INSTITUTO POLITÉCNICO
DE VIANA DO CASTELO
ESTG

Abordagens de *Machine Learning* para previsão de curto prazo de níveis de insulina ativa e de glicose no sangue

Abordagens de *Machine Learning* para previsão de curto prazo de níveis de insulina ativa e de glicose no sangue

Daniel Alfredo Francisco

Escola Superior de Tecnológica e Gestão

2024

INSTITUTO POLITÉCNICO
DE VIANA DO CASTELO

Instituto Politécnico de Viana do Castelo

Daniel Alfredo Francisco

Abordagens de *Machine Learning* para previsão de curto prazo de níveis de insulina ativa e de glicose no sangue

Trabalho de Projeto
Mestrado em Engenharia Informática

Trabalho efetuado sob a orientação de
Professor Doutor António Miguel Rosado da Cruz
Professor Doutor Jorge Manuel Ferreira Barbosa Ribeiro

Janeiro de 2024

RESUMO

Para se ter um controlo minimamente eficaz à diabetes, os doentes diabéticos precisam de controlar o nível de glicose no sangue (BGL), monitorizando-o e prevendo os seus valores futuros. Isto permite evitar BGL alto ou baixo, tomando as ações recomendadas com antecedência. Por outro lado, nos últimos anos, vários estudos científicos e de aplicabilidade na área da saúde têm evidenciado resultados promissores na aplicação de técnicas/abordagens de *Machine Learning* (*ML*), em particular para a monitorização e previsão da glicose no sangue.
Os métodos tradicionais de monitorização dos níveis de glicose, por vezes apresentam informações limitadas e não permitem previsões precisas. Isso dificulta a planificação de refeições, atividades físicas, levando a o organismo a produzir de hiperglicemia e hipoglicemia, com graves consequências para a saúde.

Neste trabalho, partindo de um estudo do estado da arte das abordagens de *ML* aplicadas ao contexto da diabetes (*mellitus*) pretende-se simular e aplicar abordagens de *ML* para a previsão de níveis de glicose em pacientes com diabetes, nomeadamente através das redes *Long Short-Term Memory* (LSTM) para o problema da previsão de níveis de insulina ativa e de glicose no sangue. Para alcançar este objetivo é apresentada uma arquitetura projetada para criação do modelo preditivo e implementação de uma aplicação web para o teste e exploração dos resultados do modelo de previsão, permitindo o uso da web pelos pacientes para monitorizarem e preverem ocorrências das flutuações de açúcar no sangue. Os resultados obtidos com *LSTM* mostraram desempenho superior. O modelo *LSTM Bidirecional* destacou-se com uma precisão de aproximadamente 82%, superando modelos comparativos como *Random Forest* 77% e Análise Discriminante Linear 77%. *F1 Score do LSTM bidirecional* atingiu cerca de 66%, enquanto *Precision* e *Recall* atingiram 77% e 57*%,* respetivamente. Estes resultados indicam a eficiência do *LSTM* na previsão precisa das flutuações nos níveis de glicose, fornecendo uma base sólida para futuras implementações.

Palavras-chave: Long Short-Term Memory, *Machine Learning*, *Deep Learning,* diabetes, glucose, previsão.

ABSTRACT

To have minimally effective control of diabetes, diabetic patients need to control their blood glucose level (BGL) by monitoring it and predicting future BGL. It allows to avoid high or low BGL by taking the recommended actions in advance. On the other hand, in recent years, several scientific and applicability studies in the health scientific area have shown promising results in the application of *ML* techniques/approaches, for the monitoring and prediction of blood glucose levels. The traditional methods of monitoring glucose levels sometimes provide limited information and do not allow for accurate predictions. This makes it difficult to plan meals and physical activity, leading the body to produce hyperglycaemia and hypoglycaemia, with serious health consequences.

In this work, starting from a state-of-the-art study of *ML* approaches applied to the context of diabetes (mellitus), it is intended to simulate and apply *ML* approaches for the prediction of glucose in patients with diabetes, namely through the *Long Short-Term Memory Networks* for the problem of prediction of active insulin and BGL. To achieve this goal, this work presents an architecture designed to create the predictive model and implement a web application for testing and exploring the results of the prediction model, allowing the use of the web by patients to monitor and predict blood glucose fluctuations. The results obtained with *LSTM* showed superior performance. The *Bidirectional LSTM model* stood out with an accuracy of approximately 82%, outperforming comparative models such as *Random* Forest 77% and Linear Discriminant Analysis 77%. Bidirectional LSTM's F1 Score reached around 66%, while Precision and Recall reached 77% and 57%, respectively. These results indicate the effectiveness of LSTM in accurately predicting fluctuations in glucose levels, providing a solid foundation for future implementations.

Keywords: Long Short-Term Memory (LSTM), Machine learning, Deep Learning, diabetes, glucose, prediction.

Índice

Conteúdo

Índice de Figuras

Índice de Tabelas

Lista de Acrónimos

RNN - Rede neuronal recorrente

LSTM - Long Short-Term Memory Networks

ML - Machine Learning

DL - Deep learning

NAR - Redes neuronais autorregressivas não lineares

BGL - Blood Glucose Level

FNN - Feed-forward

CGM - Monitorização contínua de glicose

DM - Diabetes Mellitus

AUC - Area Under the Curve

ROC - Receiver Operating Characteristic

1. INTRODUÇÃO

A ciência realça que no nosso corpo existe uma medida que afeta todos os sistemas do organismo. Se somos capazes de entender e tomar decisões que o otimizem, poderemos melhorar o nosso bem-estar físico e mental. Esta medida corresponde a quantidade de açúcar ou glicose no sangue. A glicose é a principal fonte de energia para o corpo.

Quando o assunto a ser tratado se refere a glicose, pensa-se automaticamente de pessoas que sejam diabéticos, no entanto a realidade é totalmente diferente pois isso afeta todos nós. Porém, para alguém com Diabetes de tipo 1, o organismo não vai conseguir compensar o excesso de glicose no sangue com a produção natural de insulina. Por esse motivo, a administração de insulina torna-se o meio de tratamento mais habitual. Assim, para uma pessoa com diabetes, ser tratada com insulina injetável, controlar o nível de glicose no sangue, assim como do nível de insulina ativa, é algo importante, uma vez que o excesso de qualquer um deles pode ser fatal.

Por sua vez, a Diabetes *Mellitus* (DM) é definida como um grupo de distúrbios metabólicos principalmente causado por secreção anormal de insulina. Estima-se que o aparecimento da diabetes aumente drasticamente nos próximos anos. A DM pode ser dividida em vários tipos distintos. No entanto, existem dois grandes tipos clínicos, diabetes tipo 1 (T1D) e diabetes tipo 2 (T2D). O T2D parece ser a forma mais comum de diabetes (90% de todos os pacientes diabéticos), caracterizado principalmente pela resistência a insulina. As principais causas do T2D incluem o estilo de vida, a atividade física, os hábitos alimentares e a hereditariedade. No entanto, pensa-se que o T1D se deve à destruição auto imunológica dos ilhéus de Langerhans que albergam células pancreáticas. Estes ilhéus têm funções endócrinas e podem ser divididos em duas variedades celulares principais: células A (ou alfa), que segregam glucagom (glicose), e células B (ou beta), que segregam insulina [1]. O T1D afeta quase 10% de todos os pacientes diabéticos em todo o mundo, com 10% deles a desenvolver diabetes idiopática [2] .

1.1. CONTEXTUALIZAÇÃO

A monitorização dos níveis de insulina ativa e de glicose no sangue é importante para controlar a diabetes. A previsão de curto prazo desses níveis pode ajudar os pacientes a tomar decisões sobre a administração de insulina e a ingestão de carboidratos. Têm sido aplicadas abordagens de *ML* para prever esses níveis com base em dados históricos [3].

Por outro lado, uma pessoa pode desenvolver diabetes em qualquer idade, pelo fato de também ser uma doença genética. O tratamento é normalmente feito através da aplicação de insulina. Dessa forma, a pessoa aplica uma quantidade de insulina basal relativa ao seu metabolismo, e efetua outras aplicações (*bolus*) referentes à sua alimentação, levando em conta

a quantidade de carboidratos ingeridos, o nível de glicose no sangue, e a quantidade de insulina ativa.

Segundo as estatísticas em [4], existem cerca de 347 milhões de diabéticos no mundo, razão pela qual o controlo da glicose no sangue é essencial no cuidado de indivíduos com diabetes pois em casos extremos, níveis altos ou baixos de glicose no sangue (BG, *blood glucose*) são letais. Para se evitar esses eventos adversos, foram desenvolvidos equipamentos que permitem medir os níveis de glicose no sangue, tipicamente a partir de uma gota de sangue.

Nos últimos anos, tornaram-se disponíveis dispositivos eletrónicos para monitorizar a glicose sanguínea de forma simples e contínua, dispensando as frequentes punções de dedo para aferir o nível de glicose no sangue. Estes dispositivos de medição da glicose, também conhecido como medidores contínuos de glicose, correspondem a um dos métodos de verificação da quantidade de glicose no sangue. Estes dispositivos são usados na parte interna do braço para rastrear a glicemia substituindo assim as picadas nos dedos dos pacientes. Com este tipo de dispositivos, a glicemia pode ser medida em pequenos intervalos e armazenar essas informações a fim de cuidar e prevenir riscos para a saúde [5]. Esta tecnologia permite que os indivíduos controlem facilmente os seus níveis de glicose no sangue com intervenção precoce, evitando a necessidade de visitas aos hospitais, por exemplo. Os dados recolhidos desses sensores fornecem um excelente conjunto de informação, no sentido de se aplicar algoritmos de *ML*para descobrir padrões e prever valores futuros, tanto do nível de glicose no sangue como do nível de insulina ativa.

Atualmente, tem-se assistido a um aumento no uso da tecnologia para ajudar o paciente a controlar a doença. Essas tecnologias incluem sistemas de circuito fechado, alarmes de eventos de glicose no sangue e sistemas de decisão personalizados [6]. Isso, aliado ao alto impacto que os eventos hipoglicémicos podem ter na vida de uma pessoa diabética, particularmente em pacientes com diabetes tipo 1, serve de motivação para desenvolver um sistema de classificação capaz de prever a curto prazo os níveis de glicose e de insulina ativa no sangue. Com este tipo de sistema também se quer ajudar a evitar hipoglicemias graves em pacientes que sofrem de desconhecimento da hipoglicemia.

1.2. OBJETIVOS

Os objetivos definidos para o desenvolvimento desta dissertação são os seguintes:

- Estudar abordagens para previsão de curto prazo, usando técnicas de *ML*;
- Identificar variáveis para a previsão de curto prazo de níveis de insulina ativa e de glicose no sangue, com base em séries temporais que incluem medições desses valores, entre outras variáveis;

- Comparar e selecionar abordagens para previsão de curto prazo aplicadas ao caso de estudo;
- Desenvolver, como prova de conceito, um protótipo de software baseado em *ML*, para previsão do nível de glicose e de insulina ativa, no sangue, conduzindo o sujeito alvo (paciente) a corrigir esse nível de glicose.

1.3. METODOLOGIA DE INVESTIGAÇÃO UTILIZADA

A metodologia seguida para a elaboração desta dissertação foi a *Design Science Research (DSR)*. A *DSR* é frequentemente aplicada no desenvolvimento e design de artefactos para resolver problemas de organizações, possibilitando a melhoria progressiva na forma de resolução de um dado problema. Esta metodologia de pesquisa aborda o desenvolvimento de uma solução de um problema previamente identificado através de um ciclo de construção/avaliação de um artefacto [7]. Isto é uma metodologia em que o design é adaptado progressivamente à medida que o artefacto é testado no que respeita à sua capacidade de resolução do problema detetado. Assim, é possível avaliar quais componentes do artefacto são adequados para resolver o problema e quais não são, podendo melhorar o artefacto até chegar a uma solução adequada para o problema. Assim, um projeto de investigação que use DSR precisa da criação intencional de um artefacto inovador para resolver um problema específico num determinado domínio [8].

1.4. ESTRUTURA DO DOCUMENTO

Esta dissertação está organizada em capítulos. O capítulo 1 é o onde foi feita uma contextualização do trabalho a ser desenvolvido, foram apresentados os objetivos e motivação da investigação e foi apresentada a metodologia de investigação que vai ser usada neste trabalho. No capítulo 2 apresentam-se alguns conceitos, definições, ferramentas e tecnologias semelhantes usados neste projeto. O capítulo 3 apresenta trabalhos já feitos na área, nomeadamente trabalhos relacionados com algoritmos de *ML* e algumas soluções existentes. O capítulo 4 descreve as etapas e a forma como foi tratado o *dataset*. O capítulo 5 é composto pela seleção das *features* mais significativas, a fase de treino dos algoritmos e o impacto que a seleção de *features* tem na análise. O capítulo 6 apresenta os resultados obtidos pelos diferentes algoritmos de *ML* e considerando os diferentes algoritmos de seleção de *features*. O capítulo 7 apresenta o *deployment* do modelo treinado utilizando a *Framework Flask.* Por fim, no capítulo 8 apresentam-se as conclusões do trabalho e algumas linhas para o trabalho futuro.

2. CONCEITOS, FERRAMENTAS E TECNOLOGIAS

Este capítulo apresenta uma visão geral dos conceitos que são fundamentais para este trabalho e está dividido em três secções com o objetivo de facilitar o entendimento do problema que o trabalho procura resolver. A primeira secção discute brevemente o tratamento e os riscos para pacientes com diabetes tipo 1 mal controlado. O sistema atual para coletar os valores de glicose é apresentado na segunda secção. Para concluir, a terceira secção apresenta um conjunto de tecnologias que são usadas no sistema proposto neste trabalho de projeto.

2.1. DIABETES TIPO 1

Quando se consome carboidratos, o nível de glicose no sangue aumenta. Para regular esse crescimento, o corpo produz um hormônio chamado insulina que é gerado por células beta, estas células geralmente são destruídas pelo sistema imunológico do paciente causando T1DM. Essa deficiência, se não for tratada, pode produzir hiperglicemias, gerando momentos potencialmente fatais chamados cetoacidose diabética [9]. Para evitar as hiperglicemias, em 1921 foi descoberta a insulina, que atualmente pode ser fornecida no organismo de maneira exógena por injeção ou bomba de insulina.

Existem duas categorias de insulina:

- Ação lenta (insulina basal).

A insulina de ação lenta fica ativa 24 a 48 horas e deve ser administrada apenas uma vez ao dia pelo paciente. Ela é responsável por manter os valores de glicose estáveis durante os períodos em que o paciente não consome hidratos de carbono.

- Ação rápida (*bolus*).

Para responder ao aumento da glicemia no sangue nos horários das refeições, e para ajustes pontuais de descompensação, é usada insulina de ação rápida, cuja duração/atividade é de entre duas e quatro horas.

Ao tratar a diabetes T1D por meio de injeções ou bomba de insulina, o paciente deve escolher o tipo de *bolus* de insulina que precisará em cada momento; para isso, deve-se considerar os valores da glicose sanguínea (Blood Glucose – BG) e várias outras métricas [10]. Em pacientes com T1D, existem muitas situações contínuas de hiperglicemia que resultam em cetoacidose, que a longo prazo pode resultar em retinopatia, nefropatia e doenças cardiovasculares. Por outro lado, situações de hipoglicemia ocorrem quando os valores BG estão abaixo do limite marcado. Isso ocorre devido a uma deficiência de glicose no sangue. A falta de tratamento pode causar distúrbios cognitivos, disfunção, convulsões, coma e até mesmo morte. Ao monitorar

continuamente a BG e ajustar a dose de insulina de acordo, ambas as situações são igualmente evitadas.

O tratamento do T1D pode ser resumido como a busca constante de manter o equilíbrio entre a ingestão de alimentos e a dose de insulina injetada, levando em conta fatores transversais como desporto, ciclos de sono e outros fatores que afetam o valor de BG e a filtragem da insulina de forma indireta, portanto, seria um avanço significativo ter a capacidade de prever com precisão os níveis de BG futuros com base nos níveis atuais [11].

Por outro lado, as estratégias atuais para controlar a diabetes dependem da medicação, de um estilo de vida saudável e do monitoramento dos valores de glicose, exigindo muitos comportamentos de autocuidado que são muitas vezes um grande fardo para os pacientes [12][13]. Durante todo o dia, os níveis glicémicos vão oscilar para cima e para baixo. Até um certo intervalo, esta variação de açúcar no sangue é normal. No entanto, duas situações anormais podem ocorrer: os níveis glicémicos podem ser muito altos (> 240 mg/dL), o que é chamado de hiperglicemia, ou podem ser demasiado baixos (< 70 mg/ dL), conhecido como hipoglicemia [14][15].

2.2. SISTEMA DE MONITORIZAÇÃO CONTÍNUA DE GLICOSE (CGM)

Atualmente, existem sistemas de monitorização de glicose (*Continuous Glicose Monitoring* - CGM) para monitorizarem continuamente os níveis de glucose no sangue. Esses sistemas visualizam os níveis de glucose no sangue em tempo real e também enviam alertas e alarmes quando ocorrem hipoglicemias e hiperglicemias. Esses sistemas funcionam 24 horas por dia. Cada vez mais sofisticados, precisos, discretos, confortáveis e fáceis de usar, alguns desses dispositivos podem conectar-se à internet para armazenar ou partilhar dados [16]. Em [17] lê-se que vários estudos mostram que o uso desse sistema é diretamente proporcional aos benefícios clínicos em várias populações de pacientes com diferentes níveis de controlo glicémico.

Dependendo das características basais dos pacientes, o uso de CGM é geralmente associado a uma redução no risco de hipoglicemia. Os medidores de glicose no sangue por punção no dedo, e os CGMs relacionam-se, fornecendo informações detalhadas sobre as tendências para a tomada de decisões diárias. Além disso, os dados sobre BG podem ser automaticamente coletados e partilhados.

Um dispositivo de CGM controla de maneira contínua o nível de glicose na camada fina de líquido que rodeia as células do corpo debaixo da pele, chamado líquido intersticial ou líquido tissular. A medição do nível de glicose no líquido instersticial acarreta um atraso médio de dez minutos relativamente ao valor da glicose no sangue que seria obtido por um aparelho de punção do dedo.

O sistema CGM é estruturado como um sensor, transmissor e recetor com capacidade de armazenamento de até oito horas entre medições (ver Figura 1).

