UNIVERSIDADE FEDERAL DE SANTA CATARINA
DEPARTAMENTO DE INFORMÁTICA E ESTATÍSTICA

Desenvolvimento de modelo de machine learning para previsão das condições de ondas para a prática do surf

Jonas Lai Barbosa

Rolf Zambon

Florianópolis – SC

2023

UNIVERSIDADE FEDERAL DE SANTA CATARINA
DEPARTAMENTO DE INFORMÁTICA E ESTATÍSTICA

Desenvolvimento de modelo de machine learning para previsão das condições de ondas para a prática do surf

Jonas Lai Barbosa

Rolf Zambon

Trabalho de conclusão de curso apresentado como parte dos requisitos para obtenção do grau de Bacharel em Sistemas de Informação.

Orientador:
Prof. Dr. Elder Rizzon Santos

Florianópolis – SC
2023

Jonas Lai Barbosa Rolf Zambon

Desenvolvimento de modelo de machine learning para previsão das condições de ondas para a prática do surf

Trabalho de conclusão de curso apresentado como parte dos requisitos para a obtenção do grau de Bacharel em curso de Sistemas de Informação

Florianópolis, 12 de julho de 2023.

Banca Examinadora:

_____

Prof. Dr. Elder Rizzon Santos

Orientador

_____

Prof. Dr. Alexandre Gonçalves Silva

_____

Me. Rodrigo Rodrigues Pires de Mello

Resumo

O surf é um esporte praticado mundialmente e cada vez atrai mais a atenção e curiosidade de novas pessoas, porém, para quem está começando no esporte, pode se tornar um pouco difícil a experiência de identificar quais as melhores condições das ondas para praticá-lo. Em função disso, este trabalho propõe o desenvolvimento e o treinamento de um modelo de *machine learning* que, a partir de dados coletados em sites de previsões meteorológicas utilizando o processo de *web scraping*, conseguirá prever essas condições através da utilização de classificadores como por exemplo SVM, KNN e Árvores de Decisão. Por fim, através de experimentos realizados utilizando dados não disponíveis no conjunto de treinamento para analisar o funcionamento do modelo desenvolvido e dos resultados encontrados, o SVM apresentou as melhores métricas de qualidade, com uma acurácia de 83% e uma precisão de 93%.

Palavras-chave: Aprendizado de Máquina, Mineração de Dados, *Web Scraping*, Previsão Metereológica

Abstract

Surfing is a sport worldwide respected and increasingly attracts more attention and curiosity from new people, however, for those who are starting in the sport, it can become difficult to identify the best waves conditions to practice it. As a result, this paper proposes the development and training of a machine learning model that, from data that will be collected on weather forecasting sites using web scraping process, will be able to predict these conditions through the use of classifiers such as SVM, KNN and decision trees. Finally, through experiments carried out using data not available in the training set to analyze the functioning of the developed model and the results found, the SVM presented the best quality indicators, with an accuracy of 83% and a precision of 93%.

Palavras-chave: Machine Learning, Data Mining, Web Scraping, Weather Forecast

Lista de Figuras

Lista de Tabelas

Sumário

1. Introdução

O surf é um esporte praticado há décadas por inúmeras pessoas ao redor do mundo, e possui uma grande importância social, econômica e ambiental. Como Reineman, Koenig, Strong-Cvetiche e Kittinger (2021) apresentam, cada vez mais há um reconhecimento por parte de comunidades, pesquisadores e praticantes no valor de surf *breaks* (praias onde há ondas para a prática do surf) locais, e particularmente em economias em desenvolvimento. Estes grandes benefícios socioeconômicos eram previamente subestimados e não eram levados em consideração nos processos de planejamento em desenvolvimentos costeiros. Reineman et al. (2021) também mencionam que, globalmente, o turismo relacionado ao surf é avaliado entre 31,5 e 64,9 bilhões de dólares, e que esses benefícios fornecem o mecanismo para acelerar o crescimento econômico nas comunidades ao redor de surf breaks.

Para um surfista iniciante/intermediário, pode ser difícil identificar as condições necessárias para a prática do esporte por causa da grande quantidade de variáveis existentes na formação de ondas, sendo as principais delas (KAHALU'U BAY SURF & SEA, 2013):

- Período da ondulação: é o intervalo de tempo entre duas ondas, medido em segundos. Quanto maior o período, mais alta e mais forte a onda surfável nas praias, visto que cada onda carregará mais água;

- Altura da ondulação: é a altura da ondulação no oceano, normalmente medida em metros ou pés. Não necessariamente é o mesmo tamanho da onda surfavel nas praias, pois a altura da onda na praia depende também do período;

- Direção da ondulação: é a direção que a ondulação vem do oceano. Dependendo da praia, uma ondulação de uma certa direção pode não alcançar a praia, e não haverá ondas surfáveis nela, ou se a ondulação entrar em cheio na praia (perpendicular à faixa de areia), as ondas estarão com força total. É uma variável que depende totalmente da praia em análise;

- Velocidade e direção do vento: O ideal para a prática do surf é menos vento possível. Mas também é possível com vento fraco a moderado, desde que a direção não seja do oceano para a praia (vento maral).

A fim de analisar e entender melhor essa grande variedade e quantidade de dados, a ciência de dados é uma abordagem poderosa que auxilia nesse processo. Segundo Brodie (2019), a ciência de dados tem como objetivo realizar a análise de grandes quantidades de dados para extrair correlações com estimativas de probabilidade e erro. A ciência de dados também abrange várias outras disciplinas, como *data mining* e *machine learning*.

Para Ramageri (2010), *data mining* é o processo lógico usado para explorar grandes quantidades de dados a fim de encontrar dados úteis. De acordo com Haddaway (2015), data scraping é o termo usado para descrever a extração de dados de um arquivo eletrônico utilizando um programa de computador. *Web scraping* descreve o uso de um programa para extrair dados de arquivos HTML na internet. Normalmente, esses dados são padronizados, principalmente em listas ou tabelas.

Inteligência artificial e *machine learning* estão fortemente relacionadas, porém, essas tecnologias são diferentes em várias maneiras. Em 2007, McCarthy (2007) se referiu à inteligência artificial como a ciência e a engenharia de fazer máquinas inteligentes, especialmente programas de computador inteligentes. Está relacionada à tarefa semelhante de usar computadores para entender a inteligência humana, mas a IA não precisa se limitar a métodos que são biologicamente observáveis. Kersting (2018) acredita que o comportamento de uma máquina não é somente a saída de um programa, ele também é afetado pelo seu "corpo" e o ambiente que fisicamente ele está presente. Para mantê-lo simples, no entanto, se você puder escrever um programa muito inteligente que tenha comportamento semelhante ao humano, ele pode ser IA. Mas, a menos que aprenda automaticamente com os dados, não é *machine learning*:

> *Machine learning é a ciência que "tem como objetivo a questão de como construir programas de computador que melhoram automaticamente com a experiência," (MITCHELL, 1997)*

Finalmente, o presente trabalho propõe a criação de um modelo de *machine learning* que tem como saída a condição das ondas para a prática do surf. A criação do modelo será

possibilitada pela utilização das técnicas de *web scraping*, onde será realizada a coleta das variáveis previamente mencionadas, além da variável alvo, que é a condição das ondas, para realizar o treinamento através da técnica de *machine learning*. Ao fim, será feito um experimento onde o modelo em questão será posto em teste, e os resultados obtidos serão analisados.

1.1 Método de Pesquisa

Inicialmente, serão estudadas as tecnologias de *web scraping* e *machine learning*. Com o conhecimento adquirido, será realizado um estudo dos dados mais importantes para o modelo proposto, iniciando, assim, o scraping, que é a coleta e preparação dos dados, a partir das fontes das quais os dados serão extraídos, construindo e preparando o dataset que será utilizado no treinamento do modelo.

Após a preparação do dataset, será iniciado o treinamento do modelo de machine learning, com a implementação em código e as iterações de treinamento. Para finalizar, será feita uma análise do modelo proposto, realizando experimentos com dados fora do escopo do treinamento e dos resultados encontrados.

2. Objetivos

2.1 Objetivo Geral

O objetivo principal deste trabalho é criar e treinar um modelo de *machine learning* que consiga prever a condição das ondas para a prática do surf a partir de dados que serão coletados através de *web scrapers*, implementados também como parte deste trabalho.

O modelo utilizará os seguintes dados de entrada: direção do vento, velocidade do vento, direção da ondulação, altura da ondulação, período da ondulação e localização da praia. O valor de saída será a condição da praia para a prática do surf, podendo ser ruim, regular ou boa.

2.2 Objetivos Específicos

A seguir, os objetivos específicos deste trabalho são definidos:

- Analisar o estado da arte das áreas de *machine learning*, análise de dados e *web scrapers*.
- Propor um modelo de *machine learning* para a previsão das condições de ondas de praias de Florianópolis para a prática do surf.
- Construção do dataset para treinamento mediante dados coletados através de *web scraping*.
- Realizar o treinamento do modelo proposto.
- Analisar o funcionamento do modelo proposto realizando experimentos utilizando dados não disponíveis no conjunto de treinamento.
- Analisar os resultados oriundos do modelo proposto.

3. Fundamentação Teórica

Este capítulo tem como objetivo apresentar conceitos que são fundamentais para o desenvolvimento deste trabalho. Na seção 3.1 é apresentado o conceito de aprendizado de máquina, ou *machine learning*, e suas classificações. Na seção 3.2 é realizada uma explicação sobre mineração de dados, ou *data mining*, e, por fim, a seção 3.3 apresenta o conceito de *web scraping*.

3.1 Aprendizado de Máquina

Mitchell (1997) define aprendizagem de máquina como a ciência que tem como objetivo a questão de como construir programas de computador que melhoram automaticamente com a experiência. Já Grus (2015) se refere à aprendizagem de máquina como a criação e uso de modelos que aprendem através de dados.

Usualmente o aprendizagem de máquina é classificado em três categorias: aprendizado supervisionado, aprendizado não supervisionado e aprendizado por reforço.

3.1.1 Aprendizado Supervisionado

Aprendizagem supervisionada é a busca por algoritmos que raciocinam a partir de instâncias fornecidas externamente para produzir hipóteses gerais, que então fazem previsões sobre instâncias futuras. Em outras palavras, o objetivo da aprendizagem supervisionada é construir um modelo conciso da distribuição de categorias de acordo com características ou atributos. O classificador resultante é então usado para atribuir categorias às instâncias de teste, diferentes dos exemplos vistos previamente, onde as características e atributos são conhecidos, mas a categoria é desconhecida (KOTSIANTIS, 2007).

Na maioria dos casos, a quantidade de exemplos utilizados no aprendizado não é suficiente para se obter uma função que modele qualquer das possíveis entradas do domínio sendo tratado. Na realidade, os sistemas de aprendizado são capazes de induzir uma função que se aproxima da função conceito, chamada de hipótese (GENTLEMAN; CAREY, 2008).

De acordo com Kotsiantis (2007), o aprendizagem supervisionado é uma das técnicas mais utilizadas por sistemas inteligentes, sendo, portanto, desenvolvidos diversos tipos de algoritmos. Alguns deles são: Árvores de Decisão, *K-Nearest Neighbor* e *Support Vector Machine*.

3.1.1.1 Árvores de Decisão

As Árvores de decisão, ou *decision trees*, são árvores que classificam as instâncias ordenando com base em valores de recursos. Cada nó em uma árvore representa um dado a ser classificado, e cada ramificação representa um valor que o nó pode presumir. As instâncias são classificadas começando no nó raiz e classificadas com base em seus valores de recurso, conforme exibido nas Figuras 1 e 2 (KOTSIANTIS, 2007).

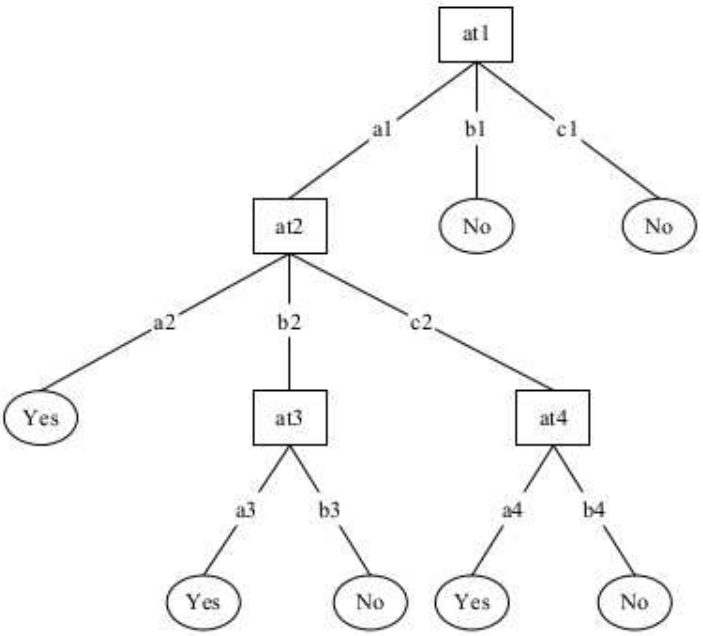

Figura 1 - Exemplo de Árvore de Decisão
(KOTSIANTIS, 2007)

at1	at2	at3	at4	Class
a1	a2	a3	a4	Yes
a1	a2	a3	b4	Yes
a1	b2	a3	a4	Yes
a1	b2	b3	b4	No
a1	c2	a3	a4	Yes
a1	c2	a3	b4	No
b1	b2	b3	b4	No
c1	b2	b3	b4	No

Figura 2 - Exemplo de dados de treinamento

(KOTSIANTIS, 2007)

Com a árvore de decisão criada, a função de classificação recebe um valor de entrada e retorna a classificação para ela, que é obtida navegando pela árvore avaliando os atributos encontrados nos nodos e seguindo pelo ramos que representa o valor do atributo até chegar em um nodo folha.

3.1.1.2 K-Nearest Neighbor

De acordo com Kotsiantis (2007), o *K-Nearest Neighbor*, ou KNN, se baseia no princípio que instâncias em um *dataset* geralmente existirão próximas a outras instâncias que têm propriedades similares. Se as instâncias estiverem categorizadas, então a categoria de uma instância não classificada pode ser determinada observando a categoria de seus vizinhos mais próximos. O KNN localiza as instâncias mais próximas da instância consultada e determina sua categoria, identificando a categoria mais frequente. A figura abaixo apresenta um exemplo de pseudocódigo para os métodos de aprendizado.

```
procedure    InstanceBaseLearner(Testing
Instances)
   for each testing instance
   {
   find the k most nearest instances of
   the  training  set  according  to  a
   distance metric
   Resulting Class= most frequent class
   label of the k nearest instances
   }
```

Figura 3 - Pseudocódigo de métodos de aprendizado para o KNN

(KOTSIANTIS, 2007)

Entretanto, o KNN pode classificar incorretamente uma instância. Quando houver ruído no local da consulta da instância, as instâncias ruidosas ganham a maioria dos votos, resultando, assim, na previsão da classe incorreta. Esse problema poderia ser evitado com um *k* maior. Outra situação é quando a região que define a classe é tão pequena que as instâncias pertencentes a classe que circunda o fragmento ganham a maioria dos votos. Um *k* menor poderia resolver este problema (KOTSIANTIS, 2007).

3.1.1.3 Support Vector Machine

O *Support Vector Machine*, ou SVM, é um método para a classificação de dados lineares e não lineares. Em poucas palavras, um SVM é um algoritmo que funciona da seguinte maneira: ele usa um mapeamento não linear para transformar os dados de treinamento originais em uma dimensão superior. Dentro dessa nova dimensão, ele procura o hiperplano separador ótimo linear (isto é, um "limite de decisão" que separa as tuplas de uma classe de outra). Com um mapeamento não linear apropriado para uma dimensão suficientemente alta, os dados de duas classes sempre podem ser separados por um hiperplano. O SVM encontra este hiperplano usando vetores de suporte (tuplas de treinamento "essenciais") e margens (definidas pelos vetores de suporte) (HAN; KAMBER; PEI, 2011).

3.1.2 Aprendizado não Supervisionado

Segundo Konar (1999), embora no aprendizado supervisionado os dados de entrada e saída são fornecidos e o sistema precisa construir uma função de mapeamento que gera a saída correta para um determinado padrão de entrada, no aprendizado não supervisionado não há um treinador. Assim, o sistema precisa construir conceitos realizando experimentos no ambiente. O ambiente responde mas não identifica quais atividades são recompensadas e quais são puníveis. Isto se deve ao fato de que os objetivos ou as saídas das instâncias de treinamento são desconhecidos; então o ambiente não pode medir o status das atividades no que diz respeito aos objetivos.

No aprendizado não supervisionado o sistema de aprendizado não necessita de um conjunto de treinamento, pois utiliza apenas as características e atributos dos dados para classificá-los, não considerando rótulos previamente definidos. O resultado esperado nesse tipo de aprendizado é a formação de agrupamentos ou clusters de acordo com a semelhança entre as características dos dados de entrada (GENTLEMAN; CAREY, 2008).

O principal método de abordagem não supervisionada é o *clustering*, que será abordado na seção seguinte.

3.1.2.1 Clustering

A técnica de *clustering*, *cluster analysis* ou, em português, agrupamento, tem como objetivo agrupar dados com base nas informações encontradas, descrevendo as suas relações. O objetivo é criar grupos de dados que possuam similaridades entre si e diferenças com dados de outros grupos (KAROUSSI, 2012). A definição desses grupos permite que novos dados sejam analisados e classificados a partir desses grupos já estabelecidos. Novos atributos podem ser descobertos com a realização de novas análises, assim, aperfeiçoando cada vez mais os grupos de dados.

Segundo Karoussi (2012), o clustering é uma técnica que vem sendo aplicada em diversos campos além do aprendizado de máquina, como por exemplo na estatística, otimização, geometria computacional, biologia, administração, psicologia e medicina.

3.1.3 Aprendizado por Reforço

De acordo com Konar (1999), no aprendizado por reforço, o sistema adapta seus parâmetros determinando o status (recompensa/punição) do sinal de feedback do ambiente. A forma mais simples de aprendizado por reforço é adotada no aprendizado de autômatos. Já Kaelbling, Littman e Moore (1995), afirmam que o aprendizado por reforço é um modelo popular de um agente que aprende o comportamento por meio de interações de tentativa e erro com um ambiente dinâmico.

No modelo padrão de aprendizado por reforço, um agente é conectado ao seu ambiente por meio de percepção e ação. A cada passo da interação, o agente recebe como

entrada alguma indicação da situação atual do ambiente; o agente então escolhe uma ação para gerar como saída. A ação altera o estado do ambiente, e o valor desse novo estado é refletido para o agente em uma entrada de reforço. O agente deve escolher ações que tendam a aumentar a soma de longo prazo dos valores do sinal de reforço; ele pode aprender a fazer isso ao longo do tempo por tentativa e erro sistemáticos (KAELBLING; LITTMAN; MOORE, 1995).

Uma maneira intuitiva de entender a relação entre agente e ambiente é através do seguinte diálogo de exemplo adaptado de Kaelbling, Littman e Moor (1995).

Ambiente: Você está no estado 65. Você possui 4 possíveis ações.

Agente: Realizarei a ação 2.

Ambiente: Você recebeu um reforço de 7 unidades. Você agora está no estado 15. Você possui 2 possíveis ações.

Agente: Realizarei a ação 1.

Ambiente: Você recebeu um reforço de -4 unidades. Você agora está no estado 65. Você possui 4 possíveis ações.

Agente: Realizarei a ação 2.

Ambiente: Você recebeu um reforço de 5 unidades. Você agora está no estado 44. Você possui 5 possíveis ações.

3.2 Mineração de Dados

Mineração de dados, ou *data mining*, é o processo de descobrir padrões a partir de grandes quantidades de dados. Como um processo de descoberta de conhecimento, normalmente envolve a coleta, limpeza, integração, seleção e transformação de dados, descoberta e avaliação de padrões e apresentação do conhecimento (HAN; KAMBER; PEI, 2011).

Um enorme número de variáveis e dados são coletados diariamente e, assim, há uma necessidade de tecnologia computacional que seja capaz de lidar com os desafios colocados por esses novos tipos de conjuntos de dados. O campo de mineração de dados cresce para extrair informações úteis dos volumes de dados em rápido crescimento, vasculhando

informações dentro dos dados que consultas e relatórios não podem revelar com eficácia. A mineração de dados ajuda a analisar relacionamentos, tendências, padrões, exceções e dados incomuns que podem passar despercebidos pelo uso de tecnologias de reconhecimento de padrões, técnicas estatísticas e matemáticas para filtrar as informações armazenadas (KAROUSSI, 2012).

Para Kalra (2013), a mineração de dados pode ser vagamente descrita como a procura de padrões em dados. Pode ser caracterizada como a extração de informações ocultas, anteriormente desconhecidas, e úteis de dados. As ferramentas de *data mining* podem prever comportamentos e tendências futuras, permitindo que questões de negócio sejam resolvidas mais rapidamente. São realizadas pesquisas em bancos de dados a procura de informações críticas que especialistas podem perder por estar fora de suas expectativas. O objetivo geral da mineração de dados é extrair informações de um banco de dados e transformá-lo em uma estrutura entendível para uso futuro.

3.3 Web Scraping

Embora *web scraping* não seja um termo novo, nos últimos anos a prática tem sido mais conhecida como *screen scraping*, *data mining*, *web harvesting*, ou variações semelhantes (MITCHELL, 2018). *Web scraping* pode ser definido como o processo de extrair e combinar conteúdos de interesse da web de maneira sistemática. Nesse processo, um agente de software, também conhecido como robô da web, imita a interação de navegação entre os servidores da web e o humano em uma travessia da web convencional. Passo a passo, o robô acessa quantos sites forem necessários, analisa seu conteúdo para encontrar e extrair dados de interesse e estrutura esses conteúdos conforme desejado (PEÑA el al., 2014).

De acordo com Mitchell (2018), *web scraping*, em teoria, é a prática de coletar dados por qualquer meio que não seja um programa interagindo com uma API. Isso é mais comumente realizado escrevendo um programa automatizado que consulta um servidor da web, solicita dados (geralmente na forma de HTML e outros arquivos que compõem páginas da web) e, em seguida, analisa esses dados para extrair as informações necessárias. Na prática, *web scraping* abrange uma ampla variedade de técnicas e tecnologias de

programação, como análise de dados, análise de linguagem natural e segurança da informação.

