PUC Rio

Machine Learning

Classificação e Categorização de Falhas em equipamentos Industriais

PROJETO DE FINALIZAÇÃO DE CURSO

Sumário

1 – INTRODUÇÃO

Quando aplicado à indústria, a Machine Learning é capaz de automatizar processos, otimizando operações e proporcionando um aumento de produtividade, qualidade e eficiência no desempenho das mais variadas funções.

Durante muitos anos essa nova vertente da tecnologia ainda passava por um processo de adaptação dentro das empresas e até um certo momento sofreu uma resistência quanto aos verdadeiros retornos com economia e custos operacionais.

Porém, assim como toda nova forma de se executar tarefas em uma corporação, essa inteligência teve que se adaptar e encontrar um ponto focal para que pudesse ser entendida e que os resultados encontrados pela IA, fosse mensurável e escalonado, foi então que houve a necessidade e criação do que é entendido como MLOps.

> MLOps (uma combinação de Machine Learning e "operações de tecnologia da informação") é uma nova disciplina / foco / prática para colaboração e comunicação entre Cientistas de Dados e profissionais de tecnologia da informação (TI), ao automatizar e produzir algoritmos de aprendizado de máquina. Por meio de práticas e ferramentas, o MLOps tem como objetivo estabelecer uma cultura e um ambiente em que as tecnologias de ML possam gerar benefícios comerciais, construindo, testando e liberando, de maneira rápida, frequente e confiável, a tecnologia de ML em produção.

1.1 – O Machine Learning atualmente

Já é sabido que a indústria como um todo está passando por diferentes transformações, advindas dos impactos da COVID-19 e foi nesse momento que a tecnologia mostrou sua real importância nesse momento em que já é entendido que o ML e IA nos auxiliou à realização de inúmeras tarefas cotidianas.

O setor de pesquisa e desenvolvimento busca constantemente encontrar soluções que possam ser a aplicação direta do ML no tanto Downstream como no Upstream dessa indústria.

De acordo com o relatório desenvolvido por Mordor Intelligence, como os custos relacionados a aquisição de sensores IoT declinou, diversas outras empresas que ainda não conseguiam adquirir estes, que são peça fundamental na comunicação e distribuição de dados entre equipamentos e interfaces de execução de IA. E então existe uma expectativa bem positiva para o setor de um aumento de cerca de 10,81% nos lucros produzidos por essa indústria entre 2021-2027, graças aos advindos proporcionados pela ML e IA.

2 – DESENVOLVIMENTO

A possibilidade de demarcar exatamente onde irá ocorrer as operações de perfuração, otimização das operações de perfuração, bem como manutenção preditivas e detecção de falhas, são algumas das possibilidades que o Machine Learning aplicado a indústria de O&G pode nos oferecer. Nesse projeto, vou demonstrar através de um modelo de Machine Learning como é possível, estruturar uma ferramenta preditiva, para evitar falhas em equipamentos, melhorando assim o seu desempenho.

No setor industrial e mais precisamente no de O&G o machine learning e a IA, se vez essencial principalmente no setor de comunicação e administração de documentações. Mas já antes dos impactos econômicos e relacionados a saúde pública que enfrentamos e ainda estamos vivenciando, existem algumas outras áreas em que o machine learning e a IA vem atuando e trazendo diversos benefícios.

2.1 - Previsão para Operações de Perfuração

RSI é um recurso disponibilizado em pgs.com contendo os principais recursos tecnológicos de imagens, para auxiliar empresas a montar e atualizar a sua estrutura de pesquisa no campo de exploração e construir modelos de ML que possam prever pontos exatos de perfuração.

2.2 - Otimizando Operações de Perfuração

Outra vantagem proporcionada pela IA, é a otimização de operações de perfuração, uma vez que a partir de dados histórico de operações já realizadas, é possível aprimorar e aprender novas formas de execução e evitar erros passados. Um bom exemplo de aplicações realizadas com IA, é a redução no tempo e custos com operações de conexão em perfuração utilizando CNN para diagnósticos, tabulação de resultados na construção de interfaces gráficas para análises cognitivas. Mais detalhes dessa tecnologia foram apresentados na Offshore Conference.