- **Sensor:** é um dispositivo que mede os níveis de glicose no fluido intersticial de forma quase contínua. Ele é instalado parcialmente debaixo da pele. Alguns sensores estão completamente debaixo da pele e são conhecidos como sensores subcutâneos.
- **Transmissor**: Responsável por transmitir informações sem fios do sensor para o destinatário ou para um dispositivo inteligente compatível. O dispositivo envia ou faz isso automaticamente ou apenas quando o sensor é digitalizado. A transmissão de informações sobre glicose é feita sem dor e pode ser feita através da roupa. O transmissor pode ser incorporado ao sensor ou separado dele.
- O transmissor possui um módulo NFC para coleta de dados e um módulo Bluetooth para enviar alarmes.
- **Leitor**: um dispositivo inteligente recarregável com tamanho menor do que um telemóvel, ou pode usar o smartphone. As informações são mostradas em forma de gráfico e são constantemente atualizadas com novos dados sobre a glicose. Também mostra a tendência atual, ajudando essa informação sobre a tomada de decisão no tratamento.

Figura 1-Esquema geral da medição da glicose

2.3. REDES NEURONAIS ARTIFICIAIS

Amey Thakur e Archit Konde, em [18] definem a rede neuronal como um tipo de algoritmo de *ML* que é inspirado na estrutura do cérebro humano. É composta por um conjunto de nós interconectados, que são chamados de neurónios. Cada neurônio recebe entradas de outros neurónios e gera uma saída que é enviada para outros neurónios. Os autores explicam que as redes neuronais artificiais aprendem a executar tarefas analisando amostras e inferindo regras sem a necessidade de envolvimento direto de um programador.

As primeiras redes neuronais foram criadas no início da década de 1940 pelos matemáticos Warren McCulloch e Walter Pitts, que criaram um sistema básico baseado em algoritmos para imitar a atividade cerebral humana. O perceptron, um algoritmo revolucionário criado para lidar

com tarefas complicadas de reconhecimento, foi inventado por Frank Rosenblatt, da Universidade de Cornell, em 1957, acelerando o trabalho no campo. A falta de capacidade informática necessária para analisar grandes volumes de dados atrasou os progressos nas quatro décadas que se seguiram. Nos anos 2000, os cientistas da computação finalmente conseguiram o que precisavam, devido ao aumento do poder de processamento e hardware sofisticado, bem como à disponibilidade de enormes conjuntos de dados dos quais tirar, e as redes neurais e a IA decolaram, sem fim à vista. Considere que 90% dos dados da Internet foram produzidos desde 2016. Devido à rápida expansão da Internet das Coisas, esta taxa continuará a aumentar [19].

Em [20], os modelos autorregressivos (AR) são definidos como classe de modelos matemáticos que descrevem a relação entre o valor atual de uma série temporal e seus valores passados. Esses modelos são baseados na suposição de que os valores de uma série temporal são influenciados por seus valores anteriores. Modelos AR são frequentemente utilizados para prever valores futuros de uma série temporal, pois consideram a dependência temporal dos dados passados. Os autores tratam os Modelos autorregressivos não lineares (NAR) como uma extensão dos modelos AR tradicionais, incorporando a possibilidade de dependência não linear entre os valores passados e o valor atual da série temporal. Esses modelos são mais flexíveis e podem capturar padrões mais complexos nas séries temporais, especialmente quando há dependência não linear entre os dados.

Long Short-Term Memory Networks (LSTM) [21] é uma rede neuronal de *Deep Learning* (DL) que permite que a informação persista, ou por outras palavras, armazena informação durante o processo de aprendizagem. É um tipo especial de Rede Neuronal Recorrente que é capaz de lidar com o problema de gradiente de desaparecimento enfrentado pela rede neuronal recorrente (RNN). LSTM foi projetado por Hochreiter e Schmidhuber que resolve o problema causado por RNN tradicionais e algoritmos de *ML* [18]. Neste contexto, LSTM é uma arquitetura de RNN amplamente utilizada em *DL*. Destaca-se na captura de dependências de longo prazo, tornando-o eficiente para tarefas de previsão de sequência. Ao contrário das redes neurais tradicionais, o LSTM incorpora conexões de feedback, permitindo processar sequências inteiras de dados, não apenas pontos de dados individuais. Isso o torna altamente eficaz na compreensão e previsão de padrões em dados sequenciais, como séries temporais [21].

As redes neuronais *feed-forward* são um tipo de rede neuronal artificial em que a informação flui em uma única direção, da camada de entrada para a camada de saída. As redes neurais de *feed-forward* são o tipo mais comum de rede neuronal, e são utilizadas em uma ampla variedade de aplicações. Estas redes são compostas de uma ou mais camadas de neurónios. Cada neurônio recebe entradas de outros neurónios, realiza uma operação matemática sobre essas entradas e gera uma saída. A saída de cada neurônio é transmitida para os neurónios da camada seguinte.

O processo de aprendizagem em uma rede neuronal de *feed-forward* é realizado por meio de um processo chamado *backpropagation*. O *backpropagation* consiste em calcular a diferença entre a saída real da rede e a saída desejada. Essa diferença é então usada para ajustar os pesos dos neurónios da rede, de forma que a saída da rede se aproxime da saída desejada [22] [23].

2.3.1 Arquitetura da Rede Neuronal

Uma rede neuronal é um método de inteligência artificial que ensina computadores a processar dados de uma forma inspirada pelo cérebro humano. É um tipo de processo de *ML*, chamada *DL*, que usa nós ou neurónios interconectados em uma estrutura em camadas, semelhante ao cérebro humano. A rede neuronal cria um sistema adaptativo que os computadores usam para aprender com os erros e melhorarem continuamente. As redes neuronais artificiais tentam solucionar problemas complicados, como resumir documentos ou reconhecer rostos com grande precisão.

Uma rede neuronal básica tem neurónios artificiais interconectados em três camadas:

- **Camada de entrada;**

As informações do mundo externo entram na rede neuronal artificial pela camada de entrada. Os neurónios de entrada processam os dados, analisam ou categorizam esses dados e os encaminham para a próxima camada.

- **Camada oculta;**

As camadas ocultas usam as entradas da camada de entrada ou de outras camadas ocultas. As redes neuronais artificiais podem ter várias camadas ocultas. Cada camada oculta analisa o resultado da camada anterior, processa-o mais um pouco e o encaminha para a próxima camada.

- **Camada de saída**

A camada de saída fornece o resultado de todos os dados processados pela rede neuronal artificial. Ela pode ter um ou vários nós. Por exemplo, se tivermos um problema de classificação binária (sim/não), a camada de saída terá um nó de saída, o que fornecerá o resultado como 1 ou 0. Porém, se tivermos um problema de classificação de várias classes, a camada de saída poderá ter mais de um nó de saída.

As *DL*, têm várias camadas ocultas, com milhões de neurónios artificiais interligados. Um número, conhecido como peso, representa as conexões entre um nó e outro. O peso será um número positivo se um nó excitar o outro, ou negativo se um nó reprimir o outro. Uma particularidade nestas estruturas é o facto dos nós com valores de peso maiores têm mais influência nos outros nós.

Teoricamente, as redes neuronais profundas podem direcionar qualquer tipo de entrada para qualquer tipo de saída. Porém, tipicamente, elas necessitam de muito mais treino do que outros

métodos de *ML*. Elas precisam de muitos exemplos de dados de treino, enquanto as redes simples talvez precisem de apenas centenas ou milhares (ver Figura 2).

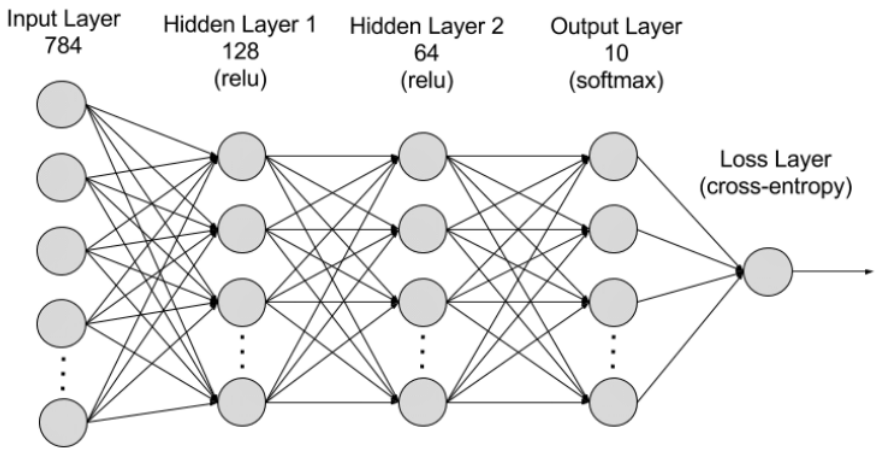


Figura 2 - Arquitetura da Rede Neuronal [75]

2.3.2 LSTM - FUNCIONAMENTO

Os autores em [76] e [31] resumidamente explicam o funcionamento e a arquitetura de uma rede de LSTM. Elas são uma espécie de *RNN* e funcionam de forma semelhante aos *RNNs* tradicionais, sendo distinguidas pelo seu mecanismo de *Gating*. Esta funcionalidade aborda o problema de "memória de curto prazo" de *RNNs*.

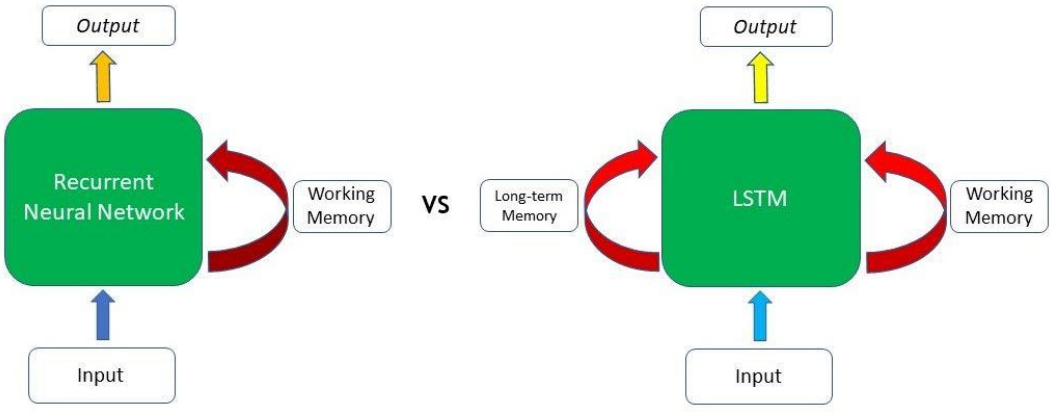
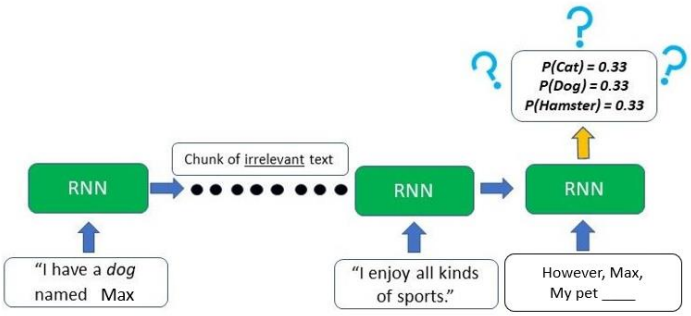


Figura 3 - LSTM vs. RNN [77]

A diferença reside principalmente na capacidade do LSTM de preservar a memória de longo prazo, como pode ser visto na Figura 3. A maioria das tarefas de Processamento de Linguagem Natural (NLP), bem como séries sequenciais e de tempo, requer isso. Por exemplo, suponhamos que temos uma rede que cria texto com base em informações que recebemos. No início do texto, é dito que o autor tem um "cão chamado Max". O autor traz o seu animal de estimação novamente depois de algumas outras frases em que não há menção a um cão ou animal de

estimação, e o modelo é obrigado a produzir a próxima palavra para "No entanto, Max, meu animal ____". Como a palavra animal apareceu logo antes do espaço em branco, um RNN pode inferir que a próxima palavra provavelmente será um animal que pode ser mantido como um



animal de estimação.

Figura 4 – Esquemático geral associado à limitação das RNNs [77]

No entanto, a Figura 4 mostra que o RNN normal só poderá usar a informação contextual do texto que apareceu nas frases mais recentes devido à memória de curto prazo, o que não é muito útil. As informações pertinentes desde o início do texto foram perdidas pelo RNN, portanto, ele não pode determinar qual animal pode ser o animal de estimação.

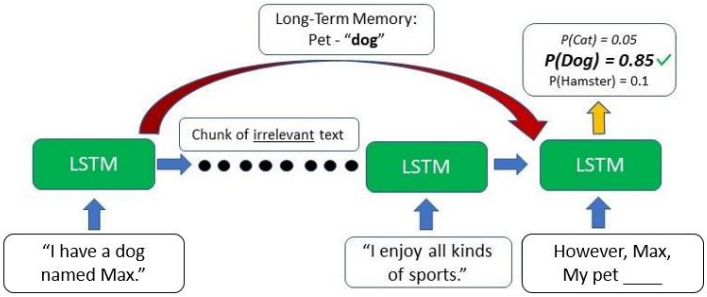


Figura 5 - Os LSTMs podem manter os recursos por um longo prazo [77]

Por outro lado, a Figura 5 retrata que as informações do autor sobre um cão de estimação podem ser armazenadas pelo LSTM, o que ajudará o modelo a escolher "o cão" quando se trata de gerar o texto nesse ponto devido às informações contextuais de um passo de tempo significativamente anterior.

O mecanismo de *gating* de cada célula LSTM é o segredo do modelo treinado com LSTM. Uma função de ativação *tanh* é usada na célula RNN padrão para transferir a entrada de um *time-step* e o estado oculto do passo anterior para criar um novo estado escondido e uma saída.

Em contraste, a estrutura dos LSTMs é mais complexa. A célula LSTM recebe três tipos diferentes de informações em cada ciclo de tempo: os dados de entrada atuais, a memória de

curto prazo da célula anterior (semelhante aos estados ocultos em RNNs) e a memória de longo prazo. O estado da célula é o termo usado para a memória de longo prazo, enquanto o estado oculto é o termo usado para a memória de curto prazo.

As portas são então usadas pela célula para controlar as informações que devem ser mantidas ou descartadas em cada etapa. Antes de transferir as informações de longo e curto prazo para a próxima célula, as portas são usadas pela próxima célula (ver Figura 6).

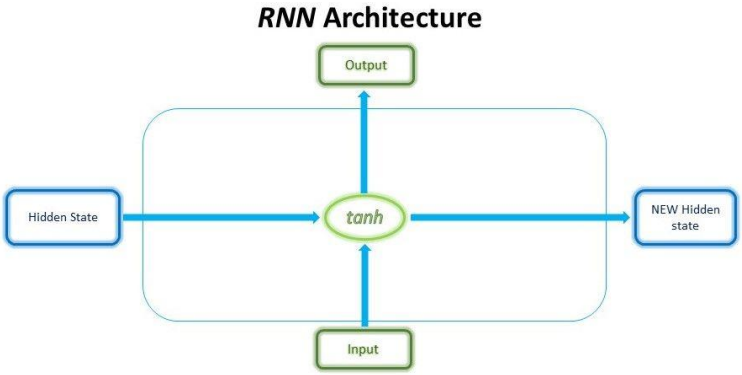


Figura 6 - Funcionamento interno de uma célula RNN [78]

É possível pensar nessas portas como filtros de água. Semelhante ao que os filtros de água fazem para impedir a passagem de impurezas, o objetivo ideal dessas comportas é remover seletivamente informações irrelevantes. Ao mesmo tempo, as comportas só armazenam informações úteis, e apenas água e nutrientes saudáveis podem passar por esses filtros. Claro, é necessário treinar esses portões para distinguir com precisão os componentes úteis (ver Figura 7).

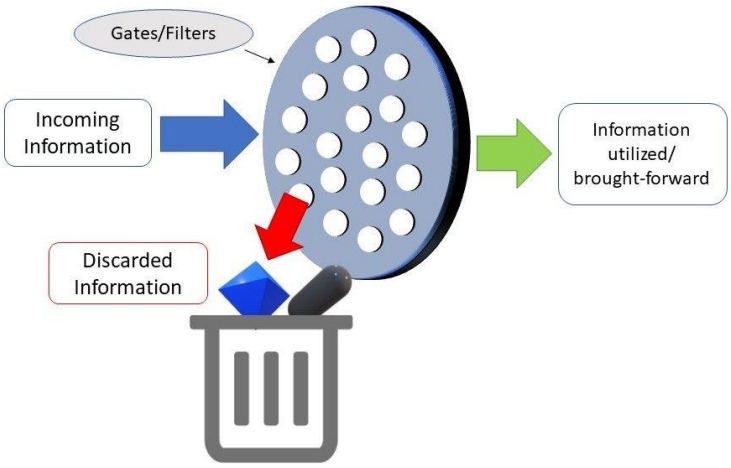


Figura 7 – Gates do LSTM podem ser vistos como filtros [79]

Os nomes desses "portões/gate" são Portão de Entrada (*Input Gate*), Portão de Esquecimento (*Forget Gate*) e Portão de Saída (*Output Gate*). Os nomes desses portões podem variar muito, mas basicamente os cálculos e como esses portões funcionam são os mesmos.

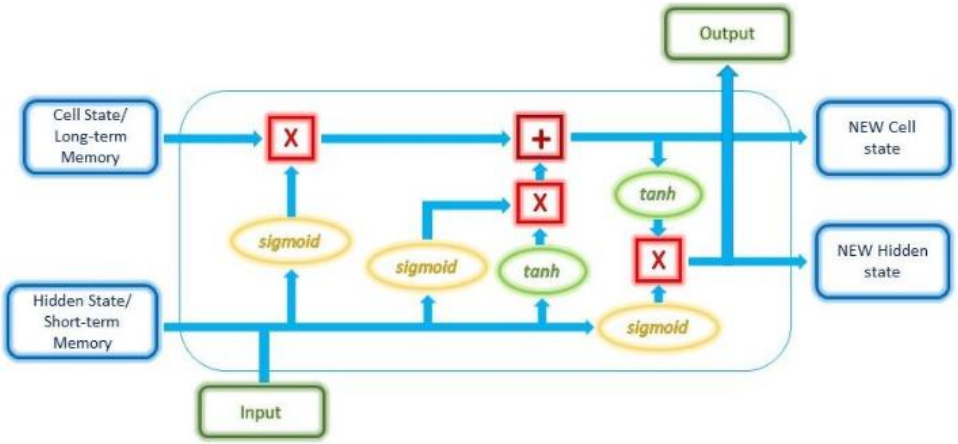


Figura 8 - Arquitectura LSTM [79]

A Figura 8 apresenta como está estruturada a arquitetura do LSTM. As arquiteturas LSTM (Long Short-Term Memory) foram introduzidas para superar limitações das redes neuronais convencionais. As LSTMs são um tipo único de RNN que foi desenvolvido para capturar dependências de longo prazo em sequências de dados. Eles são compostos por uma célula de memória e unidades de porta (*gates*), que funcionam juntas para controlar o fluxo de informações ao longo do tempo. A seguir explicam-se os principais mecanismos da arquitetura LSTM e como esses componentes permitem que as redes aprendam e mantenham os dados pertinentes durante o processamento de sequências.