4. Trabalhos Relacionados

Foram realizadas pesquisas, através das ferramentas Google Scholar, IEEE e DPLP, a fim de encontrar trabalhos sobre o surf e predição de clima, vento e ondas que pudessem se relacionar com o objetivo do presente trabalho. Essa seção apresenta detalhadamente os cinco trabalhos que melhor se relacionam com o trabalho atual, além de mencionar outros estudos que também tratam sobre os temas discutidos e, por fim, é apresentada uma análise geral sobre os trabalhos citados nesta seção.

4.1 A machine learning framework to forecast wave conditions

James, Zhang e O'Donncha (2018) iniciam o estudo com uma breve explicação do contexto do trabalho, na qual ele diz que há inúmeras razões para realizar a previsão das condições das ondas e que elas são muito importantes para a economia. Além dos surfistas, há razões para fazer a previsão para os próximos dias, por exemplo, para rotas de transporte que podem ser otimizadas evitando mar agitado e reduzindo drasticamente o tempo total da viagem. Outra indústria que se beneficia disso é a indústria de aquicultura que pode otimizar consideravelmente as suas operações de colheita. Outros motivos também podem ser destacados, como o conhecimento das condições litorâneas para operações da marinha e a previsão da produção de eletricidade para manter uma rede elétrica estável.

Há um certo interesse em estabelecer uma fonte de energia que tem como origem as ondas e o departamento de energia dos Estados Unidos está realizando pesquisas nessa área. A comercialização e a implementação dessa tecnologia não só requer questões de licenciamento e regulatórias, mas também a superação de desafios tecnológicos. Um desses desafios é ser capaz de prover uma previsão precisa da geração de eletricidade. Para isso é necessária a criação de um modelo de previsão das condições do mar que deve ser extremamente veloz e capaz de incorporar dados de condições meteorológicas relevantes para suas predições.

Pelo motivo de modelos de ondas serem computacionalmente caros, uma nova abordagem com *machine learning* é abordada nesse estudo. O objetivo dessa abordagem é

treinar modelos de *machine learning* baseado no histórico da atmosfera e dos estados do mar para representar com precisão as condições da onda.

Haas et al. (2017) define recursos de energia de ondas em função da altura significativa da onda e o período de pico da onda. Essa informação pode ser usada para calcular a densidade de potência das ondas, o que é necessária para prever o potencial de energia das ondas. O oceano costeiro apresenta um desafio complexo de modelagem, ele está conectado tanto no oceano profundo como na atmosfera. Incertezas na previsão das ondas tem origem da representação matemática do sistema, aproximações numéricas e conjuntos de dados incertos e incompletos. Estudos demonstram que as maiores fontes de incertezas na previsão de ondas operacionais são os dados de entrada do modelo. Este estudo simula condições de onda sujeitas a condições reais em um local de estudo de caso, Monterey Bay, California.

Antes de desenvolver o conjunto de dados para o aprendizado do modelo de *machine learning*, foi importante verificar se o modelo SWAN (*Simulating WAves Nearshore*) pode simular características de onda com acurácia. Dois diferentes tipos de aprendizagem de máquina supervisionado foram usados para executar duas tarefas distintas: análise de regressão para altura de onda e análise de classificação para período característico. O modelo *multilayer perceptron* (MLP) usado para replicar as alturas das ondas é vagamente baseado na anatomia do cérebro. Essa rede neural artificial é composta por nodos de processamento de informação densamente interconectados e organizados em camadas. São atribuídos pesos às conexões entre os nodos, que determinará o quanto a saída de um determinado nodo contribuirá para a computação do próximo nodo. Durante o treinamento, onde a rede é apresentada com exemplos de computação que está aprendendo a realizar (ou seja, modelos de execução SWAN), esses pesos são otimizados até a última camada de trabalho da rede que se aproxima consistentemente do resultado dos dados de treinamento definido (no caso deste estudo, alturas de onda). Uma classificação *Support Vector Machine* (SVM) analisa e constrói hiperplanos que divide o conjunto de dados de treinamento em grupos rotulados. *Machine learning* tem mostrado um enorme potencial para reconhecimento de padrões em grandes datas sets.

Um modelo MLP é organizado em camadas sequências compostas por neurônios interconectados, como apresentado na figura a seguir, que foi retirada do artigo relacionado.

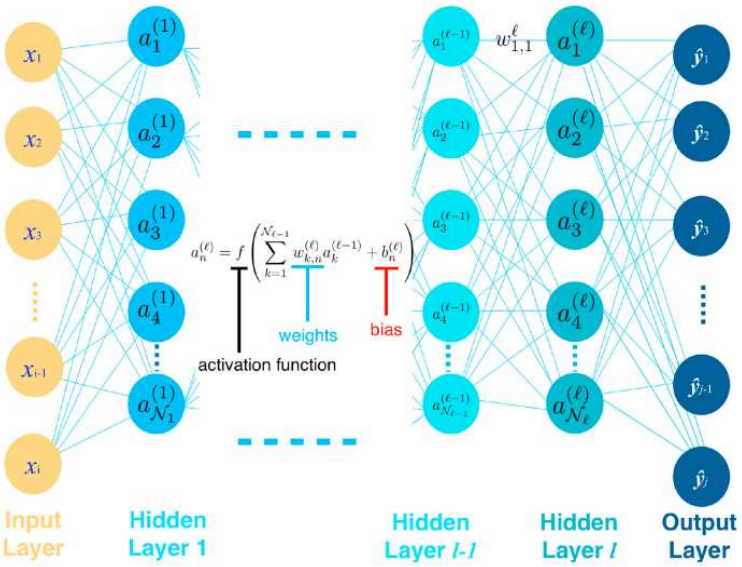

Figura 4 - Esquema uma rede de *machine learning* MLP (JAMES; ZHANG; O'DONNCHA, 2018)

Os modelos de *machine learning* podem ser executados para rapidamente gerar a altura da onda e o período característico. Para uma previsão de 48 horas (16 simulações, uma a cada 3 horas), as simulações SWAN em um processador de um único núcleo levaram 583 segundos (112 segundos com oito núcleos), enquanto que o *machine learning* levou 0,086 segundos para calcular a altura das ondas e 0,034 segundos para calcular o período característico.

James, Zhang e O'Donncha concluem afirmando que os modelos de *machine learning* foram desenvolvidos como um preciso e substituto computacionalmente eficiente para o modelo SWAN para prever a altura das ondas e o período característico. Os modelos de *machine learning* podem atuar como um sistema de previsão das condições de ondas rápido e eficiente. Essas condições de ondas podem ser usadas para estimar as condições de surf ou o potencial de geração de energia.

4.2 An integrated framework that combines machine learning and numerical models to improve wave-condition forecasts

O'Donncha, Zhang, Chen e James (2018) iniciam o estudo com uma descrição dos modelos numéricos baseados em física e eles são definidos por: (1) a fórmula física, (2)

discretização numérica e (3) dados de entrada que conduzem as simulações. Normalmente, todos os três envolvem algum grau de incerteza. A modelagem de ondas e as previsões resultam da solução do balanceamento das equações spectral-action, que são baseadas em uma aproximação da realidade derivada de um conjunto de dados incompletos.

O modelo físico Simulating WAves Nearshore (SWAN) é usado para calcular ondas aumentadas pelo vento. Os conjuntos são desenvolvidos com base em várias simulações que perturbam a entrada de dados no modelo. Uma técnica de agregação de aprendizagem usa observações históricas e previsões de modelo para calcular um peso para cada membro do conjunto. São realizadas comparações das previsões de conjunto ponderado com dados medidos para avaliar o desempenho contra o atual estado da arte. O domínio do modelo SWAN é apresentado na Figura 5.

O estudo de referência propõe um *framework* para integrar dados observacionais precisos com condições de previsão, para melhorar as capacidades preditivas. Para investigar o provável estado de solução verdadeiro do sistema, foram consideradas perturbações estatísticas de entradas (dados de limite de onda lateral) para o modelo SWAN. Isso produz um conjunto de previsões de conjunto para as próximas 48 horas. É proposta uma abordagem não invasiva de agregação de modelos que integra esses modelos com base em um conjunto de pesos aprendidos calculados, minimizando, assim, as diferenças entre os resultados do modelo e as observações em cada vez que os dados medidos se tornam disponíveis.

Esses pesos são então usados para produzir uma previsão única e determinística, ciente do melhor desempenho do modelo, tanto historicamente quanto na observação mais recente. As vantagens da estrutura proposta são: (1) não há restrições sobre quais modelos podem ser incluídos nos conjuntos e informações de modelos físicos determinísticos, modelos estocásticos ou abordagens baseadas em dados podem ser prontamente incorporadas e (2) o pesos são calculados usando os resultados do modelo, portanto, nenhuma modificação no modelo é necessária como seria necessário para assimilação de dados (DA).

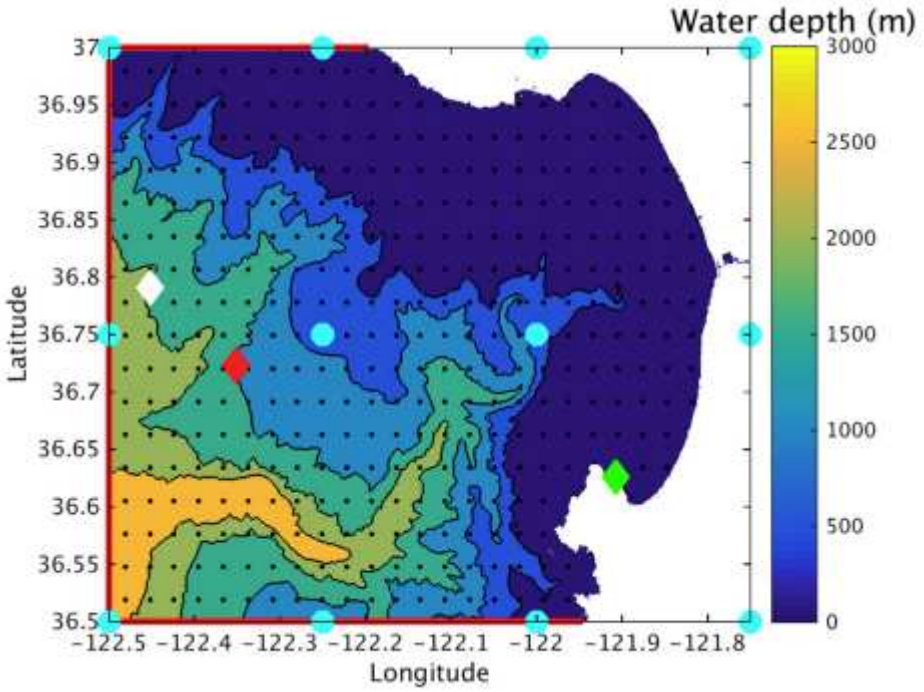

Figura 5 - Domínio do modelo SWAN com cores indicando o nível de profundidade da água (O'DONNCHA et al., 2018)

Primeiramente foi realizado um estudo utilizando o SWAN para realizar as previsões das condições das ondas, correntes oceânicas e velocidade do vento em Monterey Bay, Califórnia, que apresentaram baixa sensibilidade para perturbações dos valores de entrada do vento. Em seguida, é feita a integração desse modelo com *machine learning*, a qual apresentou resultados que demonstraram que a agregação melhorou significativamente as previsões em comparação com a abordagem tradicional.

Segundo O'Donncha et al. (2018), uma das principais vantagens desta abordagem é que ela fornece um método não invasivo para aproveitar os dados para melhorar as previsões. Como o algoritmo atua apenas nas saídas do modelo para calcular previsões de soma ponderada, ele não requer nenhuma alteração ou desenvolvimento do código-fonte, como é necessário com as abordagens tradicionais de DA. Além disso, o algoritmo pode ser facilmente substituído por abordagens alternativas de mínimos locais que refletem melhor as necessidades de um estudo específico (por exemplo, abordagens de gradiente descendente).

4.3 Artificial intelligence tools to forecast ocean waves in real time

Recentemente, países como EUA, Canadá, Austrália e Índia implementaram programas elaborados de coleta de dados oceânicos. Sob esses programas, dados de parâmetros relacionados ao oceano, incluindo alturas de ondas significativas, períodos de ondas e velocidade do vento são medidos em intervalos regulares - normalmente 1 ou 3 horas - por meio de instrumentos como bóias flutuantes e transmitidos aos usuários por meio de um esquema de disseminação de dados baseado na web. Na Índia, o Instituto Nacional de Tecnologia Oceânica (NIOT), localizado em Chennai, realiza essa coleta de dados em grande escala. Esse artigo trata de medições de ondas feitas pelo NIOT em três locais na área do Mar da Arábia. (Figura 6).

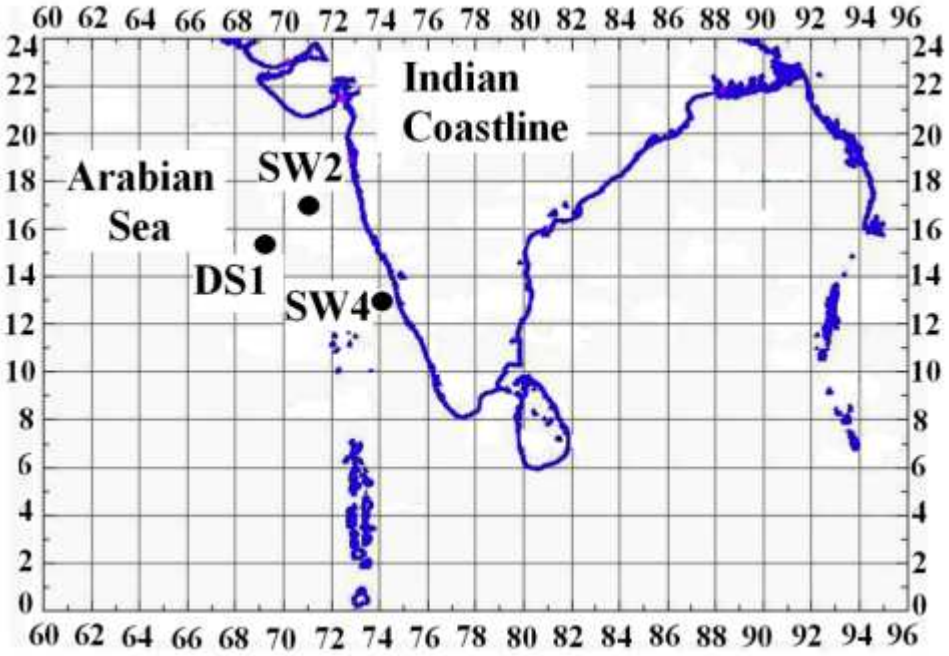

Figura 6 - Localização das bóias no Mar da Arábia
(POOJA; DEO, 2008)

Pooja e Deo (2008) buscam fornecer previsões em tempo real de alturas de ondas significativas nessas estações, considerando este exercício como um problema de previsão de séries temporais e com base em três técnicas alternativas de inteligência artificial, a saber, rede neural artificial (RNA), programação genética (GP) e árvores modelo (MT). Diferentes

ferramentas são empregadas para ver se melhores resultados são possíveis pela adoção de diferentes esquemas de aprendizagem.

Uma rede neural artificial típica consiste em uma interconexão de elementos computacionais chamados neurônios. Cada neurônio basicamente realiza a tarefa de combinar a entrada, determinando sua força comparando a combinação com um viés (ou alternativamente passando por uma função de transferência não linear) e disparando o resultado na proporção de tal força.

O GP é modelado a partir do processo de evolução que ocorre na natureza, onde as espécies sobrevivem seguindo o princípio de sobrevivência do mais apto. Essencialmente, ele transforma uma população de indivíduos em outra de maneira iterativa, seguindo as operações genéticas naturais como reprodução, mutação e crossover. Ao contrário do algoritmo genético (AG) mais conhecido, sua solução é um programa de computador ou uma equação em oposição a um conjunto de números no AG.

Em uma árvore de modelo (MT), o processo computacional é representado por uma estrutura de árvore que consiste em um nó raiz (caixa de decisão) ramificando-se para vários outros nós e folhas (Figura 7). Todo o domínio de entrada ou parâmetro é dividido em subdomínios e um modelo de regressão linear multivariado é desenvolvido para cada subdomínio (POOJA; DEO, 2008).

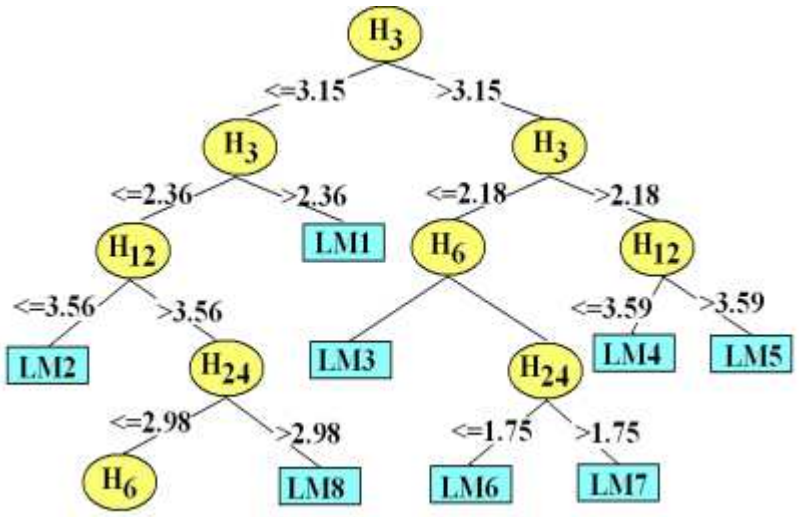

Figura 7 - Árvore de modelo utilizada

(POOJA; DEO, 2008)

Pooja e Deo afirmam que uma das vantagens do MT em relação a outras abordagens de mineração de dados como o ANN ou programas baseados em GP é que o seu resultado é compreensível e pode ser facilmente aplicado por outro usuário. As ferramentas selecionadas foram capazes de fazer previsões satisfatórias até mesmo para um lead time alto de 72 horas. No entanto, é reconhecido que essas exatidões são possíveis no atual ambiente moderado, onde as ondas-alvo eram menores do que cerca de 6 metros e 2,5 metros para as estações costeiras e offshore, respectivamente.

Uma interface gráfica foi desenvolvida a fim de que o usuário consiga acessar as previsões em qualquer um dos locais considerados por meio de uma operação baseada na web. O software cuida dos valores ausentes de uma maneira inteligente. Se um valor não for registrado, como a altura da onda, ele é imediatamente avaliado usando a regressão temporal com base em análise de série temporal univariada para cada uma das estações e posteriormente usado para o propósito de previsão. Embora a GUI atual atenda a apenas três estações ao longo da costa oeste da Índia, ela pode ser facilmente estendida para cobrir outras estações de bóia de dados.

4.4 Time-series prediction of wind speed using machine learning algorithms: A case study Osorio wind farm, Brazil

Khosravi, Machado e Nunes (2018) iniciam o artigo trazendo uma contextualização do problema que o trabalho busca minimizar, que é a escassez das fontes de energias fósseis, o aumento da poluição do ar, o aquecimento global e a crise energética. Esses fatores encorajaram a pesquisa por fontes de energias limpas e não poluentes, sendo a energia eólica a mais comum entre elas e a que mais se desenvolveu ao redor do mundo. A energia eólica é a mais acessível, inesgotável, mais barata, renovável e sustentável, além de ser ambientalmente amigável.

Desenvolver sistemas para a energia eólica pode melhorar a ideia da geração de eletricidade sem poluição no futuro. Entretanto, a integração dos parques eólicos com as redes de energia tem se tornado um problema importante para o compromisso e controle de usinas elétricas. A energia produzida pela turbina eólica está relacionada com a velocidade do vento, a qual é considerada uma das variáveis meteorológicas mais difíceis de serem

estimadas. Porém, a predição da velocidade do vento contribuirá para uma operação segura e econômica para o produtor de eletricidade (CHEN, Kuilin; YU, Jien, 2013).

Segundo Khosravi, Machado e Nunes, diversas pesquisas propuseram algoritmos de *machine learning* para prever a variação dos dados meteorológicos, como a velocidade do vento. Nesse artigo, sete tipos de algoritmos são implementados para prever os dados de velocidade do vento em série temporal (que prevê os valores futuros usando apenas os valores anteriores), sendo eles:

- MLFFNN: *Multiplayer feed forward neural network*
- GMDH: *Group method of data handling*
- SVR: *Support vector regression*
- *Fuzzy inference system* (FIS)
- ANFIS: *Adaptive neuro-fuzzy inference system*
- ANFIS-PSO: Combinação do *Particle swarm optimization* (PSO) com o modelo ANFIS
- ANFIS-GA: Interconexão entre algoritmo genérico (GA) e o modelo ANFIS

O Brasil é um dos países do mundo industrial com as maiores parcelas de energia limpa e, o estudo de caso é o parque eólico de Osório, no Rio Grande do Sul, que é conhecido por ter um grande potencial eólico, associado com uma boa condição de infraestrutura e conexão com a rede elétrica.

No estudo de referência, os métodos inteligentes são desenvolvidos baseados nos valores anteriores dos dados da velocidade do vento. Os algoritmos de *machine learning* são divididos em duas seções, o aprendizado supervisionado e o aprendizado não supervisionado. O aprendizado supervisionado utiliza técnicas de regressão (redes neurais, aprendizado *neuro-fuzzy* e SVR), e técnica de classificação (máquina de vetores de suporte, redes neurais) a fim de encontrar o modelo de previsão. Além disso, a aprendizagem não supervisionada usa a técnica de clustering, que os algoritmos mais comuns são o *fuzzy k-means*, redes neurais e mistura Gaussiana.

Em seguida, o artigo apresenta o desenvolvimento e as equações matemáticas relacionados a cada um dos sete tipos de algoritmos listados anteriormente.

A figura a seguir, retirada do artigo, ilustra o processo de previsão da velocidade do tempo em série temporal usando os algoritmos de *machine learning*. Os dados são divididos em duas seções: entradas e saídas. 70% desses dados são determinados para o treinamento do modelo, enquanto que os outros 30% são determinados para os testes.

