



**Vanessa da Silva Simoes**

**Digital Platform for Smart maintenance in  
Logistic Vessels**

**Dissertação de Mestrado**

Thesis presented to the Programa de Pós-graduação em BI  
Master da PUC-Rio in partial fulfillment of the requirements for  
the degree of Mestre em BI Master .

Advisor : Prof. Dr. Leonardo Forero Mendoza  
Co-advisor: Prof. Cristian Enrique Muñoz Villalobos

Rio de Janeiro  
Dezembro 2018



**Vanessa da Silva Simoes**

**Digital Platform for Smart maintenance in  
Logistic Vessels**

Thesis presented to the Programa de Pós-graduação em BI Master da PUC-Rio in partial fulfillment of the requirements for the degree of Mestre em BI Master . Approved by the undersigned Examination Committee.

**Prof. Dr. Leonardo Forero Mendoza**

Advisor

Departamento de Inteligência Computacional Aplicada –  
PUC-Rio

**Prof. Cristian Enrique Muñoz Villalobos**

Co-advisor

Departamento de Inteligência Computacional Aplicada –  
PUC-Rio

Rio de Janeiro, Dezembro the 1st, 2018

All rights reserved.

**Vanessa da Silva Simoes**

Bibliographic data

Vanessa da Silva Simoes

Digital Platform for Smart maintenance in Logistic Vessels / Vanessa da Silva Simoes; advisor: Leonardo Forero Mendoza; co-advisor: Cristian Enrique Muñoz Villalobos. – Rio de Janeiro: PUC-Rio , Departamento de Inteligência Computacional Aplicada, 2018.

v., 60 f: il. color. ; 30 cm

Dissertação (mestrado) - Pontifícia Universidade Católica do Rio de Janeiro, Departamento de Inteligência Computacional Aplicada.

Inclui bibliografia

1. Inteligência Computacional Aplicada – Teses. 2. ICA – Teses. 3. Manutenção Inteligente;. 4. IoT;. 5. Inteligência Artificial;. 6. Logística;. 7. Plataforma Digital;. 8. Dashboards.. I. Pontifícia Universidade Católica do Rio de Janeiro. Departamento de Inteligência Computacional Aplicada. II. Título.

CDD: 620.11

Aos meus amigos e familiares pelo apoio desde o início do curso, em especial a minha irmã que passou os finais de semana estudando ao meu lado. Aos professores por conseguirem passar um conteúdo pertinente de forma clara e objetiva.

## **Acknowledgments**

Gostaria de agradecer ao corpo docente do programa do Business Intelligence Master, em especial ao professor Leonardo Mendonza pela orientação desse projeto e pelas aulas em Big Data e Inteligência Artificial e ao coordenador do curso Marco Aurélio Pacheco, por sua visão estratégica sobre o mercado de trabalho atual que permitiu modelar ferramentas analíticas num curso de pós graduação voltado a profissionais procurando se atualizar ou especializar numa área de interesse crescente.

Gostaria de agradecer aos meus amigos e familiares por terem me dado apoio quando decidi me inscrever no curso e por serem mais flexíveis e compreensivos nesses últimos meses onde tive menos tempo livre para acompanhá-los.

## Abstract

Vanessa da Silva Simoes; (Advisor); (Co-Advisor). **Digital Platform for Smart maintenance in Logistic Vessels**. Rio de Janeiro, 2018. 60p. Dissertação de mestrado – Departamento de Inteligência Computacional Aplicada, Pontifícia Universidade Católica do Rio de Janeiro.

This work is based on a business case and data analysis developed during the Hackaton - Libra Field Challenge, which occurred between 24 and 27 of September of 2018 regarding technological challenges in the Digital Libra project.

The knowledge in data analytics, statistics and artificial intelligence acquired during the BI Master was essential during the project proposal, development and presentation, and for that reason I decided to choose this theme for my final project. This work

This hackaton was developed by Petrobras, IBP and Fábrica de Startups and Iara was the winner selected by more than 20 managers in Petrobras and partners in the Libra Consortium.

In this work I will present the challenges found during the interviews with several segments in Petrobras, the motivation for selecting the maintenance for logistic vessels, the benefits for the companies, the market analysis, and finally the data analysis for clustering boats according to equipment degradation and a dashboard containing accessible information to be shared with managers and maintenance segment.

## Keywords

Inteligente Maintenance; IoT; Artificial Intelligence; Logistics; Digital Platforms; Dashboards.

## Resumo

Vanessa da Silva Simoes; ; . **Iara – Plataforma digital para Manutenção Inteligente de Barcos de Carga**. Rio de Janeiro, 2018. 60p. Dissertação de Mestrado – Departamento de Inteligência Computacional Aplicada, Pontifícia Universidade Católica do Rio de Janeiro.

Esse trabalho de conclusão de curso contém uma aplicação de análise de dados como ferramenta na proposta de solução do projeto desenvolvido durante o Hackaton Petrobrás – desafio Campo de Libra que ocorreu durante os dias 24 e 27 de Setembro e abordou os desafios na construção de soluções tecnológicas no âmbito do projeto Libra Digital.

O conhecimento de análise de dados e Inteligência Artificial adquirido durante o BI Master foi essencial na proposição do projeto, e em seu desenvolvimento e apresentação, e por esse motivo escolhi o tema como projeto final do curso.

O hackaton foi desenvolvido pela Petrobras em conjunto com o IBP e a Fábrica de Startup e Iara – Plataforma Digital para Manutenção Inteligente foi o projeto vencedor do qual sou uma das participantes.

Nesse trabalho apresentarei os desafios que encontramos durante entrevistas com diversos setores da Petrobrás, o motivo para a escolha do tema de manutenção de barcos de carga, os benefícios para as empresas envolvidas, a análise de mercado do projeto e finalmente a análise de dados e apresentação de dashboard para um protótipo feito com dados sintéticos

## Palavras-chave

Manutenção Inteligente; IoT; Inteligência Artificial; Logística; Plataforma Digital; Dashboards.

## Table of contents

1	Introdução	14
2	Hackaton: Desafio Campo de Libra	16
2.1	Desafio de Logística	17
2.1.1	Motivação do problema escolhido – Melhoria no controle das embarcações	18
2.1.2	Organização das entregas offshore	20
2.1.3	Problema a ser atacado e solução proposta	21
3	Canvas - Análise da Solução Proposta	24
3.1	Objetivo e Ferramentas utilizadas	24
3.1.1	Análise de Mercado	26
3.2	Custos e receitas	29
4	Análise de Dados	32
4.1	Pré Processamento	34
4.2	Carregamento e Limpeza	35
4.3	Treinamento	36
4.3.1	KNN	37
4.3.2	Árvore de Decisão	38
4.3.3	Random Forest	40
4.3.4	Redes neurais	41
5	Apresentação dos Dados para o Cliente	49
5.1	Apresentação dos dados para as produtoras de Petróleo	49
6	Conclusions	54
	Bibliography	55
A	Código em Python dos algoritmos apresentados no texto	57
A.1	Pré Processamento	57
A.2	Treinamento	58
A.2.1	Modelo de Redes Neurais	58
A.2.2	Modelo de Random Forest	59
A.2.3	Modelo KNN	59
A.2.4	Avaliação dos modelos	59

## List of figures

Figure 1.1	Relatório gerado pela Iara com dados simulados de frota de embarcações em operação (Retirado de apresentação do Pitch para Petrobras)	15
Figure 2.1	Passo a passo do desenvolvimento do produto, que se inicia com a coleta de dados offline e tem como visão dar sugestões de manutenção prescritiva.	19
Figure 3.1	Curva de Valor do Iara versus curva de valor de manutenção da Caterpillar	29
Figure 3.2	Curva de Valor do Iara versus curva de valor de manutenção da Petronect.	30
Figure 3.3	Relação entre custos fixos e variáveis.	31
Figure 4.1	Distribuição das amostras em 4 classes para o estado do compressor e quatro classes para o estado da turbina.	35
Figure 4.2	Matrix de confusão utilizando o método KNN com peso dado pela distância.	38
Figure 4.3	Trecho da árvore mostrando que ela tem 13 camadas e que não está balanceada.	40
Figure 4.4	Matriz de confusão obtida nos dados de teste usando árvore de decisão com segundo grupo de parâmetros.	41
Figure 4.5	Árvore de decisão com novos parâmetros e limite mínimo de 10 elementos por folha..	42
Figure 4.6	Matrix de confusão obtida com o método Random Forest, que apresentou ótimo desempenho.	43
Figure 4.7	Modelo de redes neurais com uma camada de ativação, sendo $x_1$ e $x_2$ entradas do sistema, e $y_1$ e $y_2$ saídas do sistema..	44
Figure 4.8	Arquitetura de uma rede neural com uma camada de input, duas camadas escondidas e uma camada de saída.	45
Figure 4.9	Arquitetura de camadas de uma rede neural full connected.	46
Figure 4.10	Evolução do desempenho da rede neural com otimizador Nadam, função de perda 'binary_crossentropy', batch size=150, e arquitetura de rede mostrada na Fig. 4.9.	47
Figure 4.11	Matriz de confusão para o modelo de redes neurais com otimizador Nadam e função de perda binary_crossentropy. O modelo apresentou 94% de acurácia nos dados de treino e somente confundiu classes vizinhas.	48
Figure 5.1	: Relacionamento da base de dados pelo, com chave indicada pelo Index.	51
Figure 5.2	Dashboard para apresentação dos barcos para as empresas petrolíferas, mostrando os barcos que farão paradas na próxima semana separado por empresa e por estado de desgaste dos equipamentos..	52

Figure 5.3	Informações dos sensores de leitura de uma embarcação com comportamento semelhante a média de todas as embarcações..	52
Figure 5.4	Informação de estado de uma embarcação com baixa velocidade em relação a frota.	53
Figure A.1	Matrix de confusão plotada segundo a função <code>plot_confusion_matrix</code> , apresentada em <code>scikitlearn</code> .	60

## List of tables

*"It is a capital mistake to theorize before one  
has data."*

**Sherlock Holmes, *A Study in Scarlet*(Arthur Conan Doyle).**

## List of Abbreviations

IoT – Internet das Coisas (*Internet of Things*)

ETL – Extrair, Transoformar, Carregar (*Extract Transform and Load*)

MVP – Produto Mínimo Viável (*Minimum Viable Product*)

GINSP – Grupo de Inspeção da Petrobrás

SOM – Mercado Total Obtível (*Serviceable Obtainable Market*)

SAM – Mercado Total Atingido *Serviceable Available Market*

TAM – Mercado Total Disponível *Total Available Market*

IBP – Instituto Brasileiro de Petróleo

# 1

## Introdução

Esse trabalho de conclusão de curso contém uma aplicação de análise de dados como ferramenta na proposta de solução do projeto desenvolvido durante o Hackaton Petrobrás – desafio Campo de Libra que ocorreu durante os dias 24 e 27 de Setembro e abordou os desafios na construção de soluções tecnológicas no âmbito do projeto Libra Digital, (7).

O conhecimento de análise de dados e Inteligência Artificial adquirido durante o BI Master foi essencial na proposição do projeto, e em seu desenvolvimento e por esse motivo escolhi o tema como projeto final do curso.

O hackaton foi desenvolvido pela Petrobras em conjunto com o IBP e a Fábrica de startup e Iara – Plataforma Digital para Manutenção Inteligente foi o projeto vencedor do qual sou uma das participantes.

Nesse trabalho apresentarei os desafios que encontramos durante entrevistas com diversos setores da Petrobrás, o motivo para a escolha do tema, os benefícios para as empresas envolvidas, a análise de mercado do projeto e finalmente a análise de dados e apresentação de dashboard para um protótipo feito com dados sintéticos.

A tecnologia apresenta-se atualmente como um aspecto constante em nossas vidas, pois estamos envolvidos por diferentes aparelhos, tais como celulares e computadores, além dos novos relógios, carros e, inclusive, casas inteligentes. O mesmo ocorre com os equipamentos e máquinas existentes na indústria náutica.

Todos esses aparelhos geram uma quantidade enorme de dados que são pouco analisados por conta das limitações humanas e computacionais. O processamento de dados efetuado de maneira constante e eficiente possibilitaria a análise em tempo real das condições do maquinário contido nos barcos e equipamentos náuticos.

A Iara foi criada com a intenção de solucionar esse problema. O algoritmo escrito nessa plataforma permite o monitoramento em tempo real das condições de funcionamento dos motores e equipamentos do barco (Fig. 1.1), o que permite a geração de relatórios. Esses relatórios auxiliam a tomada de decisão e logística da manutenção desses barcos, evitando quebras e diminuindo tempo de inoperância da frota.



Figure 1.1: Relatório gerado pela Iara com dados simulados de frota de embarcações em operação (Retirado de apresentação do Pitch para Petrobras)

Além do estudo de caso para propor uma solução a um problema de interesse da Petrobrás, também analisamos como viabilizar o projeto, através de análise de mercado, análise de possíveis colaboradores, como desenvolver um protótipo e quais funcionalidades seriam mais interessantes ao cliente. As seções abaixo contêm uma descrição sobre o Hackaton – desafio de Libra, os desafios apresentados durante o evento, uma introdução da solução proposta, análise de mercado, análise de custos, análise de riscos, e finalmente um protótipo desenvolvido através de uma base sintética disponível na internet com o objetivo de superar a falta de dados reais.

## 2

### Hackaton: Desafio Campo de Libra

Esse capítulo contém uma descrição sobre o Hackaton que ocorreu entre os dias 24 e 27 de Setembro de 2018, uma descrição dos desafios apresentados durante o evento, e uma descrição do desafio associado a solução proposta pelo projeto Iara.

Esse trabalho contém a organização de ideias geradas durante o Hackaton Petrobras organizado pela Fábrica de Startups. O programa consiste na formulação de um problema e proposição de uma solução que agregue valor a Petrobras. O problema escolhido e a solução proposta foram apresentados ao final de uma semana de trabalho para uma banca contendo gerentes de setores da Petrobras e de empresas parceiras no campo de Libra.