Input Gate

O *Input Gate* determina quais dados adicionais serão inseridos na memória de longo prazo. Só usa a memória de curto prazo da etapa de tempo anterior e as informações da entrada atual. Como resultado, ele precisa filtrar as informações dessas variáveis inúteis.

Isso é matematicamente possível com duas camadas. A primeira camada funciona como um filtro, escolhendo quais informações podem e não podem passar. Para criar essa camada, transformamos a entrada de corrente e a memória de curto prazo em uma função sigmoide. Os valores de 0 a 1 serão transformados pela função sigmoide, com o valor 0 indicando que uma parte da informação não é importante, enquanto um valor de 1 indica que a informação será usada. Isso facilita a tomada de decisão sobre quais valores devem ser descartados ou mantidos.

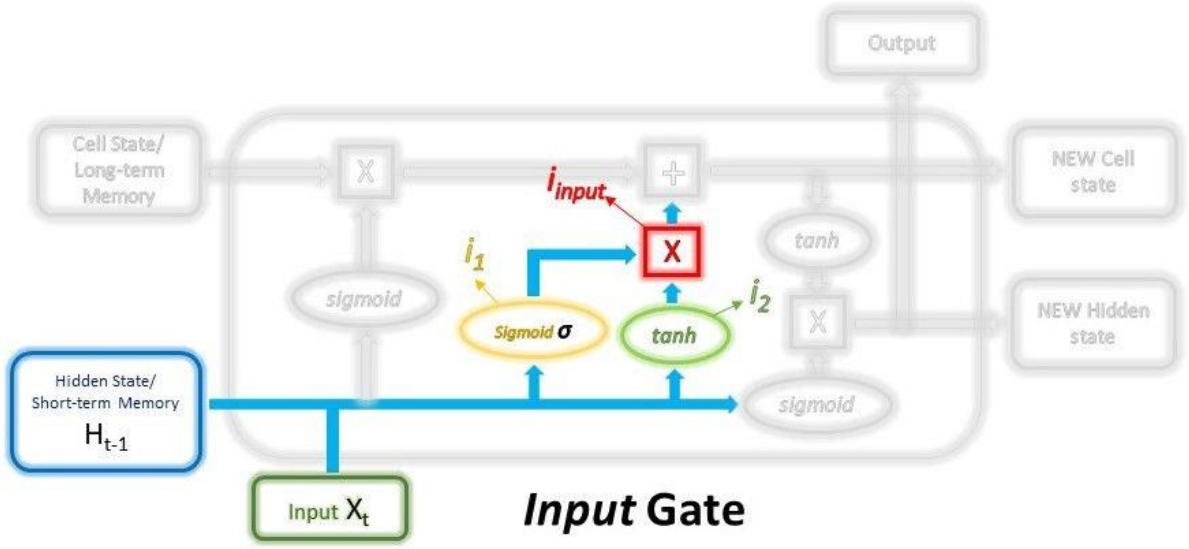


Figura 9 - Input Gate [79]

Ao treinar a camada pela retro-propagação, os pesos da função sigmoide serão ajustados para que ela aprenda a apenas deixar passar o útil enquanto descarta os recursos menos críticos (ver Figura 9).

$$eu_1 = s\,(Em_{eu_1} \times (H_{t-1}, x_t) + beuums_{eu_1})$$

A segunda camada também recebe entrada de corrente e memória de curto prazo e as passa por uma função de ativação, normalmente a função *tanh*, para administrar a rede.

$$eu_2 = tumnh\,(Em_{eu_2} \times (H_{t-1}, x_t) + beuums_{eu_2})$$

Após isso, as saídas das duas camadas são multiplicadas, dando como resultado a saída que será usada para guardar informações na memória de longo prazo.

$$eu_{eunpnat} = eu_1 * eu_2$$

FORGET GATE

O *Forget Gate* decide quais informações da memória de longo prazo devem ser mantidas ou descartadas. Isso é feito multiplicando a memória de longo prazo recebida por um vetor de esquecimento gerado pela entrada atual e pela memória de curto prazo recebida.

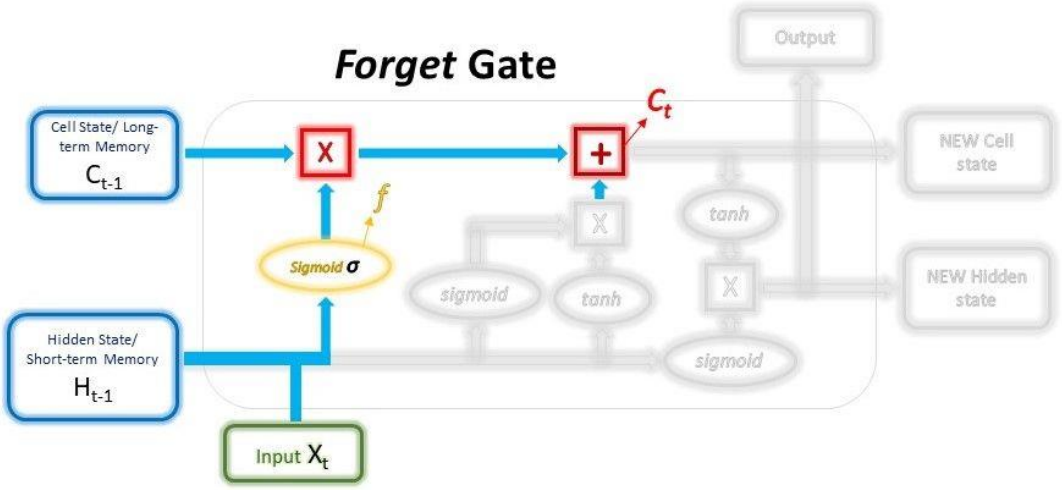


Figura 10 - Forget Gate [79]

Assim como a primeira camada na porta de entrada, o vetor *forget* também é uma camada **de filtro seletivo** (ver Figura 10). Para obter o vetor *forget*, a memória de curto prazo e a entrada de corrente são passadas através de uma *função sigmoide*, semelhante à primeira camada na porta de entrada acima, *mas com pesos diferentes*. O vetor será composto por 0s e 1s e será multiplicado com a memória de longo prazo para escolher quais partes da memória de longo prazo reter.

$$f=\sigma\left(W_{forget}\ x\ (H_{t-1}, x_t)+bias_{forget}\right)$$

As saídas do *Input Gate* e do *Forget Gate* passarão por uma adição pontual para dar uma nova versão da memória de longo prazo, que será passada para a próxima célula. Esta nova memória de longo prazo também será usada no portão final, o portão de **saída**.

$$C_t=C_{t-1}*f+i_{input}$$

OUTPUT GATE

Para criar memória de curto prazo ou estado oculto, a porta de saída recebe a entrada atual, a memória de curto prazo anterior e a memória de longo prazo recém calculada. Essa memória será enviada para a célula na próxima fase. Além disso, esse estado oculto pode ser usado para extrair a saída da etapa de tempo atual (ver Figura 11).

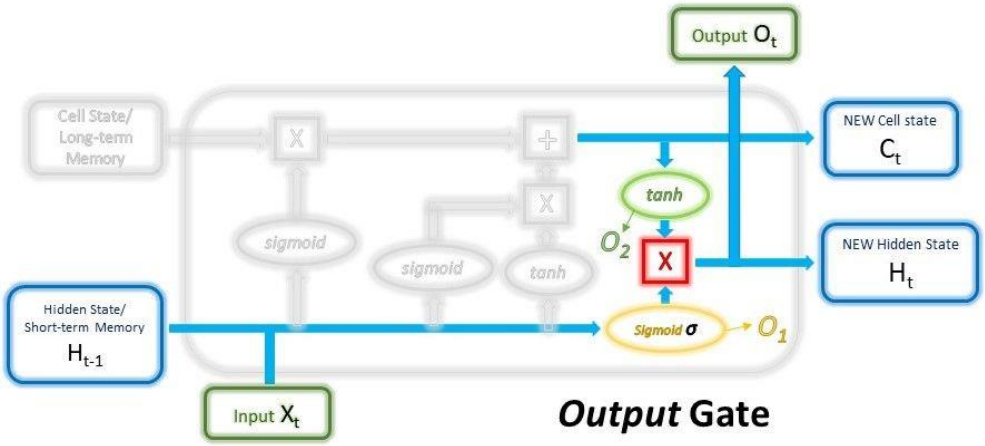


Figura 11 - Output Gate [79]

Para iniciar o terceiro e último filtro, a entrada de corrente e a memória de curto prazo anterior serão transferidas para uma função sigmoide com diferentes pesos. Coloca-se então a nova memória de longo prazo usando a função de ativação *tanh*. A nova memória de curto prazo será criada multiplicando as saídas dos dois métodos.

$$O_1=\sigma\left(W_{output_1} \times (H_{t-1}, x_t)+bias_{output_1}\right)$$

$$O_2=\tanh\left(W_{output_2} \cdot C_t+bias_{output_2}\right)$$

$$H_t, O_t=O_1 * O_2$$

Estas portas geram então memória de longo e curto prazo, que é transferida para a próxima célula para que o processo seja repetido. O estado oculto, ou memória de curto prazo, pode ser usado para obter a saída de cada passo de tempo.

2.3.3 CONJUNTO DE CASOS DE TESTE E TREINO"

A divisão do conjunto de dados em conjuntos de treino e teste é um procedimento essencial em *ML*. Isto permite avaliar o desempenho e a capacidade de generalização de modelos, como algoritmos de classificação. O *dataset* é dividido em duas partes.

Primeiro, separou-se as variáveis independentes (denominando-se pela variável "X"), correspondendo aos atributos ou características que são usados para fazer previsões ou inferências. Essas características podem incluir informações como glicose, pressão sanguínea, idade e índice de massa corporal, entre outras, para a deteção de diabetes. A variável dependente (denominando-se pela variável "y") é aquela que será classificada. Neste caso, a variável "*Outcome*" indica se uma pessoa tem diabetes (com um valor de 1) ou não (com um valor de 0).

As variáveis independentes e as variáveis dependentes são utilizadas para criar os conjuntos de treino e teste. O modelo de *ML* é treinado com o conjunto, enquanto o conjunto de teste avalia o desempenho do modelo em dados não vistos, que não foram usados durante o treino. Para criar as fases de modelação e teste do modelo, o conjunto de dados é dividido em 80% dos dados de treino e os 20% restantes do teste.

Escolher a proporção dividida entre dados de treino e teste é uma etapa importante no desenvolvimento de modelos de *ML*. No contexto deste trabalho, estabeleceu-se várias configurações para a divisão, considerando várias proporções, nomeadamente 60%-40%, 75%-25%, 80%-20% e 85%-15%. Para cada escolha realizou-se vários testes comparativos entre as configurações. O objetivo foi avaliar como o desempenho dos modelos foi afetado por cada configuração de treino e teste.

Após várias avaliações, constatou-se que o desempenho dos modelos se mantinha consistente ao longo das diferentes configurações testadas. No entanto, a configuração 80%-20% foi a escolhida por fornecer resultados semelhantes aos das configurações citadas no paragrafo anterior. Ao escolher a configuração 80%-20%, garantiu-se um equilíbrio adequado entre os dados de treino, que são essenciais para que o modelo aprenda, e os dados de teste, que são importantes para avaliar a capacidade de generalização da solução proposta.

2.4. FERRAMENTAS E TECNOLOGIAS

Nesta secção vamos falar sobre algumas das ferramentas e tecnologias usadas no desenvolvimento da solução a apresentar.

2.4.1- CSV

Em inglês, a sigla CSV significa "valores separados por virgulas". Um ficheiro CSV pode ser qualquer ficheiro de texto em que os caracteres estão separados por virgulas, criando uma forma de tabela com linhas e colunas. Cada vírgula separa duas colunas, enquanto cada linha do texto tem uma linha adicional. A criação de ficheiros CSV é muito simples com esse método, como explicaremos mais adiante. Isso explica porque a criação de tabelas de conteúdo está diretamente ligada aos ficheiros ".csv". Esses ficheiros são praticamente universais devido à facilidade de passar de uma tabela para CSV e vice-versa e ao baixo espaço de armazenamento e requisitos de computação [24].

2.4.2- PYTHON

Python é uma das linguagens de programação mais populares para *ML* devido à sua sintaxe simples e intuitiva, além da grande quantidade de bibliotecas especializadas disponíveis, como o *scikit-learn, TensorFlow e Keras*. Essas bibliotecas fornecem uma ampla gama de algoritmos de

ML, bem como ferramentas para pré-processamento de dados, validação de modelos e visualização de resultados [25].

2.4.3. GOOGLE COLAB

O Google Colaboratory é uma ferramenta do Google que compila códigos na linguagem Python. Para ter acesso basta ter uma conta do Google.

A grande facilidade do *colab* é que ao ter acesso ao site, através de qualquer máquina, é feita a conexão a um ambiente de execução remoto, do próprio Google. Esses computadores remotos são otimizados para ciência de dados e o utilizador pode até escolher algumas configurações de hardware de como o código será processado.

Outra característica importante é o facto do código ser feito com *notebooks*, o que facilita, na prática, executar linhas de código separadas e não o código inteiro em cada vez. Em algumas IDEs é necessário executar o código por inteiro, o que pode levar mais tempo do esperado muito tempo [12].

2.4.4. FLASK

Flask é uma *framework* web para *Python* que permite desenvolver aplicações web facilmente. É uma *microframework* que não inclui um ORM (*Object Relational Manager*) ou tais recursos. Um *Web Application Framework* ou simplesmente um *Web Framework* representa uma coleção de bibliotecas e módulos que permitem aos programadores de aplicações para a Web escrever aplicações sem se preocupar com muitos dos detalhes de baixo nível.

O *Flask* foi desenvolvido por Armin Ronacher, que liderou uma equipa internacional de entusiastas de *Python* chamada *Poocco*. O *Flask* é baseado no kit de ferramentas Werkzeg WSGI e no mecanismo de modelo Jinja2. Ambos são projetos *Poocco*.

Uma *microframework* é uma *framework* modularizada com uma estrutura inicial muito mais simples do que uma framework tradicional [26].

Podemos dizer que as *microframeworks* são uma versão simplificada dessas *frameworks*, e são frequentemente usadas para criar micro-serviços, como *APIs RESTful*.

Uma *microframework* pode ser vista como um *lego*(ver Figura 12). Um projeto criado com a microframework inicialmente possui apenas o necessário para funcionar (normalmente, um sistema de rotas), mas ao longo do projeto, pode ser necessário adicionar recursos adicionais, como conexão com bases de dados, sistemas de *templates*, envio de email, etc. Como resultado disso, novas bibliotecas são "encaixadas" no projeto, como uma estrutura de lego [27].

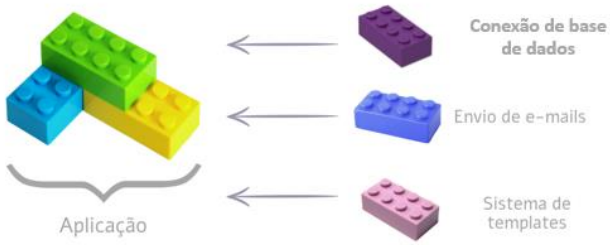


Figura 12-Flask Microframework[28]

Em termos gerais, as características do Flask são as seguintes:

- **Simplificação**

Um projeto escrito com *Flask* é mais simples se comparado com frameworks maiores, porque possui apenas o necessário para o desenvolvimento de uma aplicação e a sua arquitetura é mais simples.

- **A evolução ocorre rapidamente**

O *Flask* permite que os programadores apenas criem o que é necessário para um projeto, sem a necessidade de efetuar configurações frequentemente desnecessárias.

- **Projetos de menor porte**

Os projetos escritos em *Flask* geralmente são menores e mais leves do que os projetos escritos em frameworks maiores devido à sua arquitetura muito simples, que consiste em um único arquivo inicial.

- **Aplicações robustas**

Como é uma microframework, o *Flask* pode ser personalizado totalmente, permitindo a criação de arquiteturas mais definidas se necessário. Isso permite que ele crie aplicações robustas.

3. ESTUDO DO ESTADO DA ARTE

Pretende-se, nesta secção, apresentar abordagens de investigação e estudos e tecnologias relacionados com o tema da dissertação, com análise das suas conclusões, permitindo ter assim uma visão geral do estado da arte. No âmbito deste trabalho verificou-se que vários autores propõem projetos que visam definir e criar as condições para a previsão de curto prazo de níveis de insulina ativa e de glicose no sangue [29 - 34]. Por outro lado, vários trabalhos têm vindo a ser apresentados com sucesso envolvendo as séries temporais, a Inteligência Artificial e Machine Learnng e em particular a exploração das LSTM para objetivos de previsão.

3.1. SÉRIES TEMPORAIS

Uma série de tempo (também designada por série temporal ou “time series” na terminologia anglo-saxónica), consiste em uma coleção de observações da mesma variável sequencialmente medidas ao longo do tempo. Séries de tempo podem ser divididas em contínuas, se as medições são realizadas continuamente através do tempo, ou séries de tempo discretas, quando as medidas são feitas em pontos discretos de tempo. No entanto, a variável medida pode ser discreta ou contínua em ambos os casos, ou seja, a classificação de uma série de tempo como contínuo ou discreta refere-se exclusivamente ao eixo do tempo. Uma característica usual de dados de séries de tempo é uma dependência entre observações atuais e passadas, portanto, na sua análise, a ordem em que os pontos de tempo foram coletados deve ser levada em conta [30].

Exemplos de análise de séries cronológicas:

- Atividade eléctrica no cérebro
- Medições da precipitação
- Preços das acções
- Número de manchas solares
- Vendas anuais a retalho
- Batimentos cardíacos por minuto

Indicadores económicos e métricas de evolução da saúde dos doentes - todos são dados de séries temporais. Os dados de séries temporais também podem ser métricas de servidor, monitorização de desempenho de aplicações, dados de rede, dados de sensores, eventos, cliques e muitos outros tipos de dados analíticos. Na Figura 13 podemos ver como a temperatura varia ao longo do tempo.