2.3 - Previsão de incidente com tubulações obstruídas de poços produtores

Recentemente, dois pesquisadores, Elahifar & Hosseini, criaram um modelo de ANN chamado *rede neural de otimização para enxame de partículas híbridas,* ou em resumo (PSO-based ANN), uma tecnologia de IA capaz de prever através de utilização de algumas variáveis específicas, tais como: peso da lama, ponto de rendimento e viscosidade plástica por exemplo, onde o modelo prevê com uma acurácia de até 80%, quando uma tubulação relacionada as operações de perfuração pode vir a ser obstruída.

2.4 - Manutenção Preditiva e Detecção de Falhas em equipamentos

De acordo com a Sociedade Internacional de Automação, cerca de 647 Bilhões de dólares são perdidos anualmente devido a paradas de produção relacionadas com falhas em equipamentos. A partir dos dados gerados por sensores instalados nesses equipamentos, e utilizando de modelos de IA preditivos, empresas estão podendo detectar falhas em equipamentos antes que elas venham a ocorrer.

XGBoost, LSTM e Autoencoders, são algumas das tecnologias de IA que podem estimar e diagnosticar as condições operacionais de equipamentos, prevendo e informando qual intervalo de tempo apropriado para que um serviço de manutenção seja realizado. Todos os modelos têm como base de apoio para sua construção, bibliotecas já conhecidas como TensorFlow e Keras.

3 – MODELO DE MACHINE LEARNING

Nesse projeto, utilizei um modelo que demonstra como classificar e outro para categorizar falhas em equipamentos. Através deste relatório, demonstro como ocorreu a criação de cada sequência de notebook, onde a proposta é responder algumas perguntas que servem como direcionamento a um objetivo específico, onde a criação das inferências foi feita a partir de um conjunto de dados brutos, ou seja, utilizando uma base de dados vinda de um banco específico, já pronta para que seja construído o modelo de Machine Learning a partir da mesma.

Para criação dos notebooks, utilizei o Google Colab e algumas bibliotecas como: Pandas e Scikit-Learn, para criação dos Modelos de Machine Learning e os gráficos categorizadores, Plotly como meio de criar séries temporais, onde os ciclos de funcionamento do equipamento é a base para criação da ***time-line*** para inferência dos resultados.

3.1 – XGBOOST

“XGBoost é uma abordagem poderosa para construir modelos de regressão supervisionada. A validade dessa afirmação pode ser inferida conhecendo sua função objetivo (XGBoost).

A função objetivo contém uma função de perda e um termo de regularização. Ele informa sobre a diferença entre os valores reais e os valores previstos, ou seja, quão longe os resultados do modelo estão dos valores reais.

A função de perda mais comum no XGBoost para problemas de regressão é reg: linear, e para classificação binária é reg: logistics.”

3.2 – DECISION TREE

A Árvore de Decisão é a ferramenta mais poderosa e popular para classificação e previsão. Uma árvore de decisão é uma estrutura de árvore semelhante a um fluxograma, onde cada nó interno denota um teste em um atributo, cada ramo representa um resultado do teste e cada nó folha (nó terminal) contém um rótulo de classe.

Construção da Árvore de Decisão: Uma árvore pode ser "aprendida" dividindo o conjunto de origem em subconjuntos, com base em um teste de valor de atributo. Esse processo é repetido em cada subconjunto derivado de uma maneira recursiva chamada particionamento recursivo. A recursão é concluída quando o subconjunto em um nó tem o mesmo valor da variável de destino ou quando a divisão não adiciona mais valor às previsões. A construção de um classificador de árvore de decisão não requer nenhum conhecimento de domínio ou configuração de parâmetros e, portanto, é apropriado para a descoberta de conhecimento exploratório. As árvores de decisão podem lidar com dados de alta dimensão. Em geral, o classificador de árvore de decisão tem boa precisão. A indução da árvore de decisão é uma abordagem indutiva típica para aprender o conhecimento sobre classificação.