A Fábrica Start Up é uma empresa que tem como objetivo aproximar empresas grandes e pessoas com perfil empreendedor e inovador, eles montarão um espaço colaborativo na região do Porto do Rio de Janeiro com o objetivo de formar um ambiente para impulsionar startups e promover mentorias, trocas de contatos e apresentação a potenciais clientes e colaboradores.

O objetivo dos últimos eventos da Fábrica foi selecionar grupos de pessoas para iniciar esse projeto de colaboração entre startups e grandes empresas, e os ganhadores de cada edição foram convidados a passar 30 dias na Fábrica para ajudar a germinar startups de sucesso. Esse trabalho é fruto do projeto desenvolvido na semana do Hackaton, mais o trabalho de mentoria de um mês oferecido pela Fábrica de Start up unido a base de análise de dados adquiridos durante o BI Master, principalmente nas aulas de Redes Neurais e Big Data.

Os três desafios apresentados foram:

- Logística offshore para plataformas de Exploração, Perfuração e Produção
- Segurança Offshore
- Escritórios inteligentes

Todos os desafios estão relacionados ao campo de Libra (13), que teve seu primeiro leilão do Pré sal em 2013 e representa uma das maiores descobertas do Pré-Sal brasileiro. O campo de libra é formado por uma parceria entre a Petrobras (40

Libra está localizado em águas profundas, numa lâmina d'água de 2000 m, na Bacia de Santos, há 200 km da costa brasileira. E apresenta diversos desafios que serão enfrentados pelas empresas parceiras, e algumas das iniciativas vão em direção a utilização de tecnologias digitais (Libra digital) para melhoria de processos, para aumento de segurança operacional e para redução de custos.

Dentre esses temas o que mais chamou minha atenção foi Logística OffShore, devido ao potencial de utilização de ferramentas da indústria 4.0 de forma a reduzir custos e aumentar a eficácia do setor. A partir dessa escolha e de meu perfil acadêmico e profissional, foi selecionado um grupo formado por 4 pessoas, sendo duas pessoas com perfil comercial e duas com perfil técnico.

## 2.1

### Desafio de Logística

A Logística Offshore de plataformas de Exploração, Perfuração e Produção engloba a parte de armazenamento em terra, separação dos itens, unitização, transporte terrestre, armazém no porto, transporte por barcos de carga até a plataforma e transporte da carga de retorno da plataforma até terra, separação dos itens que voltaram da plataforma no armazém, para manutenção ou para descarte.

Nesse processo, alguns itens que representam problemas e seriam bons candidatos de um projeto para agregar valor são:

- os registros de acidentes e desvios
- cadastro de materiais
- condições marítimas
- falta de espaço no convés da unidade marítima
- estoque da unidade marítima
- rastreamento de carga
- gestão de contratos
- inoperância das embarcações
- falta de espaço no barco de carga para trazer a carga que não será mais utilizada na plataforma

Dentre os possíveis pontos apresentados, o que mostrou maior potencial de inovação através da indústria 4.0 foi a inoperância das embarcações. Tanto pelo possível impacto financeiro e de segurança, quanto pela falta de empenho nessa área no molde atual de contratação e planejamento.

### 2.1.1

#### Motivação do problema escolhido – Melhoria no controle das embarcações

A logística é um setor que não gera receitas e tem como objetivo ser eficaz e reduzir custos gerando o menor impacto negativo possível nos outros setores. Alinhados com esses objetivos decidimos focar em um sistema para aumentar a informação gerada pelos dados e para facilitar a transparência entre as empresas de barco e a Petrobras.

Para a entrega de cargas para as plataformas offshore, a logística da Petrobrás contrata 130 embarcações de carga que se dividem em 3 grupos regionais:

- Norte e Nordeste: 25 embarcações;
- Norte Fluminense: 76 embarcações;
- Sul: 37 embarcações.

Esses números são referentes a Dezembro de 2017 e sofreram alteração durante o ano de 2018. Esses barcos tem um custo médio de diárias de US 20000, além de diesel, equipe para revisão da embarcação no mar entre programações e barcos extras em regime de contratos temporários. O custo total com barcos representa 80

Ao conversar com gerentes da plataforma, gerentes de manutenção e grupo de apoio da Petrobras, constatamos que parte da inoperância é causada por quebras e falta de manutenção preventiva e falta de controle da saúde dos equipamentos do barco, tanto por parte da Petrobras quanto por parte dos proprietários dos barcos (estaleiros).

Devido a um regime intenso de operações que não permite manutenções regulares ao longo do ano, e prevê somente paradas longas de dois em dois anos e de cinco em cinco anos para manutenção e melhorias do barco, pela falta de comunicação entre a equipe de apoio e as empresas do barco em relação a presença de horas livres para manutenção entre programação e pela falta de um sistema organizado com a saúde dos barcos, a manutenção realizada é majoritariamente corretiva e feita quando o barco já está em situação crítica.

Como resultado desse fato, ao planejar o número de barcos contratados, o analista deve levar em consideração um termo de incerteza na disponibilidade dos barcos, que representa hoje uma redundância de 8

Durante as operações, quando um barco quebra e não pode operar, a Petrobrás não paga pela diária desse barco, portanto desses US80 M de inoperância calculados de antemão, parte é subtraído durante o ano e o custo gerado com diárias por inoperância no contrato comum é menor. No

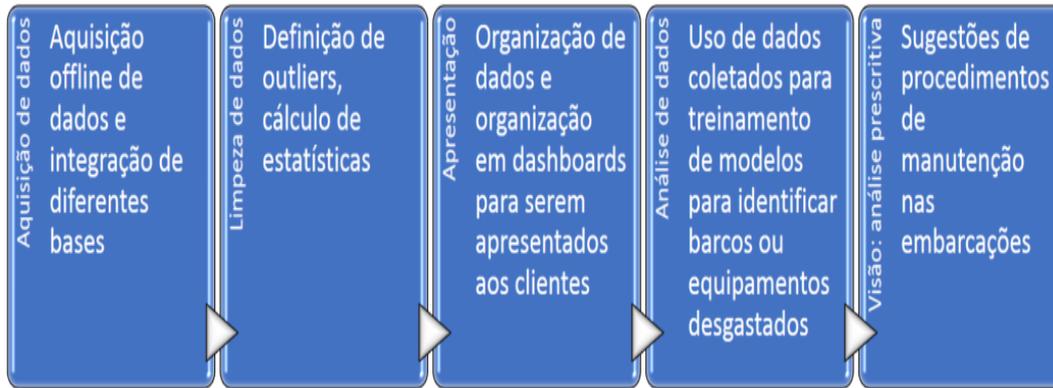


Figure 2.1: Passo a passo do desenvolvimento do produto, que se inicia com a coleta de dados offline e tem como visão dar sugestões de manutenção prescritiva.

entanto, custos adicionais tais como, possível contratação de barcos em caso de emergência por falta de redundância, atrasos nas entregas, e acidentes representam um custo adicional para a logística e para toda a Petrobrás difícil de mensurar e com alto valor potencial.

Especialmente em condições de mar adversas, os barcos precisam de potência total para não serem levados pela corrente e ir na direção da plataforma, causando abalroamento entre barco e plataforma. O capitão do barco tem mais informações sobre a saúde do motor que o informado a Petrobras e quando ele percebe que o barco não está apto para a operação ele comunica a impossibilidade de operar pelas condições de mar adversa que acata as recomendações sem oneração para a proprietária o barco quando as condições de mar realmente não são apropriadas para o tipo de embarcação em questão.

Analisando a conversa com os gerentes de plataforma, entre os itens acima, as dificuldades operacionais devido as condições de mar e as condições de barcos tiveram maior destaque que o custo com a logística, pois atrasos em entregas podem atrasar operações críticas nas embarcações, esses atrasos ficam mais críticos durante o início do segundo semestre do ano, pois nessa época as condições de mar são mais adversas, sendo imprescindível ter barcos com motores com potência total para manter a posição durante as operações.

Outro impacto levantado pelos gerentes de plataforma foi o efeito de atrasos nas próprias operações de logística, pois quando existe atraso nas entregas, o restante do estoque que deveria ser levado de volta a terra continua em uso até que o barco chegue. Essa parte é preparada, para transporte no momento de chegada do barco, como o convés não fica preparado como deveria, isso gera maior atraso de embarque e desembarque de cargas, e possivelmente

volume de carga de volta para terra maior que o previsto antes da saída do barco, o que desorganiza o planejamento feito antes da saída do porto.

Esse será um ponto crítico para as plataformas em Libra, pois algumas possuem convés ainda menor que as plataformas atuais. E além disso, devido a posição geográfica, o campo de Libra terá os mesmos problemas devido a condições de mar, e é imperativo que os barcos sejam mantidos em boas condições.

### 2.1.2

#### **Organização das entregas offshore**

Com a finalidade de entender melhor como e realizado a manutenção das embarcações, o que existe de dados disponíveis sobre os barcos, e como esses dados são utilizados, fizemos entrevistas com o gerente de manutenção de uma empresa de barcos de carga prestadora de serviço, com o gerente do transporte marítimo de Macaé, e com um contratado do grupo de inspeção de embarcações (GINSP).

Esse grupo de inspeção faz inspeções nas embarcações com a finalidade de verificar se o barco está em condições de fazer a viagem e se a documentação está em dia. Existe uma equipe com escala de trabalho no campo com adicional que poderia ser mais efetiva se houvesse um controle de desempenho do barco e estado do equipamento.

Com essas entrevistas ficou claro que em algumas ocasiões existem dados de sensores nos barcos tais como orímetro, temperatura e pressão, mas que esses dados não são utilizados para auxiliar na manutenção e acompanhamento dos barcos. Além disso, que a manutenção é feita em mais de 90

Parte do problema está no sistema de contratação dos barcos que poderia ter mais informações de indicadores, como nota de desempenho organizada por embarcação e por empresa. Isso ajudaria a filtrar empresas dentro de um certo nível de qualidade mínimo e ajudaria a assegurar a manutenção do nível de qualidade durante a duração do contrato.

Outro fator decisivo é a falta de comunicação entre o pessoal que capta os dados e a equipe de manutenção da empresa de barcos, sem esse repasse de dados a manutenção não é otimizada, o que poderia causar uma redução nas quebras de barcos e manutenção da potência dos motores dos barcos em níveis próximos ao ótimo.

Hoje em dia um na empresa que perguntamos, existe um rastreamento de quebras que tenta investigar o que ocorreu, mas devido a necessidade de colocar o barco para funcionar o mais rápido possível, a análise é feita durante o conserto e em algumas vezes isso atrapalha a investigação. Outro fator

importante na coleta dos dados, seria investigar o histórico de alterações de sensores até a quebra, o que ajudaria na investigação.

Durante as entrevistas, as principais causas para quebras de barcos, que causaram mais de 80

- defeito no motor principal
- defeito no motor auxiliar
- defeito na parte de radar e navegação
- defeito no sistema de geração de energia
- defeito no sistema de refrigeração

A manutenção da empresa também indicou que os dados dos sensores chegam como planilha de Excel mais tarde e não há equipe de análise para interpretação desses dados e que corte de custos é uma das restrições das empresas, que geralmente são empresas pequenas funcionando com o menor número de funcionários possível.

### 2.1.3

#### **Problema a ser atacado e solução proposta**

Como problema vimos o custo com frotas redundantes devido a uma inoperância de 8

Para reduzir esse problema propomos uma integração de dados dos sensores de temperatura e pressão já instalados nos barcos em tempo real, juntamente com os dados de localização dos barcos e condição de mar e dados com gastos de diesel.

Com esses dados pretendemos durante o primeiro ano de projeto coletar séries históricas e fazer uma análise estatística com indicadores de distribuição de temperatura, pressão e consumo de diesel em diferentes condições de mar e diferentes velocidades. Analisar como esses valores variam de acordo com o tipo de barco e a idade dos barcos, e verificar quais sinais são apresentados antes de quebra de partes do equipamento.

Além disso, propomos um sistema para o grupo de apoio que planeja as programações de barcos, para que eles reportem quando ocorrerá pausas de horas entre programações, e propomos que a Petrobras se comprometa a permitir pausas para manutenções programadas e que a empresa se comprometa a devolver o barco na hora especificada de antemão.

A equipe de manutenção de barcos receberá relatórios das condições dos sensores, da distribuição de velocidades, da presença de trabalhos em condições de mar adversa, e gastos com diesel, de forma organizada em gráficos

ilustrativos, para que essa informação ajude a definir quais equipamentos e quais barcos merecem maior atenção. Eles também receberão a informação do grupo de apoio com as pausas previstas entre programação para que eles tenham tempo de fazer melhorias nos barcos sem multas por inoperância.

O grupo de inspeção de embarcações (GINSP) também receberá essas informações diárias dos barcos, em especial dos barcos que atracarão naquele dia e dos que terão horas vagas entre programação, para certificar que as empresas estão fazendo a manutenção devida e que essa será reportada no sistema.

Durante o primeiro ano de coleta de dados, as informações dos diferentes barcos serão utilizadas para montar um banco de dados com:

- dados dos sensores de temperatura e pressão;
- dados de velocidades;
- dados de localização;
- dados de condições de mar;
- dados de consumo de diesel;
- dados de capitão do barco;
- dados de trocas de óleo;
- dados de manutenção de histórico de quebras.

O primeiro objetivo será treinar uma máquina para medir desgaste em partes da máquina do barco com o objetivo de auxiliar a manutenção preventiva.

O segundo objetivo será treinar uma máquina para prever futuras quebras em partes do barco com o objetivo de auxiliar na manutenção preditiva.

O terceiro objetivo será treinar uma máquina com possíveis soluções para a equipe de manutenção e para o capitão para aumentar a vida útil dos equipamentos e facilitar a realização e controle das manutenções.

O objetivo do trabalho é transformar os dados dos sensores em informação e aumentar a eficiência da frota e a Fig. 2.1 contém um passo a passo da visão que temos desde o protótipo até o objetivo final.