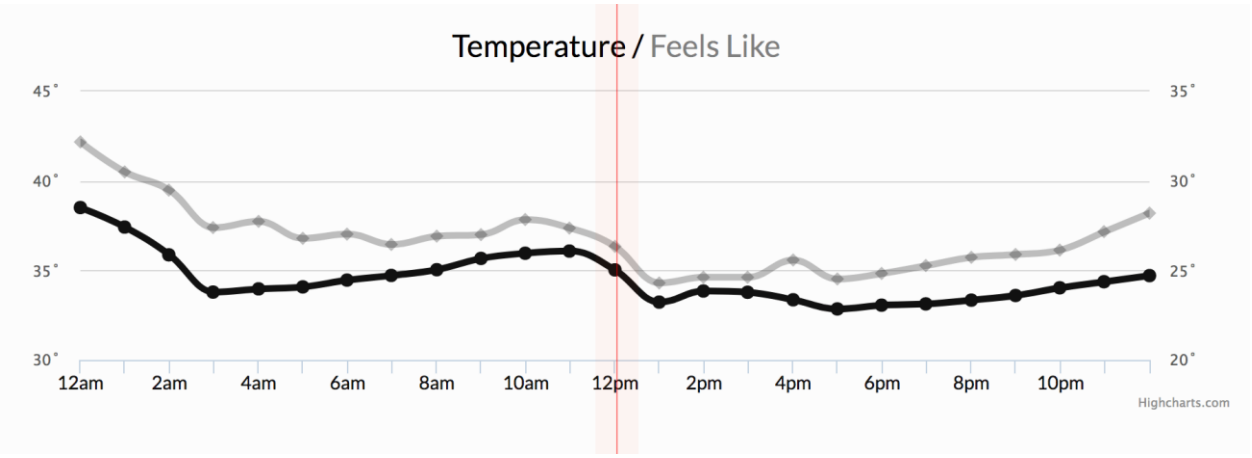


Figura 13 - Variação do climática [29]

Segundo os autores em [30], os principais objetivos ao realizar uma análise de séries temporais são:

- **Descrição**: representar os dados utilizando métodos gráficos e/ou estatísticas sumárias;
- **Modelação**: encontrar um modelo estatístico adequado que possa descrever o processo de geração de dados;
- **Previsão:** estimar os valores da série no futuro;
- **Controlo**: se a previsão for boa, o analista pode agir em conformidade e controlar um determinado processo;

Alguns aspetos importantes a serem levados em consideração ao modelar uma série temporal são [29]:

- **Tendência**: representa a mudança na média dos dados a longo prazo, o que pode afetar o desempenho do modelo, caso em que deve ser removido. A tendência de uma série temporal representa a mudança na média dos dados a longo prazo. Essa mudança pode ser causada por vários fatores. Em geral, a tendência deve ser removida de uma série temporal antes de modelá-la. Isso ocorre porque a tendência pode afetar o desempenho do modelo de séries temporais de várias maneiras. Em primeiro lugar, a tendência pode mascarar os padrões reais da série temporal. Por exemplo, se a série temporal está a aumentar a longo prazo, o aumento da tendência pode fazer com que os padrões sazonais ou cíclicos sejam menos visíveis. Em segundo lugar, a tendência pode fazer com que o modelo seja menos preciso na previsão de valores futuros. Isso ocorre porque o modelo pode aprender a tendência dos dados históricos e, portanto, projetar essa tendência para o futuro. No entanto, a tendência pode não continuar no futuro, o que pode levar a previsões incorretas;
- **Sazonalidade:** corresponde às variações regulares e previsíveis num determinado período. Por exemplo, espera-se que as vendas de espumantes sejam maiores no período

do Ano Novo. Tal como a tendência, também a sazonalidade pode afetar o desempenho do modelo, caso em que também deve ser removida;

- **Padrões cíclicos:** correspondem quando não há um período fixo para oscilações previsíveis. Um exemplo comum é o ciclo económico, em que há períodos de recessão e períodos de crescimento;
- **Resíduos (ou componente irregular):** corresponde à parte restante após a remoção de tendência, sazonalidade e padrões cíclicos, e que também pode ter impacto na dinâmica das séries temporais;
- **Auto correlação**: avalia o grau de dependência da série temporal em relação aos seus dados históricos. Como mencionado anteriormente, o ponto atual de uma série temporal está geralmente relacionado com o que aconteceu no passado. A auto correlação avalia a dependência ponto-a-ponto do ponto de tempo atual em relação aos anteriores. Se em todos os desfasamentos a série temporal apresentar baixos valores de auto correlação, é considerado ruído branco.

Assim, em qualquer momento, os autores em [29] formulam a série temporal ***y*** da seguinte forma:

$$y = Tendência + Sazonalidade + Padrão\ cíclico + Resíduos$$

3.2. Inteligência Artificial e *Machine Learning*

Atualmente, falar sobre Inteligência Artificial (IA) é muito popular porque muitos meios de comunicação abordam o assunto. Por outro lado, a IA entrou nas nossas vidas de várias maneiras, como por meio de tecnologia, escrita, cinema, televisão, rádio, etc.

A inteligência está diretamente ligada às ações dos humanos, pois é necessária para várias tarefas, como compreender a linguagem, expressar-se e aprender. Antes de falar sobre IA, precisamos entender o que significa.

Segundo o Dicionário de Língua Portuguesa, a inteligência é o conjunto de todas as áreas de conhecimento (memória, imaginação, juízo, raciocínio, abstração e conceito). Por outro lado, pode ser definido como faculdade de adquirir e aplicar conhecimentos e habilidades [35]. Ainda em [35], os autores definem a Inteligência Artificial como uma área de estudo no campo da informática que estuda como *hardware* e *software* se estão a desenvolver para que possam fazer coisas como os humanos, como aprender, raciocinar ou realizar tarefas. Existem muitas coisas que distinguem a inteligência artificial da inteligência humana, sendo que a consciência é o que mais se destaca.

A definição de IA é flexível e tem mudado ao longo dos tempos. A inteligência artificial é definida como "a capacidade de um sistema para interpretar corretamente dados externos, para aprender com esses dados, e para utilizar essas aprendizagens para alcançar objetivos e tarefas

especificas por meio da adaptabilidade flexível", de acordo com a definição de Kaplan e Haenlein e outros autores [36], [37] .

Por sua vez, o termo "inteligência artificial" refere-se ao uso de tecnologia e computadores para reproduzir pensamentos críticos e comportamentos inteligentes como os de um ser humano. Em 1956, John McCarthy usou o termo IA para descrever a engenharia e a ciência da fabricação de máquinas inteligentes.

A IA está a crescer no sector da saúde pública e terá um grande impacto em todos os aspetos dos cuidados primários. As aplicações de computador habilitados para IA ajudarão os médicos de atenção primária a identificar melhor os pacientes que necessitam de atenção extra e a fornecer protocolos personalizados para cada indivíduo. Os médicos de cuidados primários podem usar IA para fazer anotações, analisar suas discussões com os pacientes e inserir as informações necessárias diretamente nos sistemas desenvolvidos. Essas aplicações coletarão e analisarão dados de pacientes e os apresentarão aos médicos de atenção primária, juntamente com informações sobre as necessidades médicas do paciente [38].

Em síntese pode se dizer que “A inteligência artificial é a ciência e a engenharia para construir máquinas inteligentes, especialmente programas de computador inteligentes". Ela está relacionada à tarefa de entender a inteligência humana por meio do uso de computadores. No entanto a inteligência artificial não está intrinsecamente limitada a métodos biologicamente observáveis. Em [39], [40], os autores descrevem a inteligência artificial como "a ciência e a engenharia de produzir máquinas inteligentes". Isso permite que sistemas tomem decisões com base em dados e de forma independente. Com base na definição dos autores pode definir-se que a inteligência artificial é uma área que estuda e constrói entidades artificiais com habilidades cognitivas semelhantes às dos seres humanos.

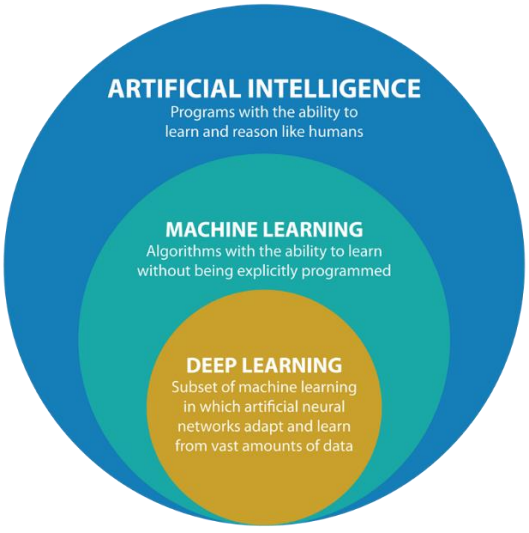


Figura 14-Campos da Inteligência Artificial [41]

O *ML*, é um subcampo da Inteligência Artificial (ver Figura 14) que se concentra na criação de algoritmos que usam a experiência em relação a uma classe de tarefas e *feedback* para melhorar a execução das tarefas usando métricas de desempenho [36].

Os cientistas de computação têm investigado desde a década de 1950 o domínio do *ML*. Nas últimas décadas, enormes esforços têm sido feitos nesta área de investigação, conduzindo a maiores expectativas. O *DL*, um subconjunto do *ML*, é uma tentativa nessa direção. O trabalho de *ML* é apresentado em muitas áreas novas e a aplicabilidade de áreas mais novas é sempre uma tarefa em curso na comunidade de investigação [42].

Em 1959, o termo *ML* surgiu pela primeira vez, descrevendo um sistema que dá aos computadores a capacidade de aprender algumas funções sem que sejam programados para tal fim. Traduzindo, alimentar um algoritmo com dados, para que a máquina aprenda a executar algo automaticamente. Já em 1964, o mundo conheceu o primeiro *chatbot*. Era a ELIZA, criado por Joseph Weizenbaum no MIT. Ela conversava de forma automática usando respostas baseadas em palavras-chave e estrutura semântica e sintática.

Em [43], os autores relatam como os tipos de algoritmos de *ML* são divididos, afirmam que há principalmente quatro categorias: aprendizagem supervisionada, aprendizagem não supervisionada, aprendizagem semi supervisionada e aprendizagem por reforço (ver Figura 15).

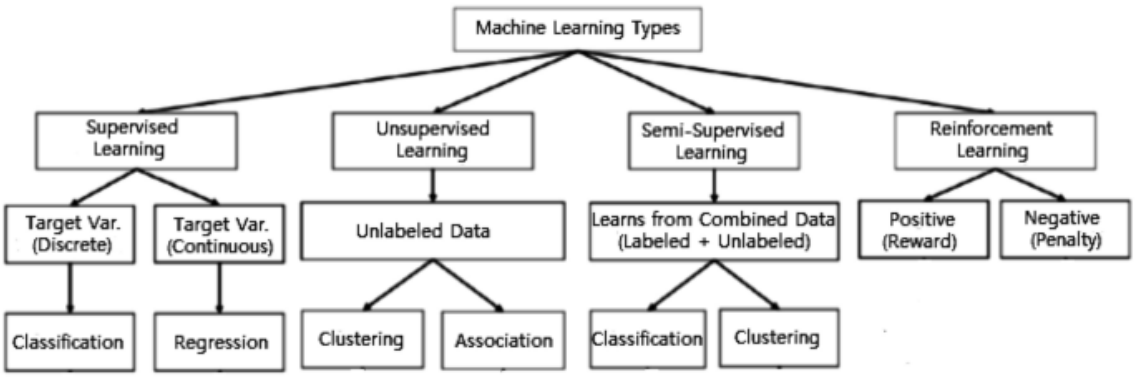


Figura 15- Tipos de Machine Learning [44]

Supervisionado: a aprendizagem supervisionada é normalmente a tarefa de *ML* para aprender uma função que mapeia uma entrada para uma saída com base em pares de entrada-saída de amostra [45]. Esta abordagem usa
dados de treino rotulados e uma coleção de exemplos de treino para inferir uma função. A aprendizagem supervisionada é realizada quando determinados objetivos são identificados para serem alcançados a partir de um determinado
conjunto de *inputs* [46], ou seja, uma abordagem orientada a tarefas. As tarefas supervisionadas mais comuns são “classificação”, que separa os dados, e “regressão”, que ajusta os dados. Por exemplo, prever o rótulo da classe ou o sentimento de um

trecho de texto, como um *tweet* ou uma avaliação de produto, ou seja, classificação de texto, é um exemplo de aprendizagem supervisionada (ver Figura 16).

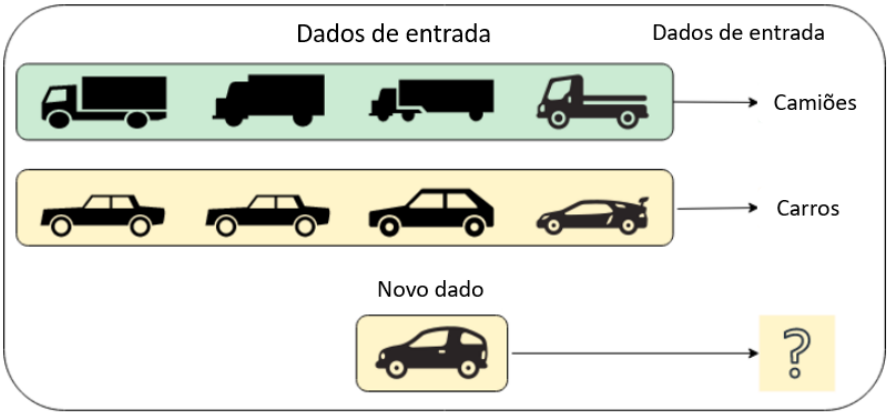


Figura 16 - ML Supervisionado

Não supervisionada: a aprendizagem não supervisionada analisa conjuntos de dados não rotulados sem a necessidade de interferência humana, ou seja, um processo baseado em dados [45]. Isto é amplamente utilizado para extrair características generativas, identificar tendências e estruturas significativas, fazer agrupamentos em resultados, e para fins exploratórios. As tarefas de aprendizagem não supervisionadas mais comuns são agrupamento, estimativa de densidade, aprendizagem de recursos, redução de dimensionalidade, localização de regras de associação, deteção de anomalias, etc (ver Figura 17).

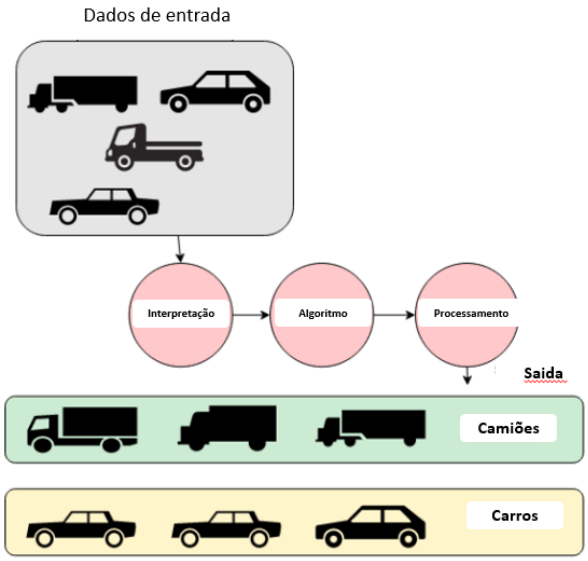


Figura 17-ML não supervisionado

Semi-supervisionada: A aprendizagem semi-supervisionada pode ser definida como uma hibridização dos métodos supervisionados e não supervisionados, mencionados acima, pois

opera em dados rotulados e não rotulados [45], [46]. Assim, situa-se entre aprender "sem supervisão" e aprender "com supervisão".

No mundo real, os dados rotulados podem ser raros em vários contextos, e os dados não rotulados são numerosos, onde a aprendizagem semi-supervisionada é útil [43]. O objetivo final de um modelo de aprendizagem semi-supervisionado é fornecer um resultado de previsão melhor do que aquele produzido usando apenas os dados rotulados do modelo. Algumas áreas de aplicação onde a aprendizagem semi-supervisionada é usada incluem tradução automática, deteção de fraudes, rotulagem de dados e classificação de texto.

Reforço: A aprendizagem por reforço é um tipo de algoritmo de *ML* que permite que agentes de software e máquinas avaliem automaticamente o comportamento ideal num contexto ou ambiente específico para melhorar a sua eficiência [47]. Ou seja, trata-se de uma abordagem orientada ao ambiente.

Este tipo de aprendizagem é baseado em recompensas ou penalidades, e o seu objetivo final é usar *insights* obtidos de agentes ou observadores ambientais para tomar medidas para aumentar a recompensa ou minimizar o risco [43]. É uma ferramenta poderosa para treinar modelos de IA que pode ajudar a aumentar a automação ou otimizar a eficiência operacional de sistemas sofisticados, como robótica, tarefas de direção autónoma, logística de fabricação e cadeia de fornecimento (ver Figura 18).

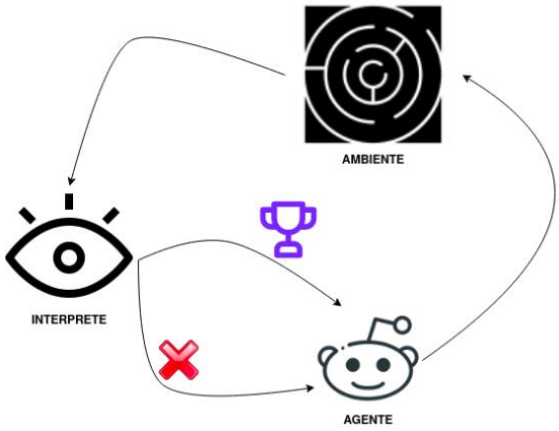


Figura 18- ML Reforço

Em [48] os autores afirmam que existem vários algoritmos de *ML* e, para avaliar o desempenho de cada um, existe um conjunto de fatores de avaliação que podem ser definidos, nomeadamente:

- **True Positive (TP):** Número de previsões que o modelo prevê corretamente a classe positiva;

- **True Negative (TN):** Número de previsões que o modelo prevê corretamente a classe negativa;
- **False positive (FP):** Número de previsões que o modelo prevê incorretamente a classe positiva;
- **False Negative (FN):** Número de prediz que o modelo prevê incorretamente a classe negativa.

Os autores em [49] abordam relatam que os fatores de avaliação podem ser usados por métricas de avaliação de desempenho ou métricas de classificação que podem ser definidas da seguinte forma:

- **Precision**: A proporção entre os verdadeiros positivos e a soma dos verdadeiros positivos e dos falsos positivos, conforme apresentado na Equação abaixo.

$$\text{Precision}=\frac{TP}{TP + FP}$$

- **Accuracy**: A ração da soma dos verdadeiros positivos e verdadeiros negativos para o número total de classificações, conforme apresentado na Equação abaixo.

$$\text{Accuracy}=\frac{TP+TN}{TP + TN + FP + FN}$$

- **Recall**: A proporção entre os verdadeiros positivos e a soma dos verdadeiros positivos e falsos negativos, conforme apresentado na equação abaixo.

$$\text{Recall} = \frac{TP}{TP + FN}$$

- **F1-score**: A média harmónica de precisão e recordação, tal como apresentada na equação abaixo.

$$\text{F1-score}= 2\times \frac{Precision\ x\ Recall}{Precision + Recall}$$

- **Confusion Matrix**: Distingue previsões verdadeiras positivas, falsas positivas, verdadeiras negativas e falsas negativas, conforme apresentado na Figura 19.

