

TÉCNICAS DE APRENDIZADO DE MÁQUINA NA ENGENHARIA MECÂNICA:

Aplicação em análise de falha de rolamentos esféricos

Pietro Santiago Alexandre Sales

RESUMO

Este trabalho visa apresentar e analisar técnicas de aprendizado de máquina com foco na aplicação de diagnóstico de falhas em elementos mecânicos. Além disso, as técnicas serão aplicadas em dados reais de vibração de rolamentos esféricos em operação. Os resultados obtidos validam a eficiência das técnicas selecionadas, apresentando um bom grau de precisão. No entanto, os valores dos parâmetros utilizados apresentam uma considerável intercessão entre duas classes de rolamentos, sendo um fator limitante para alcançar uma precisão ainda mais elevada.

Palavras-chaves: Falha. Diagnóstico. Aprendizado de Máquina.

1 INTRODUÇÃO

Ao se tratar de elementos mecânicos, a ocorrência de falhas pode resultar enormes prejuízos financeiros e humanos. Dessa forma, é papel do engenheiro mecânico garantir a integridade desses sistemas através de análises e ordens de manutenção.

Dependendo do nível de complexidade do sistema, o elevado número de variáveis pode torná-lo extremamente difícil de ser avaliado quantitativamente. Assim, a avaliação desses sistemas recai sobre a experiência do operador [1], como é o caso da análise de vibrações em rolamentos, que será discutido neste trabalho.

No entanto, com o avanço da tecnologia e da velocidade de processamento dos computadores, cada vez mais dados são gerados nos diversos tipos de processos. Quando bem utilizados, esses dados podem nos informar comportamentos inadequados com maior velocidade e confiabilidade, o que é essencial para a

prevenção de acidentes de trabalho. Uma das abordagens computacionais mais modernas e eficientes para tratar desse problema é o aprendizado de máquina, a qual o computador pode aprender os padrões de funcionamento do sistema e alertar quando algum parâmetro sair desse padrão.

O objetivo desse trabalho é apresentar algumas técnicas nessa área que podem revelar informações preciosas, tais como: regressão logística, detecção de anomalias e redes neurais. Em seguida, serão selecionadas as técnicas adequadas para o tratamento de diagnóstico de falhas em rolamentos esféricos e os resultados serão avaliados.

2 DESENVOLVIMENTO

2.1. Técnicas de aprendizado de máquina

As técnicas de aprendizado de máquina são divididas em, principalmente, dois tipos: aprendizagem supervisionada e aprendizagem não-supervisionada.

Na aprendizagem supervisionada, o computador recebe dados “rotulados”, ou seja, ele conhece tanto os parâmetros do processo quanto o significado deles. Desse modo, é possível fazer com que o computador aprenda como se comportam os parâmetros para cada condição de operação (e.g. normal, defeito 1 ou defeito 2). Com isso, ao obtermos dados da operação de um equipamento que desconhecemos seu estado, o computador pode informar sua condição de trabalho.

Em contrapartida, na aprendizagem não-supervisionada os dados não são “rotulados” e, portanto, o computador não recebe o significado dos parâmetros que recebeu. Essa abordagem pode ser utilizada para encontrar padrões nos parâmetros de trabalho e auxiliar na detecção de comportamentos anormais, mesmo que o computador não saiba informar o estado do equipamento.

2.1.1. Detecção de anomalias

O método de detecção de anomalias pode ser classificado tanto como uma técnica de aprendizagem supervisionada quanto não-supervisionada, a depender do

formato dos dados disponíveis. Esse algoritmo pode ser utilizado quando foram coletados majoritariamente dados de condições normais de operação (em diversos sistemas pode ser extremamente difícil e custoso o acontecimento de falhas) e deseja-se identificar quando houverem desvios nessas condições normais.

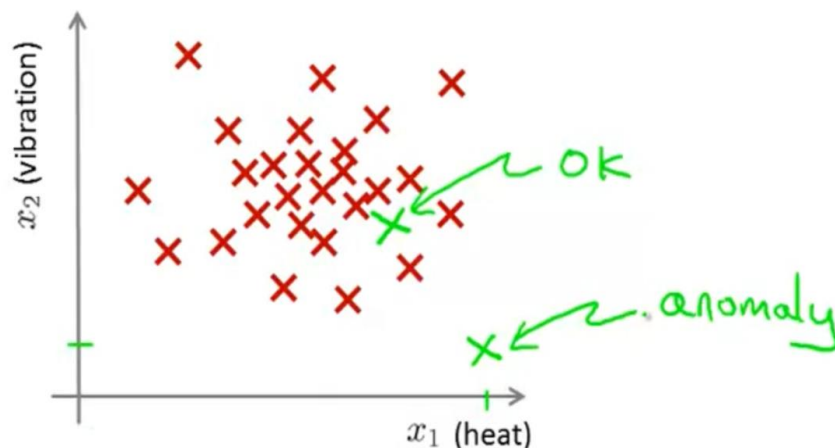


Figura 1 – Representação de um dado anômalo

A Figura 1 apresenta dados fictícios para ilustrar o que será definido como anomalia. As marcações em vermelho representam dados conhecidos que foram coletados anteriormente e servirão para treinar o algoritmo. As marcações em verde são novos pontos que deverão ser classificados pelo algoritmo como sendo pertencentes a uma condição normal ou anômala. O objetivo do algoritmo é determinar um limiar (representado por ϵ) a partir do qual os pontos passam a serem considerados anômalos.