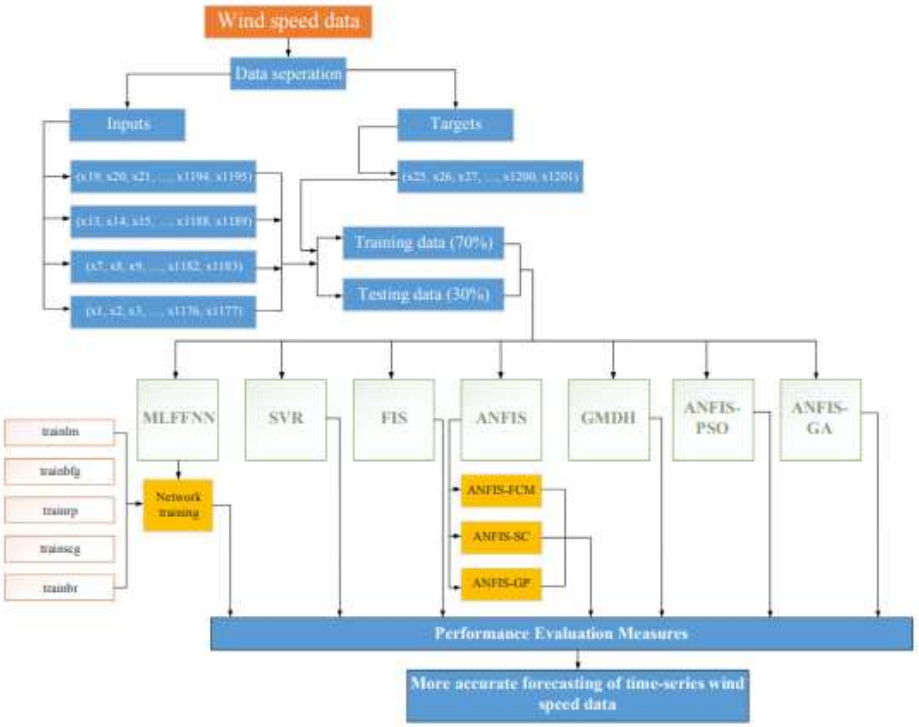

Figura 8 - Procedimento para previsão da velocidade do vento
(KHOSRAVI; MACHADO; NUNES, 2018)

Foram realizados testes utilizando quatro intervalos diferentes (5, 10, 15 e 30 minutos) e obtidos a performance dos modelos desenvolvidos durante o treinamento e os testes do dataset. Para o intervalo de 5 minutos, todos os modelos puderam com sucesso prever a velocidade do tempo na série temporal. No intervalo de 10 minutos, concluiu-se que os modelos MLFFNN, SVR, ANFIS-PSO, ANFIS-GA e GMDH performaram melhor do que os demais. Já no de 15 minutos, os modelos que apresentaram os melhores resultados foram o MLFFNN e o GMDH. Por fim, no intervalo de 30 minutos, o modelo SVR foi o destaque.

Khosravi, Machado e Nunes encerram o trabalho apresentando as principais conclusões do estudo a respeito dos diferentes tipos de algoritmos implementados, informando os pontos fortes e pontos fracos deles. Existe uma influência direta da

velocidade do vento na potência gerada de turbinas eólicas e, neste estudo, um conhecimento adequado de algoritmos de *machine learning* foi empregado para prever a velocidade do vento na região do parque eólico de Osório.

4.5 Análise da estabilidade da forma em planta e perfil nas praias da Barra da Lagoa, Moçambique e Ingleses, Florianópolis - SC: Aplicações em análise de perigos costeiros

DALINGHAUS, Charline (2016) inicialmente apresenta a área de estudo do trabalho, onde é descrita a localização que o trabalho teve como foco, assim como a geologia e geomorfologia, clima e ventos, maré, ondas e deriva litorânea, morfodinâmica e a erosão costeira. Logo após, é realizada uma síntese da área de estudo. A localização que teve como foco o trabalho pode ser observada na Figura 9.

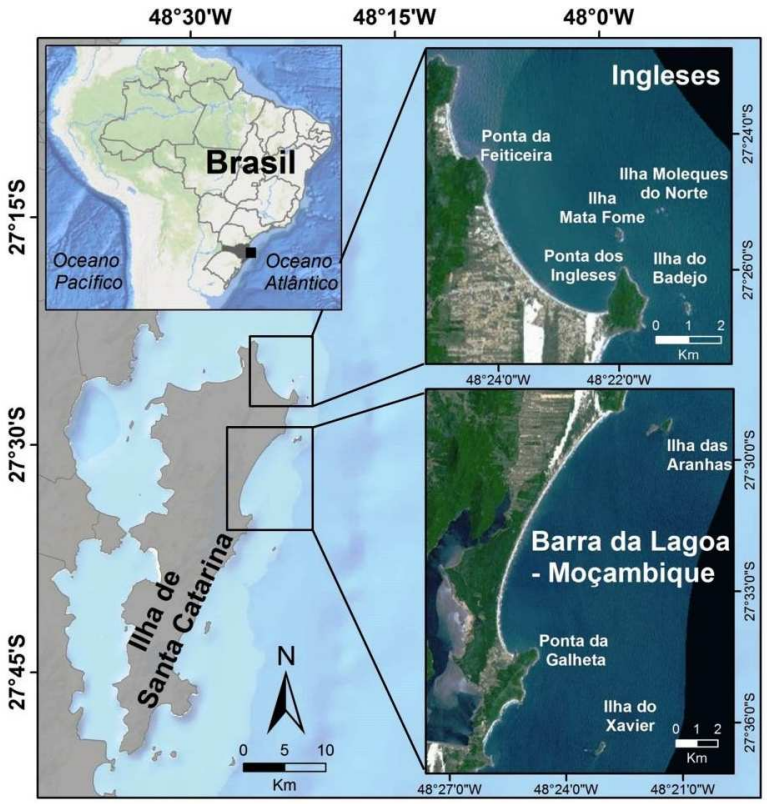

Figura 9 - Localização das praias da área de estudo

(DALINGHAUS, 2016)

Ondas são manifestações de força que agem sobre um fluido na tentativa de deformá-lo contra a ação da gravidade e a tensão superficial (DEAN; DALRYMPLE, 1991). Se uma força, como o vento, alcança tal objetivo, são geradas ondas que se propagam através de uma superfície por distâncias e tempo muito maior que o seu comprimento e período original (HOLTHUIJSEN, 2007).

Segundo a teoria linear das ondas, ao se propagarem sobre a superfície do oceano, as ondas não transportam massa, mas sim energia através do fluido (DEAN; DALRYMPLE, 1991). A energia de uma onda é a soma de duas energias: (1) cinética, que é a energia referente ao movimento orbital das partículas de água e; (2) potencial, energia das partículas como resultado do deslocamento de sua posição de equilíbrio (OPEN UNIVERSITY, 1999). Já a taxa de transmissão de energia pelo fluido é denominada fluxo de energia (HOLTHUIJSEN, 2007).

Dalinghaus revisa os conceitos bases de seu trabalho, como as características das ondas, do transporte de sedimentos, a dinâmica praial e suas formulações. Também faz uma introdução ao modelo SMC (Sistema de Modelagem Costeira) e uma explanação sobre sua base de dados. Logo após comenta sobre a hidrodinâmica básica de ondas na zona costeira, onde ela também detalha sobre o fluxo médio de energia de onda e a variabilidade do clima de ondas e sua relação com índices climáticos.

Morfodinâmica é o ajuste mútuo entre a topografia e a dinâmica do fluido envolvendo o transporte de sedimento (WRIGHT; THOM, 1977). Tais autores definem o ambiente costeiro como um sistema geomorfológico dinâmico, onde há entradas e saídas de energia e matéria bem definidas, sendo estas controladas pelas condições ambientais, onde a evolução costeira é o produto destes processos, tanto atuais como passados (herança geológica), em resposta às mudanças nas condições externas. (DALINGHAUS, 2016)

Dalinghaus comenta sobre as praias de enseada, e sobre a estabilidade praial em perfil e em planta, onde é detalhado o perfil de equilíbrio, a forma em planta das praias, e a equação da forma em planta. Comenta também sobre o Sistema de Modelagem Costeira, onde aprofunda sobre os modelos utilizados, sendo eles:

- Modelo de propagação de ondas (oluca)
- Modelo de correntes por quebra em praias (copla)

Ambos utilizam dados da base de dados SMC - Brasil, que contém dados sobre ondas, nível do oceano e batimetria.

A metodologia aplicada neste trabalho, conforme apresentada na Figura 10, consistiu em dois passos principais: i) Análise do clima de ondas em águas profundas; ii) Execução dos modelos de propagação de ondas, correntes geradas pela quebra, transporte de sedimentos, perfil de equilíbrio e forma em planta. (DALINGHAUS, 2016).

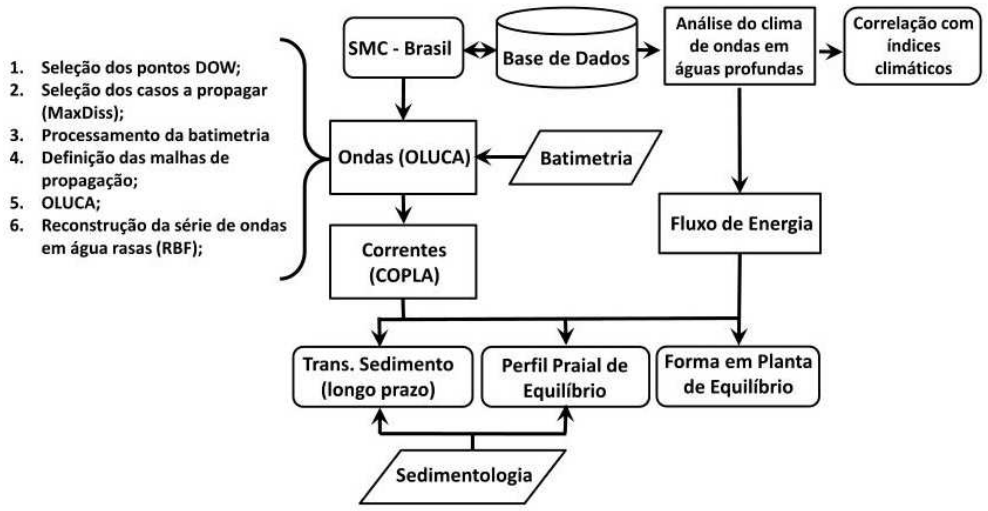

Figura 10 - Etapas da metodologia seguida no trabalho
(DALINGHAUS, 2016)

Dalinghaus comenta sobre os resultados alcançados neste trabalho. Ela discorre sobre o clima de ondas da região, assim como a modelagem de ondas e correntes realizados e suas implicações nos resultados do transporte de sedimentos, forma em planta e perfil das praias.

Logo após os resultados, apresenta uma discussão aprofundada sobre os resultados encontrados e se estes estão de acordo com o que encontra-se na literatura. Dalinghaus discute sobre a análise da variabilidade do clima de ondas em águas profundas, onde aprofunda sobre a correlação com índices climáticos, e sobre o transporte de sedimentos litorâneos e análise da estabilidade das praias da Barra da Lagoa - Moçambique e dos Ingleses.

Seu trabalho teve como objetivo analisar a estabilidade das praias da Barra da Lagoa – Moçambique e Ingleses através da utilização de modelagem numérica, diante do

conhecimento prévio do clima de ondas local e do transporte litorâneo (DALINGHAUS, 2016).

4.6 Outros Trabalhos

Para o desenvolvimento desta seção de trabalhos relacionados, foram analisados diversos trabalhos através das ferramentas Google Scholar, IEEE e DPLP. Foram utilizadas as seguintes palavras-chaves, com suas variações linguísticas (plural e Inglês/Português): surf, predição de vento, clima e onda, aprendizagem de máquina, inteligência artificial e mineração de dados. Dentre diversos trabalhos encontrados, há alguns que são importantes a serem mencionados neste trabalho.

Scarge, Elwany e Mead (2003) escreveram um estudo sobre a criação das ondas e a influência da forma da praia nesse processo. Este artigo não tem nenhuma associação com o aprendizado de máquina ou inteligência artificial, porém traz informações importantes referentes ao assunto abordado pelo atual trabalho.

Stetler e Saxton (1997) realizaram uma detalhada análise do vento, calculando a energia gerada por ele e destacando a sua relação com a velocidade e duração do evento, dados que podem ser utilizados para a predição de ondas.

No estudo de Cai *et al.* (2019) foram utilizados dados de satélites e dados climáticos para construir modelos estatísticos para previsão de rendimento de trigo na Austrália. Três métodos de machine learning (*Random forest* (RF), *Support Vector Machine* (SVM) e rede neural) e um método avançado de regressão (LASSO) foram utilizados e seus resultados comparados e validados.

Bertotti e Cavaleri (2009) fazem uma análise da qualidade da predição do vento e de ondas no Mar Adriático, na costa italiana. Embora o estudo não esteja diretamente ligado ao aprendizado de máquina ou inteligência artificial, ele nos traz informações relevantes a respeito do domínio estudado.

Apesar de serem poucos os trabalhos encontrados que se relacionam diretamente com mais de um tópico deste estudo, as informações obtidas através de cada um dos trabalhos apresentados nesta seção são de grande importância para que seja possível entender melhor a respeito do domínio estudado e das metodologias abordadas.

4.7 Considerações

A Tabela 1 faz uma comparação entre os cinco trabalhos apresentados em relação aos modelos de *machine learning* utilizados, se utilizam inteligência artificial e quais métricas de avaliação dos modelos foram utilizadas.

Tabela 1 - Comparação dos tipos, modelos de IA e métricas de avaliação utilizados dos trabalhos relacionados

Autor	**Utiliza IA**	**Tipo**	**Quais modelos**	**Métricas de avaliação**
James, Zhang e O'Donncha	Sim	Machine learning, supervisionada	Regressão e classificação. MLP e SVM	RMSE (Root Mean Square Error)
O'Donncha et al.	Sim	Machine learning	SWAN, Ridge Regression e Exponentiated Gradient	RMSE e MAPE (Mean Absolute Percentage Error)
Pooja e Deo	Sim	Machine learning, supervisionada	RNA, GP e MT	RMSE e MAE (Mean Absolute Error)
Khosravi, Machado e Nunes	Sim	Machine learning, supervisionada e não supervisionada	MLFFNN, GMDH, SVR, Fuzzy inference system (FIS), ANFIS, ANFIS-PSO: Combinação do Particle swarm optimization (PSO) com o modelo ANFIS, ANFIS-GA: Interconexão entre algoritmo genérico (GA) e o modelo ANFIS	RMSE, MSE (Mean Squared Error) e R (Correlation Coefficient)
Dalinghaus	Não	N/A	N/A	N/A

A Tabela 2 compara os trabalhos, verificando se utilizam *web scrapping* e o tamanho dos seus *datasets*.

Tabela 2 - Comparação dos métodos e datasets dos trabalhos relacionados

Autor	Utiliza Web Scraping	Tamanho do Dataset
James, Zhang e O'Donncha	Não	Matriz com 11078 linhas e 741 colunas
O'Donncha et al.	Não	N/A
Pooja e Deo	Não	3 locais, 3 a 7 anos, com intervalo de 3 horas
Khosravi, Machado e Nunes	Não	N/A
Dalinghaus	Não	N/A

A Tabela 3 compara os trabalhos em relação aos seus domínios, se são aplicados no surf e quais são os aspectos climáticos em que foram realizadas as predições.

Tabela 3 - Comparação dos domínios dos trabalhos relacionados

Autor	Domínio aplicado no Surf	Domínio envolve ventos	Domínio envolve ondulações
James, Zhang e O'Donncha	Não	Sim	Sim
O'Donncha et al.	Não	Sim	Sim
Pooja e Deo	Não	Não	Sim
Khosravi, Machado e Nunes	Não	Sim	Não
Dalinghaus	Sim	Sim	Sim

Assim como o *framework* de *machine learning* desenvolvido por James, Zhang e O'Donncha (2018), que utiliza informações como altura e período das ondas para prever as condições das mesmas, o presente trabalho também tem como finalidade construir um

modelo de predição de ondas utilizando aprendizado de máquina, porém, especificamente para a prática do surf, diferenciando-se dos trabalhos relacionados que abordaram outros domínios.

5. Desenvolvimento

O objetivo deste trabalho, conforme descrito no Capítulo 2, é criar um modelo de *machine learning* que consiga avaliar a condição da praia para a prática do surf, considerando-se os dados de entrada para o treinamento e construção do modelo, que são:

- Período da ondulação
- Altura da ondulação
- Direção da ondulação
- Velocidade do vento
- Direção do vento
- Localização (praias de Florianópolis)

Como explicado anteriormente na seção de introdução do atual trabalho, essas são as principais variáveis que influenciam na classificação da condição de uma praia para a prática do esporte. Além disso, esses dados são de fácil acesso, com dados históricos auxiliando na construção da base de dados, no site Windguru, enquanto que os dados resultantes são encontrados no site Waves, que serão apresentados nas seções seguintes.

Para cumprir esse objetivo, o desenvolvimento do modelo proposto segue um fluxo composto por quatro etapas, onde é realizada a coleta dos dados, a preparação dos dados, o treinamento do modelo, e o teste/análise. Pode-se verificar o fluxograma destas etapas na Figura 11.

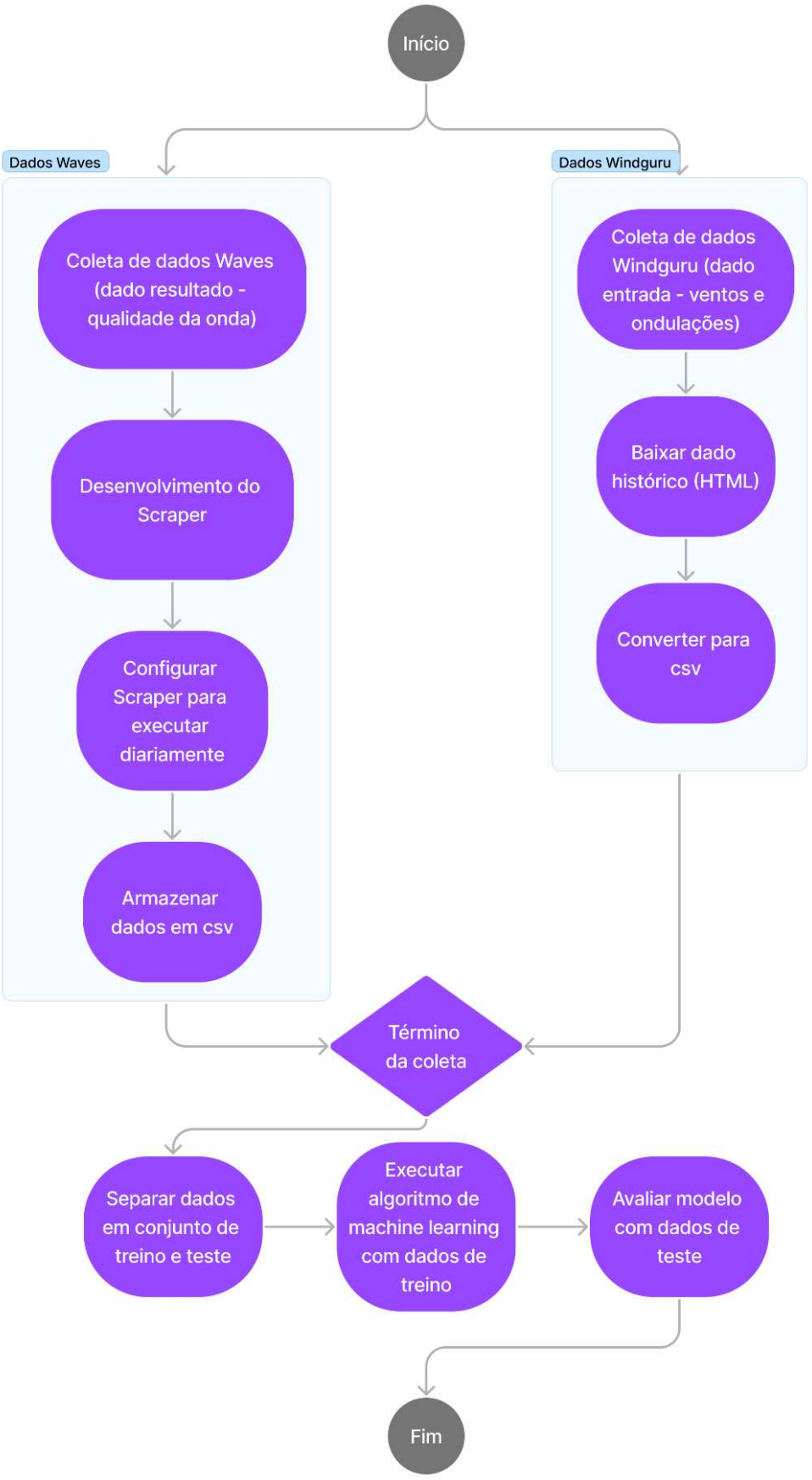

Figura 11 - Fluxograma do desenvolvimento da solução

5.1 Ferramentas

Para o desenvolvimento do Scraper foi utilizado o C# (C Sharp), que é uma linguagem de programação orientada a objetos criada pela Microsoft e que faz parte da sua plataforma .Net. A Microsoft baseou o C# na linguagem C++ e Java e é utilizada em diversos tipos de aplicações.

Para a configuração do Scraper, foi utilizada a AWS (Amazon Web Services) que é um serviço de computação em nuvem desenvolvido pela Amazon. Ela oferece mais de 200 serviços completos de data centers por todo o mundo e, por este motivo, acaba trazendo mais recursos do que outros provedores de nuvem.

Para o treinamento e teste do modelo, foi utilizado o Python, que é uma linguagem de programação de alto nível, com tipagem dinâmica e forte, multiplataforma e orientada a objetos, uma forma específica de organizar softwares onde, os procedimentos estão submetidos às classes, o que possibilita maior controle e estabilidade de códigos para projetos de grandes proporções.

5.2 Coleta e Preparação de Dados

Na etapa de coleta de dados, duas fontes são utilizadas, uma para os dados de entrada, Windguru, e uma para os dados de saída, Waves.

5.2.1 Windguru - Dados de Entrada

Windguru[1] é uma plataforma que descreve e detalha informações sobre o vento, clima e ondulações de uma determinada região, conforme apresentado na Figura 12, havendo previsões para até 7 dias. Há múltiplas métricas sobre o vento, clima e ondulações. Para este trabalho será utilizado um subconjunto destas métricas.