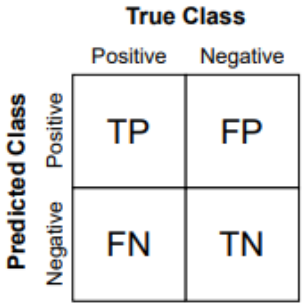


Figura 19 - Confusion Matrix

Os algoritmos de *ML* amplamente utilizados são:

- Classificador Linear;
- Regressão Logística;
- Naïve Bayes (NB);
- Rede Bayesiana;
- Máquinas de Vetores de Suporte (SVM);
- Árvore de Decisão;
- Random Florest;
- AdaBoost;
- Agregação Bootstrapped (Bagging);
- Rede Neuronal Artificial (RNA).

Há uma ampla gama de estruturas de *ML* de código aberto disponíveis no mercado, que permitem aos engenheiros de *ML* criar, implementar e manter sistemas de *ML*, gerar novos projetos e criar sistemas com impacto.

Tendo como premissa a definição sobre *ML* e suas vertentes, está na hora de saber o que fazer para começar e o que deve ser levado em consideração quando começar. Os algoritmos que geram modelos treináveis em *ML* não são tão inteligentes quanto a maioria das pessoas pensam. Eles precisam aprender e para aprender precisam de dados. Um modelo treinável precisa também de ser testado e avaliado para que possa ser viável.

O processo de preparação de dados, criação de modelos preditivos, avaliação, otimização e previsão é chamado de ***ML Workflow*** (ver Figura 40) e é a sequência primordial para o processo em questão.

Embora relevantes, as abordagens convencionais de *ML* apresentam limitações ao processar dados naturais em sua forma bruta. *DL* permite que modelos computacionais compostos de várias camadas de processamento aprendam representações de dados com vários níveis de abstração. Essas técnicas permitiram melhorarar significativamente diferentes tecnologias assim como a descoberta de medicamentos[50] [51].

Usando o algoritmo de retropropagação, a *DL* permite descobrir estruturas complexas em grandes conjuntos de dados. Isso mostra como uma máquina deve alterar os seus parâmetros

internos para calcular a representação em cada camada a partir da representação na camada anterior. As redes recorrentes melhoram dados sequenciais, como texto e fala, enquanto as redes convolucionais profundas melhoram o processamento de imagens, vídeo, fala e áudio [52].

No contexto deste trabalho, será demonstrado por que é que o *DL* é uma opção melhor para prever os níveis de insulina no sangue. Vários argumentos sólidos do estado de arte sustentam esta afirmação, e enfatizam a eficácia e a importância do DL na área médica[10], [11], [13], [14], [15], [16].

A capacidade do *DL* de detectar conexões complexas nos dados é um dos principais pontos a considerar. A dieta, a atividade física, os padrões de sono e a genética são algumas das muitas coisas que afetam a insulina no sangue. Existe a possibilidade de que modelos mais simples, como a regressão linear, não consigam reconhecer adequadamente essas interações complexas. Além disso, o *DL* pode aprender representações importantes de informações de entrada. Isso sugere que o modelo tem a capacidade de extrair automaticamente características pertinentes, como padrões temporais, sazonalidades e correlações entre variáveis. Isso torna a previsão mais precisa e informada.

3.3. LONG SHORT-TERM MEMORY (LSTM)

Na última década, houve resultados promissores na aplicação de técnicas, abordagens e métodos de *ML* para prever e diagnosticar diabetes, prever complicações de saúde causadas pela evolução de certos sintomas relacionados com a diabetes, avaliar antecedentes genéticos e fatores ambientais e mesmo ajudar a melhorar a saúde por meio de meios complementares de apoio a decisões na área de aprendizagem automático [53].

Os autores em [54], afirmam que os pacientes com diabetes tipo 1 têm uma tarefa difícil. Eles precisam de monitorizar continuamente os seus níveis de glicose no sangue (*Blood Glucose Level* - BGL) e ajustá-los quando os níveis ficam muito altos (hiperglicemia) ou muito baixos (hipoglicemia). Atingir e manter um controlo adequado da glicose (açúcar no sangue) é essencial para evitar complicações graves da doença. Os pacientes podem prever o BGL com 30 a 60 minutos de antecedência, prevenindo a hiperglicemia e a hipoglicemia, melhorando o controlo glicémico e, em certa medida, melhorando a saúde dos pacientes.

Aliberti et al. [55] comparam dois tipos de soluções que funcionaram bem para vários problemas de previsão de séries de tempo. Essas soluções eram baseadas em NAR e em redes de memória de curto prazo (LSTM). Os métodos baseados em redes neuronais de *feed-forward* (FNN), modelos autorregressivos (AR) e *redes neuronais recorrentes* (RNN) foram comparados experimentalmente com essas soluções. Embora o NAR tenha obtido uma boa precisão apenas para previsões de curto prazo, ou seja, com um horizonte de 30 minutos, o LSTM foi muito melhor em inferência de sinais de glicose a curto e longo prazo, superando todos os outros métodos em

termos de correlação entre o sinal de glicose medido e previsto, bem como em termos de resultados clínicos.

Na pesquisa de Bevan et al. [56], os autores usaram uma variedade de fontes de dados e períodos históricos para prever os níveis futuros de glicose no sangue. Essas classes de modelos incluem modelos lineares, redes neuronais *feed-forward*, redes neuronais recorrentes. Como resultado, foram encontrados resultados melhores com "durações de história relativamente curtas": A duração de 30 minutos demonstrou ser ideal para as suas experiências. Na mesma investigação, os autores descobriram que as redes LSTM eram melhores do que as redes neuronais lineares e de *feed forward*. Os autores também descobriram que a inclusão de um mecanismo de atenção no modelo LSTM melhorou ainda mais o desempenho, mesmo ao processar sequencias com comprimento relativamente curto. Observaram que os modelos treinados em nível individual superaram os modelos treinados com todos os dados disponíveis.

Em [57], Wang et al. avaliaram uma abordagem usando vinte casos do sistema de pâncreas artificial *open source* (OpenAPS), com 878.000 valores de glicose. A abordagem utilizou um modelo baseado em LSTM e permitiu uma previsão e precisão clássica melhoradas para os níveis de glicose no sangue a longo prazo. Os autores propuseram um método para prever os níveis diários de açúcar no sangue de pacientes com diabetes [58]. O método é treinado de ponta a ponta e usa uma rede neuronal recorrente. O único requisito é o histórico de glicose no sangue do paciente. O modelo não apenas fornece previsões, mas também fornece uma estimativa de exatidão, ajudando os utilizadores a entender melhor os seus níveis de previsão.

No entanto, devido à sua capacidade de modelar e prever a dinâmica de sistemas não lineares variantes no tempo, têm sido amplamente estudadas nos últimos anos, pois conseguem aprender dependências de longo prazo. Em 1997, Hochreiter e Schmidhuber introduziram este algoritmo. O LSTM é amplamente usado na indústria porque funciona bem em uma ampla gama de problemas sequenciais [59].

Em [60], os autores sugerem treinar um modelo de *ML* para prever níveis futuros de glicose com alta precisão usando uma coleção OhioT1DM e uma LSTM [11]. O conjunto de dados foi processado em três variáveis diferentes. O primeiro usou dados originais não processados, o segundo usou interpolação linear e o terceiro usou o método de séries temporais. Os conjuntos de dados coletados foram inseridos no modelo sugerido depois de serem divididos em horizontes de previsão de tempo de 5, 30 e 60 minutos. Dos três tipos diferentes de processamento de conjuntos de dados, o processamento de séries temporais foi o mais preciso. O processamento com interpolação linear foi o seguinte mais preciso.

Em [61], Lindemann et al. fornecem uma visão geral da abrangência dos derivados de células LSTM existentes, bem como estruturas de rede, para previsão de eventos temporais. Os autores apresentam um resumo das arquiteturas LSTM que foram criadas para prever o comportamento não linear das séries de tempo. Eles afirmam que as arquiteturas disponíveis têm sido usadas em

várias áreas de uso, como processamento de imagens, fabricação e sistemas autónomos. Os aspetos seguintes de LSTM foram destacados no estudo dos autores em [61]:

- LSTM com representações otimizadas do estado das células, tais como LSTM hierárquico que mostram uma capacidade melhorada de processar dados multidimensionais;
- LSTM com estados celulares interagindo, tais como *Grid* e LSTM cross-modal, são capazes de prever cooperativamente múltiplas quantidades com alta precisão;
- Seq2Seq LSTM pode prever várias quantidades em termos de previsão de vários passos à frente com propagação de erros minimizados;

De acordo com Lindemann et al., o Seq2Seq LSTM parcialmente condicionado demonstra a melhor adaptação para modelar dependências de curto e longo prazo. Em [62], é desenvolvido um estudo sobre algoritmos de rede neuronal artificial baseados no conjunto de dados de treino OhioT1DM com dados de 12 pessoas. O conjunto de dados contém informações como identificadores de indivíduos, dados de monitorização de glicose contínuos, obtidos a cada cinco minutos, taxa de infusão de insulina, etc. Seis modelos de *DL* individualizados foram desenvolvidos, tais como:

- LSTM (*Long Short-Term Memory*);
- BiLSTM (*Bidirectional LSTM);*
- CNN-LSTM (convolutional neuronal network LSTM);
- TCN (Redes Convolucionais Temporais/*Temporal Convolutional Networks*);
- Modelos Seq-to-Seq;
- *Transfer Learning*.

Os autores criaram modelos de aprendizagem de transferência que foram identificados por um algoritmo de aumento de gradiente tendo em conta as características mais significativas dos dados. Estes modelos foram testados usando um conjunto de dados de teste OhioT1DM, que continha seis dados individuais do sujeito. O modelo com os valores RMSE (*Root Mean Square Error*) mais baixos para os trinta e sessenta minutos foi escolhido para ter o melhor desempenho.

Em [62], os autores também explicam por que os dados de texto de entrada são estáticos e que toda a sequência pode ser usada simultaneamente. Eles também explicam por que se implementou um modelo BiLSTM para ver como o processamento da sequência em ambas a direção afeta a precisão. Por ser semelhante ao modelo LSTM, a estrutura do modelo BiLSTM é útil na previsão de séries temporais

Uma nova maneira de prever o nível de glicose no sangue foi abordada em [34]. Aqui, é levada em consideração a falha do sensor e é usado um modelo de LSTM, empilhado baseado na rede neuronal profunda. Para corrigir as leituras incorretas de monitoramento contínuo de glicose, causadas pelo erro do sensor, foi usado o método de suavização Kalman. O objetivo do método estudado pelos autores foi reduzir a diferença entre os valores previstos de CGM e o indicador

de leitura de glicose no sangue. Rabby et al. mostraram que a nova técnica sugerida pode melhorar a precisão da previsão da glicose no sangue processando as leituras de CGM usando a técnica de Kalman para correção de erros de sensor.

Em [63], usando um conjunto de dados de registo de data e hora de um paciente gerado aleatoriamente de um período típico de um paciente em convalescença hospitalar, os autores criaram um modelo de previsão do tempo de espera de uma emergência médica nas próximas duas horas usando redes neuronais recorrentes de LSTM e regressão linear. Os autores afirmam com esses dados que, em aproximadamente três minutos, o modelo de LSTM conseguiu reduzir o erro médio absoluto LSTM em 9,7% em comparação com o modelo de regressão linear (LR). Devido à sua capacidade de aprender relações entre passos de tempo consecutivos, tanto o modelo LR quanto o modelo LSTM são estatisticamente superiores. No entanto, ambos os modelos podem ser úteis na previsão do tempo de espera. O algoritmo LSTM, demonstrou ser mais eficaz em lidar com dados médicos irregulares do que outros modelos, de acordo com vários estudos [64].

Em [59], os autores empregam dados de UTI (Unidade de Tratamento Intensivo), que são extremamente complexos devido aos vários padrões de doenças.

Um LSTM convolucional (Conv-LSTM) foi usado para classificar pacientes com diabetes em um novo modelo de classificação [33]. O "algoritmo de Boruta" é usado para identificar pacientes diabéticos com alta precisão. Este algoritmo extrai dados específicos, como insulina, IMC, pressão arterial, idade e açúcar no sangue. Aqui, a otimização de alguns Hiper parâmetros é feita usando um "algoritmo de pesquisa em grid". Isso é feito para identificar os parâmetros ideais para o método que foi desenvolvido. Uma abordagem preditiva da diabetes usando métodos de *ML* foi proposta em [32]. Essa abordagem poderia ser usada em sistemas médicos simulados e sistemas clínicos automatizados. Como era muito caro, foi considerado um modelo ineficaz em termos de custo.

A tabela a seguir apresenta as comparações entre diferentes classificadores LSTM e estudados neste estado de arte:

Classificador	**Pontos fortes**	**Pontos fracos**
LSTM	Bom para capturar dependências de longo prazo.	Pode ser computacionalmente caro.
BiLSTM	Ainda melhor para capturar dependências de longo prazo.	Pode ser mais difícil de treinar.
CNN-LSTM	Pode lidar com dados espaciais e temporais.	Pode ser mais complexo de treinar.

TCN	Pode processar dados muito mais rápido que as LSTMs.	Pode ser menos preciso que as LSTMs.
Seq-to-Seq	Pode ser usado para uma variedade de tarefas, como tradução automática, resumo de texto e resposta a perguntas.	Pode ser mais difícil de treinar do que outros tipos de classificadores.
Transferência de aprendizagem	Pode melhorar significativamente o desempenho de um classificador em uma nova tarefa.	Requer uma grande quantidade de dados rotulados para treinar o classificador inicial.

Tabela 1-Comparação dos classificadores

Zhang et al., em [65], afirmam que as BiLSTM são "ainda melhores" para capturar dependências de longo prazo do que as LSTMs, os autores basearam-se no facto de que as BiLSTM podem processar informações em ambas as direções, do passado para o futuro e do futuro para o passado. Isso permite que o modelo aprenda padrões que podem ser difíceis de serem aprendidos por LSTMs convencionais.

Por exemplo, Xiaodan Zhu et al., em [65] explicam que as BiLSTMs foram usadas com êxito nos modelos para o reconhecimento da voz e tradução automática, onde o contexto é importante. Em ambos modelos, as BiLSTM foram capazes de melhorar o desempenho em comparação com as LSTMs convencionais.

Classificador	Eficácia	Precisão
LSTM	Médio	Alto
BiLSTM	Alto	Alto
CNN-LSTM	Alto	Alto
TCN	Alto	Médio
Seq-to-Seq	Alto	Alto
Transfer Learning	Alto	Alto

Tabela 2 - Comparação de eficácia vs Precisão

Para perceber o desempenho dos classificadores na Tabela 2, os autores em [66] definem a eficácia e a precisão como duas medidas importantes da performance de um modelo de *ML*. Os autores afirmam que eficácia e precisão são dois conceitos importantes em *ML*, mas que são diferentes. A eficácia é uma medida mais geral da capacidade de um modelo, enquanto a precisão é uma medida mais específica do desempenho do modelo para uma tarefa específica, também discutem como eficácia e precisão podem ser afetadas por uma série de fatores, incluindo o

tamanho do conjunto de dados de treino, a complexidade do modelo e a tarefa específica que está sendo realizada.

Uma comparação simplificada entre diferentes classificadores é apresentada na Tabela 2, focando principalmente a sua precisão na previsão de séries temporais. Os métodos que foram considerados estão listados na coluna Classificador. Por outro lado, as colunas "Precisão" e "Eficácia" mostram a capacidade de cada algoritmo em termos de previsão e classificação de séries temporais.

Valores médios ou baixos indicam um desempenho moderado ou insatisfatório para um algoritmo, enquanto um algoritmo com alta precisão é preferível. Mas a escolha do melhor algoritmo deve ser feita após uma análise cuidadosa que leve em consideração interpretação, escalabilidade e complexidade do modelo.

3.4. DISPOSITIVOS EXISTENTES PARA A MONITORIZAÇÃO CONTÍNUA DE NÍVEIS DE GLICOSE

Atualmente, existem diversas tecnologias de hardware e software que desempenham um papel crucial na prevenção de hipoglicemias. Exemplos são o dispositivo G6 da *Dexcom* (ver Figura 20) *e o Guardian Connect* da *Medtronic (*ver Figura 21), que operam em smartphones e estabelecem uma conexão com o CGM para a coleta de dados. Estas aplicações desempenham um papel vital ao utilizar os dados de glicose recolhidos para antecipar e prever variações nos níveis de glicose sanguínea, incluindo episódios de hipoglicemia ou hiperglicemia.

Estas soluções representam avanços significativos na gestão da diabetes, permitindo aos indivíduos diabéticos uma maior autonomia e controlo sobre a sua saúde. Além disso, a integração com smartphones torna a monitorização da glicose mais acessível e conveniente, o que pode contribuir para uma maior adesão ao tratamento. Esta evolução tecnológica tem o potencial de melhorar significativamente a qualidade de vida das pessoas com diabetes, reduzindo os riscos associados às variações nos níveis de glicose sanguínea e promovendo uma gestão mais eficaz da doença. Portanto, o estudo aprofundado e a análise destas tecnologias

desempenham um papel fundamental na investigação relacionada com a diabetes e na busca por soluções cada vez mais eficazes no tratamento da doença.

Figura 20 - Dexcom G6 device [67]

Figura 21 - Guardian Connect da Medtronic [68].

A Dexcom propõe um produto que assegura a monitorização consistente dos níveis de glicose, alertando com 20 minutos de antecedência para valores elevados ou baixos de glicemia, independentemente do tipo de diabetes. A sua patente faz menção a uma tecnologia que depende de variações temporais nos níveis de glicose. Por outro lado, a Medtronic utiliza o denominado sistema de prevenção da hipoglicemia baseado em CGM, o qual se apoia numa fórmula matemática previamente definida, considerando o histórico de valores de glicose, para monitorizar os dados do CGM e emitir alertas com uma antecedência que pode variar de 10 a até 60 minutos para valores elevados ou baixos de glicose.

Estas abordagens distintas revelam diferentes estratégias de prevenção e gestão da glicemia. A Dexcom enfatiza a importância da análise temporal da glicose, enquanto a Medtronic baseia-se em cálculos matemáticos para determinar o risco de ocorrência de hipoglicemia ou hiperglicemia. Esta diversidade de métodos demonstra a complexidade do campo da monitorização contínua de glicose e a constante busca por soluções mais eficazes e personalizadas para as necessidades individuais dos pacientes diabéticos. Como tal, a

investigação nesta área desempenha um papel crucial na evolução e melhoria das tecnologias disponíveis para o tratamento e gestão da diabetes [68] [69] [67] .