Representação da Árvore de Decisão: As árvores de decisão classificam as instâncias através da raiz até algum nó folha, que fornece a classificação da instância. Uma instância é classificada iniciando-se no nó raiz da árvore, testando o atributo especificado por este nó e, em seguida, descendo a ramificação da árvore correspondente ao valor do atributo. Este processo é então repetido para a sub árvore enraizada no novo nó.

4 – INTERPRETAÇÃO DOS DADOS

A base de dados contém 10 atributos, onde o atributo "Fail" foi o utilizado como label para criação dos modelos classificadores.

5 – ANÁLISE EXPLORATÓRIA DA BASE DE DADOS

Após a realização da primeira análise exploratória, três principais aspetos foram observados na base de dados:

Shape da base de dados: 10 atributos e 800 linhas.

Missing Values: não existem dados faltantes na base.

Característica da base: Dados categóricos e dados numéricos na mesma base.

6 – CRIAÇÃO DE UM OBJETIVO ATRAVÉS DE QUESTIONAMENTOS

6.1 – Calcular quantas vezes ocorreu falhas no equipamento durante seu funcionamento.

Foi utilizado o recurso value_counts, bem como um gráfico de pizza com as proporções de falhas. Onde de acordo com o Dataset, ***False*** significa que não houve a falha e True onde houve a falha.

```
# Fazendo a contabilidade dos valores presentes no atributo antes da codificação
print(df['Fail'].value_counts())
```

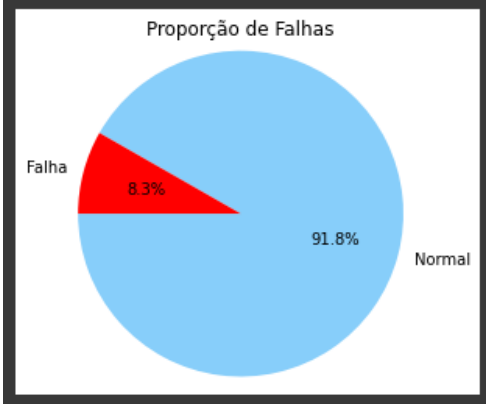
```
False    734
True      66
Name: Fail, dtype: int64
```

```
df_f = pd.DataFrame(df.Fail.value_counts())

status = ['Normal', 'Falha']

df_f.insert(loc=1,column='status', value=status)

labels = df_f.status
colors = ['lightskyblue', 'red']
plt.pie(df_f['Fail'], labels= labels,colors=colors, radius= 1.2, startangle=180, autopct='%.1f%%')
plt.title("Proporção de Falhas")
plt.show()
```



6.2 – Categorizar as falhas do equipamento de acordo com os atributos Preset_2 e Prest_1.

Para responder a essa pergunta, utilizei antes das inferências, um trecho de código, que me demonstraria qual é a ***correção*** entre esses dois atributos, antes de começar na criação das inferências e então escolher algum modelo de Machine Learning classificador mais apropriado com relação tanto a Classificação como também Categorização.

Como é possível observar na imagem abaixo, o grau de correção que esses dois atributos apresentam com relação a saída (label) que indicará a falha no equipamento é mínima. Logo, o mais coerente é utilizar um classificador numérico que possa categorizar as falhas por escala de criticidade.