O aumento do controle da frota trará uma redução de custos para a Petrobrás, pois isso levará a uma redução com a contratação de barcos devido a menor incerteza de inoperância, também levará a um aumento da segurança e confiabilidade nas entregas, pois ocorrerá uma redução de quebras de barcos em trânsito e também a redução de chance de colisão entre plataformas e barcos causados por falta de potência nos motores.

Para as empresas de barcos, os relatórios e as pausas para manutenção representarão uma redução de custos com barcos redundantes e evitará o não recebimento de diárias em casos de inoperância.

Devido a queda de operações no Brasil, o mercado de aluguel de barcos de transporte de cargas está competitivo e a Petrobras tem poder de negociação para especificar no contrato o nível de qualidade exigido durante o contrato e os dados necessários, e como as empresas querem reduzir custos e quebras e manter o cliente, é importante que elas se adaptem a uma modernização no sistema de manutenção.

Como ambas as partes serão beneficiadas pelo maior controle da frota, haverá corte de custos juntamente com aumento da qualidade do serviço prestado pela logística, essa é uma proposta sustentável e pode ser implementada em momentos de cortes de gastos como o momento atual e mantido durante um longo tempo desde que ambas as partes cumpram com as responsabilidades da solução.

## 3

### Canvas - Análise da Solução Proposta

#### 3.1

##### Objetivo e Ferramentas utilizadas

Essa seção contém a proposta de valor, contendo o mapeamento dos clientes mais importantes e possíveis colaboradores.

A proposta de valor é oferecer uma plataforma integrada de apresentação de informações organizadas obtidas através de sensores dos barcos de carga para evitar quebras inesperadas durante operação e para otimizar a manutenção dos barcos.

O problema a ser resolvido é a falta de eficiência na gestão de manutenção de barcos de logística offshore, pois tanto as empresas de barco quanto o setor de logística e manutenção da Petrobrás reclamaram sobre quebra de barcos frequentes e falta de transparência entre o capitão do barco com a equipe onshore sobre a real situação do barco. Isso acarreta em quebras inesperadas, abalroamento devido a motor com potência reduzida e frota excedente para evitar transtornos.

O objetivo final da primeira fase do projeto é obter uma forma integrada para que os dados dos sensores sejam apresentados tanto as empresas de barco quanto a operadora de forma simplificada em dashboard e com indicativos de alertas simples para viabilizar a manutenção preditiva das embarcações pelos armadores e controle da situação dos barcos pela operadora.

O objetivo final após a construção dessa plataforma é desenvolver um Market place para aluguel de barcos, facilitando a relação entre armadores e produtoras offshore e durante o período de aluguel manter ambos os lados informados sobre o estado dos barcos e nossa fonte de receita desejada nesse sistema é uma comissão da diária do barco.

Os recursos chave são os equipamentos de sensores e transmissão dos barcos, espaço em nuvem para armazenamento e processamento, plataforma digital para integração de dados, e programadores para limpeza, integração, processamento e organização de dados.

As atividades chaves do negócio são: integração, processamento, organização e apresentação e distribuição de dados.

Para comunicar a proposta ao cliente será apresentado o MVP (Minimum Viable Product) ao cliente em Fevereiro de 2019, esse resultado será obtido através de programação em Python e apresentação dos dados em dashboard em Power BI.

A comunicação com clientes e possíveis parceiros será via e-mail, primeiramente para um dos mentores do hackaton que trabalha com logística e para um dos juris da Petrobrás que nos incentivou durante o Hackaton. Após o feedback, pretendemos fazer um website apresentando nossos serviços e fazer parte de eventos marítimos e submeter o projeto para edital de inovação em 2019.

A comunicação com parceiros até o momento foi através de conversa durante a Rio Oil and Gas, posterior troca de e-mail e visita ao estaleiro Mauá e tele-conferência por internet com a Athos. Até o momento os parceiros se mostraram interessados e dispostos a colaborar

Como potenciais parceiros no nosso projeto temos as plataformas digitais com capacidade para IOT e os armadores que já oferecem serviço de manutenção e podem otimizar esse serviço através da utilização de dados: Possíveis parceiros em plataformas digitais e IoT:

- Psystem- Osisoft: já utilizada pela Petrobras;
- Matrikon: para ler o protocolo;
- Mindsphere - athos; para IoT, ETL e normalização de dados.

Estaleiros:

- Estaleiro Atlântico Sul;
- Vard Promar;
- Engevix Construções Oceânicas;
- Estaleiro ESA;
- Estaleiro Mauá;
- Oceano Estaleiro;
- Estaleiro São Miguel;
- Enavi Reparos Navais;
- Estaleiro Brasa;
- Estaleiro Mac Laren;
- Estaleiro Rio Maguari.

### 3.1.1

#### Análise de Mercado

Para definir o mercado, utilizamos as definições de

- SOM (Serviceable Obtainable Market);
- SAM (Serviceable Available Marke) ;
- TAM (Total Available Market).

O público Som ainda é inexistente, pois ainda não conseguimos obter clientes com as análises que já foram realizadas.

O público SAM são os barcos de transporte de carga offshore no Brasil, somente a Petrobrás aluga 123 desses barcos para a logística de plataformas, e cada barco tem uma diária de 20K dólares. E isso representa mais da metade do valor gasto com logística de plataformas.

O público TAM são todos os barcos de transporte offshore do mundo. Se o Market place funcionar. Esse é o mercado total, mas não é o interesse do projeto.

Analisando o mercado, avaliou-se que um dos desafios encontrados em relação ao mercado é o poder de barganha do maior potencial cliente: Petrobrás, por ser dominante no mercado. A Petrobras geralmente dita as regras do jogo e essas regras estão sujeitas a alterações, por isso expandir o mercado para outras empresas parece necessário para um planejamento de longo prazo.

Por outro lado, os armadores atualmente estão com pouco poder de barganha e precisando otimizar seu funcionamento devido à concorrência durante o mercado em baixa, o que torna nossa proposta mais atraente e aumenta as chances de colaboração para fazer um MVP.

Ainda em relação a situação política e econômica, estamos num ótimo período para propor esse projeto, pois o corte de custos e a necessidade de modernização da indústria se torna mais evidente durante os períodos de crise, e nós estamos propondo uma forma sustentável de reduzir custos e aumentar a qualidade do serviço prestado pelos armadores para a Petrobrás, e pelo setor de logística para as plataformas.

Como principais clientes, colocamos as empresas de extração de petróleo com atividades offshore do Brasil e os armadores com barcos de carga:

Cliente com interesse em obter controle sobre produtos alugados com impacto em operações: Extratoras e Operadoras:

- Petrobras
- Chevron

- Shell
- Modec
- Andrade Gutierrez
- Queiroz Galvão
- Teekay
- Sete Brasil
- Seadrill
- BP
- Total
- Exxon

Cliente com interesse em obter redução de custo com manutenção e reduzir riscos de pagar multas por quebras inesperados: Armadores:

- Transpetro
- Aliança Navegação
- Maersk
- Arsenal da marinha
- CBO
- CIA de Navegação Norsul
- Empresa de Navegação Elcano
- Norskan/DoF
- Wilson Sons Armadores:
- Transpetro
- Aliança Navegação
- Maersk
- Arsenal da marinha
- CBO
- CIA de Navegação Norsul
- Empresa de Navegação Elcano
- Norskan/DoF
- Wilson Sons

Riscos do projeto	Nota associada ao risco
Poder de barganha dos clientes	5
Poder de barganha dos fornecedores	4
Ameaça de novos entrantes	4
Ameaça de produtos substitutos	1
Rivalidades entre concorrentes	1

Para mitigar os riscos durante o projeto, foi feita um mapeamento de riscos de mercado do projeto e os de maior impacto para o sucesso do projeto estão apresentados na tabela abaixo:

Para comparar o projeto concorrentes diretos, utilizou-se a definição de curva de valor, que foi apresentada pelos autores W. Chan Kim e Renee Mauborgne no livro *Blue Ocean Strategy* (2), e descreve como oceano vermelho as empresas que trabalham com commodities oferecendo produtos semelhantes a concorrência e mantendo clientes através de menor valor, enquanto no oceano azul as empresas podem agregar valor ao produto devido ao diferencial, tal como o Cirque de Soleil que não concorre diretamente com outros circos.

Como concorrentes diretos do projetos existem dois seguimentos já atuantes no mercado, primeiramente a Caterpillar que fabrica grande parte dos motores dos barcos no Brasil e possui serviço de manutenção desses motores e segundo a Petronect que é um portal para vendas para a Petrobrás e tem como foco principal a venda de produtos consumíveis. A integração dos dados que desejamos oferecer e a transparência sobre o estado das embarcações ainda não é oferecida por nenhuma dessas empresas no momento atual, e é nesse gap que o projeto se mostra útil.

A curva de valor em relação ao concorrente Caterpillar está esquematizada a na Fig. (3.1):

O grande diferencial entre o projeto e a Caterpillar é a transparência de informação que está prevista no projeto, pois hoje em dia mesmo com os sensores lendo a informação nos barcos, essa não é passada para a equipe em terra ou para a empresa que contrata os barcos. Já nossa curva de valor em relação à Petronect encontra-se no esquema em Fig. 3.2:

Nosso principal diferencial é o design e a relação entre operadora e armadores, além do conteúdo local que é um elemento importante durante a contratação pela Petrobrás.

Nesse dia encontramos uma lista de empresas com trabalhos e patentes na áreas de manutenção preditiva através da análise de dados de sensores que está listada abaixo:

- Smartsignal Corporation;
- Hitachi, Ltd.System;

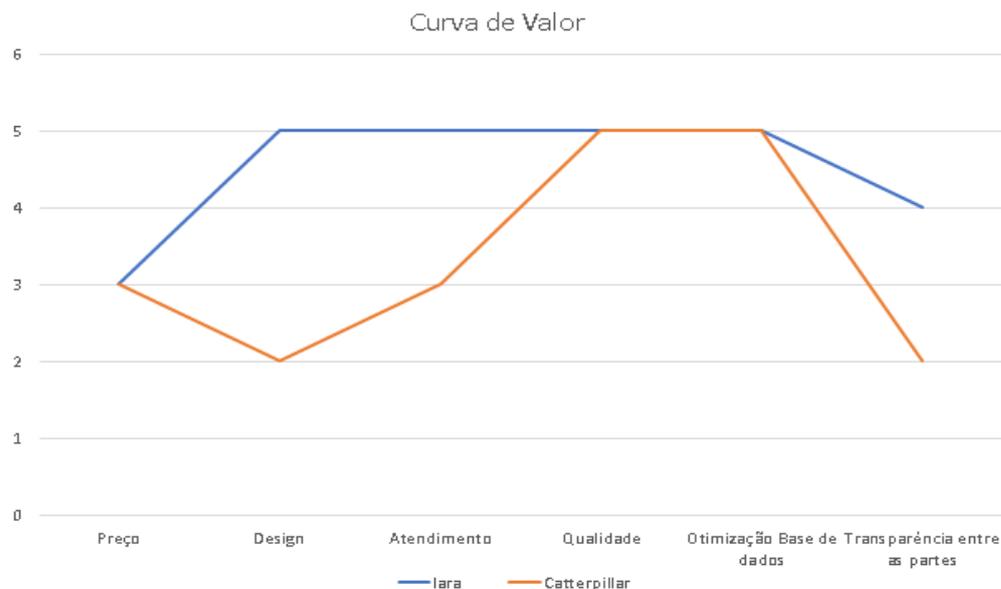


Figure 3.1: Curva de Valor do Iara versus curva de valor de manutenção da Caterpillar

- Accenture Global Services Limited;
- Baiju Shah;
- Applied Materials, Inc;
- Caterpillar Inc;
- The Boeing Company -System health operations analysis model.

Dessas empresas aparentemente só a caterpillar trabalha com análise para manutenção preditiva de barcos, sendo que a Caterpillar trabalha com manutenção preditiva de motores fabricados pela própria empresa, conforme as informações obtidas no estaleiro, a Caterpillar é a maior fornecedora de motores para os barcos de carga no Brasil devido ao preço mais baixo que os concorrentes, e ela possui alto custo de manutenção em comparação com os demais fornecedores.

### 3.2 Custos e receitas

Outro aspecto importante do projeto é entender quais os custos associados durante a execução, através de um passo a passo da metodologia a ser utilizada. Para entender quais os custos mínimos para iniciar o projeto e para manter, é importante dividir esses custos em fixos e variáveis, sendo que os custos variáveis dependerão da abrangência do projeto, e os custos fixos estarão

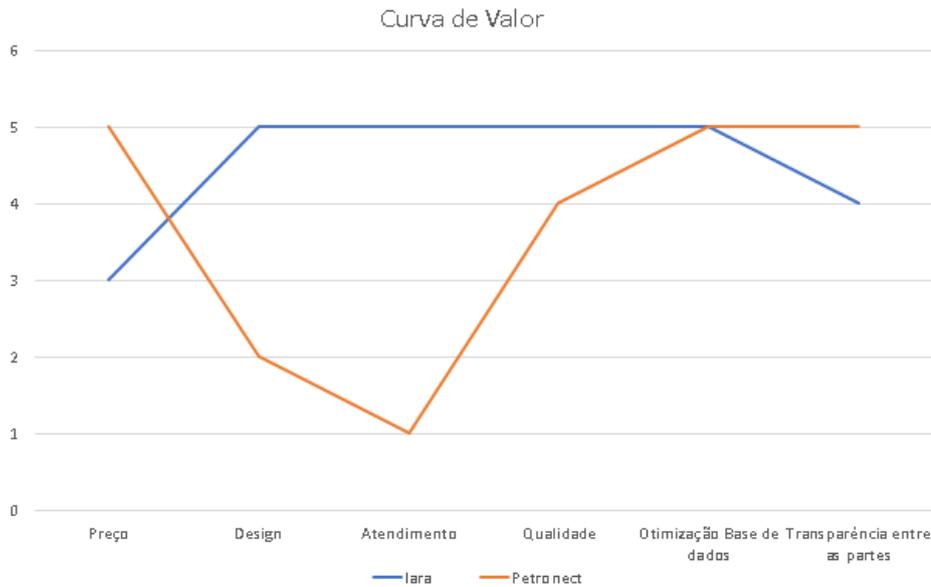


Figure 3.2: Curva de Valor do Iara versus curva de valor de manutenção da Petronect.

presentes mesmos quando não houver clientes, e por isso devem ser reduzidos ao mínimo

Os custos iniciais para a primeira fase do projeto de captação de dados, análise e apresentação dos dados está apresentado no esquema abaixo.

