso também será utilizado o label encoding.

4.2.3. Padronização e Balanceamento dos Dados

As colunas numéricas que não sofreram categorização (Age, TP e A_G_Ratio) devem ser padronizadas devido às suas diferentes escalas. O método utilizado foi o standard scaler, pois, como demonstrado na análise descritiva, essas variáveis apresentam uma distribuição próxima da normal.

A separação dos dados, em treino e teste será feita em uma proporção de 80/20, sendo 80% dos dados para treino e 20% para teste. Essa separação é feita de maneira aleatória, porém deve ser estratificada de acordo com a variável alvo. Isso quer dizer que a aleatoriedade na divisão dos dados irá levar em consideração a proporção de observações que apresentam a doença e as que não apresentam.

Essa estratégia evita que o grupo de dados de treino, por exemplo, tenha uma quantidade muito baixa, ou até nenhuma, da classe da variável alvo com menos frequência.

Após a separação dos dados, foi feito o balanceamento dos dados de treino, utilizando a técnica SMOTE. O balanceamento de classes é feito somente nos dados de treino, pois não há necessidade em balancear os dados de teste, já que somente o grupo de treino é necessário para o desenvolvimento do modelo.

4.3. Aplicação dos Modelos de Machine Learning

Para a aplicação dos algoritmos de Machine Learning, utilizou-se as técnicas de Grid Search. Essa técnica testa diversas combinações de hiperparâmetros do algoritmo, com o objetivo de encontrar aqueles que apresentem a melhor performance, para o conjunto de dados em questão.

Outra tentativa de otimização de performance dos modelos foi a avaliação de atributos por meio da estratégia de importância de permutação (permutation importance). Nela, a importância do atributo é medida de acordo com o aumento do erro na predição do modelo, ao permutar esse atributo. Características são mais importantes se a alteração de seus valores gerar mais erros na predição. É possível existir valores negativos, o que representa uma diminuição do erro devido a alteração dos valores desse atributo. Isso pode ser interpretado como um evento aleatório (causado por “sorte”), mais comum em conjunto de dados menores.

Para cada algoritmo, foram desenvolvidas duas versões de modelos, a primeira considerando todos os atributos e a outra, retirando os 3 atributos menos importantes. A seguir, seguem os gráficos da importância dos atributos, para cada um dos algoritmos utilizados:

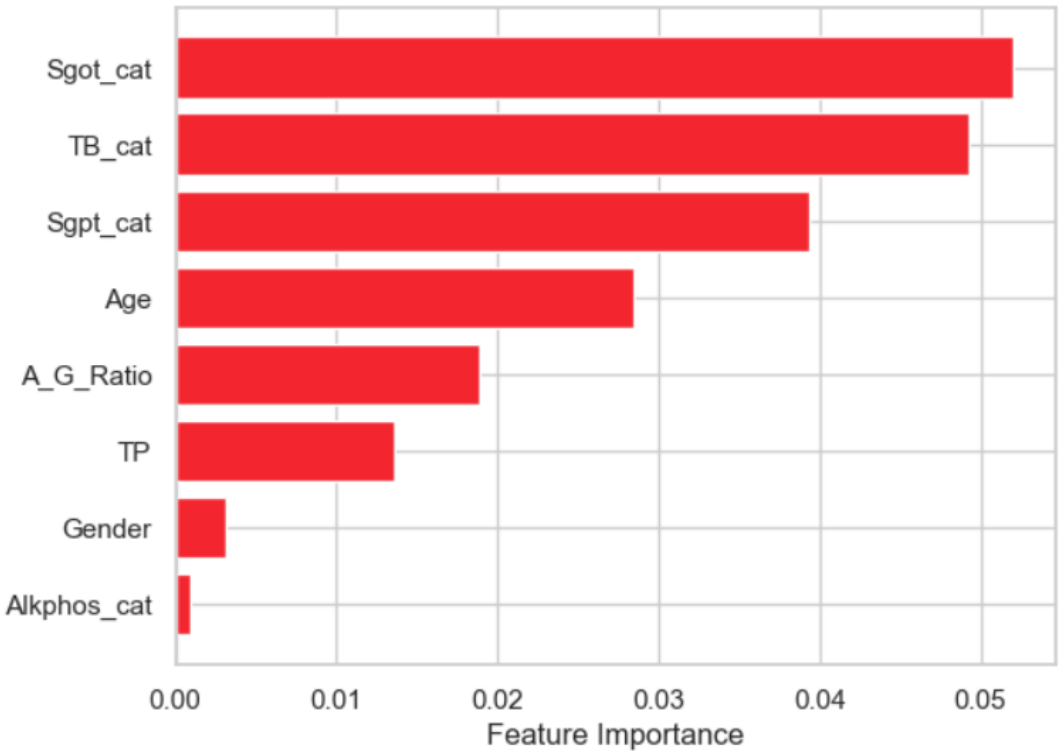

Figura 10 – Gráfico de Feature Importance para Regressão Logística
Fonte: gráfico produzido pelo autor (2024)

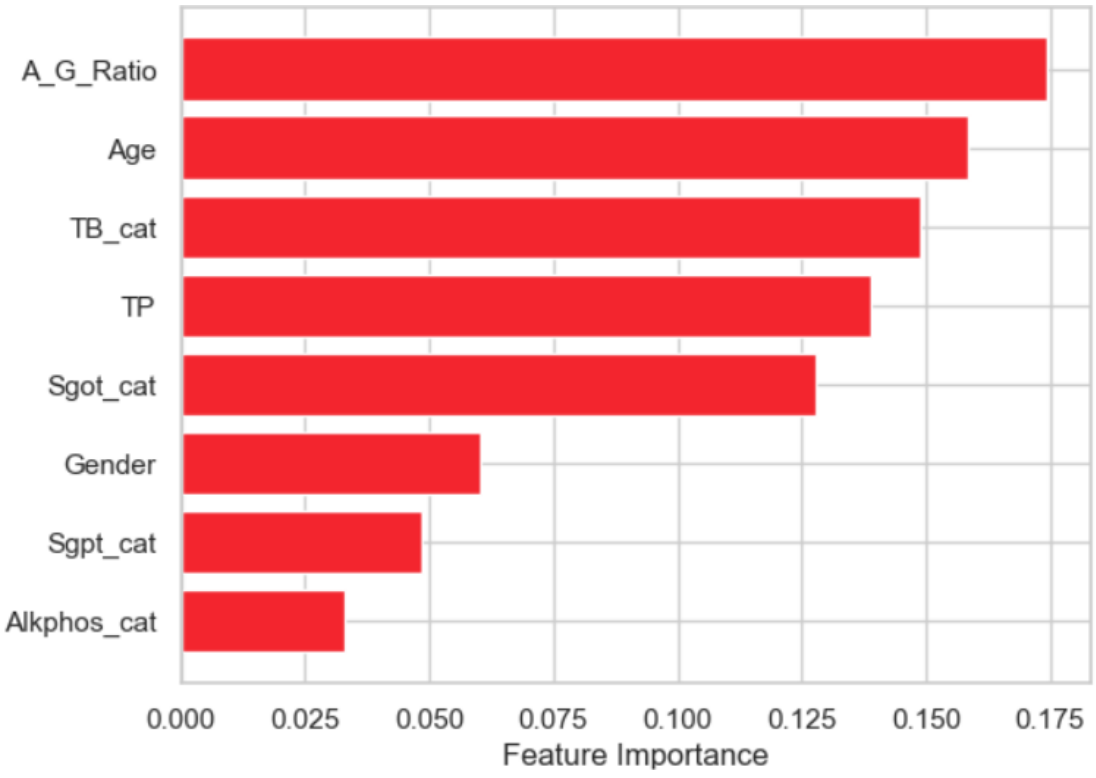

Figura 11 – Gráfico de Feature Importance para Random Forest
Fonte: gráfico produzido pelo autor (2024)