4. ANÁLISE ESTATÍSTICA E PRÉ-PROCESSAMENTO DO CONJUNTO DE DADOS

Para este projeto, o conjunto de dados *Pima Indians* Diabetes [70] foi usado como fonte de dados. É um conjunto de dados amplamente reconhecido e acessível no *Kaggle*[1]. O papel deste conjunto de dados (*dataset*), gentilmente oferecido pelo Instituto Nacional de Diabetes e Doenças Digestivas e Renais dos Estados Unidos, têm-se mostrado crucial em investigação médica e de saúde, sendo frequentemente usado para criar modelos de previsão que podem ajudar a identificar pacientes com risco de desenvolver diabetes. Este *dataset* é composto por vários parâmetros de entrada, como idade, nível de glicose, pressão arterial, insulina, índice de massa corporal (IMC) e outros fatores relacionados à saúde. O objetivo principal é usar esses dados para desenvolver modelos de *ML* capazes de prever a probabilidade de um paciente desenvolver diabetes [71].

Além da criação de modelos preditivos e da análise de dados, o *dataset* está diretamente relacionado à saúde e ao bem-estar das pessoas, oferecendo uma ferramenta útil para identificar rapidamente os riscos de diabetes. Isso pode resultar em intervenções mais eficazes, tratamentos mais rápidos e melhores condições de vida para esses pacientes.

Como resultado, o conjunto de dados de diabetes do *Pima Indians* é um recurso vital para investigadores e profissionais de saúde que trabalham com diabetes e medicina preventiva, pois ajuda a melhorar o entendimento da doença e a desenvolver estratégias de tratamento mais eficazes.

4.1. CARACTERÍSTICAS DO CONJUNTO DE DADOS

O conjunto de dados (ou *dataset*) contém 768 dados individuais com 9 atributos definidos. A descrição detalhada de todos os atributos é a seguinte:

- **Gestações**: indica o número de gestações;
- **Glicose**: indica a concentração de glicose plasmática (sanguínea);
- **Pressão arterial**: indica a pressão arterial diastólica em mm/Hg;
- **Espessura da pele**: indica a espessura da dobra cutânea do tríceps em mm;
- **Insulina**: indica a insulina em U/*ML*;
- **IMC**: indica o índice de massa corporal em kg/m2;
- **Diabetes Pedigree Function**: indica a função que pontua a probabilidade de diabetes com base no histórico familiar;
- **Idade**: indica a idade da pessoa;

[1] https://www.kaggle.com

- **Classe**: indica se o paciente tinha diabetes ou não (1 = sim, 0 = não).

Todas as variáveis são autoconhecidas ou estão disponíveis em um simples exame de sangue e a classe é uma espécie de "Resultado" e é o que se precisa para prever.

4.2 Análise Estatística do Conjunto de Dados

Inicialmente importa-se as bibliotecas de *python* que serão imprescindíveis para desenvolver o modelo preditivo, sendo elas de grande ajuda para manipular o conjunto de dados e extremamente importante para a visualização dos dados manipulados de forma gráfica (ver Figura 22).

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import tensorflow as tf
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import MinMaxScaler
from sklearn.preprocessing import StandardScaler
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
from tensorflow.keras.callbacks import EarlyStopping
```

Figura 22 - Importação das bibliotecas

Na Figura 23, carregou-se o conjunto de dados disponível para dar o início ao processo de previsão, usando a biblioteca pandas que nos permite carregar um conjunto de dados em diversos formatos transformando-os em um *dataframe*, no caso do *dataset* implementado está no formato *csv*. Inicialmente ela mostra apenas os 5 primeiros elementos quando não se define a quantidade de elementos por parâmetro. Assim procedeu-se a uma primeira análise do *dataframe* carregado:

```
df = pd.read_csv('/content/diabetes.csv')
df.head()
```

	Pregnancies	Glucose	BloodPressure	SkinThickness	Insulin	BMI	DiabetesPedigreeFunction	Age	Outcome
0	6	148	72	35	0	33.6	0.627	50	1
1	1	85	66	29	0	26.6	0.351	31	0
2	8	183	64	0	0	23.3	0.672	32	1
3	1	89	66	23	94	28.1	0.167	21	0
4	0	137	40	35	168	43.1	2.288	33	1

Figura 23 - Leitura do csv

A seguir, para colunas representadas numericamente, algumas métricas consideradas podem ser significativas (ver Figura 24).