Com o objetivo de minimizar custos fixos iniciais, pode-se utilizar espaço em nuvem somente para aluguel durante treinamento e usar programadores independentes no desenvolvimento.

Um modelo proposto para receita é o modelo de assinaturas, onde os armadores param uma taxa para obter as análises por um serviço básico, o que nos tornaria capazes de captar uma quantidade de dados necessárias para fazer análise em dados reais e oferecer um serviço mais avançado com análise preditiva e prescritiva.

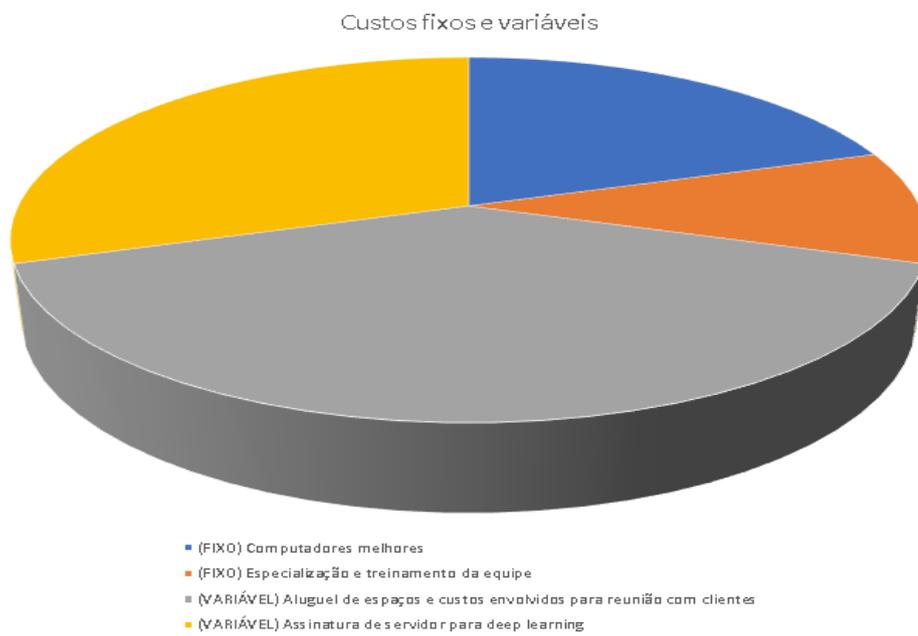


Figure 3.3: Relação entre custos fixos e variáveis.

## 4

### Análise de Dados

Apesar dos encontros com a Petrobrás e com o estaleiro Mauá e de grande interesse por parte dos mesmos, não foi possível obter dados reais de sensores de embarcações somente uma base de dados sintética obtida no site [www.cbm.smartlab.ws](http://www.cbm.smartlab.ws). Os dados foram gerados por Andrea Coraddu(b), Luca Oneto(a), Alessandro Ghio(a), Stefano Savio(b), Davide Anguita(a), Massimo Figari(a), através de um simulador numérico de barcos caracterizado por uma planta de propulsão com Turbina a Gás, onde os diferentes blocos do equipamento formam um simulador completo (propulsor, casco, caixa de velocidade, e controlador). Os resultados foram comparados com dados reais e é esperado que esses dados reflitam a realidade de embarcações.

Esses dados podem ser utilizados com a condição de citação de duas referências (3) e (4) onde eles foram utilizados:

Nesse simulador, o comportamento do sistema de propulsão foi descrito com os parâmetros:

- Velocidade do barco (lp)
- Coeficiente de degradação do compressor (kMc)
- Coeficiente de degradação da turbina (kMt)

E nos modelos de análise de dados, focamos nos coeficientes de degradação da turbina e do compressor, pois a velocidade é um dado que pretendemos obter de GPS a partir do sistema de posicionamento já disponível pela Petrobrás, e será usado para criar um alerta automático. Na base original, os coeficientes de degradação foram apresentados no intervalo  $[0.95, 1]$  e  $[0.975, 1]$  e com alta granularidade.

Para cada medida está disponível 16 séries de inputs, dentre esses 16 utilizamos 14 séries como entradas do modelo de inteligência artificial. Os 14 dados estão listados a seguir:

- posição da alavanca;
- torque do eixo da Turbina a Gás;
- taxa de revolução do gerador de gás;

- torque de hélice de estibordo;
- torque da hélice de porta;
- Saída de temperatura da turbina de alta pressão;
- Temperatura do ar de entrada do compressor;
- Temperatura do ar de saída do compressor;
- Saída de pressão do compressor;
- Saída de pressão da turbina;
- Pressão do ar de entrada no compressor;
- Pressão do ar de saída do compressor;
- Pressão do gás de escape;
- Controle de Injeção de Turbina;
- Fluxo de combustível.

Como saída temos os estados de decaimento do compressor e da turbina

- Coeficiente de degradação do compressor;
- Coeficiente de degradação da turbina.

O dataset foi distribuído em 3 arquivos de texto, e toda a descrição é baseada na informação encontrada no `Readme.txt` e da descrição da série em `Features.txt`.

Os dados são apresentados como uma matriz de 18 colunas no arquivo `data.txt` e não estão normalizados.

Estado do Compressor	Estado da Turbina	
Velocidade	0	0
GTT	0.003	0.000
GTh	0.001	0.000
GGn	-0.019	0.010
kN	0.001	0.000
turbtemp	0.001	0.000
comprTemp	-0.04	-0.038
GTinletTemp	-0.047	-0.017
GtCompOutTemp	0.008	-0.003
CompressAirPre	0.008	-0.018
Pexh	0.035	0.012
TIC	-0.032	-0.019
FuelFlow	-0.014	0.017
Estado do compressor	1	0
Estado da Turbina	0	1

## 4.1

### Pré Processamento

Todo o pré-processamento, limpeza e análise de dados foi feita em Python utilizando Jupyter, durante o pré-processamento os dados foram armazenados num data frame em pandas.

O primeiro passo foi analisar a matriz de correlação linear entre as variáveis de input e os estado do compressor e da turbina. Essa matriz de correlação está apresentada abaixo e é possível notar que a velocidade, o estado da turbina e o estado do compressor não tem correlação linear nesse dataset, e que a dependência entre as variáveis de input e output não são lineares:

Após fazer o carregamento dos dados, foi necessário separar entre input e output, X e Y.

Além disso, o estado da turbina e compressor são curvas contínuas no dataset original e para facilitar a modelagem. As duas curvas foram divididas em 4 classes cada, aproximadamente balanceada, sendo que todas as classes possuem pelo menos 2500 amostras como pode ser visto na Fig. 4.1 contendo o histograma de distribuição das amostras por cada uma das classes.

As 4 classes de desgaste podem ser pensadas como novas, quase novo, gastos e muito gastos. E uma maior atenção deve ser dada a equipamentos muito gastos. O valor do output, dividido em classes foi atribuído a variável Ycat.

Outro fator necessário é a normalização dos inputs, para isso utilizamos a função  $\text{Normalize}(X)$  apresentada no Appendix, que [e uma função afim que coloca todos os pontos entre -0.5 e 0.5.

Para validação do modelo fizemos uma divisão do input em 66% treino

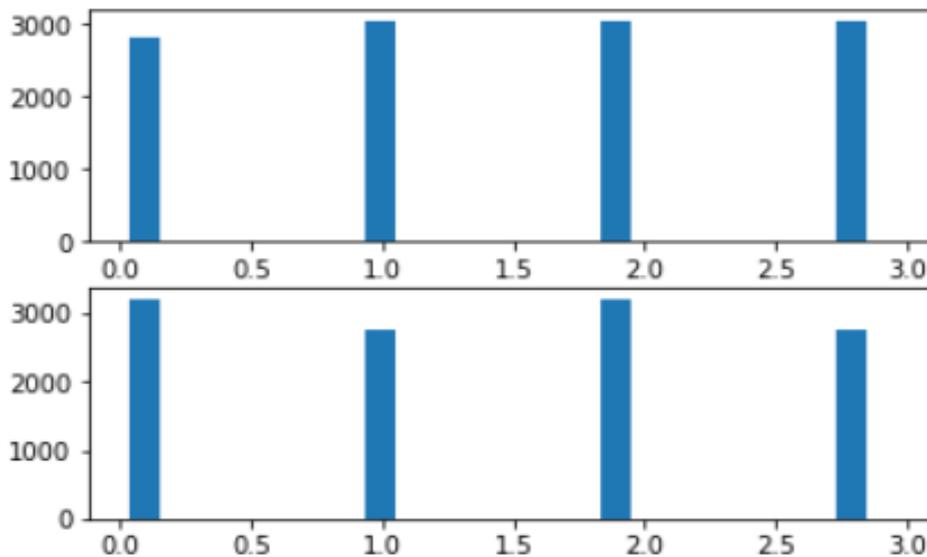


Figure 4.1: Distribuição das amostras em 4 classes para o estado do compressor e quatro classes para o estado da turbina.

e 33% teste utilizando a função `train_test_split` do `sklearn.model_selection`, como abaixo.

```
Xtrain, Xtest, ytrainCat, ytestCat = train_test_split(Xnorm, Ycat,
test_size=0.33, random_state=42).
```

A última etapa do pré-processamento foi transformar o output categórico em vetores binários usando `to_categorical` do `keras.utils`. Essa etapa foi necessária para treinamento de uma rede neural para agrupamento supervisionado com 4 classes.

## 4.2

### Carregamento e Limpeza

Nesse projeto, a primeira etapa será coletar os dados, organizar, filtrar e apresentar.

Num primeiro momento quero obter os dados reais para aprender a carregar dentro do sistema, montar uma estatística de distribuição das variáveis, entender a correlação entre os diferentes inputs, tais como velocidade e diferentes medidas tais como consumo do veículo, posicionamento da alavanca, temperatura do equipamento, e taxa de rotação. Nos dados sintéticos, esses dados eram bastante correlacionados e queremos estudar se desvios nessas correlações podem estar associados a condição do barco.

Enquanto essas correções não forem obtidas, podemos apresentar aos

clientes apenas as leituras, assim como apresentado na seção: “Apresentação dos Dados para o Cliente”, o que já seria mais que o disponível no sistema atual.

Durante essa etapa além de tratar os dados, é importante aprender como integrar dados disponibilizados através do Protocolo OPC, e como utilizar a Mindsphere: Plataforma, que pode servir como acelerador para a pré análise de dados.

Enquanto não temos acesso a uma base de dados reais, utilizaremos a base sintética já descrita aqui. Para carregar os dados, filtrar as colunas que não tinham variação, normalizar, transformar em categórico e separar em treino e teste utilizamos os comandos em Python apresentados no Apêndice.

A partir desses comandos conseguimos montar as variáveis `X_train`, `Y_train`, `X_test` e `Y_test` a serem utilizados nos modelos de inteligência artificial propostos a seguir.

### 4.3 Treinamento

A próxima etapa depois de obter dados históricos suficientes para montar um modelo é iniciar o treinamento para conseguir indicar quando uma parte do equipamento está gasta.

Através de conversa com especialistas, ficou evidente que as partes mais críticas que causam parada da embarcação em alto mar são:

- defeito no motor principal;
- defeito no motor auxiliar;
- defeito na parte de radar e navegação;
- defeito no sistema de geração de energia;
- defeito no sistema de refrigeração.

Além disso, algumas peças têm duplicação para que o barco não pare em alto mar, mas o funcionamento do barco fica comprometido quando uma peça de duplicação não funciona, e para evitar que o barco tenha que ser parado a tripulação do barco não informa sobre as quebras.

Identificar o mal funcionamento dos itens acima, e quebra de partes não informadas pela tripulação será nosso com a base de dados históricos que coletaremos.

Por enquanto, para montar o protótipo utilizaremos a base de dados sintéticos, cujo pré-processamento já foi detalhado em seções anteriores. Nessa base o input serão as curvas lidas com temperatura, pressão e posicionamento,

e como output temos o estado do compressor e da turbina. Portanto, temos duas variáveis de output, e cada uma delas multi-classe com 4 classes.

Para modelar esse problema e montar um modelo que pudesse ser aplicado em novas amostras vamos testar os seguintes algoritmos:

- KNN: agrupamento não supervisionado
- Árvore de Decisão: agrupamento supervisionado
- Redes neurais de convolução: agrupamento supervisionado
- Random Forest: agrupamento supervisionado

Utilizamos como referência para montar os modelos o livro (1) e os sites com documentação do pandas, sklearn e keras.

Os métodos que deram melhores resultados foram as redes neurais e o Random Forest, e vamos apresentar todos os modelos utilizados e o desempenho no conjunto de teste nas seções a seguir:

### 4.3.1 KNN

O algoritmo KNN (K-Nearest Neighbors) ou k-vizinhos mais próximos, é um algoritmo de reconhecimento de padrão. Esse algoritmo faz o agrupamento dos vizinhos mais próximos segundo a norma especificada, o algoritmo é iterativo e divide a amostra em K grupos, sendo K o número de classes pré-especificado e fixo. Portanto, temos como grau de liberdade o número de classes, a escolha dos primeiros elementos de cada grupo, e a norma a ser utilizada e o algoritmo de otimização. Essa norma pode ser escolhida de acordo com a natureza do problema, e a normal quadrática é a mais comum, e gera elipsoides com centroides definidas como o centro de massa de cada agrupamento.