	Cycle	Preset_1	Preset_2	Temperature	Pressure	VibrationX	VibrationY	VibrationZ	Frequency	Fail
Cycle	1.0	0.051	0.072	0.097	0.1	0.097	0.091	0.088	0.013	0.15
Preset_1	0.051	1.0	0.055	-0.0021	0.008	0.032	-0.063	-0.044	-0.055	-0.047
Preset_2	0.072	0.055	1.0	0.034	-0.013	-0.02	-0.0049	0.00018	0.03	-0.017
Temperature	0.097	-0.0021	0.034	1.0	0.42	0.24	0.36	0.38	0.17	0.27
Pressure	0.1	0.008	-0.013	0.42	1.0	0.44	0.39	0.26	0.38	0.35
VibrationX	0.097	0.032	-0.02	0.24	0.44	1.0	0.22	0.51	0.015	0.25
VibrationY	0.091	-0.063	-0.0049	0.36	0.39	0.22	1.0	0.51	0.47	0.46
VibrationZ	0.088	-0.044	0.00018	0.38	0.26	0.51	0.51	1.0	0.26	0.37
Frequency	0.013	-0.055	0.03	0.17	0.38	0.015	0.47	0.26	1.0	0.33
Fail	0.15	-0.047	-0.017	0.27	0.35	0.25	0.46	0.37	0.33	1.0

Antes de utilizar o modelo de regressão que consiga categorizar as falhas, entendi que poderia ser interessante codificar os dois atributos ***Preset 1&2*** utilizando o recurso ***one – hot – encoder***.

E logo após realizar essa codificação de transformação dos atributos, também codifiquei de categórico para numérico o atributo ***Fail.*** Essa última transformação de dados é de extrema importância, pois o modelo escolhido para a categorização foi o ***Gradient Boosting Regressor***, que realiza todas as inferências apenas com dados numéricos.

- **Gradient Boosting Regressor**

Este modelo de machine learning cria uma escala numérica entre as falhas e as categoriza em grupos distintos.

De acordo com a média absoluta entre os possíveis erros, é possível quando comparando ao anterior, gerar uma escala de criticidade entre as falhas observadas.

A partir da imagem abaixo é possível verificar qual foi a Média Absoluta entre os erros encontrada, e então, salvar o modelo de machine learning criado, para ser utilizado em produção como nova fonte de dados para inferências.

```
# Verificando o quanto de melhora foi gerada após a normalização.

Y_pred = best_regressor.predict(X_test)
mean_absolute_error(Y_test, Y_pred)
```

```
/usr/local/lib/python3.7/dist-packages/sklearn/base.py:451: UserWarning:
  "X does not have valid feature names, but"
0.1737645348837209
```

Como o objetivo do modelo é categorizar somente as falhas, realizei uma filtragem de forma criar um Dataset contendo apenas as falhasse verificar os resultados, logo, abaixo é possível verificar quais foram os valores de inferência encontrado.

```
df_f = pd.DataFrame(df_2.Resultado.value_counts().sort_index())

df_f = df_f.reset_index()

df_f = df_f.rename({'index': 'Fail_Category'}, axis=1)
df_f
```

	Fail_Category	Resultado
0	1.046296	12
1	1.072727	4
2	1.081967	29
3	1.110465	21

Esse resultado pode ser interpretado, de forma que quanto maior o valor para cada categoria criada pelo modelo, significa que essas falhas são menos “severas” pois esse valor leva em consideração os atributos escolhidos para criação do modelo (Preset_1 e Preset_2).

6.3 – Categorizar as falhas do equipamento de acordo com a causa natural dos parâmetros medidos (temperatura, pressão e outros).

- **Gradient Boosting Regressor**

Nessa parte do Projeto utilizei algumas das variáveis que depois de uma análise exploratória inicial foram consideradas mais críticas quanto a geração de uma falha pontual.

Possuindo média absoluta entre erros de 9,1%, o GBR, conseguiu categorizar as 66 falhas em três escalas de criticidade. Existe a possibilidade de utilizar a função ***loc*** (localizador), para localizar e analisar melhor as características de cada grupo distinto de falhas, e de alguma forma identificar uma maneira de diminuir essa recorrência, ou até mesmo evitar que elas voltem a ocorrer, levando assim a uma melhora e performance do equipamento avaliado.

```
# Verificando o quanto de melhora foi gerada após a normalização.

Y_pred = best_regressor.predict(X_test)
mean_absolute_error(Y_test, Y_pred)
```

```
/usr/local/lib/python3.7/dist-packages/sklearn/base.py:451: UserWarning:
  "X does not have valid feature names, but"
0.09189618644067812
```

```
df_f = pd.DataFrame(df_2.Resultado.value_counts().sort_index())
df_f = df_f.reset_index()
df_f = df_f.rename({'index': 'Fail_Category'}, axis=1)
df_f
```

	Fail_Category	Resultado
0	1.012712	8
1	1.174242	30
2	1.827586	28

6.4 – Criar um modelo de Aprendizado de Máquina Classificador e utilizar uma métrica para avaliação.