2.1.1.1. Base matemática

Ainda em relação ao exemplo hipotético da Figura X, é possível observar que a marcação verde superior está próxima ao aglomerado de marcações vermelhas, enquanto o ponto verde inferior está distante delas. Assim, deve-se definir formalmente o que significa “distante” em problemas reais.

Na detecção de anomalias supervisionada, a tarefa de determinar o limiar é facilitada pelo fato de ser conhecida a condição de cada ponto. Em contrapartida, a detecção não-supervisionada segue a premissa que a maioria dos dados de treino

são considerados normais¹ e, assim, é possível gerar uma curva gaussiana sobre esses pontos para ser utilizada de base na determinação do limiar.

Após selecionar os parâmetros representativos, para montar a curva gaussiana é necessário obter a média (μ) e a variância (σ^2) de cada parâmetro. Em seguida, obtém-se a função densidade de probabilidade ($p(x_j; \mu_j; \sigma_j^2)$) para cada parâmetro. A produtória desses valores ($p(x)$) será a probabilidade representativa do conjunto. Assim,

$$\mu_j = \frac{1}{m} \sum_{i=1}^m x_j^{(i)} \quad (1)$$

$$\sigma_j^2 = \frac{1}{m} \sum_{i=1}^m \left(x_j^{(i)} - \mu_j \right)^2 \quad (2)$$

$$p(x_j; \mu_j; \sigma_j^2) = \frac{1}{\sqrt{2\pi} \sigma_j} \exp \left(-\frac{(x_j - \mu_j)^2}{2\sigma_j^2} \right) \quad (3)$$

$$p(x) = \prod_{j=1}^n p(x_j; \mu_j; \sigma_j^2) \quad (4)$$

Onde o sobrescrito (i) representa o valor de cada ponto individual do parâmetro j, m representa a quantidade de dados e n o número de parâmetros.

Desse modo, os pontos mais periféricos possuem menor valor de $p(x)$ e àqueles com valor abaixo de um limiar (ϵ) são considerados anômalos ($p(x) < \epsilon$). Uma boa forma de determinar o limiar na aprendizagem não supervisionada é coletar ao menos uma pequena quantidade de dados em condição anômala, obter seus valores de $p(x)$ e selecionar ϵ de modo a classificar corretamente esses pontos.

2.1.2. Regressão logística

A regressão logística é um método de aprendizagem supervisionada utilizada para classificação dos dados (na aplicação deste trabalho, têm-se as classes: saudável, defeito nas esferas, defeito na pista interna ou defeito na pista externa).

A rigor, a regressão logística é uma classificação binária dos dados (positiva ou negativa para uma classe). No entanto, uma simples superposição de regressões é capaz de classificar um dado em qualquer número de classes. Por exemplo, se existem as classes A, B, C e D, é possível criar quatro regressões referentes a cada

uma das classes individualmente. Então, cada dado da amostra receberá uma classificação binária para cada uma das classes e será identificado de acordo.

O objetivo deste algoritmo, então, é traçar uma curva de decisão para separar os dados como sendo pertencentes a cada uma das classes. As Figuras 2 e 3 representam essa situação.

Considere os pontos brancos como dados coletados e utilizados para treinar o algoritmo (criar a curva de decisão). Esses dados são rotulados de uma forma binária como sim ($y=1$) ou não ($y=0$). O objetivo é traçar uma curva de decisão que defina a região onde os dados são positivos ou negativos, possibilitando classificar dados futuros nunca antes vistos. Note que existe uma superposição de exemplos positivos e negativos na parte central, o que resulta na impossibilidade de obter uma classificação com 100% de precisão.

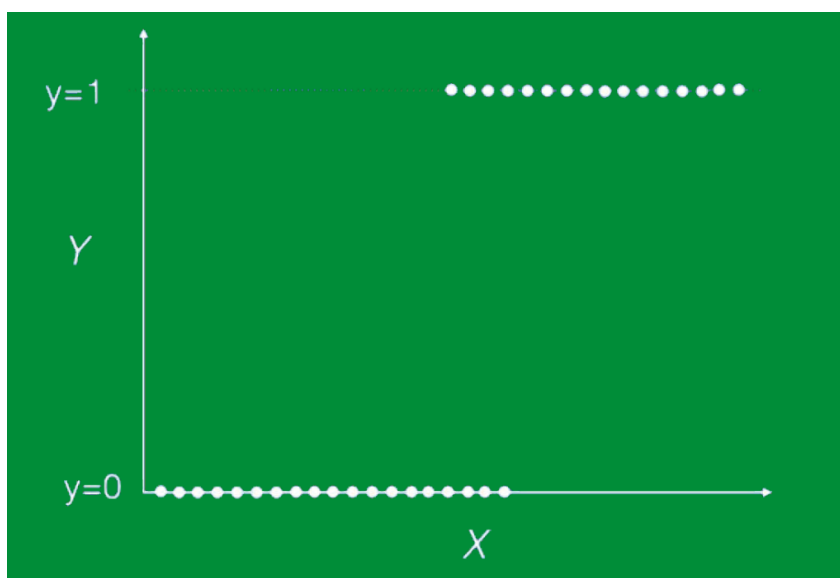


Figura 2 – Dados rotulados de forma binária. Adaptada de [4].