[1] https://www.windguru.cz

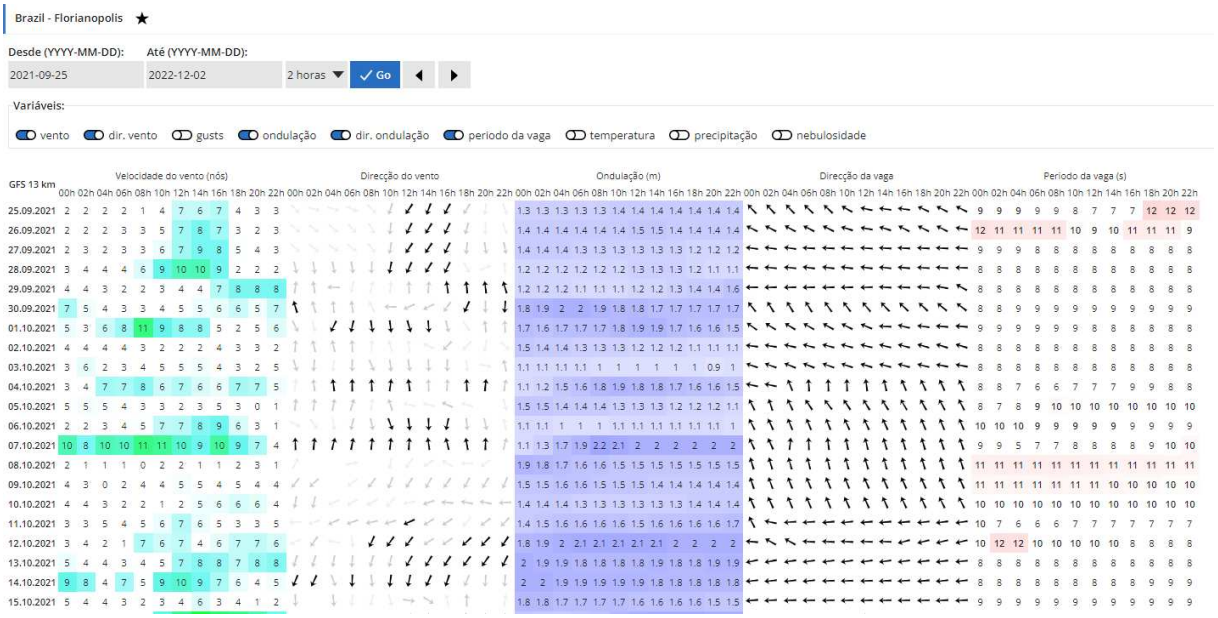

Figura 12 - Windguru

(https://www.windguru.cz/archive.php?id_spot=105160&id_model=3&date_from=2022-11-02&date_to=2022-12-02, acesso em 02/12/2022)

No Windguru há dados históricos que são atualizados diariamente e com intervalos de 2 horas. Como os dados do Waves são atualizados todos os dias entre às 7 e 9 horas da manhã, serão utilizados os dados do Windguru das 8 horas da manhã. Será necessário baixar o histórico em HTML, e desenvolver um conversor para CSV. O conversor será implementado utilizando .Net Core com C#.

5.2.2 Waves - Dados de Saída

Waves[2] é uma plataforma que tem como objetivo informar os praticantes do surf sobre a condição das ondas no dia atual. Na plataforma é possível escolher o estado desejado, e visualizar praia a praia a altura das ondas, e a condição, podendo ser ruim, regular ou boa, conforme apresentado na Figura 13. A plataforma é atualizada diariamente, havendo pessoas que inserem a informação para cada praia individualmente, classificando a altura e a qualidade da onda.

[2] https://www.waves.com.br

Figura 13 - Waves

(https://www.waves.com.br/surf/ondas/condicao/santa-catarina/, acesso em 15/10/2021)

No Waves, não há dados históricos, os dados são atualizados diariamente e, por esse motivo, tornou-se necessário o desenvolvimento de um Scraper, que acessa o site, baixa, formata e salva os dados. O Scraper foi implementado utilizando a linguagem .Net Core 3.1 com C#, publicado na AWS Lambda, e os dados coletados são armazenados no AWS S3. Também foi necessário fazer a configuração para o Scraper ser executado diariamente, esta que foi realizada na própria AWS, no serviço EventBridge.

Esta etapa da coleta foi feita o mais rápido possível, para obter a maior quantidade de dados, e aprimorar a precisão do modelo.

5.2.3 Dados Coletados

Foram publicados dois scripts escritos em C# .Net Core 3.1 na Lambda AWS, um para acessar a plataforma Waves e salvar o HTML em um bucket no S3, e outro para acessar a plataforma Waves e salvar o CSV em um bucket no S3. Esses passos para a coleta dos dados são exibidos a seguir nas Figuras 14, 15, 16 e 17.

Figura 14 - Funções Lambda AWS

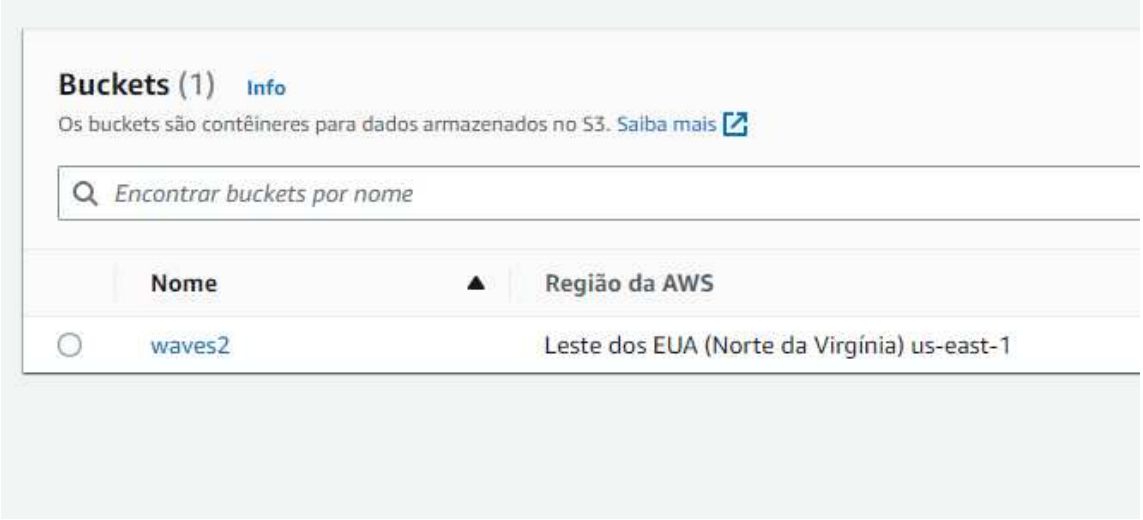

Figura 15 - Lista de Buckets do S3, contendo o bucket utilizado (waves2)

Nome	Tipo	Última modificação	Tamanho	Classe de armazenamento
waves-20211026.csv	csv	26 Oct 2021 05:15:54 PM -03	578.0 B	Padrão
waves-20211026.html	html	26 Oct 2021 05:15:54 PM -03	184.3 KB	Padrão
waves-20211027.csv	csv	27 Oct 2021 05:15:54 PM -03	647.0 B	Padrão
waves-20211027.html	html	27 Oct 2021 05:15:54 PM -03	183.7 KB	Padrão
waves-20211028.csv	csv	28 Oct 2021 05:15:54 PM -03	523.0 B	Padrão
waves-20211028.html	html	28 Oct 2021 05:15:54 PM -03	183.9 KB	Padrão
waves-20211029.csv	csv	29 Oct 2021 05:15:53 PM -03	463.0 B	Padrão
waves-20211029.html	html	29 Oct 2021 05:15:53 PM -03	177.3 KB	Padrão
waves-20211030.csv	csv	30 Oct 2021 05:15:54 PM -03	411.0 B	Padrão
waves-20211030.html	html	30 Oct 2021 05:15:53 PM -03	178.2 KB	Padrão
waves-20211031.csv	csv	31 Oct 2021 05:15:54 PM -03	156.0 B	Padrão

Figura 16 - Bucket waves2 do S3, com os dados coletados

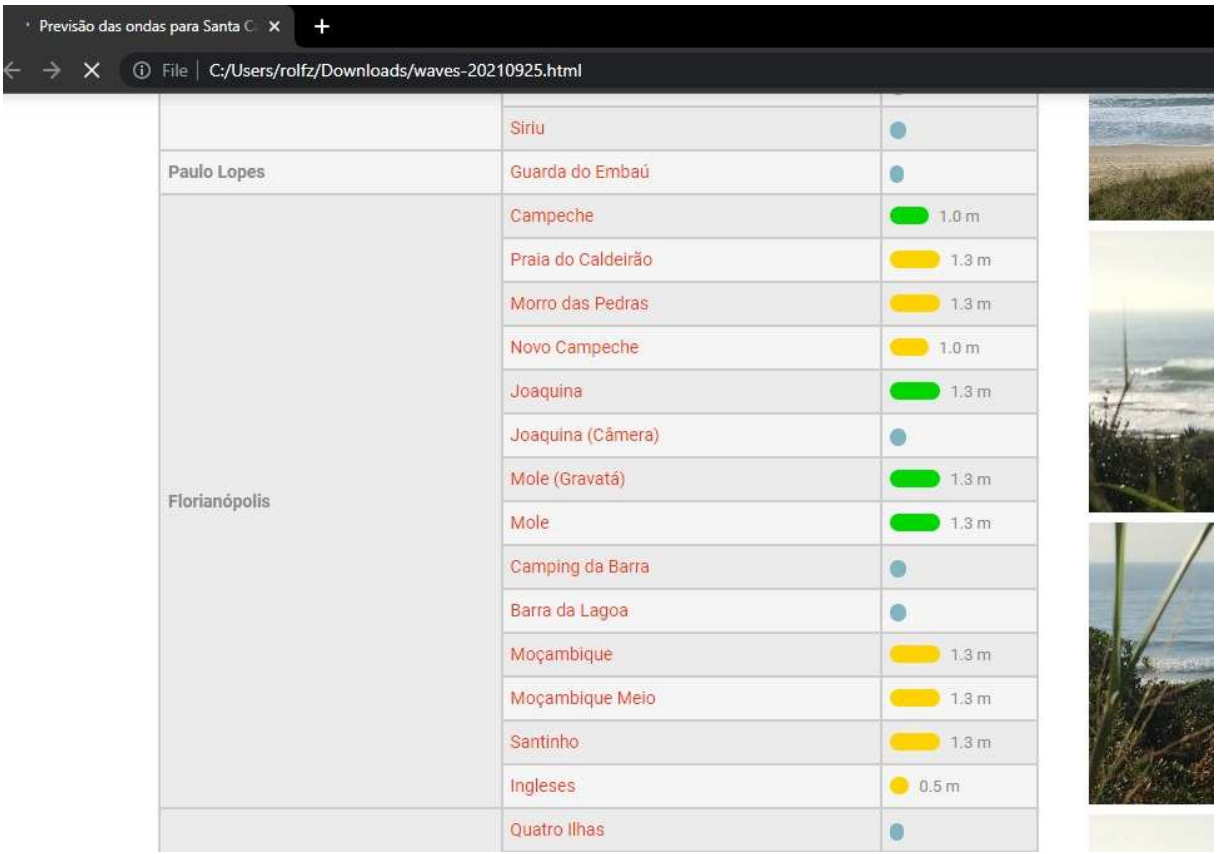

Figura 17 - HTML salvo do Waves

Para programar a execução da função Lambda diariamente, foi criada uma regra no EventBridge AWS, que, todos os dias às 20:15 UTC, dispara ambas as funções Lambda (WavesHTML e WavesCSV). Abaixo, as figuras 18, 19 e 20 apresentam os passos citados para a realização deste processo.

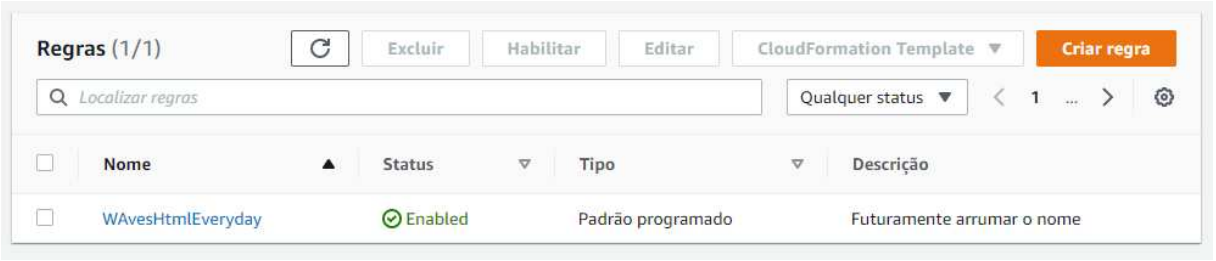

Figura 18 - Lista de regras EventBridge AWS, contendo a regra utilizada (WavesHtmlEveryday)

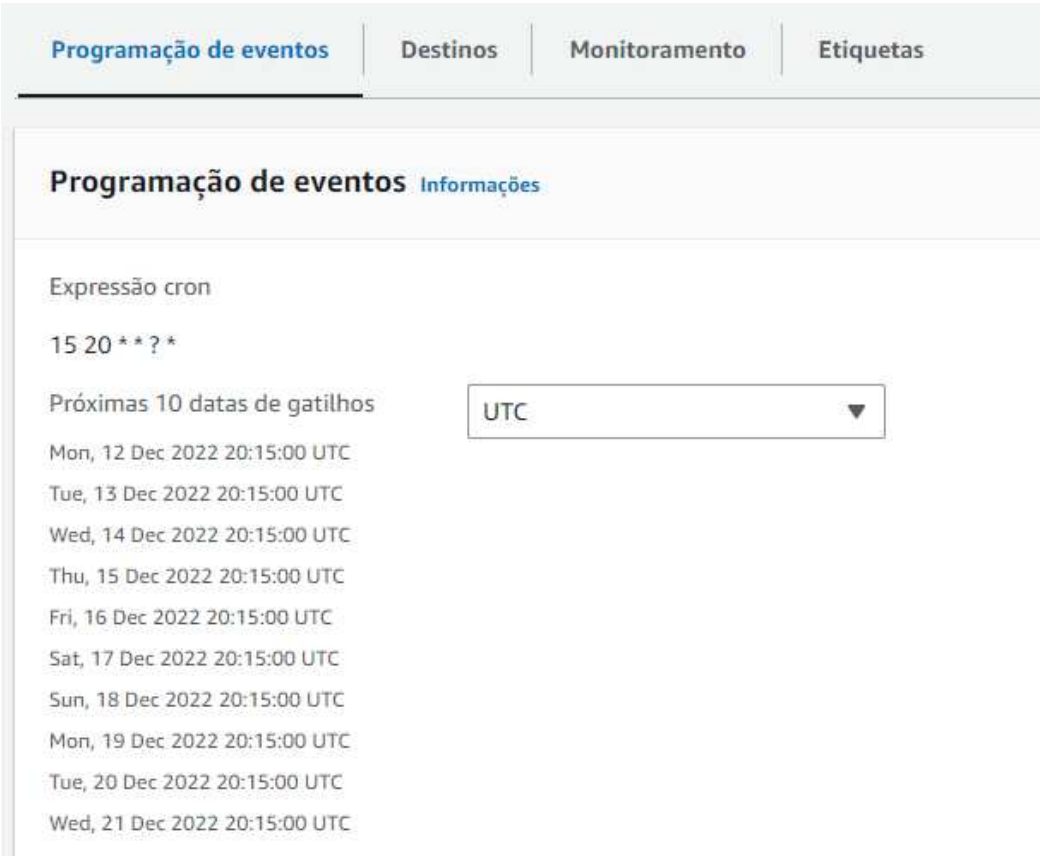

Figura 19 - Programação para execução diária

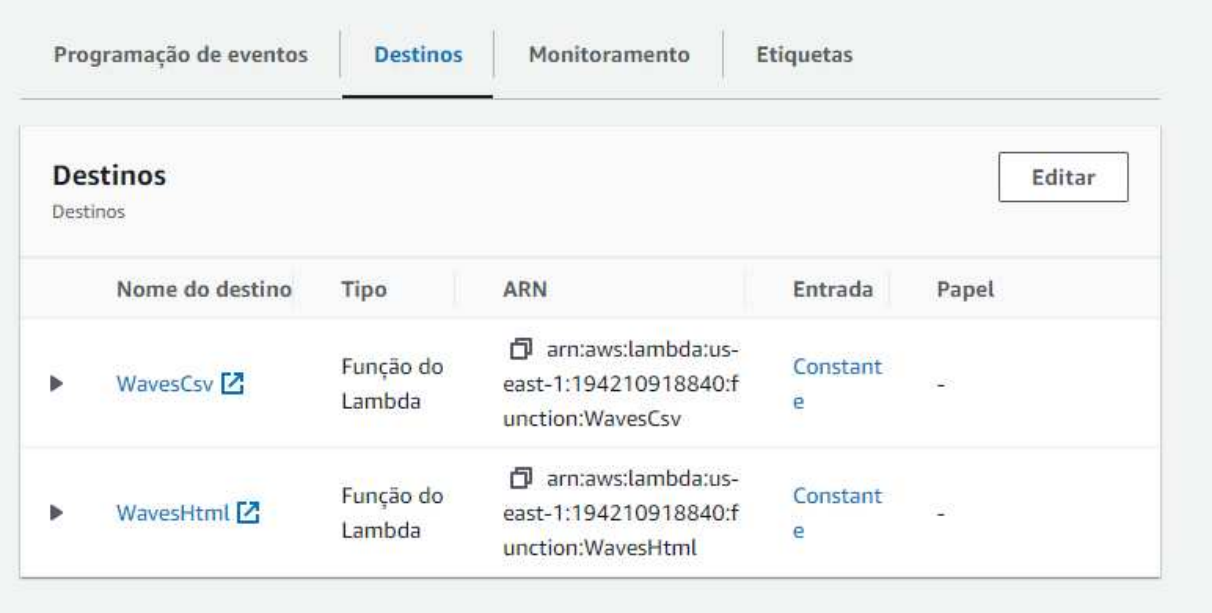

Figura 20 - Funções Lambda que são disparadas

Com esta infraestrutura suportada pela AWS, foi possível realizar a coleta dos dados diariamente automatizada da plataforma Waves, que permitiu a obtenção de dados de saída de qualidade para a nossa aplicação.

Os dados foram carregados de CSVs para *DataFrames*, utilizando a biblioteca Pandas em Python, e organizados para o treinamento e teste do modelo, através da associação entre os mesmos pela data. Na seção a seguir será realizada uma análise descritiva das colunas que compõem os *DataFrames*. Em sequência, a Figura 21 apresenta o *DataFrame* dos dados coletados do Windguru, a figura 22 exibe o *DataFrame* dos dados coletados do Waves da praia de Moçambique, praia em que foi possível coletar a maior quantidade de dados e, na figura 23, é possível observar o *DataFrame* dos dados fundidos e preparados para o treinamento do modelo proposto.

	Data	Hora	Velocidade vento (em knots)	Direcao vento	Tamanho onda (em metros)	Direcao onda	Periodo onda (em segundos)
0	2021/09/25	0	2	NW	1.3	SE	9
1	2021/09/25	1	2	NW	1.3	SE	9
2	2021/09/25	2	2	W	1.3	SE	9
3	2021/09/25	3	2	W	1.3	SE	9
4	2021/09/25	4	2	W	1.3	SE	9
...	...	...	...	...	...	...	...
13195	2023/03/30	19	2	S	1	SE	10
13196	2023/03/30	20	0	-	1	SE	10
13197	2023/03/30	21	1	-	0.9	SE	10
13198	2023/03/30	22	2	W	0.9	SE	10
13199	2023/03/30	23	2	W	0.9	SE	10

13200 rows × 7 columns

Figura 21 - DataFrame dos dados coletados do Windguru

	Praia	Condicao	Tamanho (em metros)	Year	Month	Day	Date	datetime
2032	Moçambique	regular	1.3	2021	09	25	2021/09/25	2021-09-25
2217	Moçambique	regular	1.7	2021	09	26	2021/09/26	2021-09-26
2070	Moçambique	regular	1.5	2021	09	27	2021/09/27	2021-09-27
495	Moçambique	regular	0.7	2021	09	28	2021/09/28	2021-09-28
720	Moçambique	regular	0.5	2021	09	29	2021/09/29	2021-09-29
...	...	...	...	...	...	...	...	...
1008	Moçambique	boa	1.0	2023	03	26	2023/03/26	2023-03-26
1134	Moçambique	ruim	0.3	2023	03	31	2023/03/31	2023-03-31
2401	Moçambique	regular	3.0	2023	04	03	2023/04/03	2023-04-03
1868	Moçambique	regular	1.5	2023	04	04	2023/04/04	2023-04-04
2038	Moçambique	regular	1.5	2023	04	05	2023/04/05	2023-04-05

314 rows × 8 columns

Figura 22 - DataFrame dos dados da praia de Moçambique coletados do Waves

	Velocidade vento (em knots)	Direcao vento	Tamanho onda (em metros)	Direcao onda	Periodo onda (em segundos)	Condicao
0	1	W	1.3	SE	9	regular
1	3	NW	1.4	SE	11	regular
2	2	NW	1.3	E	8	regular
3	4	N	1.2	E	8	regular
4	2	SW	1.1	E	8	regular
...	...	...	...	...	...	...
305	1	SW	0.9	SE	11	ruim
306	3	S	0.9	S	9	ruim
307	2	NW	1.4	SE	12	regular
308	2	SW	1.4	SE	7	regular
309	4	N	1.3	SE	13	boa

293 rows × 6 columns

Figura 23 - DataFrame dos dados fundidos e preparados para o treinamento

5.3 Análise Descritiva

A fim de realizar as análises deste trabalho, tornou-se necessário organizar as informações mais importantes para determinar a condição das ondas para o surf. Assim, as análises utilizaram os dados físicos coletados diariamente às 7 horas da manhã, horário em que usualmente são lançados os boletins das ondas para os surfistas se informarem sobre as melhores praias do dia para praticar o esporte. Os seguintes dados de entrada foram armazenados em um *DataFrame*, utilizando a linguagem de programação Python, para melhor manipulação dos mesmos:

- Velocidade do vento: velocidade do vento em *knots* (unidade de medida de velocidade equivalente a uma milha náutica por hora, ou seja, 1,852 km/h)

- Direção do vento: direção do vento, podendo ser os 8 pontos cardeais (N - NE - E - SE - S - SW - W - NW)

- Tamanho da onda: tamanho da onda medida em metros

- Direção da onda: direção da onda, podendo ser os 8 pontos cardeais

- Período da onda: tempo necessário para a formação de um comprimento de onda, medido em segundos

- Localização: nome da praia na qual foram coletados os dados

Foi adicionado também o dado de saída para que fosse possível treinar o modelo de *machine learning* e posteriormente realizar as previsões das condições da onda sem qualquer tipo de supervisão humana.

- Condição da onda: condição da onda para a prática do surf, podendo ser: ruim, regular ou boa

5.4 Análise Exploratória

A análise exploratória é a etapa da estatística na qual são gerados novos conhecimentos através da análise de gráficos e matrizes gerados a partir dos dados coletados no experimento. Para realizar essa fase de análise, foram utilizados gráficos que buscam relacionar a quantidade total de ocorrências de um dado, onde quanto mais forte a cor maior a quantidade total, com o valor final da condição da onda.