Esses centroides são recalculados durante a execução do algoritmo, toda vez que um ponto novo é adicionado aos grupos, o que torna o algoritmo de iteração um pouco caro e suscetível a erros quando existem ruídos ou outliers no conjunto de treino, o que pode viesar os centroides e mudá-los para fora da região real de interesse.

Para esse método, fiz testes na base de dados trocando o número de vizinhos  $K$  a ser levado em consideração, a métrica, e o peso a ser dado, e a matriz de confusão apresentada em Fig. 4.2 se refere ao melhor resultado obtido, com 92% de acurácia no conjunto de teste.

Para obter esse resultado utilizei a seguinte função de keras:

```
KNeighborsClassifier(algorithm='ball_tree', leaf_size=20, metric='minkowski',
metric_params=None, n_jobs=1, n_neighbors=5, p=2, weights='distance')
```

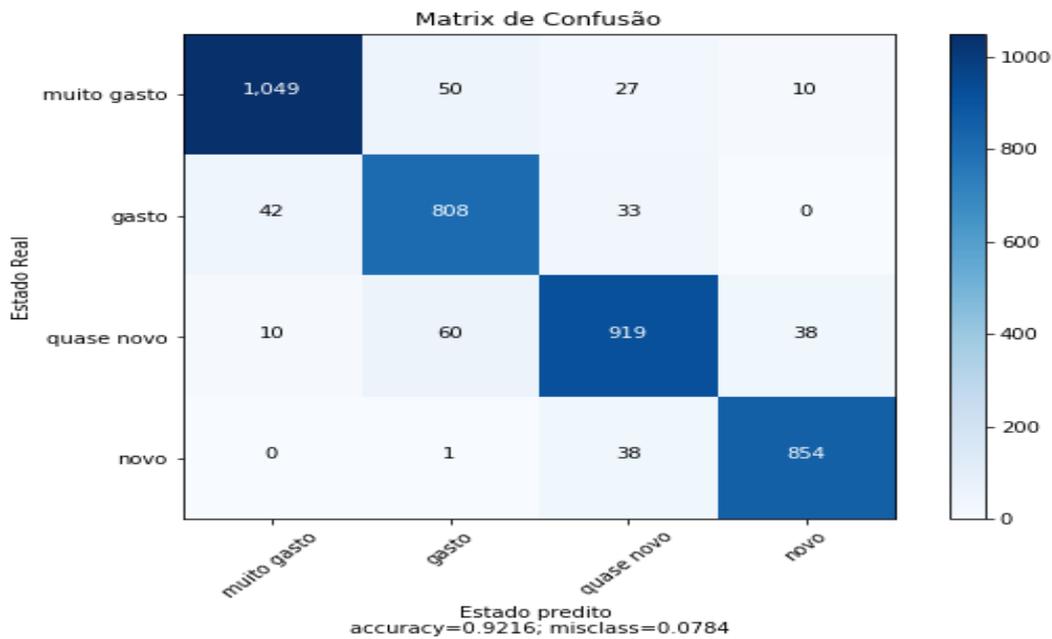


Figure 4.2: Matrix de confusão utilizando o método KNN com peso dado pela distância.

Esse método tem um treino computacionalmente barato e teve alta acurácia no conjunto de teste, mas teve algumas classificações erradas entre classes não vizinhas, o que é altamente não desejado.

### 4.3.2

#### Árvore de Decisão

Árvore de decisão é um algoritmo de classificação supervisionada onde todas as amostras iniciam no mesmo nó e a partir dos valores de input essas amostras são subdivididas em classes, a árvore pode ter várias profundidades e cada nó da árvore da decisão é otimizado de forma que minimizar o erro de má classificação no conjunto de treino, um problema comum com esse algoritmo é o overfitting, o que é obtido quando a árvore tem erro pequeno no conjunto de treino, mas não tem um desempenho bom no conjunto de teste.

Para utilizar o método da árvore de decisão utilizamos a função da biblioteca do scikitlearn em Python, cuja documentação pode ser encontrada em (8). Como parâmetros da árvore de decisão temos a seguinte lista:

- criterion: é possível escolher a função ‘gini’ ou ‘entropy’, e por padrão é gini;

- splitter: divisor das folhas, que por padrão é best ; item max\_depth: profundidade máxima da árvore, padrão é sem limite;
- min\_samples\_split: número mínimo para que ocorra uma divisão da folha, padrão é 2;
- min\_samples\_leaf: número mínimo de amostras em cada folha, padrão é 1;
- min\_weight\_fraction\_leaf: o peso mínimo associado a cada node em relação ao todo, padrão é 0;
- max\_features: número máximo de características a ser usada em cada divisão, padrão são todas os inputs;
- random\_state: estado randômico inicial, para que o algoritmo possa ser reproduzido;
- max\_leaf\_nodes: número máximo de folhas, padrão é sem limites;
- min\_impurity\_decrease: divisão ocorre se se a impureza for reduzida, padrão é 0;
- presort: reordenamento antes do treinamento, o que pode acelerar o treinamento, mas consumir tempo com o ordenamento caso o conjunto de treino seja grande, o padrão é sem reordenamento.

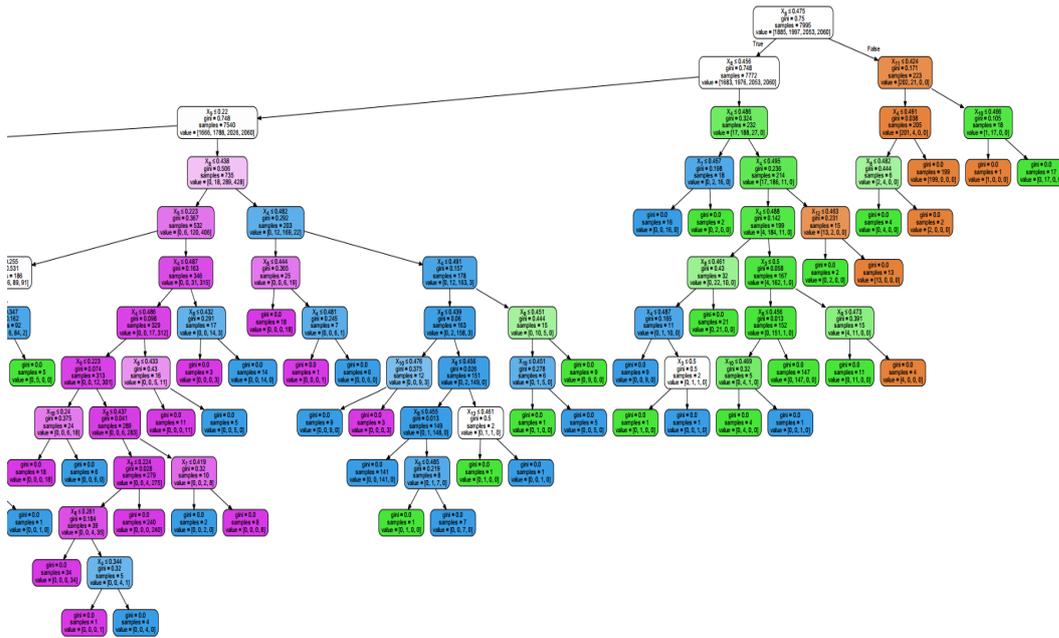
Na primeira modelagem utilizada, separamos o conjunto em treino e teste como descrito acima e temos 4 classes, a primeira tentativa foi deixar todos os parâmetros com o valor padrão

```
class sklearn.tree.DecisionTreeClassifier(criterion='gini', splitter='best', max_depth=None, min_samples_split=2, min_samples_leaf=1, min_weight_fraction_leaf=0.0, max_features=None, random_state=None, max_leaf_nodes=None, min_impurity_decrease=0.0, min_impurity_split=None, class_weight=None, presort=False)
```

Uma segunda tentativa para balancear mais a árvore, foi limitar o número mínimo de amostras para fazer a divisão de nós para 10, fazer um balanceamento dos pesos de cada classe pelo número de representantes no conjunto de treino , fazer um pré reordenamento, e trocar a função de split para entropia, segundo a função abaixo.

```
dtree = tree.DecisionTreeClassifier(criterion='entropy', min_samples_split=10, class_weight='balanced', random_state=42, presort=True)
```

Com esse novo modelo, o número de folhas pequenas sendo redivididas foi reduzido sem que a acurácia fosse afetada, a matriz de confusão desse novo modelo é apresentada na Fig. 4.4.



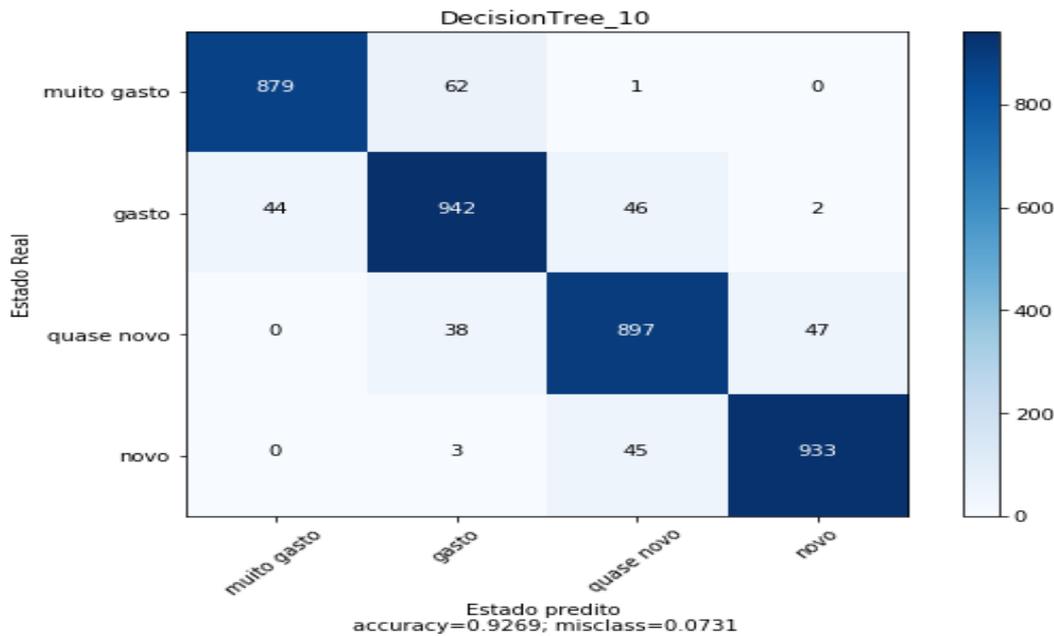


Figure 4.4: Matriz de confusão obtida nos dados de teste usando árvore de decisão com segundo grupo de parâmetros.

#### 4.3.4

##### Redes neurais

O modelo de redes neurais são formados a partir de camadas que tem como objetivo extração de características, mapeamento de características e subamostragem (redução de parâmetros) e está se mostrando muito útil nos problemas de análise de imagens por ter a capacidade de lidar com grandes quantidades de dados mantendo a estrutura dada pela localização dos pixels através da montagem de camadas que combinam extração, mapeamento e sub amostragem.

O elemento básico de redes neurais são os perceptrons, que são elementos básicos de aprendizagem supervisionada, cujo funcionamento é baseado na forma como os neurônios lidam com informação, sendo ativos ou não ativos a partir de vetores afins de entrada no formato  $wx + b$ . Sendo  $w$  os pesos associados a cada input e  $b$  o bias. A equação abaixo apresenta a equação para a função de ativação de Heaveside. Ambos  $w$  e  $b$  dessa equação são parâmetros otimizados durante o treinamento, obtidos através de um problema de minimização da função de loss (*loss function*) definida durante o treino.

$$f(x) = \begin{cases} 1, & \text{se } wx + b > 0, \\ 0, & \text{se } wx + b \leq 0. \end{cases}$$

Além da função de ativação Heaviside, é possível utilizar outras funções

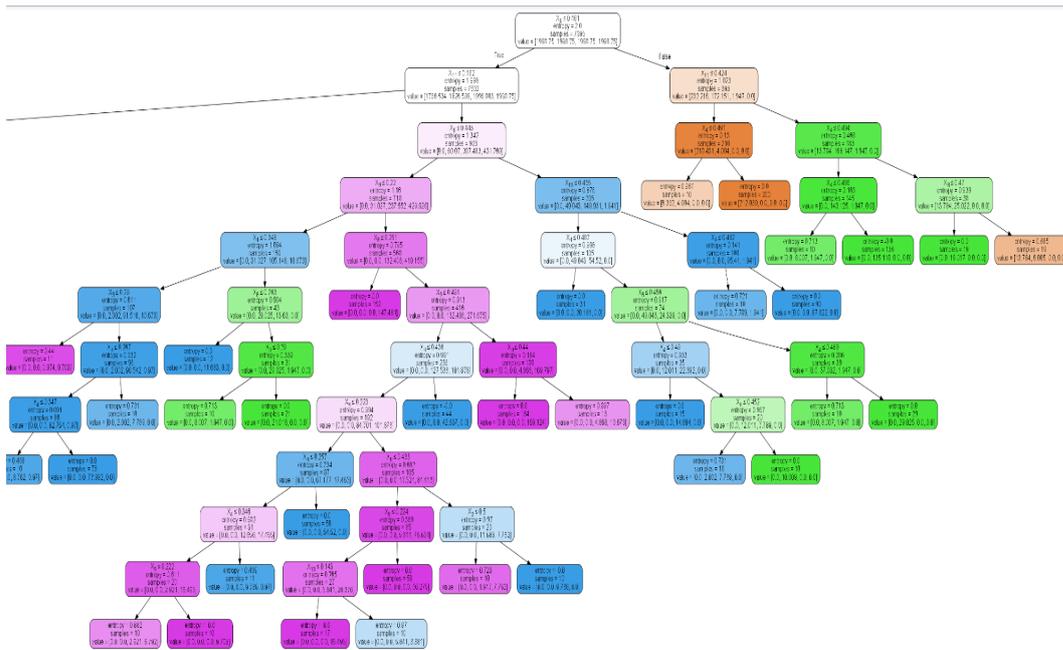


Figure 4.5: Árvore de decisão com novos parâmetros e limite mínimo de 10 elementos por folha..