A curva em azul da Figura 3 é a regressão sobre os pontos em branco e representa a probabilidade de um dado valor de x ser da classe positiva ($y=1$). Assim, pode-se definir a curva de decisão como sendo tal que todos os valores de x que resultam em uma probabilidade maior ou igual a 50% são classificados como positivos. Note que, com esse critério, a curva de decisão é exatamente a linha vertical preta na Figura 3.

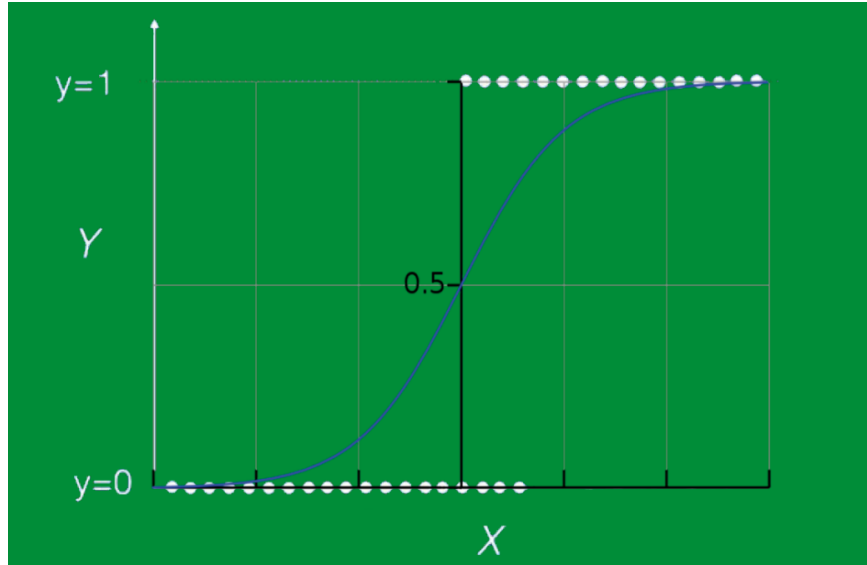


Figura 3 – Regressão logística e curva de decisão. Adaptada de [4].

2.1.2.1. Base matemática

A função mostrada na Figura 3 é a função de sigmoid, dada pela equação (5):

$$f(X) = \frac{1}{1+e^{-\theta^t X}} \quad (5)$$

Onde,

$$\theta = \begin{bmatrix} \theta_0 \\ \theta_1 \\ \vdots \\ \theta_n \end{bmatrix} \quad X = \begin{bmatrix} 1 \\ x_1 \\ \vdots \\ x_n \end{bmatrix}$$

n: quantidade de parâmetros

Logo, a multiplicação matricial que aparece na função é:

$$\theta^t X = \theta_0 + \theta_1 x_1 + \cdots + \theta_n x_n \quad (6)$$

Como mencionado anteriormente $f(x)$ representa a probabilidade do dado pertencer a classe positiva a partir de seu respectivo X e Θ . Em termos matemáticos:

$$f(X) = p(y = 1|X; \theta) \quad (7)$$

Visto que X é uma matriz em R^n com o valor de cada ponto, o objetivo do algoritmo é encontrar os valores de $\theta \in R^n$ que melhor ajusta a curva $f(x)$ aos pontos dados. A partir disso, os novos dados serão classificados de acordo com o seguinte critério:

$$\begin{cases} y_p = 1, \text{ se } f(X) \geq 0.5 \\ y_p = 0, \text{ se } f(X) < 0.5 \end{cases} \quad (8)$$

Onde y_p é a previsão do algoritmo para o valor de y (classe positiva ou negativa).

Em resumo, a partir de uma aprendizagem prévia (que encontra os valores adequados de Θ) e dado um ponto qualquer (representado por uma matriz X), o ponto será classificado como positivo quando $f(X) \geq 0.5$ e como negativo quando $f(X) < 0.5$.

Finalmente, Θ é encontrado minimizando a seguinte função de custo:

$$J(\theta) = -\frac{1}{m} \sum_{i=1}^m [y^{(i)} \log(f(x)) + (1 - y^{(i)}) \log(1 - f(x))] \quad (9)$$

Onde,

J : função custo

m : quantidade de dados de treino

$y^{(i)}$: classe do i -ésimo exemplo

Quanto menor o valor de $J(\theta)$, mais ajustada a curva está em relação aos dados utilizados para treinar o algoritmo. O método de minimização mais utilizado nesses casos é o Gradiente Descendente, que pode ser visto com detalhes em [5].