Na Figura 24, podemos observar a quantidade total das ocorrências de cada uma das classificações das condições da onda. Dos dados coletados, a grande maioria foram de ondas classificadas como regulares, em seguida, ondas com a condição ruim e, por último com menos ocorrências, ondas boas, totalizando 293 entradas.

É possível verificar na Figura 25 que as ondas com as melhores qualidades para praticar o esporte apresentaram tamanhos entre 1 metro e 1,8 metros, ressaltando que tamanhos extremos de onda, tanto quanto ondas muito pequenas quanto ondas muito grandes, não são vantajosas para este fim. Nesse gráfico, o eixo X representa o tamanho da onda em metros e o eixo Y é a condição da onda.

Em relação à velocidade do vento, podemos observar que o valor ideal seja de 1 a 4 knots. Neste caso, valores muito altos tendem a diminuir a qualidade da onda. Como visto na Figura 26, onde o eixo X representa a velocidade do vento em *knots* e o eixo Y a qualidade da onda.

O período da onda tende a possuir a mesma característica do que a do tamanho, onde valores extremos não são vantajosos. O tempo ideal para a formação da onda tende a ser de 6 a 13 segundos, conforme exibido na Figura 27, que o eixo X é representado pelo período da onda em segundos e o eixo Y a condição da onda para o surf.

Já as Figuras 28 e 29, apresentam as direções do vento e da onda, representadas pelo eixo X, que apresentaram os maiores níveis de ocorrência em relação a cada uma das condições de onda, representada pelo eixo Y. Sendo elas:

- Vento: oeste, noroeste e norte
- Onda: Sudeste e leste

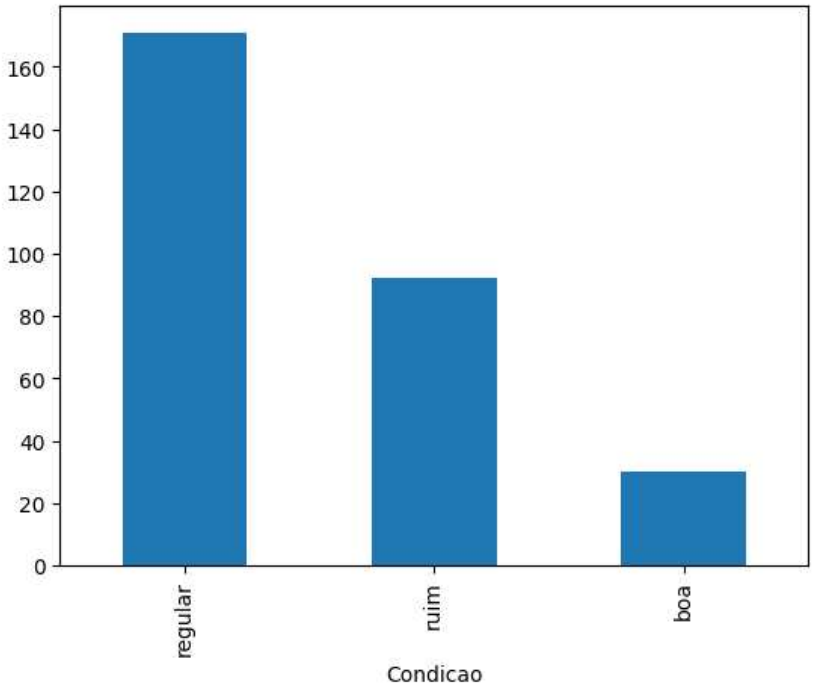

Figura 24 - Gráfico da quantidade total de ocorrências da classificação das condições da onda

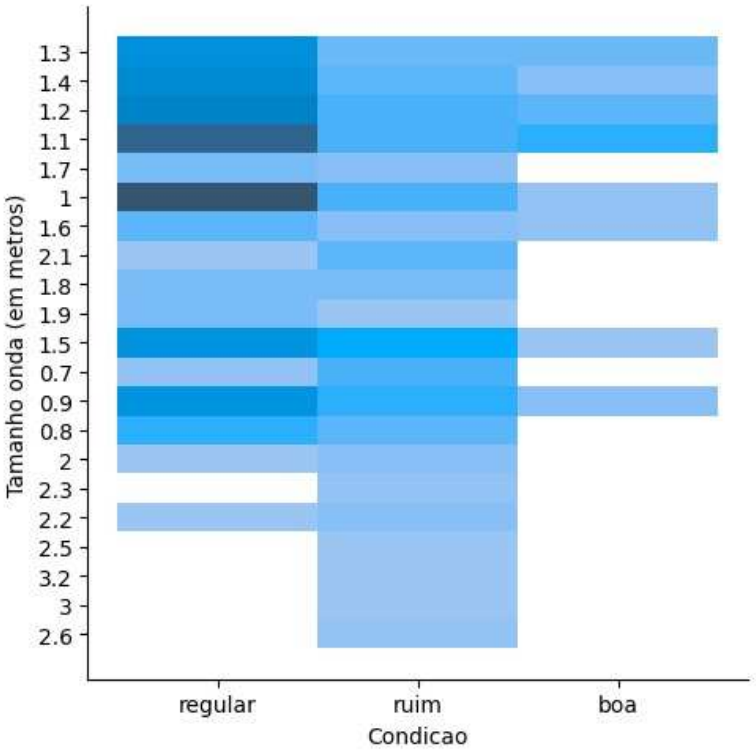

Figura 25 - Gráfico das ocorrências dos tamanhos das ondas em relação à condição da onda para o surf

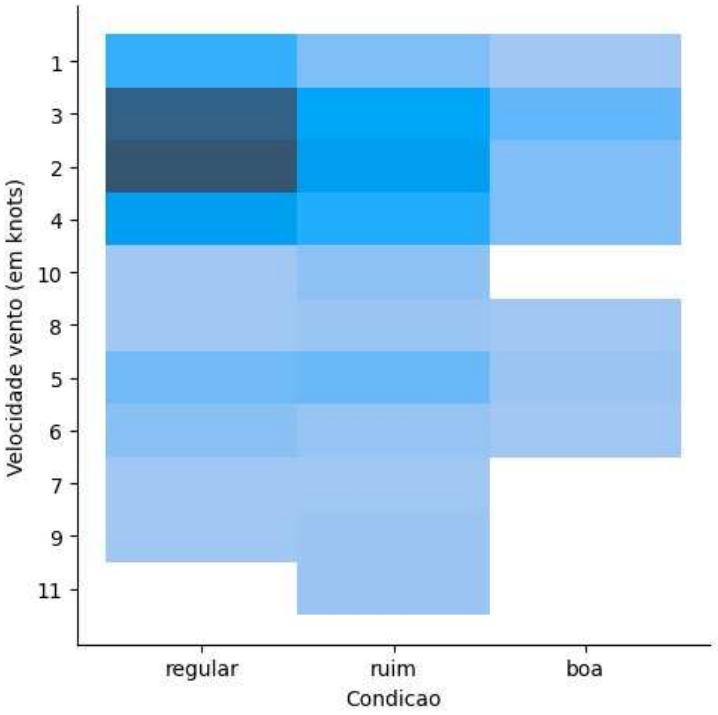

Figura 26 - Gráfico das ocorrências da velocidade dos ventos em relação à condição da onda para o surf

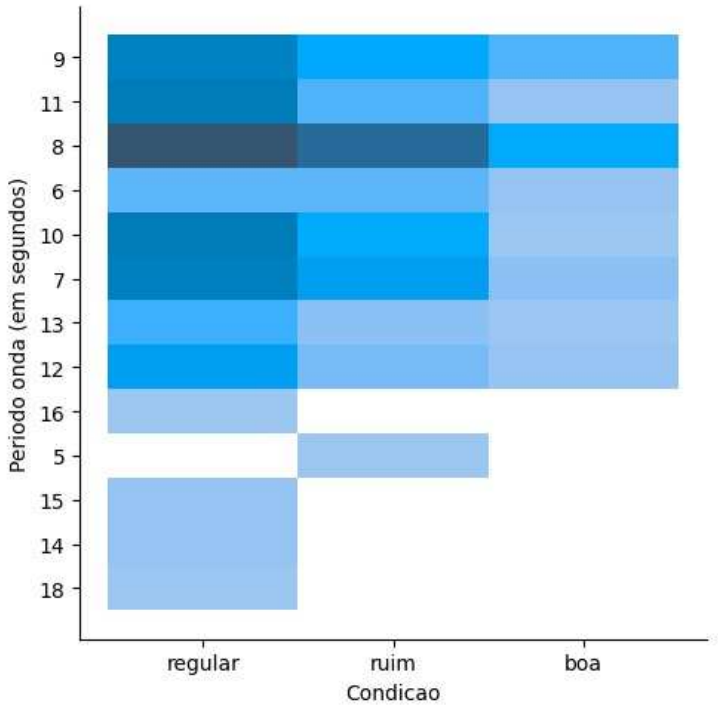

Figura 27 - Gráfico das ocorrências dos períodos das ondas em relação à condição da onda para o surf

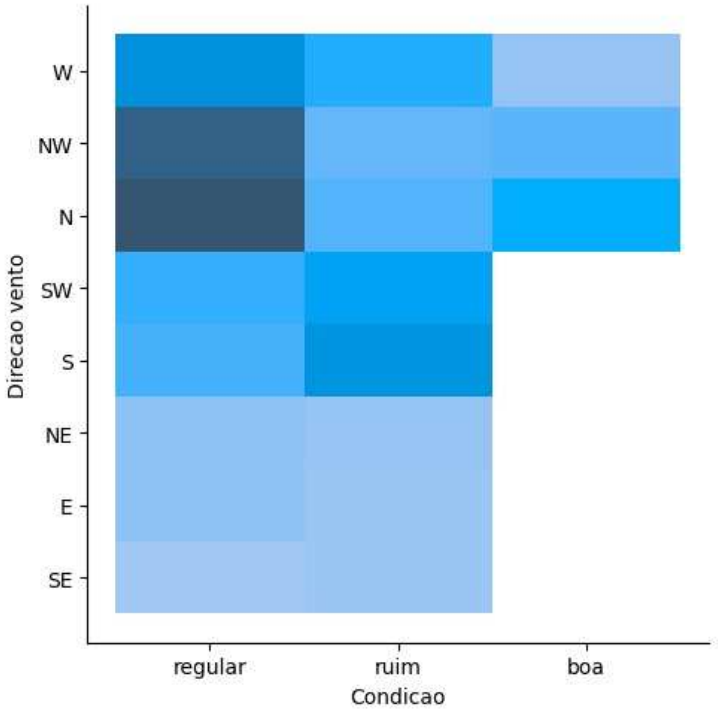

Figura 28 - Gráfico das ocorrências das direções dos ventos em relação à condição da onda para o surf

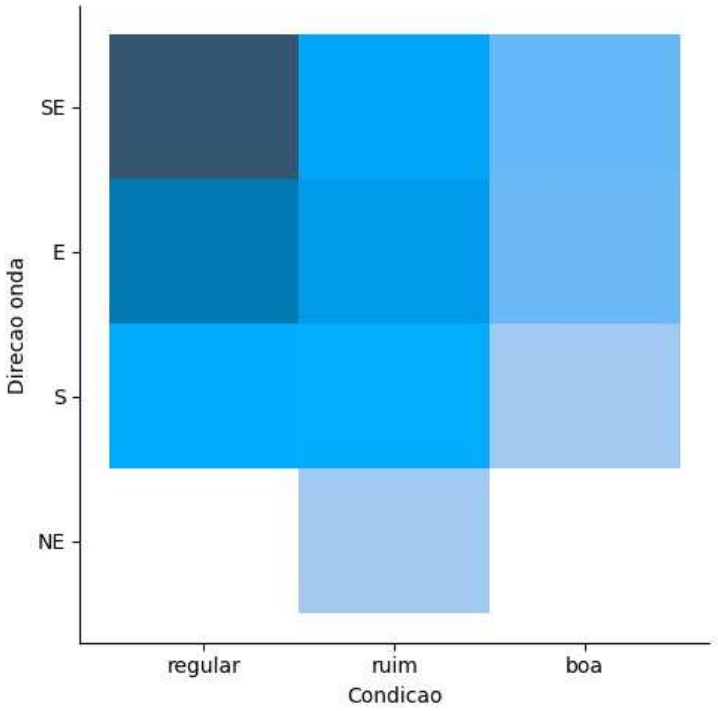

Figura 29 - Gráfico das ocorrências das direções das ondas em relação à condição da onda para o surf

É também importante observarmos a correlação entre as variáveis. O coeficiente de correlação é uma medida estatística que mede a associação entre variáveis através de valores entre -1 e 1. Quanto mais próximo de 1, podemos definir que as variáveis possuem uma forte correlação e nota-se um aumento no valor de uma variável quando a outra também aumenta. Quanto mais próximo de -1, as variáveis também possuem uma forte correlação, entretanto, uma variável tende a diminuir enquanto que o valor da outra aumenta. Um coeficiente de correlação próximo de zero indica que há uma baixa correlação entre as duas variáveis.

A Figura 30 é uma matriz de correlação entre as variáveis utilizadas no presente trabalho, indicando que poucas delas possuem uma correlação direta. Como podemos visualizar, a direção e o período da onda são as variáveis mais correlacionadas, enquanto que a direção da onda e a velocidade do vento foram as variáveis com o menor índice de correlação.

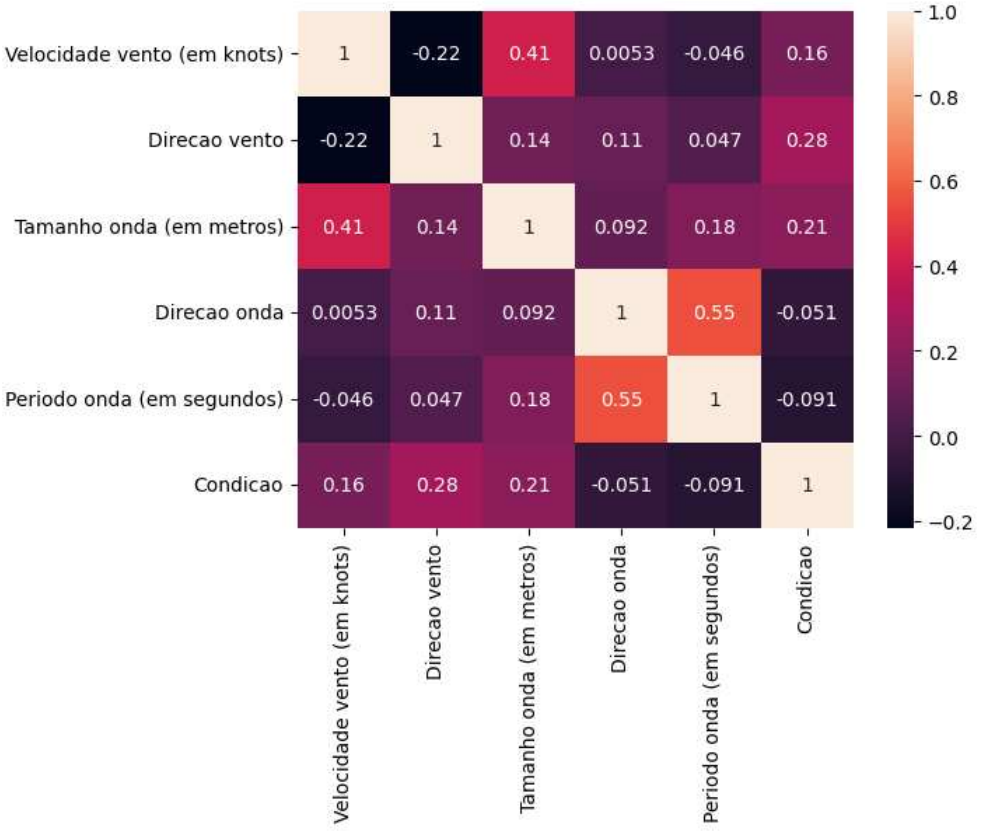

Figura 30 - Matriz de correlação entre as variáveis

5.5 Implementação dos Classificadores

Após realizada a coleta, transformação e análises descritivas e exploratórias dos dados, é o momento da implementação do modelo em si. O modelo tem o objetivo de classificar em categorias (ruim, regular ou boa) para a prática do surf através de variáveis que se referem às condições físicas e meteorológicas coletadas.

Foram realizados testes com quatro classificadores, sendo eles: SVM, Árvores de Decisão, KNN e RF (Random Forest). Assim como James, Zhang e O'Donncha (2018) e Cai *et al.* (2019), trabalhos relacionados citados neste estudo, foi utilizado o classificador SVM para construir o modelo de predição. O SVM é um método de aprendizado supervisionado para a classificação de dados lineares e não lineares, conforme explicado na fundamentação teórica deste trabalho.

Também foram utilizadas as Árvores de Decisão para realizar os testes de predição. Conforme explicado anteriormente, as Árvores de Decisão são árvores que classificam as instâncias ordenando com base em valores de recursos, onde cada nó em uma árvore representa um dado a ser classificado, e cada ramificação representa um valor que o nó pode presumir. Outro classificador testado foi o Random Forest que, de acordo com Breiman (2001), é uma combinação de preditores de árvores de modo que cada árvore depende dos valores de um vetor aleatório amostrado independentemente e com a mesma distribuição para todas as árvores da floresta.

Por fim, foram realizados testes com o classificador KNN que, conforme apresentado anteriormente, se baseia no princípio que instâncias em um *dataset* geralmente existirão próximas a outras instâncias que têm propriedades similares. Se as instâncias estiverem categorizadas, então a categoria de uma instância não classificada pode ser determinada observando a categoria de seus vizinhos mais próximos.

Para o treinamento e teste foi utilizada a PyCaret (Figura 31), que é uma biblioteca de aprendizado de máquina em Python que permite ir desde a preparação dos dados até a implantação do modelo, facilitando este processo. Os dados, que possuem 293 entradas, foram divididos aleatoriamente em conjuntos de treino, com 205 entradas, e conjuntos de teste, com 88 entradas.

	Description	Value
0	Session id	2112
1	Target	Condicao
2	Target type	Multiclass
3	Target mapping	boa: 0, regular: 1, ruim: 2
4	Original data shape	(293, 6)
5	Transformed data shape	(293, 16)
6	Transformed train set shape	(205, 16)
7	Transformed test set shape	(88, 16)
8	Numeric features	3
9	Categorical features	2
10	Preprocess	True
11	Imputation type	simple
12	Numeric imputation	mean
13	Categorical imputation	mode
14	Maximum one-hot encoding	25
15	Encoding method	None
16	Fold Generator	StratifiedKFold
17	Fold Number	10
18	CPU Jobs	-1
19	Use GPU	False
20	Log Experiment	False
21	Experiment Name	clf-default-name
22	USI	bb63

Figura 31 - Preparação dos dados para o treinamento pelo PyCaret

O processo utilizado para o teste dos classificadores envolve as etapas de inicializar a biblioteca PyCaret com os dados obtidos, realizar o treinamento do modelo, otimizar os hiperparâmetros utilizados pela biblioteca, visualizar e analisar métricas do treinamento, testar o modelo e verificar métricas com o modelo finalizado. Os hiperparâmetros do modelo são ajustados pela biblioteca PyCaret, que realiza um processo automático, executando diversas iterações onde são feitos testes, comparações, e, finalmente, os melhores são escolhidos. Pode-se verificar o fluxograma destas etapas na Figura 32.

Os resultados obtidos a partir da implementação deste processo são detalhados na seção a seguir onde, ao fim, um modelo é escolhido e suas métricas são apresentadas.

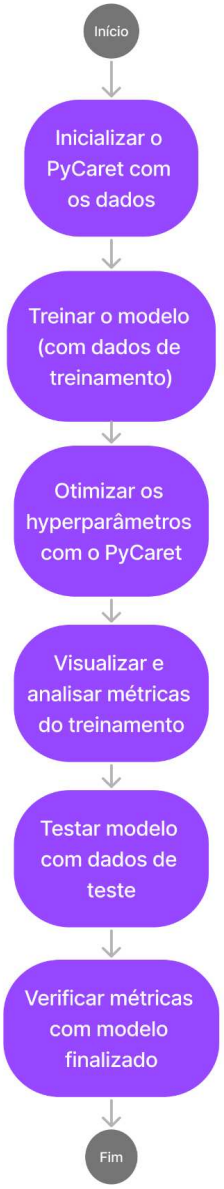

Figura 32 - Fluxograma do processo de treinamento e teste dos classificadores

5.6 Análise dos Resultados

Foram escolhidas as seguintes métricas a fim de medir a performance dos modelos desenvolvidos: acurácia, precisão, recall e f1-score. Essas métricas auxiliam na avaliação e para realizar a comparação entre si dos modelos, com o objetivo de escolher o que apresentasse o melhor resultado.

A acurácia é definida como a fração das predições corretas, ou seja, de todas as classificações, quais o modelo classificou corretamente. A precisão mede quão assertiva foram os verdadeiros positivos, isto é, de todas as praias que o modelo classificou como

boas, quantas realmente apresentavam aquela condição. O recall mede, de todas as praias que possuem boa como condição, quantas estão corretas e, por último, o f1-score é a média harmônica entre precisão e recall (JÚLIA, 2019).

Durante os primeiros treinamentos e testes dos modelos foi percebido um grande underfitting em todos os modelos. A acurácia da classificação estava abaixo do esperado, aproximadamente 59% para o melhor caso, que foi com o classificador KNN, como pode ser visto na Tabela 4. O KNN obteve também uma precisão de 0.56, recall de 0.59 e f1-score de 0.56. Já o segundo melhor resultado foi obtido através das Árvores de Decisão, com uma acurácia de 0.53, precisão de 0.57, recall de 0.53 e f1-score 0.54. Por último, o SVM com acurácia, precisão e recall de 0.50 e f1-score de 0.46.

Tabela 4 - Métricas do primeiro teste com três classes (ruim, regular e boa)

Classificador	**Acurácia**	**Precisão**	**Recall**	**F1-Score**
Árvores de Decisão	0.53	0.57	0.53	0.54
KNN	0.59	0.56	0.59	0.56
SVM	0.50	0.50	0.50	0.46
RF	0.66	0.65	0.66	0.65

Analisando a Figura 33, pode-se perceber a matriz de confusão das Árvores de Decisão, onde 0 é o valor dado para a classificação ‘boa’, 1 para a classificação ‘regular’ e 2 para a classificação ‘ruim’. Apenas em 2 dos 15 casos o modelo classificou corretamente as praias com classificação boa, um resultado muito abaixo do esperado. Para a classificação regular, o modelo acertou 27 dos 43 casos e, para a classificação ruim, o modelo conseguiu acertar 18 dos 30 casos.