Nessa etapa, todos os atributos foram utilizados para o desenvolvimento do modelo classificador e para que o modelo pudesse entender a relação entre os dados, alguns deles tiveram que ser codificados e transformados. Nessa etapa o principal objetivo é criar um modelo que possa classificar as falhas dentro de um conjunto de dados geral.

- **Árvore de Decisão**

Abaixo é possível observar quais foram as métricas encontradas com relação ao modelo após o mesmo passar por teste, onde a acurácia aprimorada, bem como as outras duas métricas kappa e F1 podem ser observados na imagem abaixo.

```
predict_and_evaluate(X_test, Y_test)
```

```
Acurácia:  0.925
Kappa:  0.4597636465953855
F1:  0.4999999999999999
```

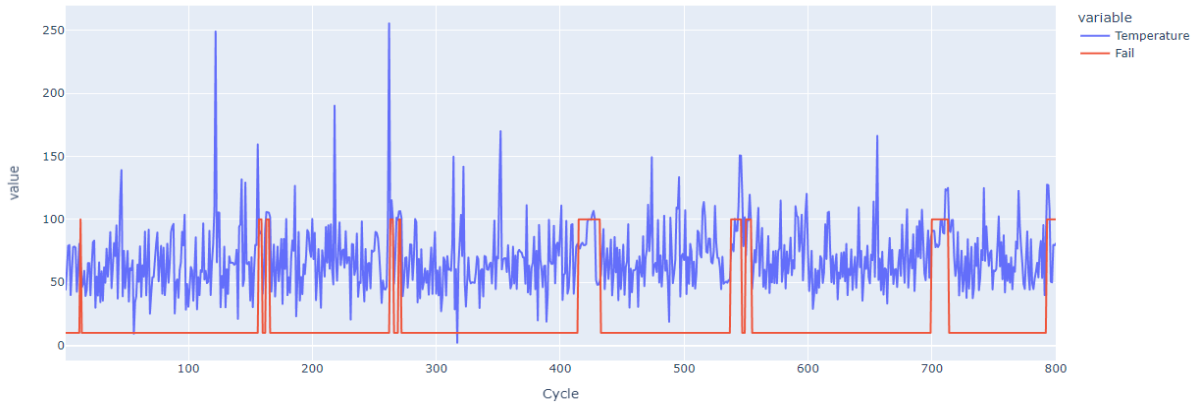
O modelo conseguiu classificar quase que totalmente as falhas.

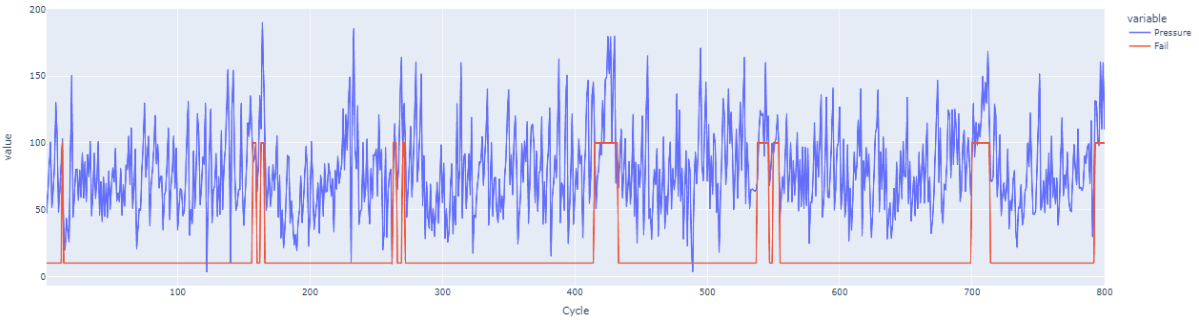
```
df_t = pd.DataFrame(df.Resultado.value_counts().sort_index())
df_t = df_t.reset_index()
df_t = df_t.rename({'index': 'Fail_Category'}, axis=1)
df_t
```