2.1.3. Redes Neurais

As Redes Neurais são algoritmos muito poderosos que podem ser usados com diferentes finalidades. Para a aplicação desse artigo, será utilizada para classificação, assim como a regressão logística. Assim, serão discutidas as diferenças no uso e resultado desses dois algoritmos.

2.1.3.1. Funcionamento das Redes Neurais

A Figura 4 representa uma forma genérica de uma Rede Neural, onde é possível identificar os elementos básicos: neurônio², camada de entrada, camada oculta e camada de saída.

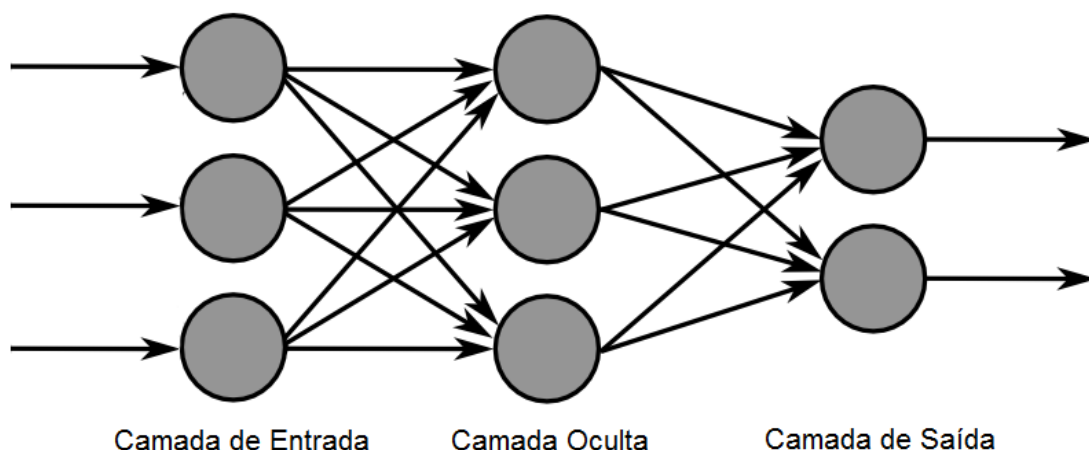


Figura 4 – Rede neural genérica

Os neurônios são a unidade básica da rede. Individualmente, eles recebem dados de entrada, realizam um processamento e enviam um dado de saída correspondente. Sendo utilizada para classificação, o processamento realizado por cada neurônio é aplicar a função de sigmoid (outras funções com características semelhantes podem ser usadas) para os dados de entrada. Assim, essas redes neurais funcionam como sucessivas etapas de regressão logística em série e em paralelo.

A vantagem aqui, no entanto, é a possibilidade de formar parâmetros mais complexos que podem ajudar na identificação da correlação entre dados de entrada e dados de saída. Por exemplo, caso essa correlação seja dada em termos de um polinômio de 3º grau, o pesquisador precisa passar isso explicitamente para o algoritmo de regressão logística. Por outro lado, devido às combinações dos resultados na rede neural, ela é capaz de formar automaticamente essas correlações mais complexas. A saída da rede continua sendo, como na regressão logística, a probabilidade do dado de entrada pertencer a uma determinada classe.

As camadas da Rede Neural Artificial definem sua topologia, que é crucial para a obtenção de um resultado satisfatório. A Figura 4 representa uma topologia genérica, visto que o número de camadas e o número de neurônios por camada pode ser ajustado conforme necessário.

A camada de entrada possui o número de neurônios igual a quantidade de parâmetros dos dados de entrada e a camada de saída igual ao número de possíveis classes (para apenas duas classes é possível utilizar somente um neurônio de saída).

A camada oculta é a parte mais flexível e aceita diferentes formas em busca de um resultado satisfatório. Podem existir qualquer número de camadas ocultas com qualquer número de neurônios em cada uma delas. Diz-se que a Rede Neural é mais profunda quanto maior o número de camadas ocultas e mais densa quanto maior o número de neurônios por camada.

Tornando a rede mais profunda e mais densa, é possível combinar os parâmetros cada vez mais e traçar curvas de decisões mais complexas. Isso vem com a limitação de aumentar o custo de processamento computacional e com o chamado sobreajuste. O sobreajuste acontece quando a quantidade de dados de entrada não é grande suficiente para a complexidade dos parâmetros atingidos pela Rede. Assim, qualquer dado anormal dentro do conjunto acaba tendo uma grande influência no aprendizado, pois a Rede tenta ajustar a curva de decisão de modo a cobrir todos os dados, resultando em uma curva que não reflete realmente a realidade. A seção 2.4 cita uma técnica simples utilizada para a escolha da topologia da Rede Neural.

2.2. Diagnóstico de falha em rolamentos esféricos

Para aplicar as técnicas anteriormente mencionadas, é necessário obter dados de funcionamento do sistema a ser estudado. O estudo sobre falhas em rolamentos esféricos que será apresentado foi facilitado devido ao experimento realizado na Case Western Reserve University [2] e ao tratamento dos dados proposto por [1].