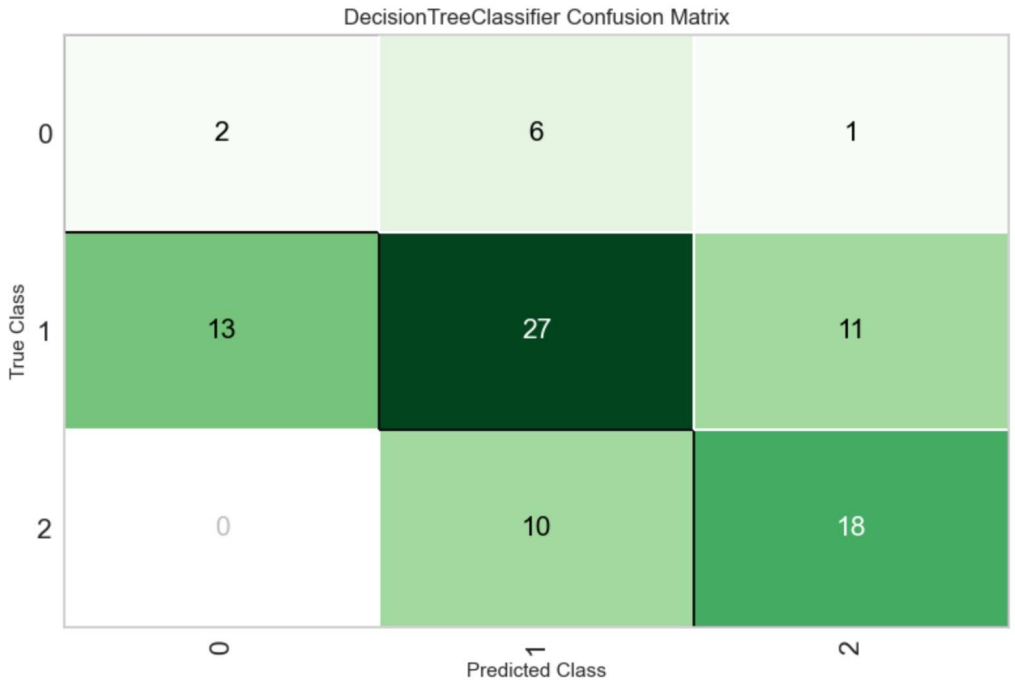

Figura 33 - Matriz de confusão das Árvores de Decisão do primeiro teste, com três classes

Os resultados obtidos no primeiro teste não foram satisfatórios e, a fim de aprimorar a acurácia do modelo, a classificação final foi reduzida de uma classificação de três classes, ruim, regular ou boa, para uma classificação binária, onde as amostras regulares passaram a ser consideradas boas, resultando em classificações ruim ou boa. Outro ponto a favor desta transformação é que praias tanto com condições regulares quanto com condições boas são propícias para praticar o surf, portanto, é uma alteração que se justifica para o atual caso de uso. Após aplicar todo o processo com os dados ajustados para a classificação binária, as métricas melhoraram consideravelmente, como é possível verificar nas Tabelas 5 e 6.

Tabela 5 - Métricas obtidas com o conjunto de treinamento no segundo teste

Classificador	**Acurácia**	**Precisão**	**Recall**	**F1-Score**
Árvores de Decisão	0.82	0.82	0.63	0.68
KNN	0.74	0.58	0.23	0.32
SVM	0.77	0.77	0.36	0.46
RF	0.79	0.74	0.58	0.60

Tabela 6 - Métricas obtidas com o conjunto de teste no segundo teste

Classificador	**Acurácia**	**Precisão**	**Recall**	**F1-Score**
Árvores de Decisão	0.75	0.67	0.43	0.52
KNN	0.70	0.62	0.18	0.28
SVM	0.83	0.93	0.50	0.65
RF	0.72	0.62	0.29	0.39

As métricas com o conjunto de teste foram satisfatórias, principalmente com o SVM, que apresentaram os melhores resultados com acurácia de 0.83, precisão de 0.93, recall de 0.50 e f1-score de 0.65. A Árvore de Decisão apresentou acurácia de 0.75, precisão de 0.67, recall de 0.43 e f1-score de 0.52. Enquanto que o KNN obteve acurácia de 0.70, precisão de 0.62, recall de 0.18 e f1-score de 0.28. E a RF obteve acurácia de 0.72, precisão de 0.62, recall de 0.29 e f1-score de 0.39.

Portanto, o modelo escolhido foi o SVM, que obteve o melhor desempenho durante os testes realizados, sem o overfitting observado na Árvore de Decisão e sem o underfitting observado no KNN. Observando a matriz de confusão na Figura 34, onde 0 é 'boa' e 1 é 'ruim', pode-se verificar que o modelo acertou 59 dos 73 casos de teste onde a classificação era 'boa', e acertou 14 dos 15 casos onde a classificação era 'ruim'.

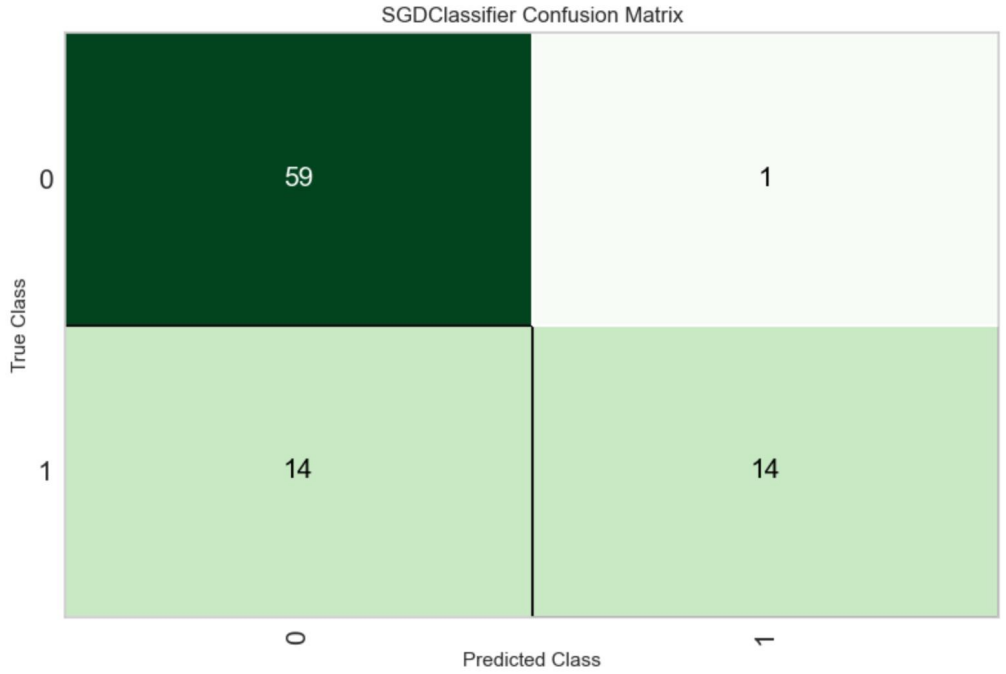

Figura 34 - Matriz de confusão do SVM no segundo teste, com duas classes

Na Figura 35, podemos verificar a importância de cada feature durante os testes realizados com o SVM. O modelo avaliou a direção do vento como a feature de maior importância, aparecendo diversas vezes na Figura 35, com o tamanho da onda como segunda feature de maior importância, e a direção da onda como a terceira feature de maior importância.

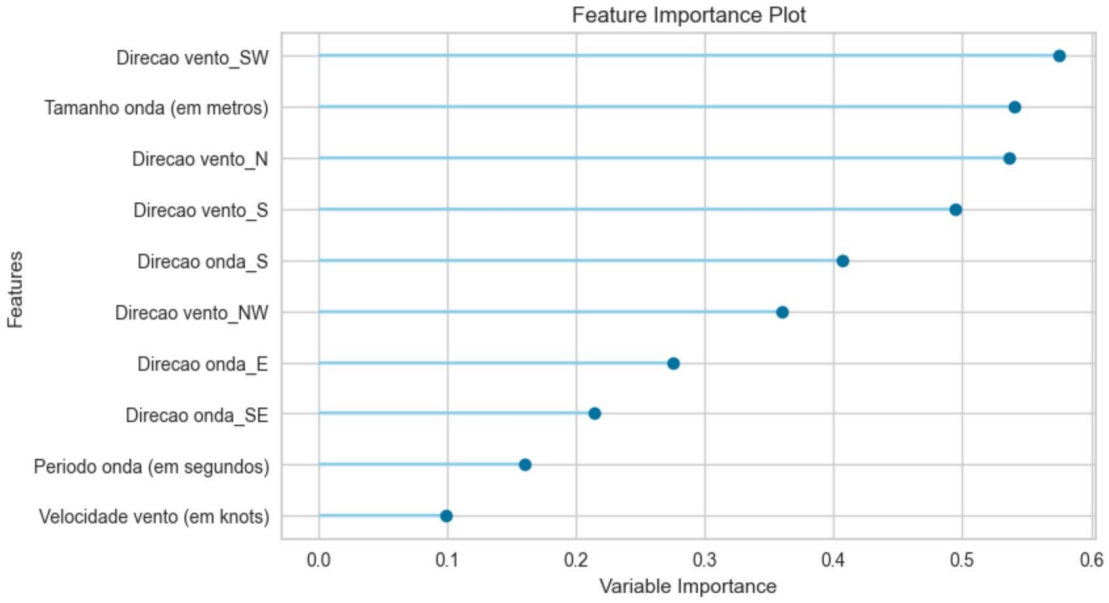

Figura 35 - Gráfico de importância de features com o classificador SVM

É possível considerar que, através dos resultados obtidos nos testes realizados, o SVM atingiu o objetivo geral deste estudo pois, conforme analisado nesta seção, apresentou resultados satisfatórios. O modelo escolhido alcançou os seguintes marcos no teste final com o conjunto de teste:

- Acurácia de 83%
- Precisão de 93%
- Recall de 50%
- F1 score de 65%

5.7 Comparação com Trabalhos Relacionados

Ao realizar uma comparação com os trabalhos relacionados, observando as informações apresentadas na Tabela 7, é possível perceber que apenas o primeiro trabalho, de James, Zhang e O'Donncha (2018), apresentou um modelo de classificação utilizado também neste trabalho, o SVM. Todos os trabalhos, com exceção do estudo de Dalinghaus (2016), fizeram uso de *machine learning*, assim como o atual trabalho.

Tabela 7 - Comparação dos tipos e modelos de IA utilizados dos trabalhos relacionados com o atual

Autor	**Utiliza IA**	**Tipo**	**Quais modelos**
James, Zhang e O'Donncha	Sim	Machine learning, supervisionada	Regressão e classificação. MLP e SVM
O'Donncha et al.	Sim	Machine learning	SWAN, Ridge Regression e Exponentiated Gradient
Pooja e Deo	Sim	Machine learning, supervisionada	RNA, GP e MT

Khosravi, Machado e Nunes	Sim	Machine learning, supervisionada e não supervisionada	MLFFNN, GMDH, SVR, Fuzzy inference system (FIS), ANFIS, ANFIS-PSO: Combinação do Particle swarm optimization (PSO) com o modelo ANFIS, ANFIS-GA: Interconexão entre algoritmo genérico (GA) e o modelo ANFIS
Dalinghaus	Não	N/A	N/A
Lai e Zambon (proposto)	Sim	Machine learning, supervisionada	SVM, árvores de decisão, KNN e RF

Na Tabela 8, são comparados os datasets dos trabalhos relacionados com o atual. Diferente dos trabalhos relacionados, o atual trabalho utilizou métodos de *web scraping* para realizar a coleta de seus dados, já que tornou-se necessário a construção de uma base de dados com as informações precisas para o estudo e, por esta razão, mostra que o trabalho atual possui uma base de dados relativamente menor em comparação aos outros trabalhos.

Tabela 8 - Comparação dos métodos e datasets dos trabalhos relacionados com o atual

Autor	**Utiliza Web Scraping**	**Tamanho do Dataset**
James, Zhang e O'Donncha	Não	Matriz com 11078 linhas e 741 colunas
O'Donncha et al.	Não	N/A
Pooja e Deo	Não	3 locais, 3 a 7 anos, com intervalo de 3 horas
Khosravi, Machado e Nunes	Não	N/A
Dalinghaus	Não	N/A
Lai e Zambon (proposto)	Sim	DataFrame com 314 linhas

Por fim, é realizada, na Tabela 9, a comparação dos domínios dos trabalhos relacionados com o atual. Nenhum dos trabalhos que utilizam inteligência artificial abordou os três domínios estudados no presente trabalho. Apenas o trabalho de Dalinghaus (2016) realizou o estudo envolvendo o surf, ventos e ondulações, porém, teve um foco maior na geologia e geomorfologia das praias e condições das marés.

Tabela 9 - Comparação dos domínios dos trabalhos relacionados com o atual

Autor	**Domínio aplicado no Surf**	**Domínio envolve ventos**	**Domínio envolve ondulações**
James, Zhang e O'Donncha	Não	Sim	Sim
O'Donncha et al.	Não	Sim	Sim
Pooja e Deo	Não	Não	Sim
Khosravi, Machado e Nunes	Não	Sim	Não
Dalinghaus	Sim	Sim	Sim
Lai e Zambon (proposto)	Sim	Sim	Sim

As métricas para a avaliação dos modelos deste trabalho não foram utilizadas nos modelos dos trabalhos correlatos, impossibilitando a comparação de desempenho entre eles. Nos estudos relacionados foram implementados modelos de regressão, tendo como dados de saída variáveis com valores numéricos e, no presente trabalho, foi implementado um modelo de classificação apresentando como resultado final classificações.

6. Considerações Finais

No decorrer deste trabalho, foram estudados conceitos, abordagens, estratégias e ferramentas para o processo de análise de dados. Além disso, foi construído também um modelo preditivo que busca prever condições das ondas para a prática do surf a partir de dados coletados de websites utilizando *web scraping*.

Sendo assim, é possível concluir que o trabalho conseguiu atingir o seu objetivo geral, assim como também os seus objetivos específicos, propostos anteriormente. O estado da arte referente a utilização de dados relacionados ao vento, onda e mar no contexto de inteligência artificial foi identificado e estudado nas análises realizadas dos trabalhos correlatos. Além disso, foi realizada a análise descritiva e exploratória dos dados coletados e dos resultados finais do modelo desenvolvido neste estudo, assim como a comparação com os trabalhos relacionados.

Algumas dificuldades foram encontradas ao longo do desenvolvimento deste trabalho. A falta de um dataset com os dados necessários para a aplicação de modelo tornou necessária a construção do mesmo. Através de técnicas de *web scraping* foi possível coletar dados referentes a um período maior do que de um ano, sendo eles de grande importância para conseguirmos realizar uma análise mais precisa do modelo proposto.

Outro ponto a se destacar foi a ausência de trabalhos relacionados que apresentassem modelos de classificação. Os trabalhos encontrados durante a pesquisa bibliográfica implementaram modelos de regressão e estudaram a respeito de estimar variáveis com valores numéricos, diferentemente do estudo atual que buscou realizar classificações.

Apesar de este trabalho abranger técnicas e estratégias variadas, muitas outras também podem ser utilizadas e avaliadas. Há também diversas outras oportunidades de trabalhos a serem desenvolvidos em relação ao aprimoramento do atual trabalho, sendo algumas delas:

- Realizar testes com uma base de dados mais aprimorada
- Realizar testes com outros classificadores
- Buscar outras variáveis que possam ter relação com a classificação das condições das ondas para a prática do surf

- Desenvolver uma aplicação que, de forma automática, realize a classificação das ondas diariamente

Referências

BRODIE, M. What Is Data Science?. Computer Science and Artificial Intelligence Laboratory, MIT, jun. 2019.

HADDAWAY, N. The Use of Web-scraping Software in Searching for Grey Literature. The Grey Journal, out. 2015.

HOW to Read a Surf Report. Kahalu'u Bay Surf & Sea, set. 2013. Disponível em: <https://learntosurfkona.com/featured/how-to-read-a-surf-report/>. Acesso em 16 set. 2021.

KERSTING, K. Machine Learning and Artificial Intelligence: Two Fellow Travelers on the Quest for Intelligent Behavior in Machines. Frontiers in Big Data, nov. 2018. Disponível em: <https://www.frontiersin.org/articles/10.3389/fdata.2018.00006/full>. Acesso em 16 set. 2021.

MCCARTHY, J. What Is Artificial Intelligence? Technical report, Stanford University, nov. 2007. Disponível em: <http://jmc.stanford.edu/artificial-intelligence/what-is-ai/index.html>. Acesso em 16 set. 2021.

MITCHELL, T. Machine Learning. New York, NY: McGraw-Hill, 1997. ISBN 0-07-042807-7.

RAMAGERI, B. Data Mining Techniques and Applications. Indian Journal of Computer Science and Engineering, nov. 2010

REINEMEN, D.; KOENIG. K.; STRONG-CVETICH, N.; KITTINGER, J. Conservation Opportunities Arise From the Co-Occurrence of Surfing and Key Biodiversity Areas. Frontiers in Marine Science, mar. 2021.

JAMES, Scott C.; ZHANG, Yushan; O'DONNCHA, Fearghal. A machine learning framework to forecast wave conditions, Coastal Engineering, Volume 137, 2018, Pages 1-10, ISSN 0378-3839.

DALINGHAUS, Charline. Análise da estabilidade da forma em planta e perfil nas praias da Barra da Lagoa, Moçambique e Ingleses, Florianópolis - SC: aplicações em análise de perigos costeiros. 2016. 200 p. Dissertação (Mestrado) - Universidade Federal de Santa Catarina, Centro de Filosofia e Ciências Humanas, Programa de Pós-Graduação em Geografia, Florianópolis, 2016.

O'DONNCHA, Fearghal; ZHANG, Yushan; CHEN, Bei; JAMES, Scott C. An integrated framework that combines machine learning and numerical models to improve wave-condition forecasts, Journal of Marine Systems, Volume 186, 2018, Pages 29-36, ISSN 0924-7963.

JAIN, Pooja; DEO, M.C. Artificial Intelligence Tools to Forecast Ocean waves in Real Time, The Open Ocean Engineering Journal, 2008, 1: 13-20.

KHOSRAVI, A.; MACHADO, L.; NUNES, R.O. Time-series prediction of wind speed using machine learning algorithms: A case study Osorio wind farm, Brazil. Applied Energy, Volume 224, 2018, Pages 550-566, ISSN 0306-2619.

SCARFE, B. E.; ELWANY, M. H. S.; MEAD, S. T.; BLACK, K. P. (2003). The Science of Surfing Waves and Surfing Breaks - A Review. UC San Diego: Scripps Institution of Oceanography.

STETLER, Larry; SAXTON, Keith. (1997). Analysis of Wind Data Used for Predicting Soil Erosion.

CAI, Yaping; GUAN, Kaiyu; LOBELL, David; POTGIETER, Andries B.; WANG, Shaowen; PENG, Jian; XU, Tianfang; ASSENG, Senthold; ZHANG, Yongguang; YOU, Liangzhi; PENG, Bin. Integrating satellite and climate data to predict wheat yield in

Australia using machine learning approaches. Agricultural and Forest Meteorology, Volume 274, 2019, Pages 144-159, ISSN 0168-1923.

BERTOTTI, Luciana; CAVALERI, Luigi. Wind and wave predictions in the Adriatic Sea. Journal of Marine Systems, Volume 78, Supplement, 2009, Pages S227-S234, ISSN 0924-7963.

MITCHELL, T. M. Machine Learning. [S.l.]: McGraw-Hill Science/Engineering/Math, 1997.

GRUS, J. Data Science from Scratch, First Principles With Python. 1005 Gravenstein Highway North, Sebastopol, CA 95472: O'Reilly Media, Inc., 2015.

KOTSIANTIS, S. Supervised machine learning: A review of classification techniques. Proceedings of the 2007 conference on Emerging Artificial Intelligence Applications in Computer Engineering: Real Word AI Systems with Applications in eHealth, HCI, Information Retrieval and Pervasive Technologies, p. 3–24, 2007.

KONAR, A. Artificial Intelligence and Soft Computing Behavioral and Cognitive, Modeling of the Human Brain. Boca Raton, Florida: CRC Press LLC, 1999.

KAROUSSI, E. Data Mining: K-Clustering Problem. Agder, Noruega. 2012.

KALRA, P. Text mining: Concepts, process and applications. Journal of Global Research in Computer Science, 2013.

HAN, J.; KAMBER, M.; PEI, J. Data Mining Concepts and Techniques. 225 Wyman Street, Waltham, MA 02451, USA: Morgan Kaufmann, 3rd edition, 2011.

MITCHELL, R. Web Scraping with Python. 1005 Gravenstein Highway North, Sebastopol, CA 95472, USA: O'Reilly Media, Inc., 2nd edition, 2018.

GENTLEMAN, R.; HUBER, W; CAREY, V. J. Supervised machine learning. Bioconductor case studies . New York, NY: Springer New York, 2008, p. 121–136.

GENTLEMAN, R.; CAREY, V. J. Unsupervised machine learning. Bioconductor case studies . New York, NY: Springer New York, 2008, p. 137–157.

KAELBLING, L.; LITTMAN, M.; MOORE, A. An Introduction to Reinforcement Learning. In: Steels, L. (eds) The Biology and Technology of Intelligent Autonomous Agents. NATO ASI Series, vol 144. Springer, Berlin, Heidelberg. 1995.

PEÑA, D.; LOURENÇO, A.; FERNÁNDEZ, H.; JATO, M.; RIVEROLA, F. Web scraping technologies in an API world, Briefings in Bioinformatics, Volume 15, Issue 5, September 2014, Pages 788–797.

NAKAYAMA, Júlia. Modelo de Detecção de Depressão através das Mídias Sociais. Orientador: Elder Rizzon. 2019. TCC (Graduação) - Universidade Federal de Santa Catarina. Centro Tecnológico. Sistemas de Informação. Disponível em: https://repositorio.ufsc.br/handle/123456789/202444

BREIMAN, Leo. Random Forests. Machine Learning 45, 5–32 (2001).