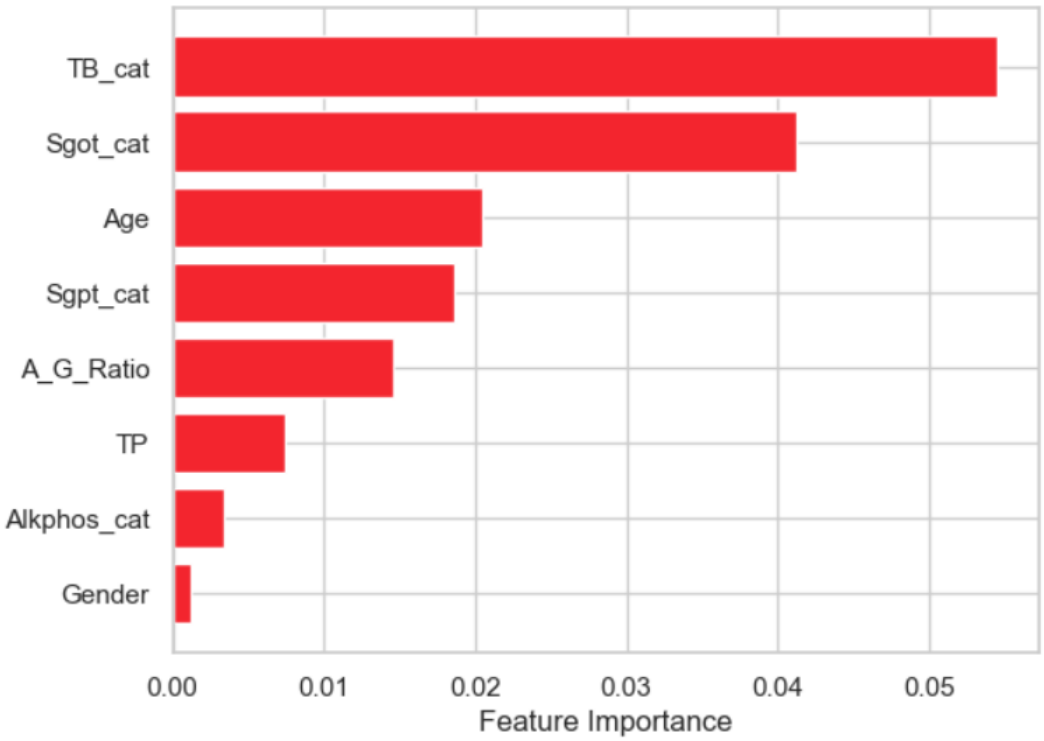

Figura 12 – Gráfico de Feature Importance para SVM
Fonte: gráfico produzido pelo autor (2024)

Apesar de haver atributos com importâncias inferiores à outros, não houve ganhos de performance com a aplicação da técnica.

4.4. Resultados

A primeira forma de comparar os modelos foi utilizado a técnica de validação cruzada. Nessa técnica, o conjunto de dados é separado em K grupos (foi utilizado um K = 5). Um grupo é utilizado para teste, enquanto os outros farão parte do treinamento do modelo. Isso é feito até que todos os grupos sejam utilizados para teste. Cada iteração retornará uma métrica de avaliação, nesse caso a acurácia, e o resultado final será a média das métricas. Ao aplicar a validação cruzada no conjunto de dados, os valores obtidos foram:

- Regressão Logística: 0.73
- Random Forest: 0.78
- SVM: 0.73

Porém, ao aplicar os dados de teste para esses modelos, houve uma grande queda na performance. É comum que os dados de teste apresentem um acurácia menor, porém o decréscimo foi muito superior ao esperado. A acurácia para os modelos, em teste foi de:

- Regressão Logística: 0.64
- Random Forest: 0.68

- SVM: 0.63

É possível que essa queda de desempenho seja devido à baixa quantidade de dados. A razão adotada entre treino e teste, neste trabalho, foi de 80/20, ficando, portanto, poucas observações para testar os modelos. Foram aplicadas diferentes proporções de separação nos dados, porém, um maior grupo de teste impactava no desempenho dos modelos ainda em treino, causando underfitting.

Vale destacar que, a aplicação da seleção de atributos pelo método do permutation importance não trouxe resultados significativos para a performance dos modelos.

Portanto, apesar do Random Forest apresentar um bom desempenho em treino, os valores para teste são considerados performances baixas. Além da acurácia, as matrizes de confusão de cada um dos modelos foram analisadas, porém todas apresentavam valores altos para os erros (falsos positivos e negativos).