2.2.1. Experimento e coleta de dados

A Figura 5 mostra o aparato utilizado para a coleta de dados experimentais realizado na Case Western Reserve University [2]. O equipamento mostrado a esquerda é um motor de 2 hp, no centro há um transdutor de torque e na direita há um dinamômetro. Os rolamentos de teste foram montados acoplados ao eixo do motor. Também foram instalados controles eletrônicos que não são possíveis de visualizar na figura.

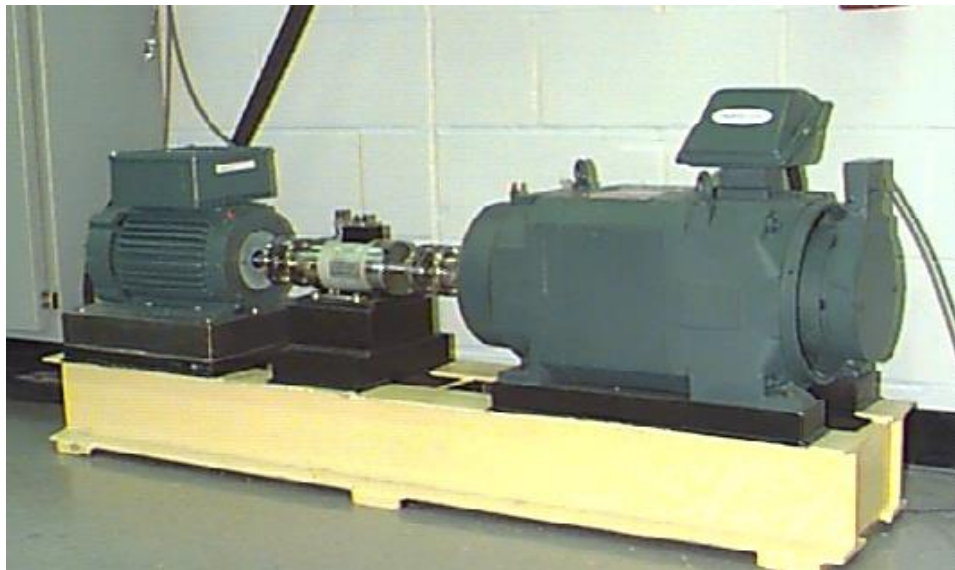


Figura 5 – Aparato experimental

O procedimento foi realizado utilizando rolamentos esféricos em quatro possíveis estados de funcionamento: I) Saudável; II) Defeito nas Esferas; III) Defeito na pista interna; IV) Defeito na pista externa. Os defeitos foram induzidos artificialmente através da usinagem por eletroerosão.

Desse modo, dados de vibração foram coletados para as quatro possibilidades em igual quantidade, o que permite analisar se há relação direta entre o perfil de vibrações e os defeitos mencionados. Além disso, visto que os dados serão rotulados (de acordo com o estado de funcionamento), será possível utilizar técnicas de aprendizagem supervisionada (a detecção de anomalias não se aplica a esse caso, pois não há uma classe majoritária de dados).

2.3. Tratamento dos dados

Como tarefa preliminar à escolha e utilização do algoritmo, é imprescindível conhecer e entender fisicamente o problema que está sendo tratado, a fim de selecionar e preparar os dados de forma a obter o melhor resultado possível.

Para o problema do diagnóstico de falha em rolamentos externos, os dados “crus” de vibração coletados [2] podem ser convertidos para medidas estatísticas específicas, tais como: “Fator de assimetria”, “Fator de forma”, “Curtose”, “Fator de crista”, “Fator de Impulso” e “Fator de margem” [1]. Assim, os dados que serão utilizados para a aprendizagem dos modelos neste trabalho são provenientes das tabelas fornecidas por [1].

Este tratamento é uma parte crucial para uma aprendizagem correta, sendo mais importante até mesmo que a escolha do algoritmo. É importante salientar que a utilização de mais parâmetros pode, em alguns casos, piorar os resultados. Portanto, é mais importante a escolha coerente dos parâmetros do que a quantidade de parâmetros.

2.4. Validação dos resultados

Para validar os resultados, uma técnica simples é aplicada. A partir do conjunto de dados coletados, divide-se esse conjunto em duas partes: Dados de treino e Dados de teste (tipicamente 60% e 40% respectivamente).

Assim, ao invés de treinar o algoritmo com 100% dos dados coletados, utiliza-se apenas 60% deles para isso. Dessa forma, é possível utilizar o conjunto de Dados de Teste para estimar a precisão do algoritmo na classificação de dados nunca antes vistos, pois, apesar dos Dados de Teste serem conhecidos para o pesquisador, são desconhecidos para a máquina.

Também é comum dividir o conjunto em três grupos (tipicamente 60%, 20% e 20%), sendo 60% para treino, 20% para teste e os outros 20% para ajustar os hiperparâmetros referentes ao algoritmo (como o número de camadas ocultas para Redes Neurais).