7. Anexos

7.1 Código-fonte

AwsLambdaWavesCsv

Function.cs

```
using Amazon;
using Amazon.Lambda.Core;
using Amazon.S3;
using Amazon.S3.Transfer;
using HtmlAgilityPack;
using System;
using System.Collections.Generic;
using System.IO;
using System.Linq;
using System.Net;
using System.Net.Http;
using System.Text.RegularExpressions;
using System.Threading.Tasks;

// Assembly attribute to enable the Lambda function's JSON input to be
converted into a .NET class.
[assembly:
LambdaSerializer(typeof(Amazon.Lambda.Serialization.SystemTextJson.DefaultLambd
aJsonSerializer))]

namespace AWSLambdaWavesCsv
{
   public class Function
   {
        private const string bucketName = "waves2";
        private string keyName = $"waves-{DateTime.UtcNow:yyyyMMdd}.csv";
        // Specify your bucket region (an example region is shown).
        private static readonly RegionEndpoint bucketRegion =
RegionEndpoint.USEast1;


        private static async Task<string> CallUrl(string fullUrl)
        {
            HttpClient client = new HttpClient();
            ServicePointManager.SecurityProtocol = SecurityProtocolType.Tls13;
```

```
            client.DefaultRequestHeaders.Accept.Clear();
            return await client.GetStringAsync(fullUrl);
        }
        /// <summary>
        /// A simple function that takes a string and does a ToUpper
        /// </summary>
        /// <param name="input"></param>
        /// <param name="context"></param>
        /// <returns></returns>
        public async Task<string> FunctionHandler(string input, ILambdaContext
context)
        {
            try
            {
                var url =
"https://www.waves.com.br/surf/ondas/condicao/santa-catarina/";
                var html = await CallUrl(url);

                HtmlDocument htmlDoc = new HtmlDocument();
                htmlDoc.LoadHtml(html);

                var allElementsWithClassSpotBar =
htmlDoc.DocumentNode.SelectNodes("//*[contains(@class,'spot_bar')]");

                var beachNames = allElementsWithClassSpotBar.Where(x =>
x.InnerHtml.Contains("<a href="));
                var conditions = allElementsWithClassSpotBar.Where(x =>
!x.InnerHtml.Contains("<a href="));


                var beaches = new List<Beach>();
                for (int i = 0; i < beachNames.Count(); i++)
                {
                    var beach = new Beach();

                    var beachEl = beachNames.ElementAtOrDefault(i);
                    if (beachEl == null)
                        throw new ApplicationException($"ERRO - Erro nos 'ndices
do array de beachNames");

                    var conditionEl = conditions.ElementAtOrDefault(i);
                    if (conditionEl == null)
                        throw new ApplicationException($"ERRO - Erro nos 'ndices
do array de conditions");

                    var condition = Regex.Match(conditionEl.InnerHtml,
@"ws_ruim|ws_regular|ws_boa|ws_offline").Captures.FirstOrDefault();
                    if (condition == null)
                        throw new ApplicationException($"ERRO - Array de
condi''es continha um elemento sem condi''o:\n${conditionEl.InnerHtml}");
```

```
                    // Caso seja offline, n'o coleta o dado
                    if (condition.Value == "ws_offline")
                        continue;

                    var size = Regex.Match(conditionEl.InnerText, @"[0-9]
m|([0-9]\.[0-9]+ m)").Captures.FirstOrDefault();
                    if (size == null)
                        throw new ApplicationException($"ERRO - Array de
condi''es continha um elemento sem tamanho:\n${conditionEl.InnerHtml}");


                    beach.Name = beachEl.InnerText.Trim();
                    beach.Size = size.Value.Replace("m", "").Trim();
                    beach.Condition = condition.Value;

                    beaches.Add(beach);
                }

                var csv = "Praia,Condicao,Tamanho (em metros)\n";
                foreach (var row in beaches.Select(x =>
$"{x.Name},{x.Condition},{x.Size}\n"))
                    csv += row;

                var stream = new MemoryStream();
                var writer = new StreamWriter(stream);
                writer.Write(csv);
                writer.Flush();
                stream.Position = 0;

                var s3Client = new AmazonS3Client("AWS-KEY-HERE",
"AWS-SECRET-HERE", new AmazonS3Config { RegionEndpoint = bucketRegion });
                var fileTransferUtility = new TransferUtility(s3Client);

                await fileTransferUtility.UploadAsync(stream, bucketName,
keyName);

                return "Sucesso!!!";
            }
            catch (Exception ex)
            {
                return $"Erro :(\nException {ex.GetType()}:
{ex.Message}\nStacktrace: {ex.StackTrace}";
            }
        }
    }
}
```

Beach.cs

```
using System;
using System.Collections.Generic;
using System.Text;

namespace AWSLambdaWavesCsv
{
   public class Beach
   {
        public String Name { get; set; }
        public String Condition { get; set; }
        public String Size { get; set; }
   }
}
```

AwsLambdaWavesHtml

Function.cs

```
using Amazon;
using Amazon.Lambda.Core;
using Amazon.S3;
using Amazon.S3.Transfer;
using HtmlAgilityPack;
using System;
using System.IO;
using System.Net;
using System.Net.Http;
using System.Threading.Tasks;

// Assembly attribute to enable the Lambda function's JSON input to be
converted into a .NET class.
[assembly:
LambdaSerializer(typeof(Amazon.Lambda.Serialization.SystemTextJson.DefaultLambd
aJsonSerializer))]

namespace AWSLambdaWavesHtml
{
   public class Function
   {
        private const string bucketName = "waves2";
        private string keyName = $"waves-{DateTime.UtcNow:yyyyMMdd}.html";
        // Specify your bucket region (an example region is shown).
        private static readonly RegionEndpoint bucketRegion =
RegionEndpoint.USEast1;
```

```
        private static async Task<string> CallUrl(string fullUrl)
        {
            HttpClient client = new HttpClient();
            ServicePointManager.SecurityProtocol = SecurityProtocolType.Tls13;
            client.DefaultRequestHeaders.Accept.Clear();
            return await client.GetStringAsync(fullUrl);
        }
        /// <summary>
        /// A simple function that takes a string and does a ToUpper
        /// </summary>
        /// <param name="input"></param>
        /// <param name="context"></param>
        /// <returns></returns>
        public async Task<string> FunctionHandler(string input, ILambdaContext
context)
        {
            try
            {
                var url =
"https://www.waves.com.br/surf/ondas/condicao/santa-catarina/";
                var html = await CallUrl(url);

                HtmlDocument htmlDoc = new HtmlDocument();
                htmlDoc.LoadHtml(html);

                var stream = new MemoryStream();
                var writer = new StreamWriter(stream);
                writer.Write(htmlDoc.DocumentNode.OuterHtml);
                writer.Flush();
                stream.Position = 0;

                var s3Client = new AmazonS3Client("AWS-KEY-HERE",
"AWS-SECRET-HERE", new AmazonS3Config { RegionEndpoint = bucketRegion });
                var fileTransferUtility = new TransferUtility(s3Client);

                await fileTransferUtility.UploadAsync(stream, bucketName,
keyName);

                return "Sucesso!!!";
            }
            catch (Exception ex)
            {
                return $"Erro :(\nException {ex.GetType()}:
{ex.Message}\nStacktrace: {ex.StackTrace}";
            }
        }
    }
}
```

WindguruCsv

Program.cs

```
using HtmlAgilityPack;
using System.Reflection;
using System.Collections;
using WindguruCsv;
using System.Text.RegularExpressions;

internal class Program
{

   public static string ParseAngle(HtmlNode node)
   {
       var el = node.FirstChild?.FirstChild;

       if (el == null)
       {
           return " - ";
       }

       var myRegex = new Regex("[0-9][0-9][0-9]");
       var elText = el.Attributes.First().Value;
       var angle = int.Parse(myRegex.Match(elText).Value) - 180;

       return GetAngle(angle);
   }

   /*
    * Interval used - 23 degrees
    *
       0/360 = N - 23
       45 = NE - 68
       90 = E - 113
       135 = SE - 158
       180 = S - 203
       225 = SW - 248
       270 = W - 293
       315 = NW - 338
    */
   public static string GetAngle(int angle)
   {
       string angleStr;

       if (angle < 23)
           angleStr = "N";
       else if (angle < 68)
           angleStr = "NE";
```

```
        else if (angle < 113)
            angleStr = "E";
        else if (angle < 158)
            angleStr = "SE";
        else if (angle < 203)
            angleStr = "S";
        else if (angle < 248)
            angleStr = "SW";
        else if (angle < 293)
            angleStr = "W";
        else if (angle < 338)
            angleStr = "NW";
        else
            angleStr = "N";

        return angleStr;

    }

    public static void Main(string[] args)
    {
        Console.WriteLine("Hello, World!");


        var path = @"windguru-2023-03-30.html";
        var htmlDoc = new HtmlDocument();
        htmlDoc.Load(path);

        var table = htmlDoc.DocumentNode

.SelectSingleNode("//*[@id=\"archive_results\"]/table/tbody/tr/td/table");

        var tbody = table.FirstChild;

        var trs = tbody.ChildNodes.Where(x => x.Name != "#text");
        var headers = trs.Take(2).ToList();
        var rows = trs.Skip(2).ToList();

        var infoRows = new List<InfoRow>();

        foreach (var row in rows)
        {
            var data = row.ChildNodes.Where(x => x.Name != "#text");

            var dateEl = data.First();
            var windSpeedEls = data.Skip(1).Take(24);
            var windDirectionEls = data.Skip(24 + 1).Take(24);
            var waveEls = data.Skip(48 + 1).Take(24);
            var waveDirectionEls = data.Skip(72 + 1).Take(24);
            var wavePeriodEls = data.Skip(96 + 1).Take(24);
```

```
            var infoRow = new InfoRow
            {
                Date = DateTime.Parse(dateEl.InnerText),
                WindAndWaveInfos = new List<WindAndWaveInfo>()
            };

            for (int i = 0; i < 24; i++)
            {
                var windAndWaveInfo = new WindAndWaveInfo
                {
                    Hour = i.ToString(),
                    WindSpeed = windSpeedEls.ElementAt(i).InnerText,
                    WindDirection = ParseAngle(windDirectionEls.ElementAt(i)),
                    Wave = waveEls.ElementAt(i).InnerText,
                    WaveDirection = ParseAngle(waveDirectionEls.ElementAt(i)),
                    WavePeriod = wavePeriodEls.ElementAt(i).InnerText,
                };

                infoRow.WindAndWaveInfos.Add(windAndWaveInfo);
            }


            infoRows.Add(infoRow);
        }


        var csv = "Data,Hora,Velocidade vento (em knots),Direcao vento,Tamanho
onda (em metros),Direcao onda,Periodo onda (em segundos)\n";
        foreach (var info in infoRows.Where(x => x.WindAndWaveInfos != null))
        {
            foreach (string row in info.WindAndWaveInfos.Select(x =>
$"{info.Date:yyyy/MM/dd},{x.Hour},{x.WindSpeed},{x.WindDirection},{x.Wave},{x.W
aveDirection},{x.WavePeriod}\n"))
            {
                csv += row;
            }
        }

        File.WriteAllText(@"C:\Users\rolfz\Documents\windguruCsv.csv", csv);
        Console.WriteLine(csv);
    }
}
```

InfoRow.cs

```
using System;
using System.Collections.Generic;
using System.Linq;
```

```
using System.Text;
using System.Threading.Tasks;

namespace WindguruCsv
{
   public class InfoRow
   {
       public DateTime Date { get; set; }
       public List<WindAndWaveInfo>? WindAndWaveInfos { get; set; }
   }
}
```

WindAndWaveInfo.cs

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace WindguruCsv
{
   public class WindAndWaveInfo
   {
       public string? Hour { get; set; }
       public string? WindSpeed { get; set; }
       public string? WindDirection { get; set; }
       public string? Wave { get; set; }
       public string? WaveDirection { get; set; }
       public string? WavePeriod { get; set; }
   }
}
```

WavesIA

notebook-python-ia.ipynb

```
import pandas as pd
```

```
import numpy as np
import seaborn as sns
import glob

windguru_data = pd.read_csv('data/windguruCsv.csv')
windguru_data = windguru_data[windguru_data["Hora"] == 7]
windguru_data["datetime"] = pd.to_datetime(windguru_data["Data"])


all_files = glob.glob("data/wavesCsv/*.csv")

li = []

for filename in all_files:
   year = filename[20:24]
   month = filename[24:26]
   day = filename[26:28]
   df = pd.read_csv(filename, index_col=None, header=0)
   df['Year'] = year
   df['Month'] = month
   df['Day'] = day
   df['Date'] = f'{year}/{month}/{day}'
   li.append(df)

frame = pd.concat(li, axis=0, ignore_index=True)
frame = frame.replace({
   'ws_ruim': 'ruim',
   'ws_regular': 'regular',
   'ws_boa': 'boa',
   })

mocambique_df = frame[frame["Praia"]=="Moçambique"]
mocambique_df = mocambique_df.sort_values(by='Date')
mocambique_df['datetime'] = pd.to_datetime(mocambique_df["Date"])

joaquina_count = len(frame[frame["Praia"]=="Joaquina (Câmera)"])
mocambique_meio_count = len(frame[frame["Praia"]=="Moçambique Meio"])
mocambique_count = len(frame[frame["Praia"]=="Moçambique"])

windguru_data

mocambique_df

frame["Praia"].value_counts().plot(kind='barh')

merged_data = pd.merge(mocambique_df, windguru_data, how='inner',
on='datetime')
df = pd.DataFrame(merged_data, columns=[
 'Velocidade vento (em knots)',
 'Direcao vento',
```

```
 'Tamanho onda (em metros)',
 'Direcao onda',
 'Periodo onda (em segundos)',
 'Condicao'
 ])

df = df.drop(
 df[
   (df["Direcao vento"] == " - ") |
   (df['Direcao onda'] == " - ") |
   (df['Periodo onda (em segundos)'] == ' - ')
 ].index
)

df.columns
df

df["Velocidade vento (em knots)"].value_counts().plot(kind='bar')

df["Direcao vento"].value_counts().plot(kind='bar')

df["Direcao onda"].value_counts().plot(kind='bar')

df["Periodo onda (em segundos)"].value_counts().plot(kind='bar')

df["Velocidade vento (em knots)"].value_counts().plot(kind='bar')

sns.displot(data=df, x="Condicao", col="Direcao vento")


sns.displot(data=df, x="Condicao", y="Direcao vento")
sns.displot(data=df, x="Condicao", y="Velocidade vento (em knots)")
sns.displot(data=df, x="Condicao", y="Direcao onda")
sns.displot(data=df, x="Condicao", y="Tamanho onda (em metros)")
sns.displot(data=df, x="Condicao", y="Periodo onda (em segundos)")

df["Condicao"].value_counts().plot(kind='bar')


df_copy = df.copy()
df_copy['Direcao vento']=df_copy['Direcao vento'].astype('category').cat.codes
df_copy['Direcao onda']=df_copy['Direcao onda'].astype('category').cat.codes
df_copy['Condicao']=df_copy['Condicao'].astype('category').cat.codes


corr_matrix = df_copy.corr()
sns.heatmap(corr_matrix, annot=True)
df_copy

from pycaret.classification import *
```

```
df = df.replace({
 'regular': 'boa',
})

s = setup(df,
          numeric_features=['Velocidade vento (em knots)', 'Tamanho onda (em
metros)', 'Periodo onda (em segundos)'],
          categorical_features=['Direcao vento', 'Direcao onda'],
          target = 'Condicao'
          )

print(s)


best = compare_models(include=['dt','svm','knn', 'rf'], n_select=4)
results = pull()
results

evaluate_model(best[0])
evaluate_model(best[1])
evaluate_model(best[2])
evaluate_model(best[3])

predictions = predict_model(best[1], data=df)
predictions

dt = best[0]
rf = best[1]
svm = best[2]
df2 = pd.DataFrame({
 'Velocidade vento (em knots)': [8],
 'Direcao vento': ['N'],
 'Tamanho onda (em metros)': [1.2],
 'Direcao onda': ['S'],
 'Periodo onda (em segundos)': [11]
})

predictions = predict_model(rf, data=df2)
predictions


s2 = setup(df,
          numeric_features=['Velocidade vento (em knots)', 'Tamanho onda (em
metros)', 'Periodo onda (em segundos)'],
          categorical_features=['Direcao vento', 'Direcao onda'],
          target = 'Condicao',
          session_id=123
          )
```

```
print(s2)
compare_models()


dt = create_model('dt')
knn = create_model('knn')
svm = create_model('svm')
rf = create_model('rf')


tuned_dt = tune_model(dt)
tuned_knn = tune_model(knn)
tuned_svm = tune_model(svm)
tuned_rf = tune_model(rf)


evaluate_model(tuned_dt)
evaluate_model(tuned_knn)
evaluate_model(tuned_svm)
evaluate_model(tuned_rf)