com perfil mais suaves, o que torna o problema de minimização durante o treinamento mais rápido devido a possibilidade de aplicar gradiente. Essas funções de ativação estão documentadas em (10) e alguns exemplos são:

- Tanh;
- Softmax;
- Elu: exponential linear unit;
- Selu: scaled exponential linear unit;
- Softplus:  $\log(\exp(x) + 1)$ ;
- Softsign:  $\frac{x}{(abs(x)+1)}$ ;
- Relu: com formula apresentada abaixo;
- Sigmoid.

$$\phi(x) = \begin{cases} maxvalue, & \text{se } x \geq maxvalue, \\ x, & \text{se } threshold < x < maxvalue \\ \alpha * (x - threshold), & \text{se } x < threshold. \end{cases}$$

Tanto o peso quanto o bias são os parâmetros da rede a serem treinados de acordo com os dados de input e output do conjunto de treinamento para

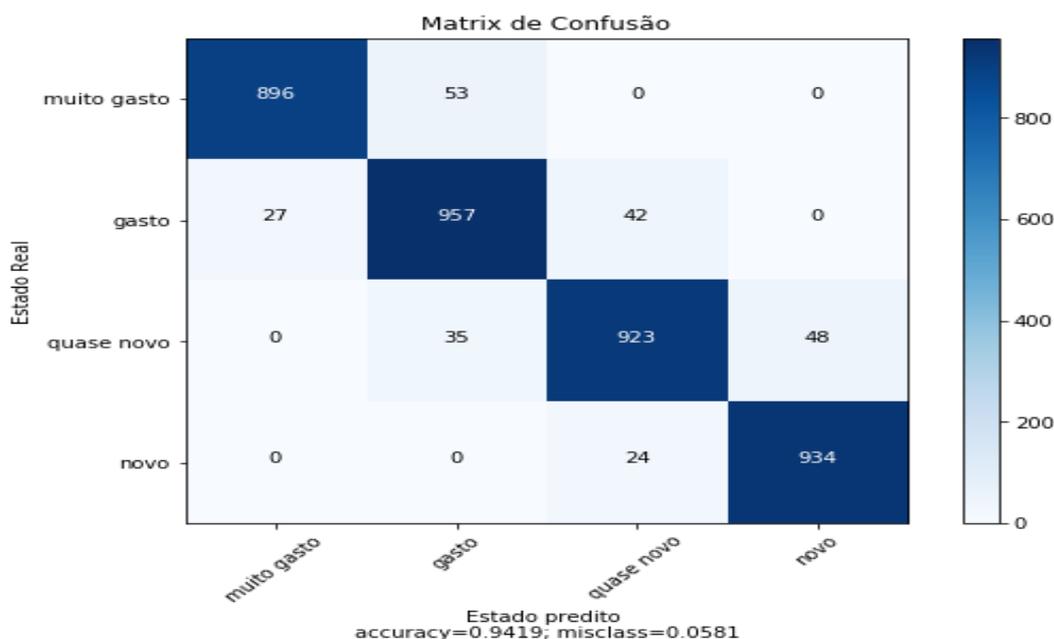


Figure 4.6: Matrix de confusão obtida com o método Random Forest, que apresentou ótimo desempenho.

reduzir a função de perda e de acordo com o otimizador definido durante a definição do modelo.

Os otimizadores disponíveis em keras são:

- SGD: otimizador estocástico baseado em gradiente descendente;
- RMSprop: recommended for recurrent neural networks;
- Adagrad: algoritmo de otimização baseado em gradiente que possui uma taxa de aprendizagem adaptável, sendo que ele faz pequenas alterações para parâmetros de características que ocorrem frequentemente e grandes alterações para características que ocorrem com baixa frequência ;
- Adadelta: uma versão mais robusta do Adagrad;
- Adam: uma versão do RMSprop com momento;
- Adamax: versão com Adam baseado na norma infinito;
- Nadam: versão do RMSProp com momento de Nestorov.

O modelo de uma camada de perceptron é descrito na figura acima, onde  $x_1$  e  $x_2$  são entradas do sistema e  $y_1$  e  $y_2$  saídas do sistema. Cada uma das conexões é associada a um peso  $w_i$  e cada nodo  $a$  um bias  $b$ .

A partir desse sistema de equações é possível treinar o modelo utilizando um problema de minimização definido por uma função de perda, geralmente como função entre o output real  $y_i$  e os outputs preditos  $y_{ti}$  e um otimizador.

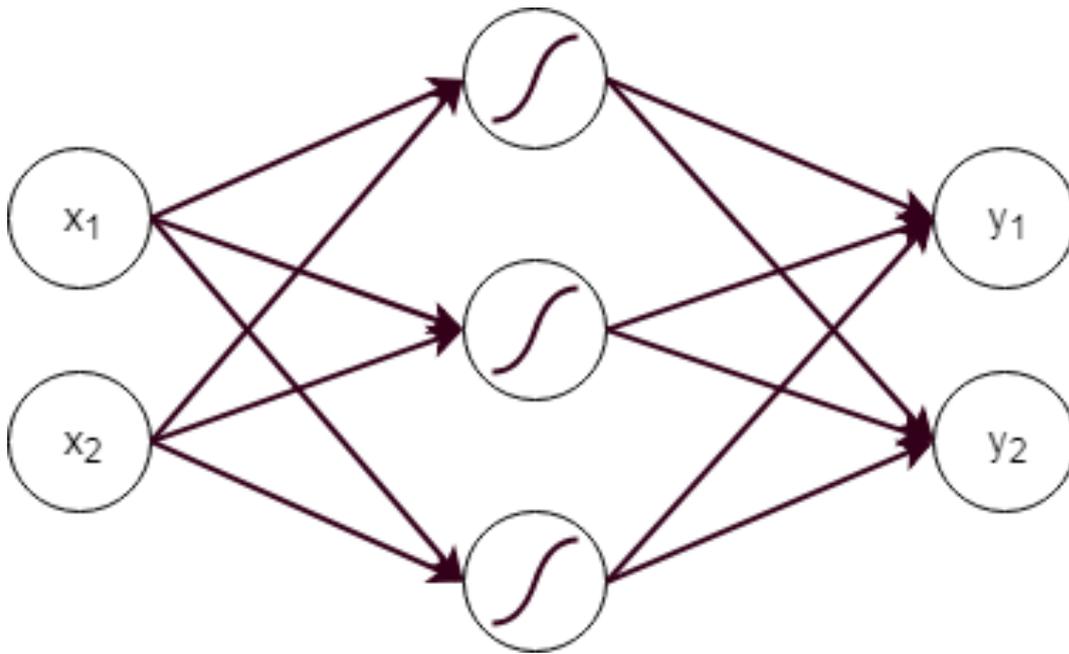


Figure 4.7: Modelo de redes neurais com uma camada de ativação, sendo  $x_1$  e  $x_2$  entradas do sistema, e  $y_1$  e  $y_2$  saídas do sistema..

Desde a formulação da ideia de perceptron por Frank Rosenblatt em 1957, o potencial desse elemento para inteligência artificial foi reconhecido, no entanto, devido a limitações para resolver problemas não separáveis essa modelagem foi abandonada por anos até que o poder de redes neurais se mostrou útil com a formulação de redes com multi-camadas, o que aumenta a possibilidade para resolução de problemas cada dia mais complexos.

Essa formulação foi potencializada pelo crescimento em poder computacional, possibilidade de computação paralela e placas gráficas que ocorreu nos últimos anos. Um dos exemplos clássicos de utilização de redes neurais é a utilização de redes neurais convolucionais combinadas com max-poolings para a montagem de redes de conhecimento profundo capazes de obter acurácia maior que humanos. A evolução dos algoritmos de reconhecimento de imagens está diretamente ligado ao desenvolvimento computacional e ao final de 2015 na competição Imagenet, patrocinada pela Nvidia, a Microsoft teve uma acurácia de 95% em reconhecimento de imagens, surpreendentemente passando a acurácia média humana. Desde então a arquitetura das redes tem evoluído com a finalidade de aumentar acurácia, identificar erros e ter mais estabilidade, e maior eficiência computacional.

Uma das formas de ser menos sensível a dados ruidosos e evitar sobre treinamento (*over fitting*) é usar *drop-out*, que faz com que uma porcentagem fixa das conexões da rede não seja ativada em cada rodada, de forma aleatória.

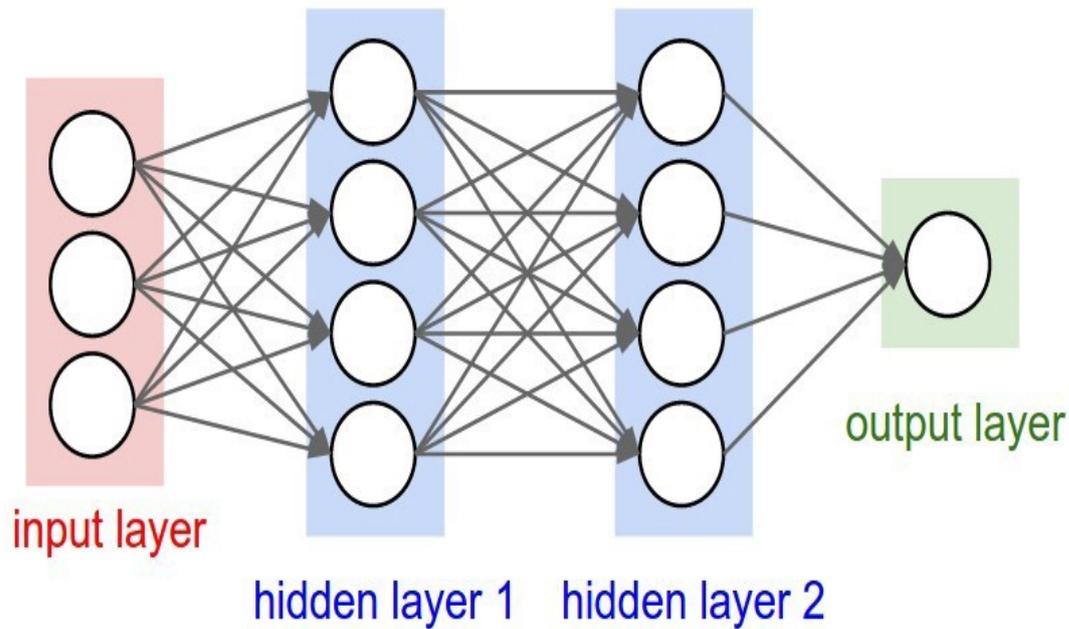


Figure 4.8: Arquitetura de uma rede neural com uma camada de input, duas camadas escondidas e uma camada de saída.

Isso torna o modelo mais estável e menos dependente dos dados de treinamento. Essa funcionalidade está disponível em keras, e é necessário definir a porcentagem de conexões que ficará inativa, por exemplo 0.2 ou 0.5 deixam 20% e 50% das conexões inativas durante o treino. Durante a predição, todas as conexões ficarão ativas. Essa camada da rede não tem parâmetros treináveis, pois sua funcionalidade é obter uma melhoria de performance na aplicação do modelo nos dados de teste, que não são usados durante o treinamento.

Nos dados sintéticos da base em questão, temos dados unidimensionais como input e não existe um número grande de inputs e, portanto, não foi necessário utilizar max-pooling para redução de parâmetros.

Depois de alguns testes variando o otimizar, o número de camadas, e o tamanho das camadas escondidas, o modelo que gerou maior acurácia foi o seguinte:

Esse modelo é baseado em camadas full dense e 4.5 possui 12852 parâmetros treináveis, foi treinado numa base com 7995 e gerou uma acurácia de 94% nos dados de treino com 3939 amostras.

Para chegar a esse modelo com alta acurácia realizei testes com os seguintes parâmetros:

- Arquitetura da rede: aumentei o número de camadas até que a acurácia estivesse alta, mantendo o número de parâmetros na mesma ordem de grandeza do número de amostras no banco de dados;

Layer (type)	Output Shape	Param #
dense_23 (Dense)	(None, 128)	1920
dropout_18 (Dropout)	(None, 128)	0
dense_24 (Dense)	(None, 64)	8256
dropout_19 (Dropout)	(None, 64)	0
dense_25 (Dense)	(None, 32)	2080
dropout_20 (Dropout)	(None, 32)	0
dense_26 (Dense)	(None, 16)	528
dropout_21 (Dropout)	(None, 16)	0
dense_27 (Dense)	(None, 4)	68
Total params: 12,852		
Trainable params: 12,852		
Non-trainable params: 0		

Figure 4.9: Arquitetura de camadas de uma rede neural full connected.

- Função de ativação da última camada: como tenho uma classificação, utilizei as funções tanh ou sigmoid, e a função sigmoid deu melhores resultados
- Batch size: 100, 150, 200 e 400, o resultado com melhor acurácia foi obtido com 200, além do mais, com batch size igual a 100, o modelo ficou mais instável.
- Função de perda, troquei a função de perda para: mean\_squared\_error, categorical\_crossentropy, binary\_crossentropy. E o melhor resultado foi obtido com binary\_crossentropy.

Além disso, realizei testes com diferentes otimizadores:

- SGD: parece ter estagnado num mínimo local;
- RMSprop: parece ter estagnado num mínimo local;
- Adagrad: algoritmo de otimização baseado em gradiente que possui uma taxa de aprendizagem adaptável, sendo que ele faz pequenas alterações para parâmetros de características que ocorrem frequentemente e grandes alterações para características que ocorrem com baixa frequência;
- Adadelta: parece ter estagnado num mínimo local;
- Adam: demorou para convergir, e mesmo com 1000 épocas ainda não tinha estabilizado, apesar de já ter chegado a uma boa acurácia

- Adamax: teve uma acurácia de 83% nos dados de teste, mas apresentou instabilidade durante o treino, com bastante variação de desempenho nos dados de validação cruzada.
- Nadam: esse otimizador teve um desempenho melhor, pois convergiu mais rapidamente e mostrou maior estabilidade no conjunto de cross validação, além de apresentar maior acurácia no conjunto de teste.