2.5. Metodologia

Os conceitos matemáticos apresentados anteriormente podem ser implementados diretamente em qualquer linguagem de programação capaz de realizar operações matriciais. No entanto, a linguagem Python oferece diversas bibliotecas prontas e otimizadas para aplicar as técnicas de aprendizagem de máquina. Desse modo, neste trabalho foram utilizadas as seguintes bibliotecas de código aberto:

1. Scikit-learn: Biblioteca de aprendizado de máquina que será utilizada para a regressão logística;
2. Tensorflow: Biblioteca adequada para desenvolvimento de redes neurais;;
3. Keras: Biblioteca que trabalha com Tensorflow e facilita a montagem da tipologia das redes neurais;
4. Numpy: Pacote que permite realizar operações matriciais;
5. Pandas: Biblioteca para análise e estruturação de dados;
6. Matplotlib: Biblioteca para representações gráficas;

Assim, o algoritmo de análise do problema foi dividido nas seguintes partes:

1. Passo preliminar: Visualização dos dados de vibração obtidos por [2];
2. Importar bibliotecas;
3. Ler e visualizar os dados tratados por [1];
4. Regressão Logística;
 - 4.1. Regressão utilizando somente Fator de Forma e Assimetria;
 - 4.2. Curva de decisão bidimensional com os Fatores de Forma e Assimetria;
 - 4.3. Regressão com todos parâmetros;
5. Rede Neural

O algoritmo completo pode ser acessado em: <https://github.com/PietroSas/analise-rolamentos> através do arquivo "analise_rolamentos.ipynb".

3 RESULTADOS

Antes de aplicar os modelos de aprendizagem de máquina, é importante tentar visualizar valores representativos da amostra de dados. Os dados de vibração obtidos diretamente no experimento não foram utilizados na aprendizagem, mas, a fim de conhecer a origem dos dados, uma pequena fração desses pontos estão apresentados na Figura 6.

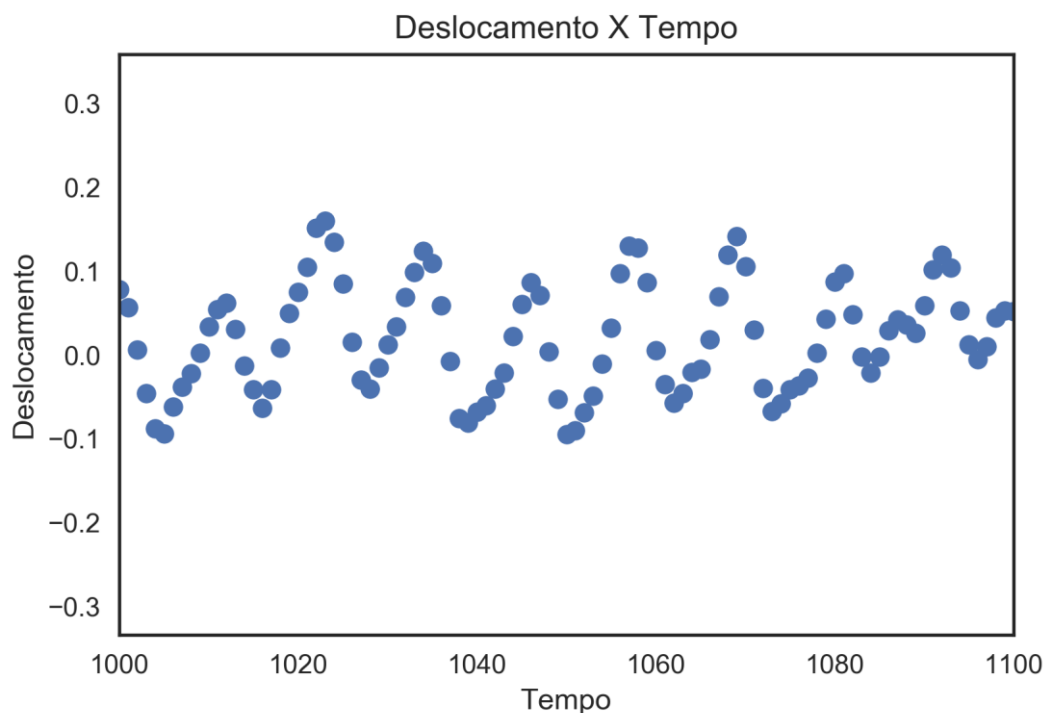


Figura 6 – Dados de vibração de um rolamento em estado normal de operação

A Figura 6 foi gerada a partir de uma pequena fração dos dados de vibração de um rolamento em estado normal (o número total de dados obtidos por teste é de aproximadamente 250 mil).

A partir do pressuposto de que cada estado de operação apresenta um padrão de vibração característico, esses dados crus foram convertidos nas medidas estatísticas mencionadas anteriormente, que são capazes de capturar as nuances desses padrões. A Tabela 1 apresenta uma pequena fração desses dados após a etapa de pré-processamento. Esses são os dados que foram diretamente utilizados nos algoritmos.