```
df.describe()
```

	Pregnancies	Glucose	BloodPressure	SkinThickness	Insulin	BMI	DiabetesPedigreeFunction	Age	Outcome
count	768.000000	768.000000	768.000000	768.000000	768.000000	768.000000	768.000000	768.000000	768.000000
mean	3.845052	120.894531	69.105469	20.536458	79.799479	31.992578	0.471876	33.240885	0.348958
std	3.369578	31.972618	19.355807	15.952218	115.244002	7.884160	0.331329	11.760232	0.476951
min	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.078000	21.000000	0.000000
25%	1.000000	99.000000	62.000000	0.000000	0.000000	27.300000	0.243750	24.000000	0.000000
50%	3.000000	117.000000	72.000000	23.000000	30.500000	32.000000	0.372500	29.000000	0.000000
75%	6.000000	140.250000	80.000000	32.000000	127.250000	36.600000	0.626250	41.000000	1.000000
max	17.000000	199.000000	122.000000	99.000000	846.000000	67.100000	2.420000	81.000000	1.000000

Figura 24 – métricas

Observando as estatísticas médicas de 768 pacientes presentes no conjunto de dados, a média de 3.85 gravidezes indica que a maioria das pacientes teve pelo menos uma gravidez. Por outro lado, a mediana de 3 indica que esse é um valor importante, com algumas pacientes tendo até 17 gravidezes. No entanto, deve-se notar que a presença de valores mínimos de zero em várias colunas, como "Glucose", "Pressão sanguínea", "Densidade da pele", "Insulina", "BMI" e "Função do diabetes pediátrico" é preocupante, pois indica que há dados insuficientes ou incorretos. Antes da realização de qualquer análise adicional, esses valores devem ser abordados.

Os níveis de glicose médios de 120.89 indicam que os pacientes têm níveis moderados de glicose em média. No entanto, um valor mínimo de zero é incomum e pode ser resultado de dados incorretos. A variação significativa nos níveis de glicose de um paciente com 199 indica a necessidade de um exame mais aprofundado.

A pressão sanguínea média dos pacientes é de 69.11, com uma mediana de 72. Tal como para a glicose, os valores mínimos de zero são problemáticos e indicam que os dados são incorretos. A variação da pressão sanguínea, que atingiu 122, requer pesquisa.

Uma média de 31.99 na coluna "BMI" indica um IMC médio. A presença de valores mínimos de zero indica, mais uma vez, problemas nos dados. Além disso, um caso de um paciente com um IMC máximo de 67.10 chama a atenção para importância de rever a precisão desses números extremos.

A coluna "Idade" abrange pessoas de 21 a 81 anos, com uma média de 33.24 anos e uma mediana de 29. A distribuição etária diversificada é demonstrada pela ausência de pacientes com menos de 21 anos e pela presença de um paciente com 81 anos.

Finalmente, o "Resultado" mostra que cerca de 34.5% dos pacientes têm diabetes e logicamente sugere que 65.5% destes pacientes estão livres da doença. Como resultado, o conjunto de dados é desequilibrado em relação aos resultados da diabetes.

Para garantir a qualidade das informações e a confiabilidade das conclusões que podem ser obtidas a partir desse conjunto de dados, é essencial tratar os valores ausentes ou incorretos

antes de realizar análises mais aprofundadas. A Figura 25 contêm a visualização gráfica de resultados após tratamento do conjunto de dados fornecido originalmente.

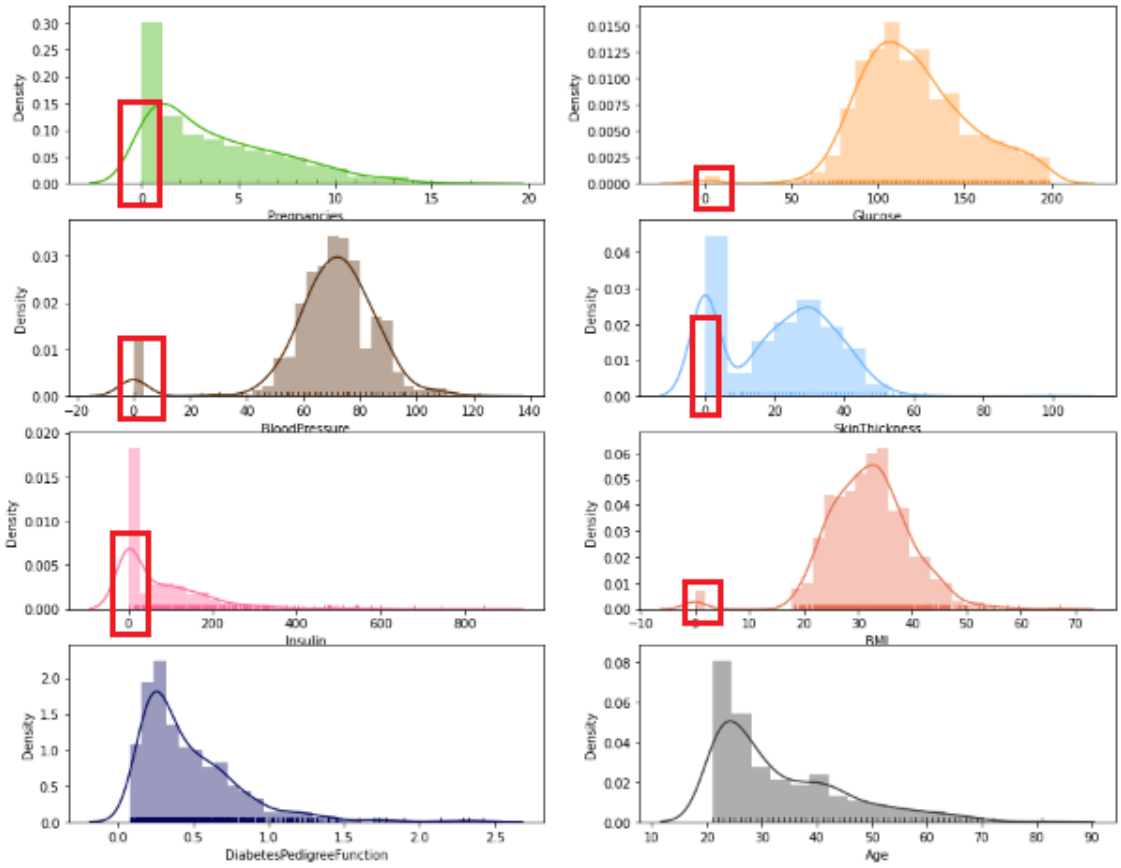


Figura 25 - Análise gráfica do conjunto de dados fornecido.

Este conjunto de dados contém valores zero como valores logicamente impossíveis, no caso da glicose, insulina, IMC e pressão arterial. É possível remover e ignorar esses valores inconsistentes ao limpar o conjunto de dados ou substituí-los por um intervalo de valores mais adequado. O conjunto de dados seria significativamente menor se as colunas "Espessura da pele" e "Níveis de insulina" fossem excluídas, pois há muitos zeros nessas colunas. Como resultado, para garantir que o tamanho do conjunto de dados permaneça inalterado para o estudo a ser realizado, se substituirá os valores zeros destas colunas pela média. Além disso, os fatores como "glicose", "pressão arterial" e "IMC" são bem atendidos pela técnica de "substituição pela média".

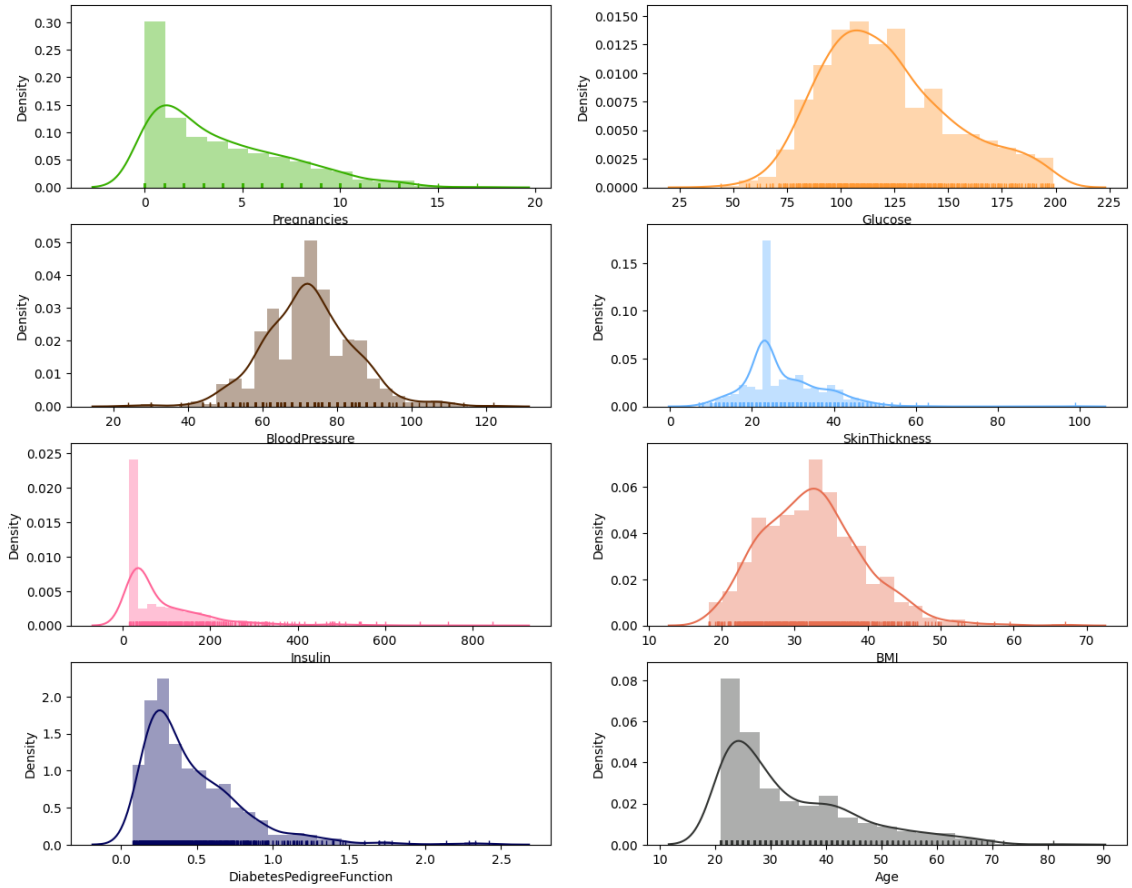


Figura 26 - Análise gráfica do conjunto de dados substituído pela mediana

Após substituir os valores zeros correspondente às colunas relativas à glicose, pressão sanguínea, IMC, espessura da pele e insulina por valores da mediana obtiveram-se os resultados da Figura 26.

4.2.1 Processo de Extração de Conhecimento

Depois de realizar a análise do conjunto de dados, passou-se para a visualização e testes semelhantes. Observa-se o número de pacientes que padecem ou não de diabetes:

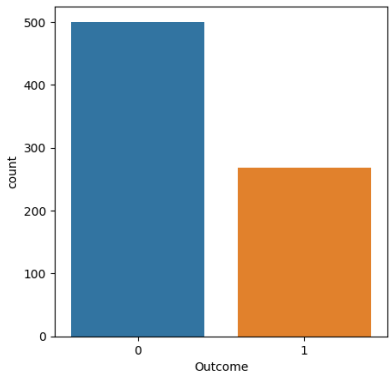


Figura 27 - Total de pacientes com diabetes e sem diabetes

Como pode se ver na Figura 27, o conjunto de dados apresenta dois terços de pacientes sem diabetes e um terço com diabetes. Este pormenor pode ter um impacto de parcialidade (*bias*) ao treinar o modelo pretendido.

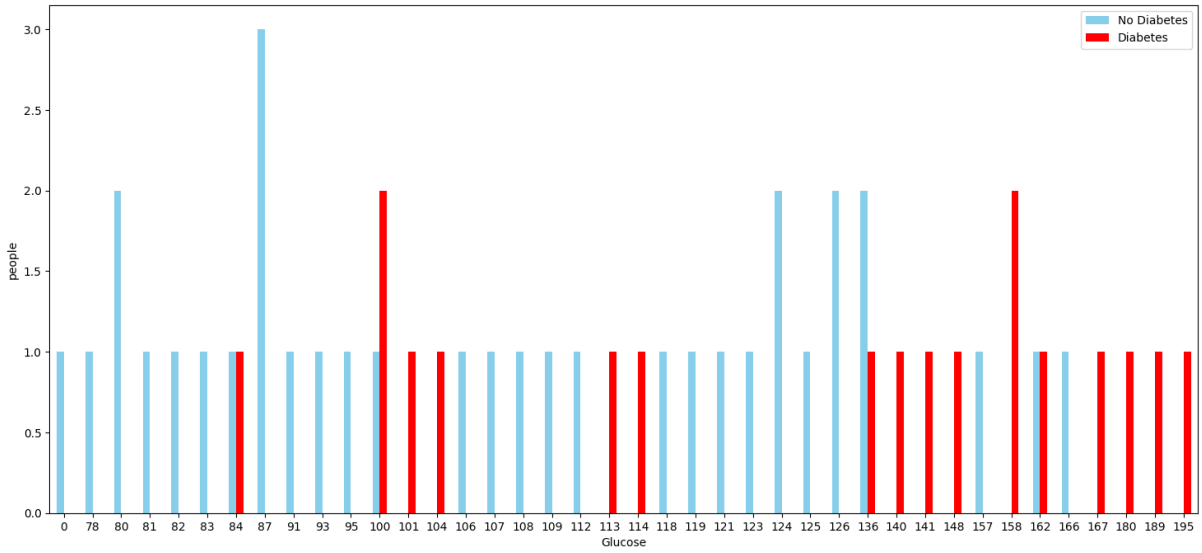


Figura 28 - Comparação da glicose entre os diabéticos e não diabéticos

Na Figura 28, gráfico de barras mostra uma comparação entre o diagnóstico de diabetes e os níveis de glicose, ou açúcar no sangue. É uma representação visual que facilita a compreensão da relação entre os níveis de glicose e a presença ou ausência de diabetes em um grupo de indivíduos.

Explorando o gráfico mais detalhadamente pode ver-se nitidamente que as barras azuis, que significa "No Diabetes" representam a quantidade de indivíduos com diagnóstico negativo, ou seja, sem diabetes e que estão em diferentes faixas de níveis de glicose. A barra azul mais alta em um intervalo de níveis de glicose indica que mais pessoas nesse intervalo não têm diabetes.

As barras vermelhas mostram o número de indivíduos com diagnóstico de diabetes em intervalos variados de níveis de glicose. Se uma barra vermelha for notavelmente alta dentro de um intervalo de níveis de glicose, isso indica que um diagnóstico de diabetes está mais relacionado a níveis de glicose mais altos.

Por fim, o gráfico mostra a relação entre os níveis de glicose no sangue e os diagnósticos de diabetes. Destaca os níveis de glicose em que a diabetes é mais comum, no *dataset*, e onde é menos comum. Isso é útil para a análise e compreensão das conexões entre essas variáveis em estudos médicos e de saúde.

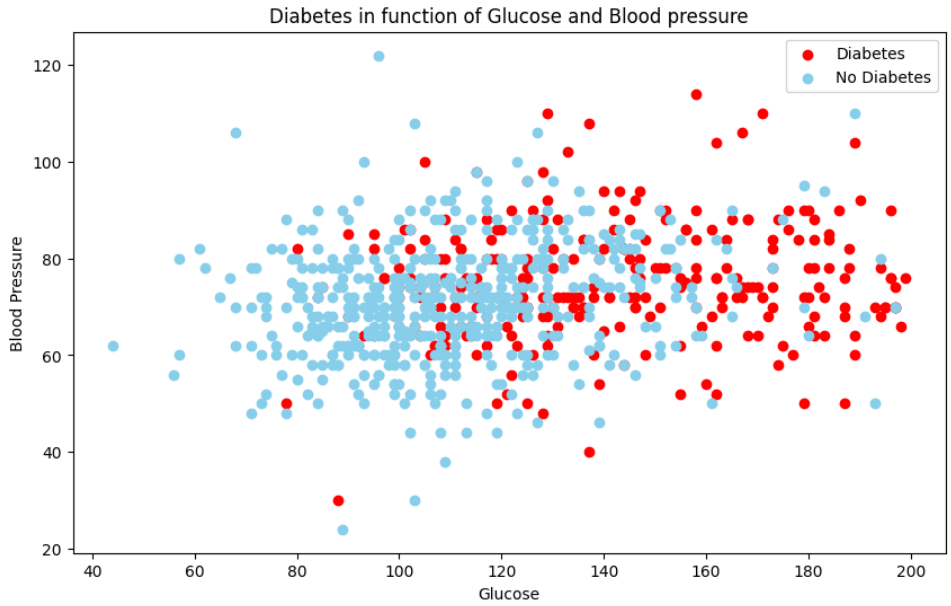


Figura 29 - Glicose e pressão

A Figura 29 mostra a relação entre os níveis de glicose e pressão sanguínea no conjunto de dados. É destacado o contraste entre pessoas com diabetes (em vermelho) e pessoas sem diabetes (em azul), relativamente ao seu nível de pressão arterial. A relação entre os níveis de glicose e pressão sanguínea entre as pessoas com diabetes é demonstrada pela posição dos pontos vermelhos no gráfico e os indivíduos sem diabetes têm pontos azuis. De forma semelhante, os pontos azuis no gráfico mostram a relação entre a pressão sanguínea e os níveis de glicose entre as pessoas sem diabetes. No gráfico, não há nenhum padrão evidente indicando uma relação linear entre os níveis de glicose e a pressão sanguínea em relação à presença de diabetes. Os pontos vermelhos e azuis se sobrepõem em várias áreas, isso indica que a relação entre glicose, pressão sanguínea e diabetes pode ser complexa, e outros fatores também podem afetar a presença de diabetes.

A Figura 30 mostra o gráfico de dispersão em pares que explora as relações entre três variáveis numéricas do conjunto de dados. A variável "*Outcome*" neste caso, indica se o indivíduo tem ou não diabetes, e determina as cores dos pontos no gráfico. Como resultado, o gráfico mostra como as variáveis numéricas se relacionam entre si e se há alguma diferença na distribuição dessas variáveis em relação à presença ou ausência de diabetes. A visualização é útil para encontrar padrões, correlações e tendências entre as variáveis e entender como elas podem estar relacionadas à diabetes.

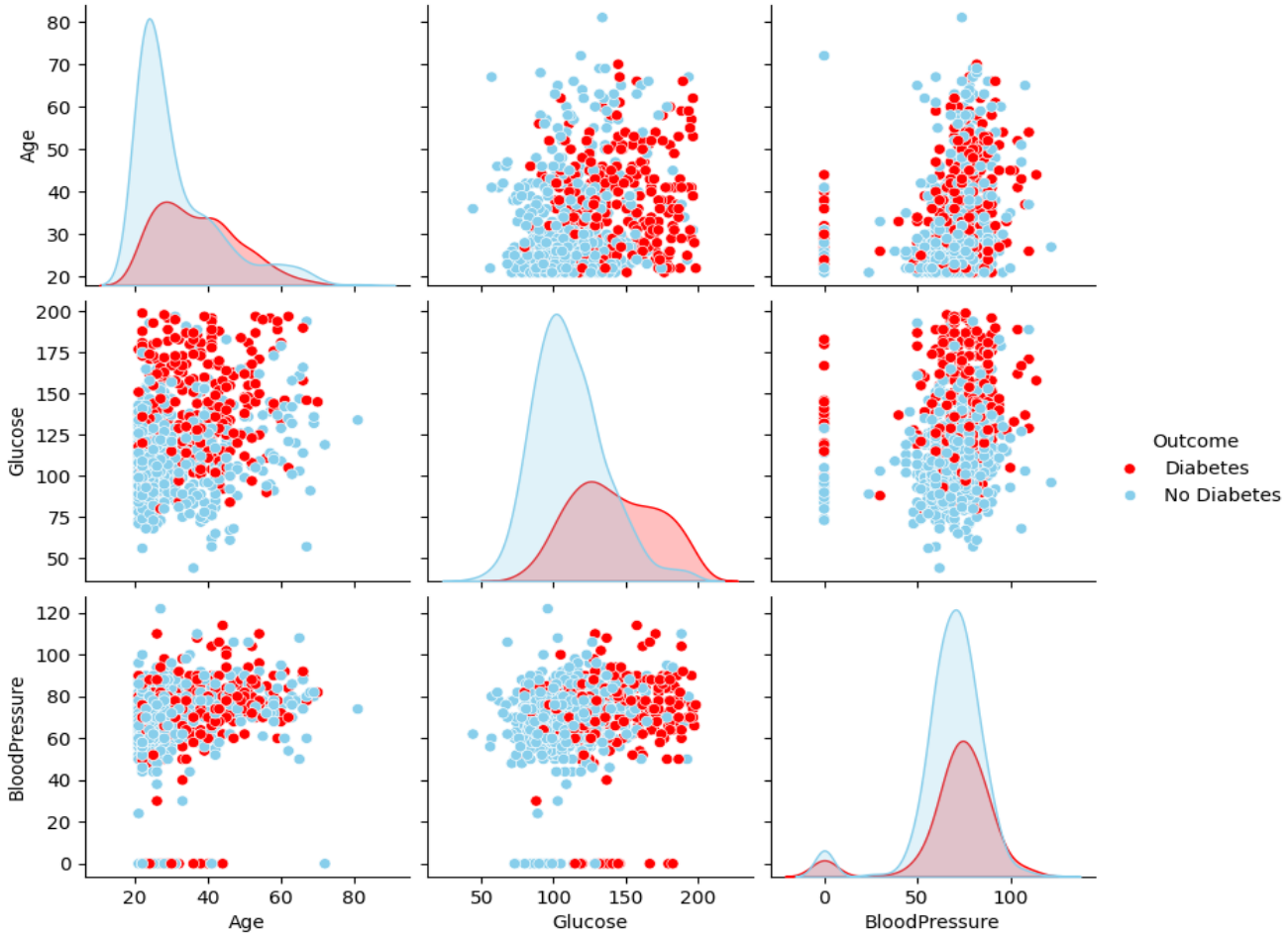


Figura 30 - Relação múltiplas variáveis

Na Figura 31 podemos ver a correlação entre todas as variáveis por exemplo, existe uma correlação negativa fraca (aproximadamente -0.13) entre gravidez e glicose. Com isto pode se dizer que mulheres com maior número de gestações podem ter níveis de glicose ligeiramente mais baixos. Se analisarmos a glicose e a idade, observa-se uma correlação positiva fraca (aproximadamente 0.26). Isso significa que, à medida que a idade aumenta, os níveis de glicose tendem a aumentar ligeiramente.

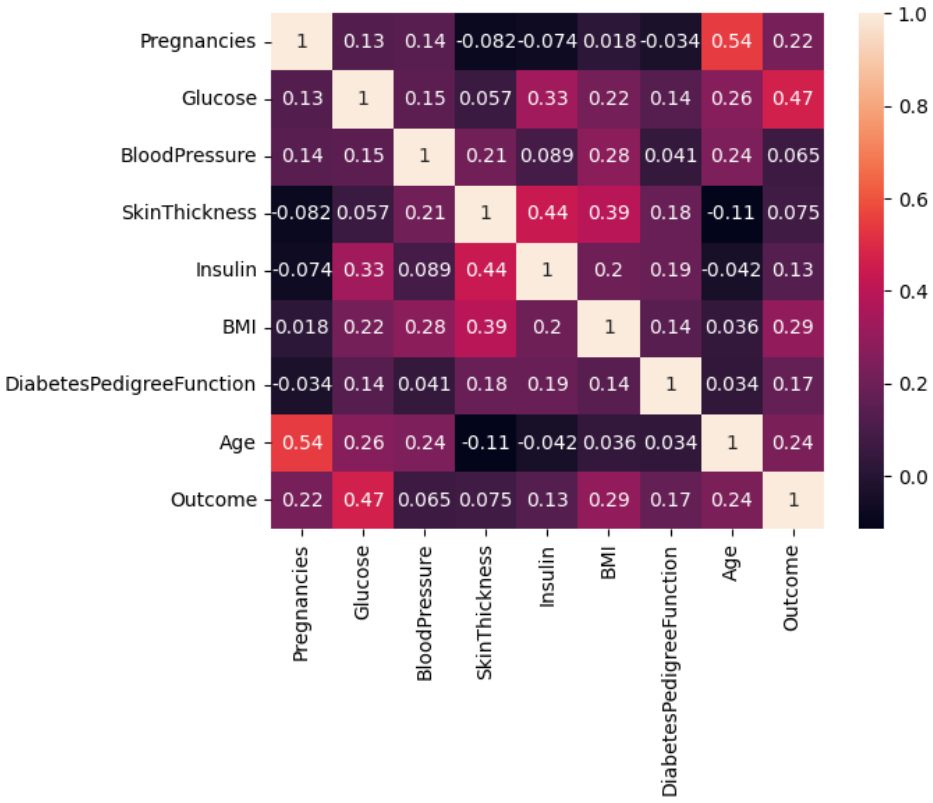


Figura 31 - Correlação entre as variaveis

Na Figura 32 faz-se a análise visual de dados é um componente crítico da pesquisa científica, pois permite uma melhor compreensão das tendências e relações presentes nos dados. No contexto deste trabalho, examinou-se um conjunto de dados sobre diabetes com foco na variável "*Outcome*", onde o valor de 1 indica casos de diabetes positivos. Utilizou-se gráficos de histograma para examinar a distribuição de várias variáveis em casos de diabetes para uma análise mais profunda.

As interpretações dos gráficos de histograma para as variáveis essenciais para os casos de diabetes estão a seguir. Essas interpretações mostrarão como as variáveis individuais se relacionam com a diabetes e podem ajudar a construir uma compreensão mais abrangente dos fatores de risco relacionados à doença.

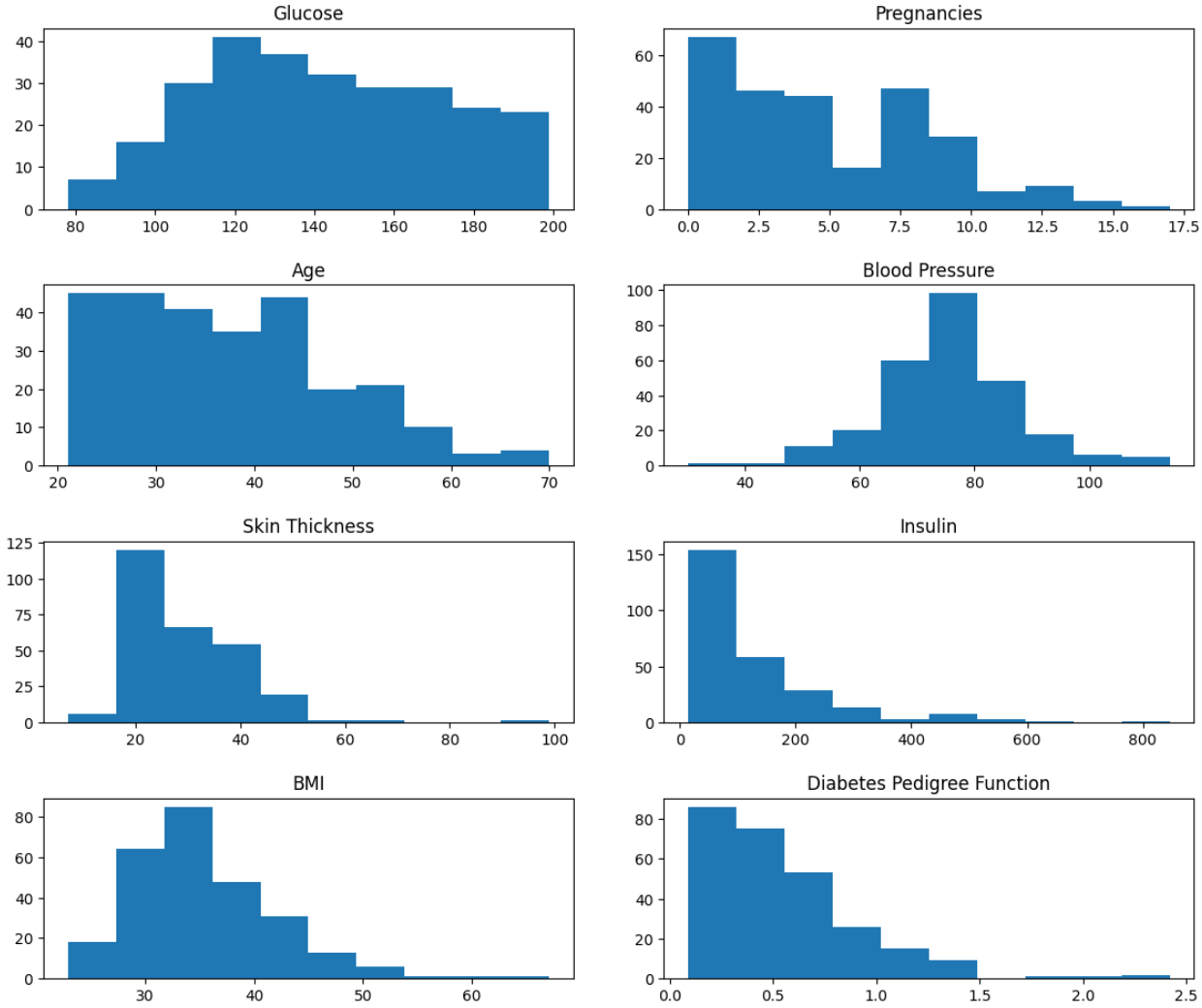


Figura 32 - Histograma outcome = 1

A distribuição dos níveis de glicose nos casos de diabetes indica que a maioria dos pacientes com diabetes tem níveis de glicose concentrados em um valor específico. Isso indica que, como a maioria dos casos apresenta valores mais elevados, a glicose desempenha um papel significativo no desenvolvimento da diabetes. Neste caso a diabetes não está fortemente correlacionada com o número de gravidezes em pacientes diagnosticados com a enfermidade. O histórico de gravidezes pode não ser um fator de risco relevante para diabetes, pois a maioria dos casos ocorre em mulheres com diferentes números de gravidez.

Os pacientes com diabetes podem ser encontrados em uma variedade de idades, mas a maioria dos casos ocorre em adultos mais jovens a meia-idade. Isso sugere que a diabetes não é apenas uma condição para pessoas mais velhas. Continuando a análise nota-se que a pressão sanguínea associada à diabetes não tem uma tendência clara, e os valores variam muito. Isso indica que a pressão sanguínea por si só pode não ser um bom indicador de diabetes.

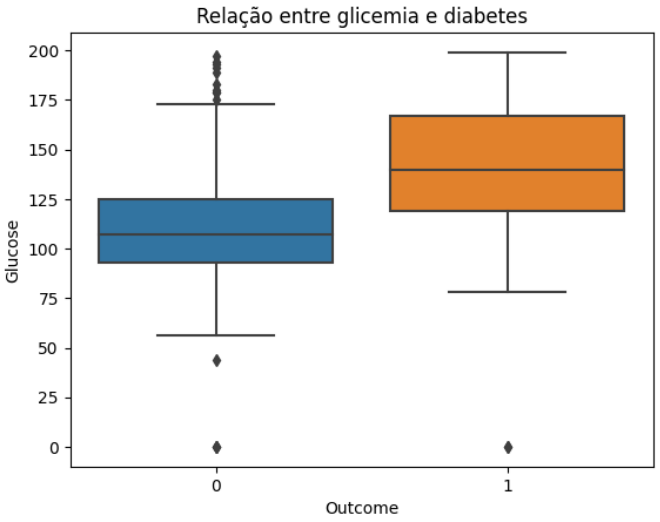


Figura 33 - Relação entre glicemia e diabetes

A média da glicose é mostrada pela linha no meio da caixa na Figura 33. O valor médio dos níveis de glicose é representado pela mediana. Se há uma diferença média significativa nos níveis de glicose entre pacientes com e sem diabetes, pode-se verificar comparando as medianas dos dois grupos (*Outcome=0 e Outcome=1*). A mediana mais alta no grupo com diabetes indica que os pacientes com diabetes têm níveis de glicose mais altos.

O primeiro quartil, que é o limite inferior da caixa, e o terceiro quartil, são separados pelo intervalo interquartil. Ele mostra como os dados e as variações nos níveis de glicose em cada grupo diferiram. Os valores que não estão nas "linhas de bigode", que são aproximadamente 1,5 vezes o intervalo interquartil. Esses valores podem indicar alguns valores atípicos.

4.3 Pré-Processamento do Conjunto de Dados

Com base no conjunto de dados inicial, foram analisadas algumas incongruências nos dados, nomeadamente, dados omissos, dados dispersos. Foi feito o pré-processamento da seguinte forma:

Para uma melhor análise faz-se uma leitura para saber os tipos de dados de cada coluna do conjunto de dados e se, por alguma eventualidade, existem valores nulos (ver Figura 34):

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 768 entries, 0 to 767
Data columns (total 9 columns):
 #   Column                    Non-Null Count  Dtype
---  ------                    --------------  -----
 0   Pregnancies               768 non-null    int64
 1   Glucose                   768 non-null    int64
 2   BloodPressure             768 non-null    int64
 3   SkinThickness             768 non-null    int64
 4   Insulin                   768 non-null    int64
 5   BMI                       768 non-null    float64
 6   DiabetesPedigreeFunction  768 non-null    float64
 7   Age                       768 non-null    int64
 8   Outcome                   768 non-null    int64
dtypes: float64(2), int64(7)
memory usage: 54.1 KB
```

Figura 34 - Tipo de dados

Explorando mais a fundo as funcionalidades do *pandas*, usou-se o método *isnull ()* junto do método *sum()* para saber de facto quais colunas apresentam valores nulos (ver Figura 35):

```
# Verificando valores nulos
print(df.isnull().sum())
```

```
Pregnancies                 0
Glucose                     0
BloodPressure               0
SkinThickness               0
Insulin                     0
BMI                         0
DiabetesPedigreeFunction    0
Age                         0
Outcome                     0
dtype: int64
```

Figura 35-Valores nulos

Após a execução da funcionalidade, nota-se que nas colunas presentes no conjunto de dados não existe sequer um valor nulo que podem afetar o modelo que se pretende treinar.