Outras estratégias podem ser buscadas para se ter uma maior performance nos modelos, como:

- Utilização de outras formas de padronização dos dados;
- Técnicas diferentes de lidar com valores outliers;
- Ajuste fino dos hiperparâmetros dos algoritmos;
- Teste com outros algoritmos de aprendizado de máquina.

5. Conclusão

O objetivo deste trabalho foi comparar diferentes algoritmos de aprendizado de máquina para o diagnóstico de doenças no fígado. Foram utilizados três algoritmos; a regressão logística, o random forest e o SVM. Além disso, o trabalho se propôs a demonstrar algumas técnicas mais utilizadas de pré-processamento de dados e métricas de desempenho de modelos de machine learning.

Considerando os dados utilizados, o modelo criado a partir do Random Forest obteve a melhor performance no treino, chegando próximo a 80% de precisão. Porém, devido à baixa quantidade de dados, os mnodelos não conseguiram atingir bons resultados em teste, tendo 68% de precisão. Algumas técnicas ainda foram implementadas para se buscar uma melhora na performance desses modelos, como a seleção de variáveis, porém, não houve aumento considerável nos desempenhos.

Apesar disso, esse estudo atingiu o seu propósito de aplicar e comparar os diferentes algoritmos apresentados, além de demonstrar etapas de pré-processamento, importantes para a aplicação de modelos de machine learning.

Referências

(IBM, 2024). **O que é Machine Learning**. Disponível em: https://www.ibm.com/br-pt/topics/machine-learning. Acesso em 20 dez. 2023.
(OMS, 2024). Hepatitis. Disponível em: https://www.who.int/health-topics/hepatitis. Acesso em 20 dez. 2023.

(Databricks, 2024). **Modelos de Machine Learning**. Disponível em: https://www.databricks.com/br/glossary/machine-learning-models. Acesso em 20 dez. 2023.

(Didática Tech, 2024). **Underfitting e Overfitting**. Disponível em https://didatica.tech/underfitting-e-overfitting. Acesso em 21 dez. 2023.
JESUS, G. C. de; SOUSA, H. H. B. A. de; BARCELOS, R. da S. S. **Principais Patologias e Biomarcadores das Alterações Hepáticas**. Revista Estudos - Vida e Saúde (Revista de Ciências Ambientais e Saúde), Goiânia, Brasil, v. 41, n. 3, 2014. DOI: 10.18224/est.v41i3.3597. Disponível em: https://seer.pucgoias.edu.br/index.php/estudos/article/view/3597. Acesso em: 21 dez. 2023.

(Awari, 2023). **SVM e Machine Learning. Conceitos e Implementações**. Disponível em https://awari.com.br/svm-em-machine-learning-conceitos-e-implementacoes. Acesso em 03 jan. 2024.

(LinkedIn, 2018). **Algoritmos de classificação pt. 2 - Árvore de Decisão**. Disponível em: https://www.linkedin.com/pulse/algoritmos-de-classifica%C3%A7%C3%A3o-pt-2-%C3%A1rvore-decis%C3%A3o-vin%C3%ADcius-gomes/. Acesso em 03 jan. 2024.

(LAPIX, 2024). **Reconhecimento de Padrões: Support Vector Machine**. Disponível em: https://lapix.ufsc.br/ensino/reconhecimento-de-padroes/reconhecimento-de-padroessupport-vector-machines/. Acesso em 03 jan. 2024.

(Sci-kit-earn, 2024). **Scikit-learn**. Disponível em: scikit-learn: machine learning in Python — scikit-learn 1.3.2 documentation. Acesso em 05 jan. 2024.

(Pandas, 2024). **Pandas documentation**. Disponível em: pandas documentation — pandas 2.1.4 documentation (pydata.org). Acesso em 05 jan. 2024.

HALL, Mark A. **Correlation-based feature selection of discrete and numeric class machine learning**. University of Waikato, Department of Computer Science, 2000. Acesso em 09 jan. 2024.

N. V. Chawla, K. W. Bowyer, L. O. Hall, W. P. Kegelmeyer. **SMOTE: Synthetic Minority Over-sampling Technique.** Access Foundation and Morgan Kaufmann Publishers, 2002. Acesso em 10 jan. 2024.