	Assimetria	Curtose	Crista	Forma	Impulso	Margem
0	-0.301250	2.481710	2.775495	1.194182	3.314447	3.826980
1	-0.192214	2.531206	2.426969	1.206328	2.927721	3.361278
2	-0.175213	2.559090	2.492546	1.215273	3.029125	3.503304
3	-0.029234	2.316718	2.469554	1.209672	2.987349	3.467161
4	0.022078	2.708726	2.815036	1.247238	3.511020	4.138479
5	-0.211558	2.755039	2.957466	1.226729	3.628010	4.241124
6	0.140896	2.794350	2.661178	1.267598	3.373305	4.018224
7	-0.316860	2.615248	2.822367	1.220262	3.444027	4.034513
8	-0.044224	2.482811	2.793800	1.218165	3.403310	3.934936
9	-0.242878	2.682910	2.700426	1.217775	3.288513	3.812402

Tabela 1 – Amostra de dados

É de grande importância visualizar esses dados graficamente, a fim de encontrar a melhor estratégia de trabalho. Segundo [1], os Fatores de Assimetria e de Forma resultam no melhor aprendizado para esse sistema, sendo, portanto, os parâmetros mais significativos. A Figura 7 mostra a distribuição bidimensional dos dados seguindo apenas esses dois parâmetros.

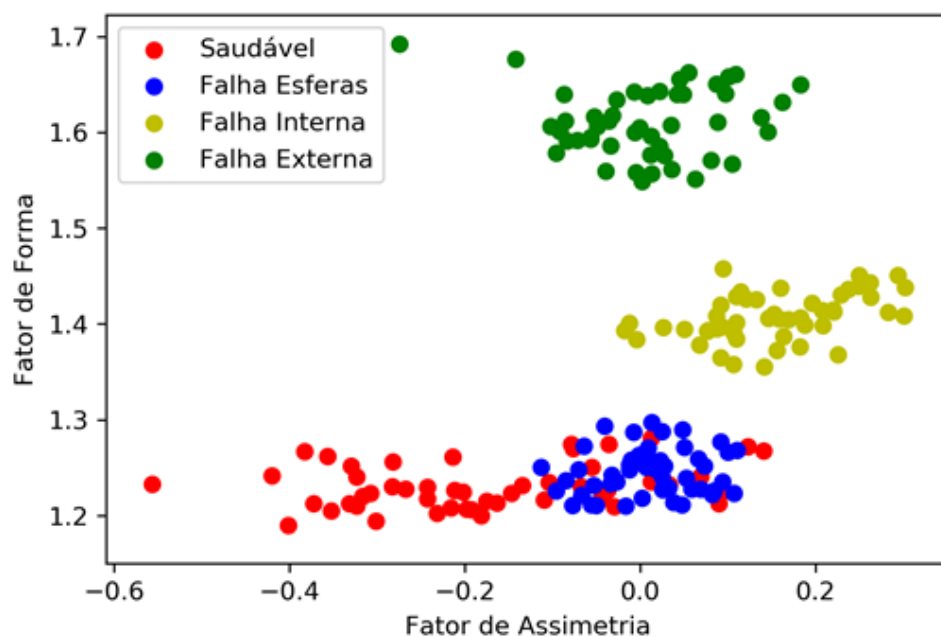


Figura 7 – Fatores de Forma e de Assimetria

Os dados apresentados na Figura 7 indicam que os valores desses parâmetros nas condições de “Falha na pista interna” e “Falha na pista externa” são bastante específicos e possíveis de distinguir dos demais. No entanto, os parâmetros das condições “Saudável” e “Falha nas esferas” se confundem, apontando que a utilização de somente esses dois parâmetros seria um fator limitante de aprendizagem.

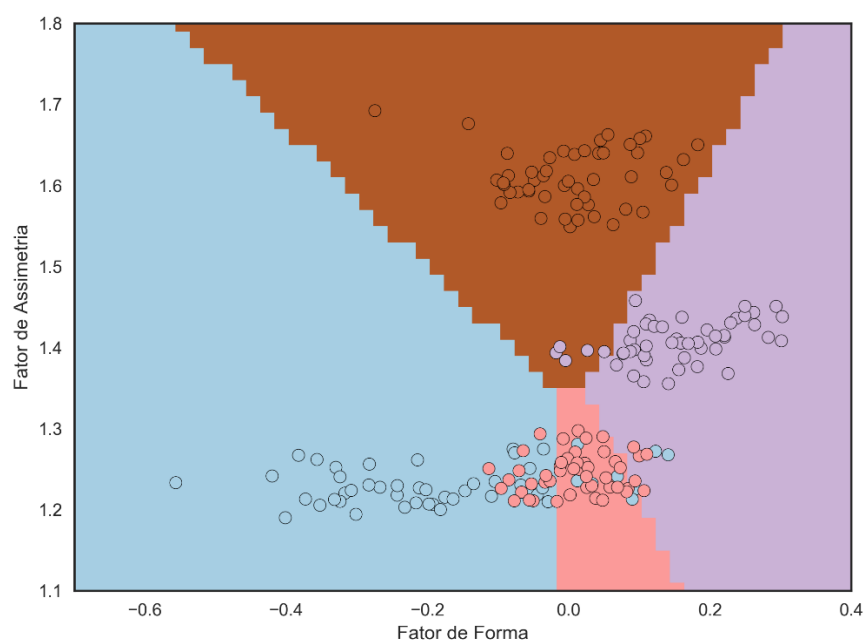


Figura 8 – Barreiras de decisão bidimensional através de regressão logística

A Figura 8 mostra os mesmo pontos da Figura 7 e as barreiras de decisão montada pelo algoritmo de regressão logística, utilizando somente o Fator de Forma e Assimetria como parâmetros. Essas barreiras dividem o gráfico em quatro partes, que representam as regiões de cada uma das condições de operação. Assim, dados nunca antes vistos serão classificados de acordo com a região que eles pertencerem.