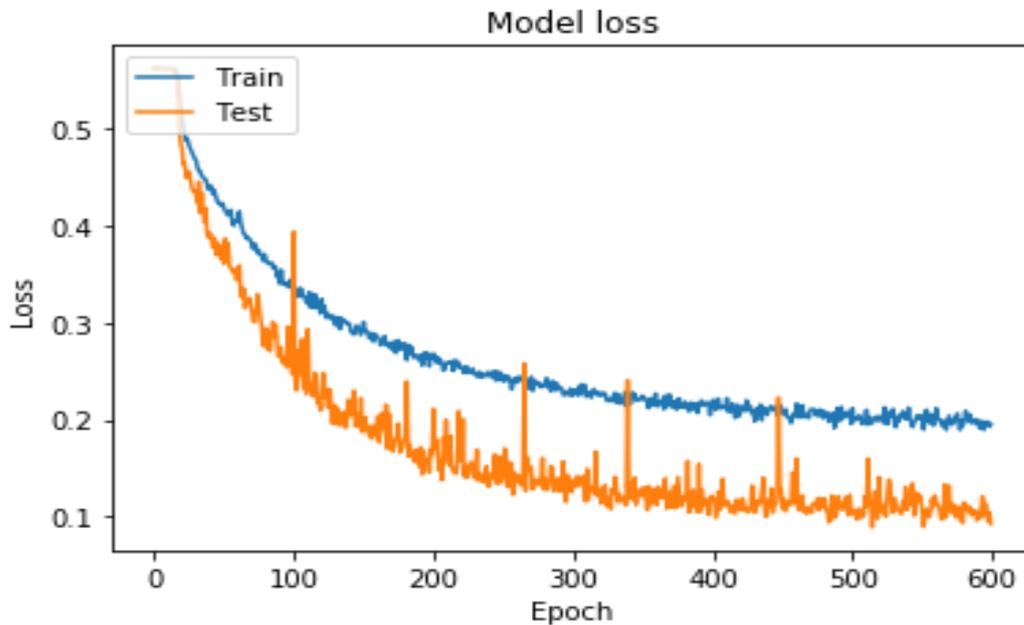


Figure 4.10: Evolução do desempenho da rede neural com otimizador Nadam, função de perda 'binary\_crossentropy', batch size=150, e arquitetura de rede mostrada na Fig. 4.9.

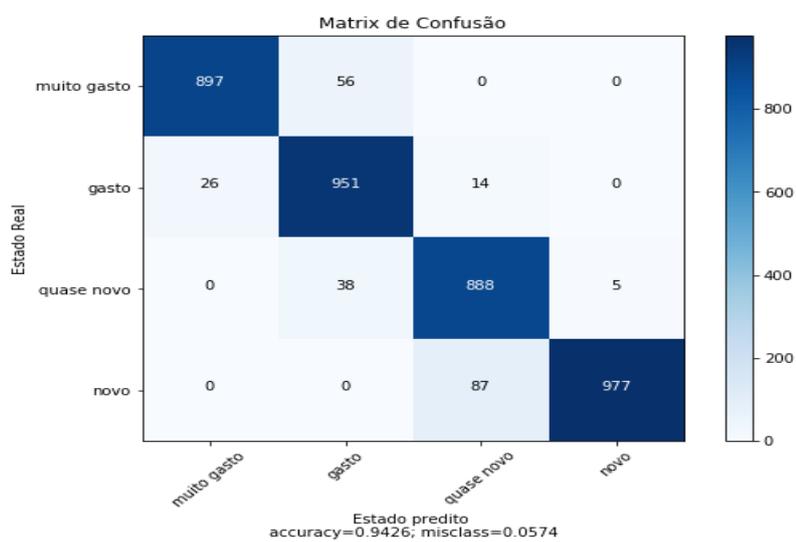


Figure 4.11: Matriz de confusão para o modelo de redes neurais com otimizador Nadam e função de perda `binary_crossentropy`. O modelo apresentou 94% de acurácia nos dados de treino e somente confundiu classes vizinhas.

## 5

### **Apresentação dos Dados para o Cliente**

Como já citado anteriormente, na primeira fase do projeto esperamos apresentar a leitura dos barcos para os armadores, que são os proprietários dos barcos e responsáveis pela manutenção e bom funcionamento dos mesmos. Durante esse tempo apresentaremos apenas uma estatística das leituras e como os barcos se comparam com o dado histórico da frota. Nesse tempo, o objetivo é coletar dados históricos para melhorar na análise estatística e posteriormente começar a identificar parâmetros.

#### **5.1**

##### **Apresentação dos dados para as produtoras de Petróleo**

Durante as entrevistas com o chefe de manutenção contratado pela Petrobrás e com o setor de logística, ficou claro que existe uma necessidade de ter maior controle sobre o estado das embarcações, e que a Petrobrás já faz esforço nessa área, fazendo revistas para conferir o funcionamento dos barcos e as documentações quando os barcos param para recarga e especialmente no período entre programações. E nosso esforço para apresentação das informações para produtores poderá ser nesse sentido de auxiliar a aumentar o controle sobre os barcos e melhorar a forma como isso é feito hoje, por pessoas que ganham adicional para trabalhar no campo.

Notou-se em todas as entrevistas que ambos os lados estão insatisfeitos com os resultados da situação atual e estão abertos a mudanças para aumentar o controle sobre o estado das embarcações e reduzir a chance de dados imprecisos passados pela tripulação. A quebra inesperada de barcos causa prejuízo a empresa de barcos que deixa de receber a diária, pode ser obrigada a pagar multa se não respeitar as cláusulas contratuais e possivelmente ficar fora de novas contratações por não cumprir pré-requisitos, e um prejuízo muito maior para a produtora de petróleo que pode ter o funcionamento de uma plataforma afetada pela quebra inesperada gerando atraso de entregas de combustível ou material essencial, tais como brocas, fluidos de perfuração e alimentos. Além de ter que manter um excesso de frota de barcos de 8% para evitar maiores prejuízos.

O regime de trabalho dos barcos funciona da seguinte forma, em cada

viagem o barco faz transporte para um cluster de 4 ou 5 plataformas e demora dias nesse trajeto, nesse tempo a tripulação se reveza e o responsável por tomadas de decisão importantes, tais como realizar carga e descarga ou não devido a condições de mar é o comandante.

Após essas entregas os barcos voltam ao porto para descarga do que voltou da plataforma e carga da nova entrega, e esse ciclo é repetido algumas vezes durante uma programação, que dura por volta de um mês. Entre programações o barco fica parado algumas horas na base, e são essas nossas horas de interesse para realização de manutenção preditiva.

O que desejamos é que exista uma transparência entre as operadoras de barco e a Petrobrás para que os armadores consigam utilizar de forma otimizada essas horas para realizar as manutenções necessárias e reduzir as quebras inesperadas e subutilização dos barcos.

Para apresentar a ideia do dashboard, escolhi 130 amostras aleatórias dentre os dados sintéticos, dividi esses barcos entre 5 empresas e dei nome a essas embarcações e dei uma chance de 25 % para que ocorra intervalo entre programação na próxima semana, baseada no fato que as programações duram um mês. Com esses dados montei três tabelas de dados.

Uma tabela com a descrição dos barcos:

- código do barco;
- nome da embarcação;
- nome da empresa;
- parada ou não na próxima semana;
- dia da próxima parada.

Uma tabela com os dados dos sensores, que são os inputs do meu problema de modelagem e são dados pelas colunas de dados das 130 amostras selecionadas. Uma tabela com os dados de desgaste do compressor e da turbina divididos em 4 categorias para facilitar a identificação dos equipamentos que merecem mais atenção.

Esses dados foram relacionados através de um índice dado a cada barco, como no esquema em Fig. 5.1.

Baseado nos resultados que obtivemos foi estimado um output de desgaste de dois equipamentos que estão indicados com números de 1 a 4, sendo:

1. novo;
2. quase novo;
3. gasto;

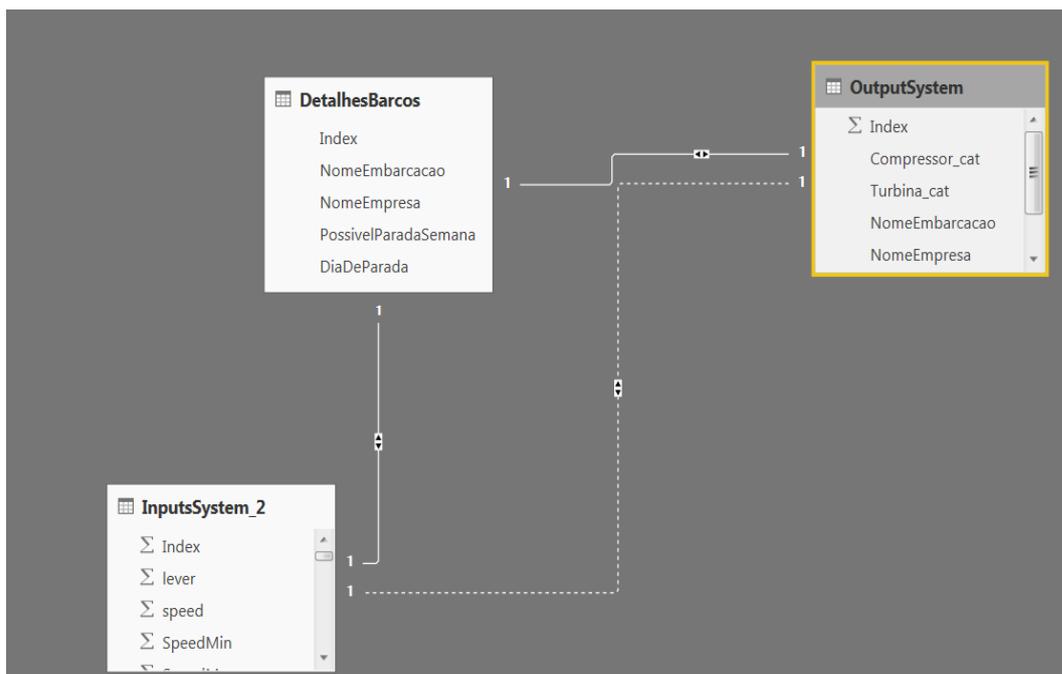


Figure 5.1: : Relacionamento da base de dados pelo, com chave indicada pelo Index.

4. muito gasto.

No dashboard abaixo apresentamos um resumo importante de informações indicando quais barcos das empresas Boat Power e Embarcações JVR com a turbina indicando muito desgaste, farão paradas na próxima semanas.

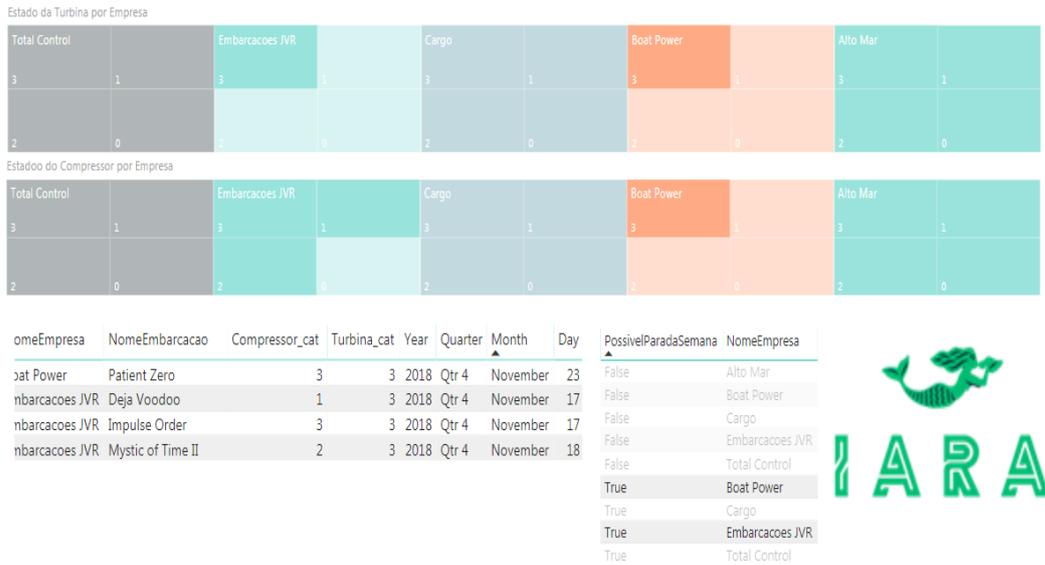


Figure 5.2: Dashboard para apresentação dos barcos para as empresas petrolíferas, mostrando os barcos que farão paradas na próxima semana separado por empresa e por estado de desgaste dos equipamentos..