predict_model(tuned_dt)
predict_model(tuned_knn)
predict_model(tuned_svm)
predict_model(tuned_rf)
```

7.2 Artigo do TCC

Desenvolvimento de modelo de machine learning para previsão das condições de ondas para a prática do surf

Jonas Lai Barbosa[1], Rolf Zambon[1], Elder Rizzon Santos[1]

[1]Departamento de Informática e Estatística – Universidade Federal de Santa Catarina (UFSC)
Caixa Postal 476 – 88040-370 – Florianópolis – SC – Brazil

jonaslaib1@hotmail.com, led479@gmail.com, elder.santos@ufsc.br

Abstract. *Surfing is a sport that increasingly attracts more attention and curiosity from new people, however, for those who are starting in the sport, it can become difficult to identify the best waves conditions to practice it. As a result, this paper proposes the development and training of a machine learning model that, from data that will be collected on weather forecasting sites using web scraping process, will be able to predict these conditions through the use of classifiers such as SVM, KNN and decision trees. Finally, through experiments the SVM presented the best quality indicators, with an accuracy of 83% and a precision of 93%.*

Resumo. *O surf é um esporte que cada vez atrai mais a atenção e curiosidade de novas pessoas, porém, para quem está começando, pode se tornar um pouco difícil a experiência de identificar quais as melhores condições das ondas para praticá-lo. Em função disso, este trabalho propõe o desenvolvimento e o treinamento de um modelo de machine learning que, a partir de dados coletados em sites de previsões meteorológicas utilizando o processo de web scraping, conseguirá prever essas condições através da utilização de classificadores como por exemplo SVM, KNN e Árvores de Decisão. Por fim, através dos experimentos realizados, o SVM apresentou as melhores métricas de qualidade, com uma acurácia de 83% e uma precisão de 93%.*

1. Introdução

O surf é um esporte praticado há décadas por inúmeras pessoas ao redor do mundo, e possui uma grande importância social, econômica e ambiental. Como Reineman, Koenig, Strong-Cvetiche e Kittinger (2021) apresentam, cada vez mais há um reconhecimento por parte de comunidades, pesquisadores e praticantes no valor de surf breaks (praias onde há ondas para a prática do surf) locais, e particularmente em economias em desenvolvimento. Estes grandes benefícios socioeconômicos eram previamente subestimados e não eram levados em consideração nos processos de planejamento em desenvolvimentos costeiros. Reineman et al. (2021) também mencionam que, globalmente, o turismo relacionado ao surf é avaliado entre 31,5 e 64,9 bilhões de dólares, e que esses benefícios fornecem o mecanismo para acelerar o crescimento econômico nas comunidades ao redor de surf breaks.

Para um surfista iniciante/intermediário, pode ser difícil identificar as condições necessárias para a prática do esporte por causa da grande quantidade de variáveis

existentes na formação de ondas, sendo as principais delas (KAHALU'U BAY SURF & SEA, 2013):

- Período da ondulação: é o intervalo de tempo entre duas ondas, medido em segundos. Quanto maior o período, mais alta e mais forte a onda surfável nas praias, visto que cada onda carregará mais água;
- Altura da ondulação: é a altura da ondulação no oceano, normalmente medida em metros ou pés. Não necessariamente é o mesmo tamanho da onda surfavel nas praias, pois a altura da onda na praia depende também do período;
- Direção da ondulação: é a direção que a ondulação vem do oceano. Dependendo da praia, uma ondulação de uma certa direção pode não alcançar a praia, e não haverá ondas surfáveis nela, ou se a ondulação entrar em cheio na praia (perpendicular à faixa de areia), as ondas estarão com força total. É uma variável que depende totalmente da praia em análise;
- Velocidade e direção do vento: O ideal para a prática do surf é menos vento possível. Mas também é possível com vento fraco a moderado, desde que a direção não seja do oceano para a praia (vento maral).

O presente trabalho propõe a criação de um modelo de *machine learning* que tem como saída a condição das ondas para a prática do surf. A criação do modelo será possibilitada pela utilização das técnicas de *web scraping*, onde será realizada a coleta das variáveis previamente mencionadas, além da variável alvo, que é a condição das ondas, para realizar o treinamento através da técnica de *machine learning*. Ao fim, será feito um experimento onde o modelo em questão será posto em teste, e os resultados obtidos serão analisados.

2. Fundamentação Teórica

A fim de analisar e entender melhor essa grande variedade e quantidade de dados, a ciência de dados é uma abordagem poderosa que auxilia nesse processo. Segundo Brodie (2019), a ciência de dados tem como objetivo realizar a análise de grandes quantidades de dados para extrair correlações com estimativas de probabilidade e erro. A ciência de dados também abrange várias outras disciplinas, como *data mining* e *machine learning*.

Para Ramageri (2010), *data mining* é o processo lógico usado para explorar grandes quantidades de dados a fim de encontrar dados úteis. De acordo com Haddaway (2015), *data scraping* é o termo usado para descrever a extração de dados de um arquivo eletrônico utilizando um programa de computador. *Web scraping* descreve o uso de um programa para extrair dados de arquivos HTML na internet. Normalmente, esses dados são padronizados, principalmente em listas ou tabelas.

Inteligência artificial e *machine learning* estão fortemente relacionadas, porém, essas tecnologias são diferentes em várias maneiras. Em 2007, McCarthy (2007) se referiu à inteligência artificial como a ciência e a engenharia de fazer máquinas inteligentes, especialmente programas de computador inteligentes. Está relacionada à tarefa semelhante de usar computadores para entender a inteligência humana, mas a IA não precisa se limitar a métodos que são biologicamente observáveis. Kersting (2018) acredita que o comportamento de uma máquina não é somente a saída de um programa, ele também é afetado pelo seu “corpo” e o ambiente que fisicamente ele está presente.

Para mantê-lo simples, no entanto, se você puder escrever um programa muito inteligente que tenha comportamento semelhante ao humano, ele pode ser IA. Mas, a menos que aprenda automaticamente com os dados, não é *machine learning*:

> *Machine learning é a ciência que "tem como objetivo a questão de como construir programas de computador que melhoram automaticamente com a experiência," (MITCHELL, 1997)*

3. Desenvolvimento

O objetivo deste trabalho, conforme descrito no Capítulo 2, é criar um modelo de *machine learning* que consiga avaliar a condição da praia para a prática do surf, considerando-se os dados de entrada para o treinamento e construção do modelo, que são:

- Período da ondulação
- Altura da ondulação
- Direção da ondulação
- Velocidade do vento
- Direção do vento
- Localização (praias de Florianópolis)

Como explicado anteriormente na seção de introdução do atual trabalho, essas são as principais variáveis que influenciam na classificação da condição de uma praia para a prática do esporte. Além disso, esses dados são de fácil acesso, com dados históricos auxiliando na construção da base de dados, no site Windguru, enquanto que os dados resultantes são encontrados no site Waves, que serão apresentados nas seções seguintes.

Para cumprir esse objetivo, o desenvolvimento do modelo proposto segue um fluxo composto por quatro etapas, onde é realizada a coleta dos dados, a preparação dos dados, o treinamento do modelo, e o teste/análise. Pode-se verificar o fluxograma destas etapas na Figura 1.

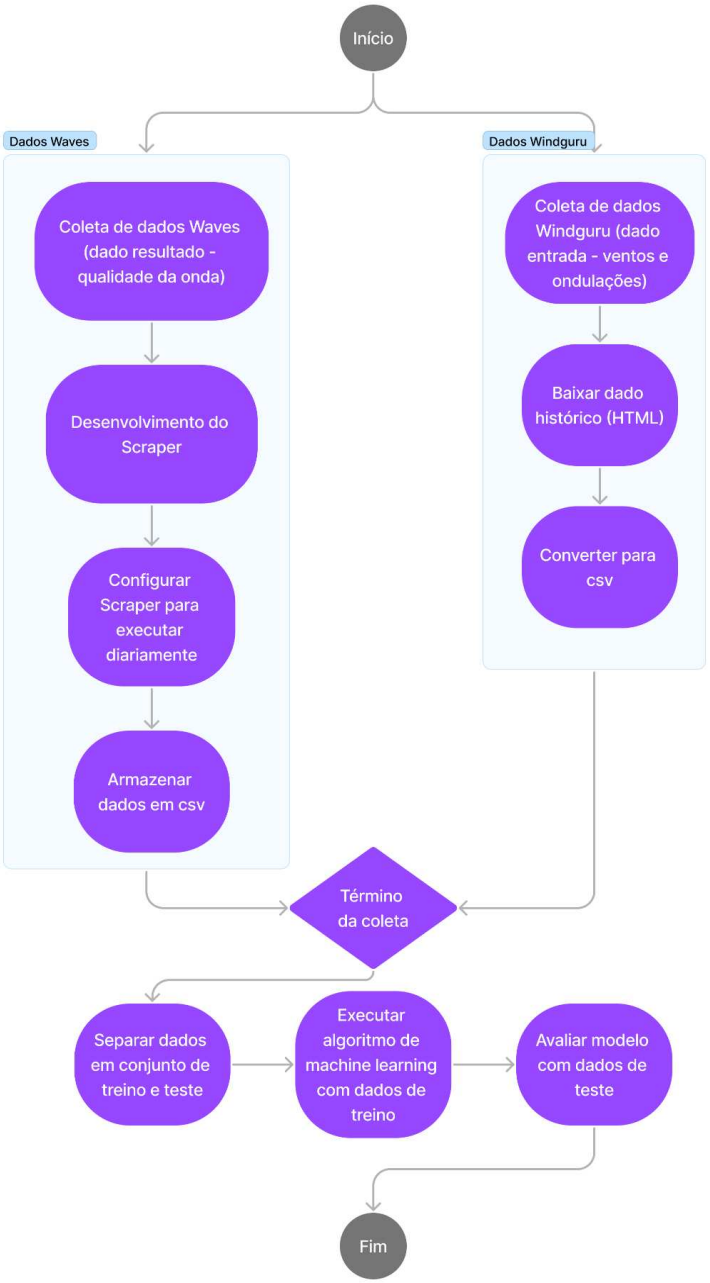

Figura 1 - Fluxograma do desenvolvimento da solução

3.1 Ferramentas

Para o desenvolvimento do *Scraper* foi utilizado o C# (C Sharp), que é uma linguagem de programação orientada a objetos criada pela Microsoft e que faz parte da sua plataforma .Net. A Microsoft baseou o C# na linguagem C++ e Java e é utilizada em diversos tipos de aplicações.

Para a configuração do *Scraper*, foi utilizada a AWS (Amazon Web Services) que é um serviço de computação em nuvem desenvolvido pela Amazon. Ela oferece mais de 200 serviços completos de data centers por todo o mundo e, por este motivo, acaba trazendo mais recursos do que outros provedores de nuvem.

Para o treinamento e teste do modelo, foi utilizado o Python, que é uma linguagem de programação de alto nível, com tipagem dinâmica e forte, multiplataforma e orientada a objetos, uma forma específica de organizar softwares onde, os procedimentos estão submetidos às classes, o que possibilita maior controle e estabilidade de códigos para projetos de grandes proporções.

3.2 Coleta e Preparação de Dados

Na etapa de coleta de dados, duas fontes são utilizadas, uma para os dados de entrada, Windguru, e uma para os dados de saída, Waves.

Windguru[3] é uma plataforma que descreve e detalha informações sobre o vento, clima e ondulações de uma determinada região, havendo previsões para até 7 dias. Há múltiplas métricas sobre o vento, clima e ondulações. Para este trabalho será utilizado um subconjunto destas métricas. No Windguru há dados históricos que são atualizados diariamente e com intervalos de 2 horas. Como os dados do Waves são atualizados todos os dias entre às 7 e 9 horas da manhã, serão utilizados os dados do Windguru das 8 horas da manhã. Será necessário baixar o histórico em HTML, e desenvolver um conversor para CSV. O conversor será implementado utilizando .Net Core com C#.

Waves[4] é uma plataforma que tem como objetivo informar os praticantes do surf sobre a condição das ondas no dia atual. Na plataforma é possível escolher o estado desejado, e visualizar praia a praia a altura das ondas, e a condição, podendo ser ruim, regular ou boa. A plataforma é atualizada diariamente, havendo pessoas que inserem a informação para cada praia individualmente, classificando a altura e a qualidade da onda. No Waves, não há dados históricos, os dados são atualizados diariamente e, por esse motivo, tornou-se necessário o desenvolvimento de um *Scraper*, que acessa o site, baixa, formata e salva os dados.

O *Scraper* foi implementado utilizando a linguagem .Net Core 3.1 com C#, publicado na AWS Lambda, e os dados coletados são armazenados no AWS S3. Também foi necessário fazer a configuração para o *Scraper* ser executado diariamente, esta que foi realizada na própria AWS, no serviço EventBridge. Esta etapa da coleta foi feita o mais rápido possível, para obter a maior quantidade de dados, e aprimorar a precisão do modelo. Foram publicados dois scripts escritos em C# .Net Core 3.1 na Lambda AWS, um para acessar a plataforma Waves e salvar o HTML em um bucket no S3, e outro para acessar a plataforma Waves e salvar o CSV em um bucket no S3. Para programar a execução da função Lambda diariamente, foi criada uma regra no EventBridge AWS, que, todos os dias às 20:15 UTC, dispara ambas as funções Lambda (WavesHTML e WavesCSV).

3.3 Análise Descritiva

A fim de realizar as análises deste trabalho, tornou-se necessário organizar as informações mais importantes para determinar a condição das ondas para o surf. Assim, as análises utilizaram os dados físicos coletados diariamente às 7 horas da manhã, horário em que usualmente são lançados os boletins das ondas para os surfistas se

[3] https://www.windguru.cz
[4] https://www.waves.com.br

informarem sobre as melhores praias do dia para praticar o esporte. Os seguintes dados de entrada foram armazenados em um *DataFrame*, utilizando a linguagem de programação Python, para melhor manipulação dos mesmos:

- Velocidade do vento: velocidade do vento em knots (unidade de medida de velocidade equivalente a uma milha náutica por hora, ou seja, 1,852 km/h)
- Direção do vento: direção do vento, podendo ser os 8 pontos cardeais (N - NE - E - SE - S - SW - W - NW)
- Tamanho da onda: tamanho da onda medida em metros
- Direção da onda: direção da onda, podendo ser os 8 pontos cardeais
- Período da onda: tempo necessário para a formação de um comprimento de onda, medido em segundos
- Localização: nome da praia na qual foram coletados os dados

Foi adicionado também o dado de saída para que fosse possível treinar o modelo de *machine learning* e posteriormente realizar as previsões das condições da onda sem qualquer tipo de supervisão humana.

- Condição da onda: condição da onda para a prática do surf, podendo ser: ruim, regular ou boa

3.4 Análise Exploratória

A análise exploratória é a etapa da estatística na qual são gerados novos conhecimentos através da análise de gráficos e matrizes gerados a partir dos dados coletados no experimento. Para realizar essa fase de análise, foram utilizados gráficos que buscam relacionar a quantidade total de ocorrências de um dado, onde quanto mais forte a cor maior a quantidade total, com o valor final da condição da onda.

Na Figura 2, podemos observar a quantidade total das ocorrências de cada uma das classificações das condições da onda. Dos dados coletados, a grande maioria foram de ondas classificadas como regulares, em seguida, ondas com a condição ruim e, por último com menos ocorrências, ondas boas, totalizando 293 entradas.

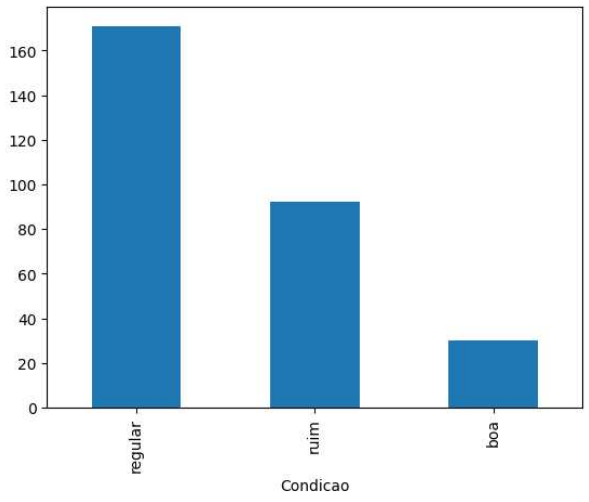

Figura 2 - Gráfico da quantidade de ocorrências da classificação das ondas

É também importante observarmos a correlação entre as variáveis. O coeficiente de correlação é uma medida estatística que mede a associação entre variáveis através de valores entre -1 e 1. Quanto mais próximo de 1, podemos definir que as variáveis possuem uma forte correlação e nota-se um aumento no valor de uma variável quando a outra também aumenta. Quanto mais próximo de -1, as variáveis também possuem uma forte correlação, entretanto, uma variável tende a diminuir enquanto que o valor da outra aumenta. Um coeficiente de correlação próximo de zero indica que há uma baixa correlação entre as duas variáveis. A Figura 3 é uma matriz de correlação entre as variáveis utilizadas no presente trabalho, indicando que poucas delas possuem uma correlação direta. Como podemos visualizar, a direção e o período da onda são as variáveis mais correlacionadas, enquanto que a direção da onda e a velocidade do vento foram as variáveis com o menor índice de correlação.

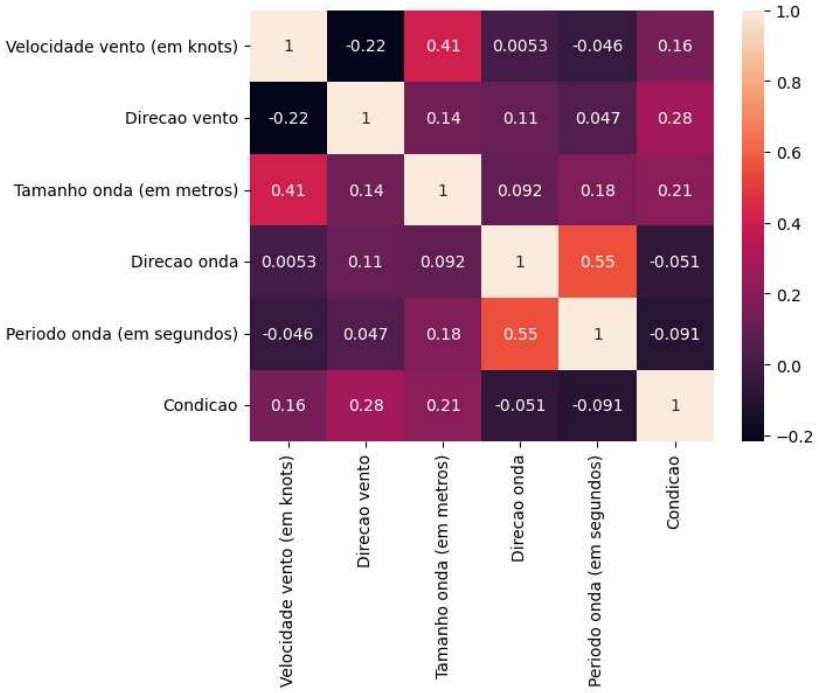

Figura 3 - Matriz de correlação entre as variáveis

3.5 Implementação dos Classificadores

Após realizada a coleta, transformação e análises descritivas e exploratórias dos dados, é o momento da implementação do modelo em si. O modelo tem o objetivo de classificar em categorias (ruim, regular ou boa) para a prática do surf através de variáveis que se referem às condições físicas e meteorológicas coletadas.

Foram realizados testes com quatro classificadores, sendo eles: SVM (*Support Vector Machine*), Árvores de Decisão, KNN (*K-Nearest Neighbor*) e RF (Random Forest). O SVM é um método de aprendizado supervisionado para a classificação de dados lineares e não lineares (HAN; KAMBER; PEI, 2011). Também foram utilizadas as Árvores de Decisão para realizar os testes de predição. As Árvores de Decisão são

árvores que classificam as instâncias ordenando com base em valores de recursos, onde cada nó em uma árvore representa um dado a ser classificado, e cada ramificação representa um valor que o nó pode presumir (KOTSIANTIS, 2007).

Outro classificador testado foi o Random Forest que, de acordo com Breiman (2001), é uma combinação de preditores de árvores de modo que cada árvore depende dos valores de um vetor aleatório amostrado independentemente e com a mesma distribuição para todas as árvores da floresta. Por fim, foram realizados testes com o classificador KNN que se baseia no princípio que instâncias em um dataset geralmente existirão próximas a outras instâncias que têm propriedades similares. Se as instâncias estiverem categorizadas, então a categoria de uma instância não classificada pode ser determinada observando a categoria de seus vizinhos mais próximos (KOTSIANTIS, 2007).

Para o treinamento e teste foi utilizada a PyCaret, que é uma biblioteca de aprendizado de máquina em Python que permite ir desde a preparação dos dados até a implantação do modelo, facilitando este processo. Os dados, que possuem 293 entradas, foram divididos aleatoriamente em conjuntos de treino, com 205 entradas, e conjuntos de teste, com 88 entradas.

3. Análise dos Resultados

Foram escolhidas as seguintes métricas a fim de medir a performance dos modelos desenvolvidos: acurácia, precisão, recall e f1-score. Essas métricas auxiliam na avaliação e para realizar a comparação entre si dos modelos, com o objetivo de escolher o que apresentasse o melhor resultado.

A acurácia é definida como a fração das predições corretas, ou seja, de todas as classificações, quais o modelo classificou corretamente. A precisão mede quão assertiva foram os verdadeiros positivos, isto é, de todas as praias que o modelo classificou como boas, quantas realmente apresentavam aquela condição. O recall mede, de todas as praias que possuem boa como condição, quantas estão corretas e, por último, o f1-score é a média harmônica entre precisão e recall (JÚLIA, 2019).

Durante os primeiros treinamentos e testes dos modelos foi percebido um grande underfitting em todos os modelos. A acurácia da classificação estava abaixo do esperado, aproximadamente 59% para o melhor caso, que foi com o classificador KNN, como pode ser visto na Tabela 1. O KNN obteve também uma precisão de 0.56, recall de 0.59 e f1-score de 0.56. Já o segundo melhor resultado foi obtido através das Árvores de Decisão, com uma acurácia de 0.53, precisão de 0.57, recall de 0.53 e f1-score 0.54. Por último, o SVM com acurácia, precisão e recall de 0.50 e f1-score de 0.46.

Tabela 1 - Métricas do primeiro teste com três classes (ruim, regular e boa)

Classificador	**Acurácia**	**Precisão**	**Recall**	**F1-Score**
Árvores de Decisão	0.53	0.57	0.53	0.54
KNN	0.59	0.56	0.59	0.56
SVM	0.50	0.50	0.50	0.46
RF	0.66	0.65	0.66	0.65

Os resultados obtidos no primeiro teste não foram satisfatórios e, a fim de aprimorar a acurácia do modelo, a classificação final foi reduzida de uma classificação de três classes, ruim, regular ou boa, para uma classificação binária, onde as amostras regulares passaram a ser consideradas boas, resultando em classificações ruim ou boa. Outro ponto a favor desta transformação é que praias tanto com condições regulares quanto com condições boas são propícias para praticar o surf, portanto, é uma alteração que se justifica para o atual caso de uso. Após aplicar todo o processo com os dados ajustados para a classificação binária, as métricas melhoraram consideravelmente, como é possível verificar nas Tabelas 2 e 3.

Tabela 2 - Métricas obtidas com o conjunto de treinamento no segundo teste

Classificador	**Acurácia**	**Precisão**	**Recall**	**F1-Score**
Árvores de Decisão	0.82	0.82	0.63	0.68
KNN	0.74	0.58	0.23	0.32
SVM	0.77	0.77	0.36	0.46
RF	0.79	0.74	0.58	0.60

Tabela 3 - Métricas obtidas com o conjunto de teste no segundo teste

Classificador	Acurácia	Precisão	Recall	F1-Score
Árvores de Decisão	0.75	0.67	0.43	0.52
KNN	0.70	0.62	0.18	0.28
SVM	0.83	0.93	0.50	0.65
RF	0.72	0.62	0.29	0.39

As métricas com o conjunto de teste foram satisfatórias, principalmente com o SVM, que apresentaram os melhores resultados com acurácia de 0.83, precisão de 0.93, recall de 0.50 e f1-score de 0.65. A Árvore de Decisão apresentou acurácia de 0.75, precisão de 0.67, recall de 0.43 e f1-score de 0.52. Enquanto que o KNN obteve acurácia de 0.70, precisão de 0.62, recall de 0.18 e f1-score de 0.28. E a RF obteve acurácia de 0.72, precisão de 0.62, recall de 0.29 e f1-score de 0.39.

É possível considerar que, através dos resultados obtidos nos testes realizados, o SVM atingiu o objetivo geral deste estudo pois, conforme analisado nesta seção, apresentou resultados satisfatórios. O modelo escolhido alcançou os seguintes marcos no teste final com o conjunto de teste:

- Acurácia de 83%
- Precisão de 93%
- Recall de 50%
- F1 score de 65%

4. Considerações Finais

No decorrer deste trabalho, foram estudados conceitos, abordagens, estratégias e ferramentas para o processo de análise de dados. Além disso, foi construído também um modelo preditivo que busca prever condições das ondas para a prática do surf a partir de dados coletados de websites utilizando *web scraping*.

Algumas dificuldades foram encontradas ao longo do desenvolvimento deste trabalho. A falta de um dataset com os dados necessários para a aplicação de modelo tornou necessária a construção do mesmo. Através de técnicas de *web scraping* foi possível coletar dados referentes a um período maior do que de um ano, sendo eles de grande importância para conseguirmos realizar uma análise mais precisa do modelo proposto.

Apesar de este trabalho abranger técnicas e estratégias variadas, muitas outras também podem ser utilizadas e avaliadas. Há também diversas outras oportunidades de trabalhos a serem desenvolvidos em relação ao aprimoramento do atual trabalho, sendo algumas delas:

- Realizar testes com uma base de dados mais aprimorada
- Realizar testes com outros classificadores
- Buscar outras variáveis que possam ter relação com a classificação das condições das ondas para a prática do surf
- Desenvolver uma aplicação que, de forma automática, realize a classificação das ondas diariamente

Referências

REINEMEN, D.; KOENIG. K.; STRONG-CVETICH, N.; KITTINGER, J. Conservation Opportunities Arise From the Co-Occurrence of Surfing and Key Biodiversity Areas. Frontiers in Marine Science, mar. 2021.

HOW to Read a Surf Report. Kahalu'u Bay Surf & Sea, set. 2013. Disponível em: <https://learntosurfkona.com/featured/how-to-read-a-surf-report/>. Acesso em 16 set. 2021.

BRODIE, M. What Is Data Science?. Computer Science and Artificial Intelligence Laboratory, MIT, jun. 2019.

RAMAGERI, B. Data Mining Techniques and Applications. Indian Journal of Computer Science and Engineering, nov. 2010

HADDAWAY, N. The Use of Web-scraping Software in Searching for Grey Literature. The Grey Journal, out. 2015.

MCCARTHY, J. What Is Artificial Intelligence? Technical report, Stanford University, nov. 2007. Disponível em: <http://jmc.stanford.edu/artificial-intelligence/what-is-ai/index.html>. Acesso em 16 set. 2021.

KERSTING, K. Machine Learning and Artificial Intelligence: Two Fellow Travelers on the Quest for Intelligent Behavior in Machines. Frontiers in Big Data, nov. 2018. Disponível em: <https://www.frontiersin.org/articles/10.3389/fdata.2018.00006/full>. Acesso em 16 set. 2021.

MITCHELL, T. Machine Learning. New York, NY: McGraw-Hill, 1997. ISBN 0-07-042807-7.

HAN, J.; KAMBER, M.; PEI, J. Data Mining Concepts and Techniques. 225 Wyman Street, Waltham, MA 02451, USA: Morgan Kaufmann, 3rd edition, 2011.

KOTSIANTIS, S. Supervised machine learning: A review of classification techniques. Proceedings of the 2007 conference on Emerging Artificial Intelligence Applications in Computer Engineering: Real Word AI Systems with Applications in eHealth, HCI, Information Retrieval and Pervasive Technologies, p. 3–24, 2007.

NAKAYAMA, Júlia. Modelo de Detecção de Depressão através das Mídias Sociais. Orientador: Elder Rizzon. 2019. TCC (Graduação) - Universidade Federal de Santa Catarina. Centro Tecnológico. Sistemas de Informação. Disponível em: https://repositorio.ufsc.br/handle/123456789/202444

BREIMAN, Leo. Random Forests. Machine Learning 45, 5–32 (2001).