É possível notar, na parte inferior da Figura 8, que o algoritmo teve dificuldades em classificar corretamente certos pontos, dado a intercessão existente. Assim, a precisão da Regressão logística nesse caso foi de 73%. Por outro lado, utilizando os seis parâmetros propostos, a precisão foi de 83%. Essa precisão representa a fração de classificações corretas dos dados de teste, que não foram apresentados à máquina em nenhum momento.

Em relação as Redes Neurais, foi construída uma Rede com três camadas ocultas, possuindo 100, 50 e 25 neurônios respectivamente. As camadas de entrada e saída possuem 6 e 4 neurônios respectivamente (de acordo com o número de parâmetros e classes de operação. A Figura 9 apresenta o resultado após 200 etapas de aprendizagem.

```
Epoch 193/200
- 0s - loss: 0.1991 - acc: 0.9050 - val_loss: 0.2538 - val_acc: 0.8850
Epoch 194/200
- 0s - loss: 0.2074 - acc: 0.9000 - val_loss: 0.2198 - val_acc: 0.9000
Epoch 195/200
- 0s - loss: 0.1983 - acc: 0.9100 - val_loss: 0.2346 - val_acc: 0.8650
Epoch 196/200
- 0s - loss: 0.2017 - acc: 0.9050 - val_loss: 0.2284 - val_acc: 0.8800
Epoch 197/200
- 0s - loss: 0.2047 - acc: 0.9000 - val_loss: 0.2222 - val_acc: 0.8950
Epoch 198/200
- 0s - loss: 0.2003 - acc: 0.9000 - val_loss: 0.2177 - val_acc: 0.9000
Epoch 199/200
- 0s - loss: 0.1988 - acc: 0.9100 - val_loss: 0.2370 - val_acc: 0.8950
Epoch 200/200
- 0s - loss: 0.2083 - acc: 0.9000 - val_loss: 0.2244 - val_acc: 0.9000
```

Figura 9 – Resultado da aprendizagem por Redes Neurais

Os campos “acc” e “val_acc” representam a precisão do modelo ao ser avaliado nos dados de treino e de teste respectivamente. Como os dados de teste são àqueles desconhecidos para a máquina, é a partir deles que pode-se tirar conclusões do algoritmo. Nesse caso, ambos apresentaram 90% de precisão, superando a precisão da regressão logística.

4 CONSIDERAÇÕES FINAIS

O objetivo deste trabalho em analisar a utilização de técnicas de aprendizado de máquina para tomada de decisões, relacionadas a prevenção e manutenção de elementos mecânicos, foi atingido.

As técnicas de Regressão Logística e Redes Neurais se mostraram eficientes no diagnóstico de falha dos rolamentos. Para o exemplo deste trabalho, o modelo por Redes Neurais apresentou melhor precisão. Além disso, a convergência em 90% dos valores de precisão utilizando os dados de treino e os dados de teste nas Redes Neurais sinaliza que não houve sobreajuste neste modelo, indicando ser um bom modelo para avaliação de dados futuros.

No entanto, os resultados apontam ser possível elevar o grau de precisão da análise através da seleção de novos parâmetros adequados, sobretudo para evitar a intercessão entre diferentes classes, como foi visto na Figura 7.

REFERÊNCIAS

1. Li Y, Yang Y, Li G, Xu M, Huang W. A fault diagnosis scheme for planetary gearboxes using modified multi-scale symbolic dynamic entropy and mRMR feature selection. *Mech Syst Signal Process.* 2017; 91: 295±312. <https://doi.org/10.1016/j.ymssp.2016.12.040>
2. Case Western Reserve University Bearing Data Centre website. Disponível em: <<http://csegroups.case.edu/bearingdatacenter>>. Acesso em: 16 jul. 2018.
3. Hui, Kar Hoou; Ooi, Ching Sheng; Lim, Meng Hee; Leong, Mohd Salman; Al-Obaidi, Salah Mahdi. **PLoS ONE**. 12/20/2017, Vol. 12 Issue 12, p1-10. 10p. DOI: 10.1371/journal.pone.0189143
4. Simplifying Logistic Regression. DZONE. Disponível em: <<https://dzone.com/articles/machinex-simplifying-logistic-regression>>. Acesso em: 16 jul. 2018.
5. Logistic Regression. ML CHEATSHEET. Disponível em: <http://ml-cheatsheet.readthedocs.io/en/latest/logistic_regression.html#gradient-descent>. Acesso em: 16 jul. 2018.

6. Logistic Regression 3-class Classifier. SCIKIT-LEARN. Disponível em: http://scikit-learn.org/stable/auto_examples/linear_model/plot_iris_logistic.html. Acesso em: 20 jul. 2018.