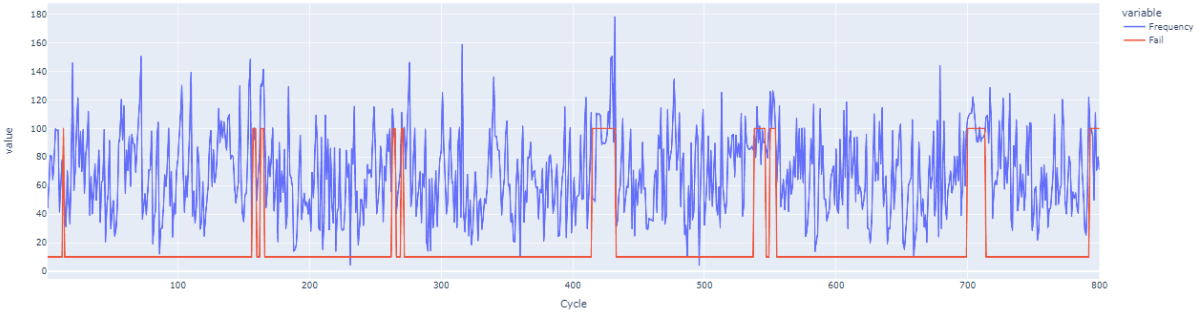
	Fail_Category	Resultado
0	False	737
1	True	63

6.5 – Avaliar a importância das variáveis.

Nessa etapa criei algumas séries temporais e contagens paralelas com análise estatística de correlação para demonstrar visualmente o comportamento de cada variável no ***time-line*** dos 800 ciclos.







```
# Valores Estatísticos da Pressão no Datased COM AS FALHA

P1 = df_WF['Pressure'].values

print('A pressão máxina é:',P1.max())
print('A Pressão mínima é:',P1.min())
print('A média da Pressão é:',P1.mean())
print('O desvio Padrão da Pressão é:',np.std(P1))
```

```
A pressão máxina é: 189.9956810944594
A Pressão mínima é: 50.8220022131057
A média da Pressão é: 116.42254028832454
O desvio Padrão da Pressão é: 29.81161064309384
```

```
# Valores Estatísticos da Pressão no Dataset SEM AS FALHAS

P2 = df_WOF['Pressure'].values

print('A pressão máxina é:',P2.max())
print('A Pressão mínima é:',P2.min())
print('A média da Pressão é:',P2.mean())
print('O desvio Padrão da Pressão é:',np.std(P2))
```

```
A pressão máxina é: 185.40640059067692
A Pressão mínima é: 3.480278691595548
A média da Pressão é: 75.63279041184637
O desvio Padrão da Pressão é: 30.54078946077085
```

7 – CONCLUSÃO

Após a criação dos dois modelos de ML, é possível entender que cada um deles conseguiu cumprir de forma adequada com sua função de categorização das falhas, onde é possível observar uma escala com relação a um valor ideal baseado no erro médio, sendo esse modelo o Gradient Boosting Regressor.

Quando foi necessário desenvolver um modelo que fosse capaz de classificar as falhas dentro de um conjunto de dados, a Árvore de Decisão conseguiu inferir e classificar falhas através do modelo criado com uma acurácia superior a 94%.

Um outro recurso de extrema importância complementando o desenvolvimento do trabalho, é a possibilidade de extrair os atributos dos grupos de falhas, em conjuntos distinto de dados e utilizar fórmulas estatísticas ou outra tecnologia para que as falhas mais graves possam ser evitadas.