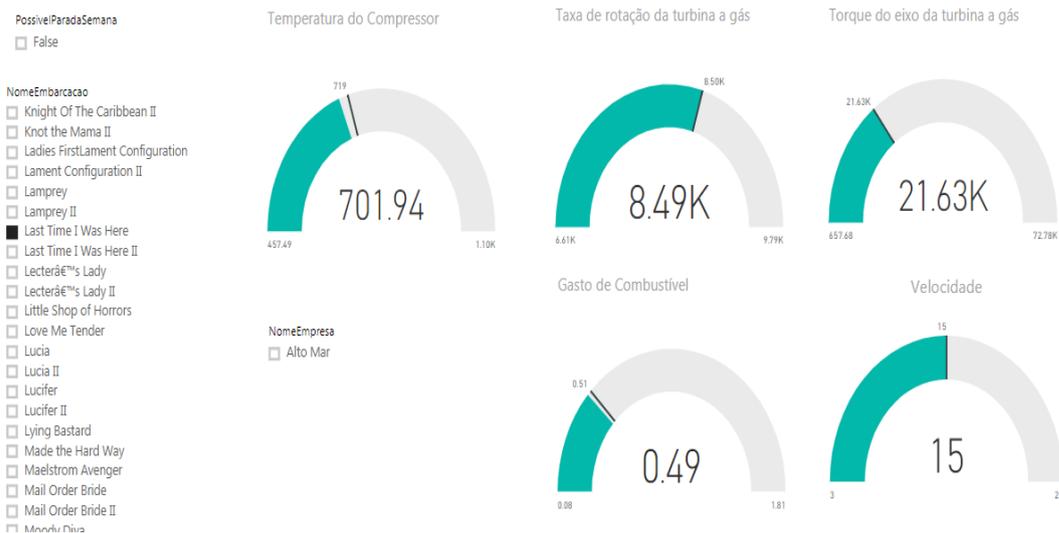


Figure 5.3: Informações dos sensores de leitura de uma embarcação com comportamento semelhante a média de todas as embarcações..

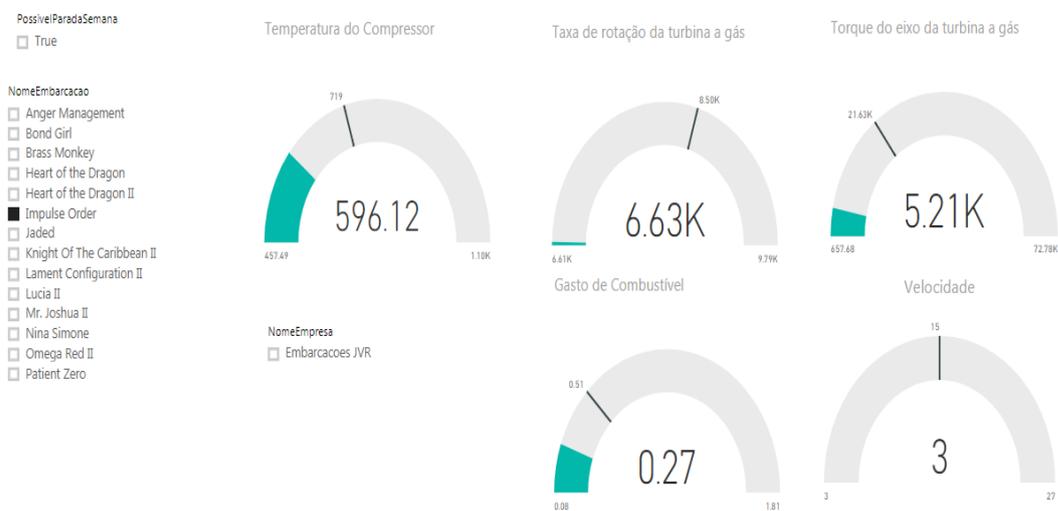


Figure 5.4: Informação de estado de uma embarcação com baixa velocidade em relação a frota.

## 6

### Conclusions

Esse trabalho discute o projeto Iara – Plataforma digital para Manutenção Inteligente de Barcos de Carga que foi apresentado no Hackaton - Desafios de Libra Digital, e contém a descrição do modelo de negócios, análise de mercado, análise de riscos, e descrição do produto.

Para apresentar o protótipo utilizamos uma base de dados sintética disponível em (14) que utilizamos para propor classificação de navios de acordo com o nível de desgastes de parte do equipamento e modelos de clusterização utilizando análise de dados com modelos de redes neurais, árvore de decisão, random forest e K- vizinhos mais próximos.

Todos os métodos apresentaram bons resultados, com acurácia maior que 80% no conjunto de teste, sendo que os melhores modelos foram Random Forest e Redes Neurais, com 94% de acurácia no conjunto de teste.

Utilizamos esses dados foram combinados a um conjunto de descrição de barcos contendo, nome da embarcação, empresa, e dia da próxima parada e combinados a uma parte da base de dados original e a informação dos barcos foi organizada em dois dashboards construído em Power BI, esses dashboards podem ser enviados aos clientes para monitoramento da saúde do barco, identificação de barcos precisando de manutenção e para organização de calendários para revisão de acordo com as paradas programadas para as embarcações.

Já identificamos uma necessidade dos clientes, tanto dos armadores quanto dos operadores em ter maior conhecimento sobre o estado das embarcações para evitar quebras inesperadas, inoperância e para otimização da manutenção. Identificamos as partes do equipamento que mais causam essas paradas inesperadas, montamos um modelo de apresentação para o cliente para apresentar um protótipo, agora o nosso próximo passo é apresentar o modelo para armadores e começar uma base de dados para apresentação e análise.

## Bibliography

- [1] GÉRON, AURÉLIEN. **Hands-On Machine Learning with Scikit-Learn and Tensor Flow: concepts, tools, and techniques to build intelligent systems**. O'Reilly, 1st edition, 2017.
- [2] KIM, W. CHAN; MAUBORGNE, RENÉE . **Blue Ocean Strategy, Expanded Edition: How to Create Uncontested Market Space and Make the Competition Irrelevant**. Harvard Business Review Press, 2014).
- [3] CORADDU, A. ; ONETO, L.; GHIO, A; SAVIO, S.; ANGUITA, D. ; FIGARI, M. . **Machine learning approaches for improving condition based maintenance of naval propulsion plants**. Journal of Engineering for the Maritime Environment, 2014.
- [4] ALTOSOLE, M.; BENVENUTO, G.; FIGARI, M.; CAMPORA, U.. **Real-time simulation of a cogag naval ship propulsion system, proceedings of the institution of mechanical engineers**. Journal of Engineering for the Maritime Environment, 223:47–62, 2009.
- [5] **Random forest**. Scikitlearn. Random Forest, 2018. Acesso em: Dezembro de 2018.
- [6] **K nearest neighbors**. Scikitlearn. KNN, 2018. Acesso em: Dezembro de 2018.
- [7] **Hackaton**. Hackaton Rio Oil and Gas. Hackaton, 2018. Acesso em: Dezembro de 2018.
- [8] **Decision tree**. Scikitlearn. Decision Tree, 2018. Acesso em: Dezembro de 2018.
- [9] **Matrix de confusão**. Scikit-learn, 2018. Acesso em: Dezembro de 2018.
- [10] KERAS. **Keras - funções de ativação**. Keras, 2018. Acesso em: Dezembro de 2018.
- [11] **Pandas**. Pandas, 2018. Acesso em: Dezembro de 2018.

- [12] **Keras**. Keras, 2018. Acesso em: Dezembro de 2018.
- [13] **Campo de libra**. CBM, 2018. Acesso em: Dezembro de 2018.
- [14] **Condition based maintenance**. CBM, 2018. Acesso em: Dezembro de 2018.
- [15] HO, T. K. . **Random decision forests**. In: PROCEEDINGS OF THE THIRD INTERNATIONAL CONFERENCE ON DOCUMENT ANALYSIS AND RECOGNITION, volumen 1, 1995.

## A

### Código em Python dos algoritmos apresentados no texto

Todo o processamento e pré processamento apresentado foi escrito em Python utilizando pandas, numpy, scikitlearn e keras. Os códigos principais estão apresentados abaixo:

#### A.1

##### Pré Processamento

Utilizamos a estrutura de dados de código aberto do Pandas em Python, mais detalhes desse projeto pode ser encontrado em (11).

```
#carregamento de dados
l=['lever', 'speed', 'GTT', 'GTn', 'GGn', 'kN', 'turbtemp', 'comprTemp',
'turbExitTemp', 'GTinletTemp', 'GtCompOutTemp', 'HPTurbPre', 'Compres-
sAirPre', 'Pexh', 'TIC', 'FuelFlow', 'GTCompressorState', 'GTTurbineState']
A= pd.read_csv(fileName, sep=' ', delimiter=None, header='infer',
names=l)
CorrMatrix=A.corr()
#divisao em input e output
X=A[l[0:14]]
Y=A[l[14:16]]
# discretização de output em 4 classes para compressor e turbina
a=pd.cut(Y[l[14]], [0.949, 0.962, 0.975, 0.988, 1.01], labels=False, ret-
bins=True, right=False, duplicates='drop')
b=pd.cut(Y[l[15]], [0.970, 0.9812, 0.988,0.995, 1.01], labels=False, ret-
bins=True, right=False, duplicates='drop')
l_cat=['GTB_cat', 'GTC_cat']
d = l_cat[0]: a[0], l_cat[1]: b[0]
Y_cat = pd.DataFrame(data=d)
Y_Cat=to_categorical(Y_cat)
#normalizacao do input
X_normalized = Normalize(X)
# divisao em treino e teste
X_train, X_test, y_train, y_test = train_test_split(X_normalized,
Y_normalized, test_size=0.33, random_state=42)
```

```
X_train, X_test, y_trainCat, y_testCat = train_test_split(X_normalized,
Y_cat, test_size=0.33, random_state=42)
```

## A.2 Treinamento

### A.2.1 Modelo de Redes Neurais

Para a estrutura de redes neurais utilizamos o API de alto nível Keras rodando em cima do TensorFlow, e para verificar a documentação e obter detalhes do otimizador, função de perda e outros parâmetros utilizei o site do projeto (12).

```
from keras.models import Sequential
from keras.layers import Dense, Dropout
#redes neurais com multi camadas e 12,852 parâmetros.
model = Sequential()
model.add(Dense(128, input_dim=14, activation='relu'))
model.add(Dropout(0.3))
model.add(Dense(64, activation='softmax'))
model.add(Dropout(0.3))
model.add(Dense(32, activation='relu'))
model.add(Dropout(0.3))
model.add(Dense(16, activation='relu'))
model.add(Dropout(0.3))
model.add(Dense(8, activation='relu'))
model.add(Dropout(0.3))
model.add(Dense(4, activation='softmax'))
OPTIMIZER=keras.optimizers.Nadam(lr=0.002, beta_1=0.9,
beta_2=0.999, epsilon=None, schedule_decay=0.004)
model.compile(loss='binary_crossentropy', optimizer=OPTIMIZER,
metrics=['accuracy'])
# Fit the model
history=model.fit(X_train, Y_trainCat2[:,0:], validation_split=0.33,epochs=600,
batch_size=150)
# evaluate the model
scores = model.evaluate(X_test, Y_testCat2[:,0:])
y_pred_NN = model.predict(X_test)
return model
```

O resumo do modelo de redes neurais criado acima pode ser encontrado na Fig. 4.9

### A.2.2

#### Modelo de Random Forest

Utilizamos o Modelo de classificação supervisionada Random Forest do sklearn, documentado em (5).

```
from sklearn.ensemble import RandomForestClassifier
rnd_clf=RandomForestClassifier()
rnd_clf.fit(X_train, yTrain0)
y_pred0_rnm=rnd_clf.predict(X_test)
```

### A.2.3

#### Modelo KNN

Utilizamos o Modelo de classificação KNN do sklearn, documentado em (6).

```
from sklearn.neighbors import KNeighborsClassifier
knn = KNeighborsClassifier(algorithm='ball_tree',
leaf_size=20,
metric='minkowski',
metric_params=None,
n_jobs=1,
n_neighbors=5,
p=2,
weights='distance')
knn.fit(X_train, Y_trainCat2[:,1,:])
y_knn=knn.predict(X_test)
B=confusion_matrix(np.argmax(y_knn,axis=1),np.argmax(Y_testCat2[:,1,:],
axis=1) )
accuracy = np.trace(B) / float(np.sum(B))
print(accuracy)
plot_confusion_matrix(B, normalize = False, target_names = ['muito
gasto', 'gasto', 'quase novo', 'novo'], title = "Matrix de Confusão")
```

### A.2.4

#### Avaliação dos modelos

Para avaliação do modelo utilizamos dois métodos, o primeiro é a matriz de confusão montada a partir do sklearn.metrics e plotada a partir da função plot\_confusion\_matrix, encontrada em (9).

Nesse plot, as linhas indicam as classes reais de cada amostra, e as colunas indicam a classe assinalada pelo método.

```
B=confusion_matrix(y_pred0_rnm,yTest0 )  
accuracy = np.trace(B) / float(np.sum(B))  
print(accuracy)
```

```
plot_confusion_matrix(B, normalize = False, target_names = ['muito  
gasto', 'gasto', 'quase novo', 'novo'], title = "Matrix de Confusão")
```

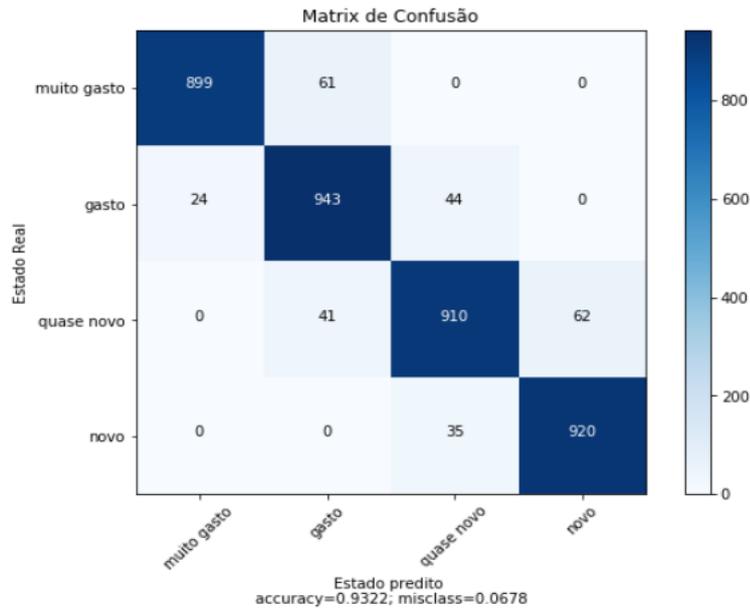


Figure A.1: Matrix de confusão plotada segundo a função `plot_confusion_matrix`, apresentada em em scikitlearn.