Na Figura 36 pode se observar que os valores mínimos da *Glucose, BloodPressure, BMI, SkinThickness, Insulin* são dados como zero, normalmente estes valores não têm valores zero , portanto, um valor de zero nessas colunas pode indicar dados ausentes ou errados [72].

Os níveis de glicose e insulina poderiam teoricamente ser zero, mas é essencial considerar o contexto e as unidades de medida usadas no conjunto de dados [73]. Para estes casos pode se utilizar a técnica de substituição da média ou mediana dos valores zero [74]

```
mediana = df.median()

colunas_zero_incorretos = ['Glucose', 'BloodPressure', 'BMI','SkinThickness','Insulin']
df[colunas_zero_incorretos] = df[colunas_zero_incorretos].replace(0, mediana)
df.describe()
```

	Pregnancies	Glucose	BloodPressure	SkinThickness	Insulin	BMI	DiabetesPedigreeFunction	Age	Outcome
count	768.000000	768.000000	768.000000	768.000000	768.000000	768.000000	768.000000	768.000000	768.000000
mean	3.845052	121.656250	72.386719	27.334635	94.652344	32.450911	0.471876	33.240885	0.348958
std	3.369578	30.438286	12.096642	9.229014	105.547598	6.875366	0.331329	11.760232	0.476951
min	0.000000	44.000000	24.000000	7.000000	14.000000	18.200000	0.078000	21.000000	0.000000
25%	1.000000	99.750000	64.000000	23.000000	30.500000	27.500000	0.243750	24.000000	0.000000
50%	3.000000	117.000000	72.000000	23.000000	31.250000	32.000000	0.372500	29.000000	0.000000
75%	6.000000	140.250000	80.000000	32.000000	127.250000	36.600000	0.626250	41.000000	1.000000
max	17.000000	199.000000	122.000000	99.000000	846.000000	67.100000	2.420000	81.000000	1.000000

Figura 36 - Substituição pela mediana

4.4. CONJUNTO DE DADOS FINAL

Após a realização da análise estatística do conjunto de dados inicial e do pré-processamento, o *dataset* inicial tinha 767 registos, não foram encontrados dados omissos, entretanto encontrou-se incongruência nas colunas *Glucose, BloodPressure, BMI, SkinThickness, Insulin* com valores zero que para o caso de estudos não representam valores substanciais [74], para esta incongruência conforme descrito na seção 4.3 substitui-se os valores das colunas pelo valor da mediana de cada coluna. Na Figura 37 pode-se observar que as colunas citadas não apresentam mais valores incongruentes pois foram substituídos com o valor da mediana como descrevem os autores em [74].

```
df
```

	Pregnancies	Glucose	BloodPressure	SkinThickness	Insulin	BMI	DiabetesPedigreeFunction	Age	Outcome
0	6	148	72	35	30.5	33.6	0.627	50	1
1	1	85	66	29	30.5	26.6	0.351	31	0
2	8	183	64	23	30.5	23.3	0.672	32	1
3	1	89	66	23	94.0	28.1	0.167	21	0
4	0	137	40	35	168.0	43.1	2.288	33	1
...	...	...	...	...	...	...	...	...	...
763	10	101	76	48	180.0	32.9	0.171	63	0
764	2	122	70	27	30.5	36.8	0.340	27	0
765	5	121	72	23	112.0	26.2	0.245	30	0
766	1	126	60	23	30.5	30.1	0.349	47	1
767	1	93	70	31	30.5	30.4	0.315	23	0

Figura 37 - Dataset Final

5. TREINO E ANÁLISE DE DESEMPENHO DAS LSTM

Para poder fazer comparações entre vários modelos avançados, ao final da aplicação, será calculada o *accuracy, recall* e a precisão de cada modelo. Para dar solução ao *pipeline*, recorreu-se à linguagem de programação *python*, fazendo a interpretação das *scripts no jupyter*.

5.1 CONFIGURAÇÃO COMPUTACIONAL

Para a realização deste trabalho e a criação dos modelos, foi utilizado uma estação de trabalho com CPU Intel(R) Core (TM) i7-6500U CPU @ 2.50GHz 2.59 GHz, RAM instalada de 8,00 GB, com GPU Intel(R) HD *Graphics* 520 com 3.9 GB. A nível de ferramentas e tecnologias foi utilizado o python versão 3.11.0, o flask versão 2.2.3 (ver Figura 38) .

```
(LSTM_ENV) PS C:\Users\Dannyel Francisco\Documents\IPVC\Tese de Mestrado\LSTM> flask --version
Python 3.11.0
Flask 2.2.3
```

Figura 38 - Versões das ferramentas e tecnologias

5.2 PROCESSO E MÉTRICAS USADAS NO TREINO

Na Figura 39 pode-se observar a implementação de modelos de *ML*, nomeadamente redes neuronais, estes modelos requerem a consideração de vários elementos. A execução de modelos envolve a definição da função de perda, optimizador e métricas de avaliação. A função de perda 'binary_crossentropy' é usada para problemas de classificação binária, enquanto o otimizador 'adam' ajusta os pesos durante o treino. Acompanhar o desempenho do modelo é essencial, usando métricas como precisão e perda na validação.

Ver: Notebook com o código python do treino dos modelos

```python
import time
import pandas as pd
import tensorflow as tf
from sklearn.metrics import f1_score, precision_score, recall_score
# Define nomes para os modelos
model_names = [
    'Simple LSTM',
    'With 2 layers',
    'Bidirectional',
    'With Dropout',
    'L2 regularization'
]
# Create an empty DataFrame to store the results
results_df = pd.DataFrame(columns=['Model Name', 'Loss', 'Accuracy', 'F1 Score', 'Precision', 'Recall', 'Training Time'])

# Train and evaluate models
for i, model in enumerate(models):
    print(f"Model: {model_names[i]}")
    model.compile(loss='binary_crossentropy', optimizer='adam', metrics=['accuracy'])
    early_stop = tf.keras.callbacks.EarlyStopping(monitor='val_loss', patience=10)
    start_time = time.time()  # Record the start time
    history = model.fit(X_train, y_train, epochs=50, batch_size=10, callbacks=[early_stop], validation_data=(X_test, y_test))
    end_time = time.time()  # Record the end time
    training_time = end_time - start_time  # Calculate the training time
    loss, accuracy = model.evaluate(X_test, y_test)
    y_pred = model.predict(X_test).round()
    f1 = f1_score(y_test, y_pred)
    precision = precision_score(y_test, y_pred)
    recall = recall_score(y_test, y_pred)
    print(f"- Loss: {loss:.2f}")
    print(f"- Accuracy: {accuracy:.2f}")
    print(f"- F1 Score: {f1:.2f}")
    print(f"- Precision: {precision:.2f}")
    print(f"- Recall: {recall:.2f}")
    print(f"- Training Time: {training_time:.2f} seconds")
    # Append the results to the DataFrame
    results_df = results_df.append({
        'Model Name': model_names[i],
        'Loss': loss,
        'Accuracy': accuracy,
        'F1 Score': f1,
        'Precision': precision,
        'Recall': recall,
        'Training Time': training_time
    }, ignore_index=True)

    # Save the trained model
    model.save(f'{model_names[i]}_model.h5')

# Display the results DataFrame
print(results_df)
```

Figura 39- Código python do treino

A técnica de *early stopping* (ver Figura 39) é implementada para evitar o *overfitting, ou seja*, interrompe o treino se a perda na validação deixar de melhorar após um determinado número de *epochs*. Isso contribui para a generalização efetiva do modelo. O treino em si é realizado ao longo de um determinado número de *epochs*, A decisão de utilizar 50 *epochs* nesta investigação é baseada na observação do desempenho do modelo evitando treino excessivo que poderia levar a ter um *overfitting*, no entanto nas fases de testes experimentou-se: 30, 70 e 100 *epochs*, neste processo observou-se as métricas como perda (*loss*), F1 Score, precisão e recall sendo para todos os casos sem variações substanciais. Além disso, a análise do tempo de treino é adicionada ao

modelo para fornecer informações sobre o tempo necessário para cada modelo durante as 50 épocas.

Model	Loss	Accuracy	F1 Score	Recall	Training Time
Simple LSTM	0.43	0.80	0.64	0.57	3.77
With 2 layers	0.42	0.79	0.64	0.60	6.7
Bidirectional	**0.42**	**0.82**	0.66	0.57	9.6
With dropout	0.48	0.77	0.42	0.28	4.3
L2 regularization	0.46	0.80	0.66	0.64	4.16

Tabela 3 - Tempo de treino dos modelos

Os resultados dos modelos LSTM mostram variações significativas em termos de precisão e tempo de treino (ver Tabela 3). O modelo bidirecional LSTM obteve a maior precisão, atingindo 82%, indicando um desempenho superior na hora de treinar em comparação com os outros modelos considerados. No entanto, é importante notar que este aumento na precisão foi acompanhado por um tempo substancialmente maior, totalizando 9.6 segundos para o *dataset* utilizado.

Por outro lado, o modelo LSTM simples e o LSTM com regularização L2 apresentaram desempenho notável em precisão, atingindo 80%, mantendo tempos de treino mais moderados, com 3.77 e 4.16 segundos, respetivamente. Esses modelos fornecem um equilíbrio eficiente entre precisão e tempo de treino, para casos em que a eficiência computacional é uma consideração importante, estes modelos são opções para considerar. O LSTM de duas camadas, embora mantendo uma precisão aceitável de 79%, aumentou consideravelmente o tempo de treino para 6.7 segundos. Por outro lado, o modelo LSTM dropout, teve uma precisão de 77% e um tempo de treino de 4.3 segundos.

5.3 Treino dos modelos usando as LSTM

Antes de começar a fase de treino, é necessário reconhecer a fonte de dados, que são os valores de glicose, que são dados exclusivos de dados históricos ou anteriores. Usando esse tipo de informação, pode-se usar uma rede neuronal, que permite fazer previsões sobre o que pode acontecer no futuro com base em dados anteriores. As redes neuronais recorrentes (RNN) podem resolver esse problema porque, além de lembrar e esquecer informações, elas também podem usar valores passados para prever futuros. As novas circunstâncias podem influenciar os padrões de comportamento que foram estudados por semanas.

Uma vez selecionada uma RNN por suas características, dentro de RNN têm-se tipos especiais de redes neuronais recorrentes:

- Redes de memória longa de curto prazo - *Long Short Term Memory* (LSTM).
- Rede *Gated Recurrent Unit* (GRU).

Para este trabalho após a análise exaustiva do estado da arte, a eleição do LSTM foi a melhor escolha para treinar os modelos.

5.3.1 LSTM UTILIZADAS PARA O TREINO

As redes neuronais, incluindo as *Long Short-Term Memory* (LSTM), têm sido usadas no campo da saúde, incluindo o tratamento da diabetes, para prever níveis de glicose no sangue, identificar padrões e auxiliar no tratamento.

No contexto do tratamento da diabetes, a escolha dos tipos de LSTM a serem utilizados desempenha um papel fundamental na busca pela precisão e eficácia dos modelos de previsão. A diabetes é uma condição médica altamente complexa, caracterizada por flutuações nos níveis de glicose no sangue, que podem ser influenciadas por diversos fatores, como dieta, exercício, medições históricas e até mesmo condições médicas subjacentes.

Diante dessa complexidade, é crucial adotar abordagens de modelação igualmente sofisticadas. Na tarefa de previsão dos níveis de glicose no sangue no tratamento da diabetes, destacam-se cinco tipos de memória longa e curta (LSTM). Isso é baseado no estado da arte da pesquisa e na eficácia comprovada dessas estruturas em questões semelhantes.

As cinco variantes de LSTM mencionadas - LSTM Simples, LSTM com 2 Camadas, LSTM Bidirecional, LSTM com *Dropout* e Regularização L2 - emergem como escolhas comuns e preferidas. Isso se deve a uma série de razões com base no estado da arte e nas necessidades deste campo específico:

- **LSTM Simples;**

Devido à sua eficácia e simplicidade, a LSTM Simples é uma opção inicial popular. Esta variante de LSTM é essencial para a previsão dos níveis de glicose no sangue em doentes diabéticos porque é capaz de aprender dependências temporais ao longo de séries temporais. Como os níveis de glicose são muito influenciados pelo tempo, sua capacidade de capturar padrões sequenciais é especialmente importante. Além disso, a LSTM Simples é uma opção fácil de entender e de interpretação, o que ajuda pacientes e profissionais de saúde a compreender as previsões.

- **LSTM com 2 Camadas;**

Uma opção estratégica é a inclusão de duas camadas LSTM. Os níveis de glicose são afetados por várias variáveis interconectadas, pois o diabetes é uma doença multifatorial. O modelo pode aprender representações de dados mais complexas usando duas camadas, capturando relações hierárquicas e nuances que uma única camada poderia não conseguir. Como resultado, o modelo

é mais capaz de lidar com as múltiplas interações que ocorrem no corpo de um indivíduo que está sofrendo de diabetes.

- **LSTM bidirecionais;**

As LSTMs bidirecionais se destacam no contexto da diabetes. Estas LSTMs processam sequências tanto no sentido temporal quanto no inverso, o que lhes permite capturar informações contextuais importantes de medições passadas e futuras. Os níveis de glicose no sangue dependem muito de padrões temporais, então este é um ponto importante para a previsão de tendências. Para antecipar mudanças nos níveis de glicose que evoluem ao longo do tempo, a abordagem bidirecional é particularmente útil.

- **LSTM com Dropout;**

Uma estratégia crucial para melhorar a capacidade de generalização do modelo é a adição de *dropout*. O *dropout* reduz o risco de o modelo se ajustar excessivamente aos dados de treino desativando unidades da rede aleatoriamente durante o treino. Isso é muito importante para a gestão da diabetes porque os dados de medição da glicose são ruidosos e variáveis, e o modelo precisa fazer previsões confiáveis para fazer escolhas de tratamento adequadas.

- **Regularização L2;**

A regularização L2 é uma maneira útil de evitar criar modelos muito complicados. A simplicidade do modelo pode ser útil no tratamento da diabetes porque ajuda os médicos e os pacientes a entender as previsões. O modelo é útil na prática clínica porque a regularização L2 ajuda a equilibrar precisão e complexidade. Ela é essencial para a criação de modelos que não desviam de dados importantes e evitam suposições infundadas.

5.3.1.1 Arquitetura do modelo de treino

A Figura 40 apresenta, de forma geral, o pipeline do processo de treino do LSTM desde a recolha de dados, o processamento até ao uso do modelo treinado. Após a definição dos tipos de LSTM para o presente trabalho passa-se então a seleção de características e análise para o processo de treino dos modelos.

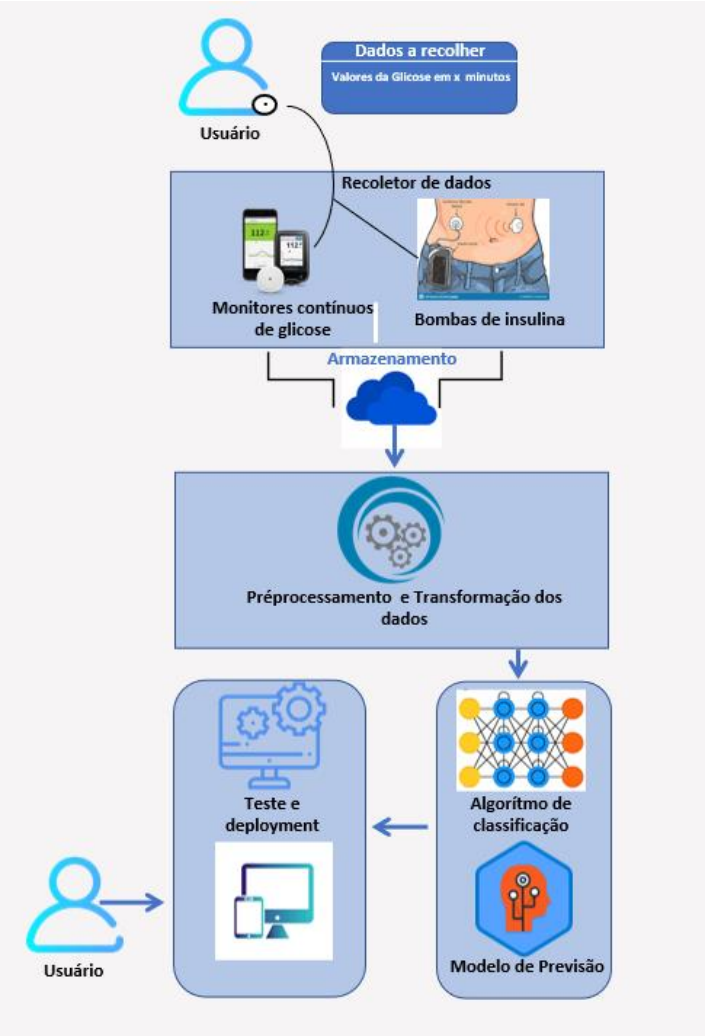


Figura 40 Arquitetura do modelo de treino

```
# Load diabetes dataset
df = pd.read_csv('/content/diabetes.csv')

# Split into training and test sets
X = df.drop('Outcome', axis=1).values
y = df['Outcome'].values
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=0)
```

Figura 41 - Carregamento e divisão dos dados

No código da Figura 41, faz-se a leitura do conjunto de dados relacionado com a diabetes, que contém informações como níveis de glicose, pressão arterial e outros fatores relevantes. O conjunto de dados é carregado a partir de um ficheiro CSV denominado 'diabetes.csv'. Em seguida, dividimos os dados em dois conjuntos: treino e teste. Os dados são divididos para permitir que o modelo seja treinado em uma parte dos dados e, em seguida, testado em outra parte para avaliar o desempenho.

```
# Normalize features
scaler = MinMaxScaler()
X_train = scaler.fit_transform(X_train)
X_test = scaler.transform(X_test)

# Reshape data for LSTM input
X_train = X_train.reshape((X_train.shape[0], 1, X_train.shape[1]))
X_test = X_test.reshape((X_test.shape[0], 1, X_test.shape[1]))
```

Figura 42 - Normalização de características

Após a divisão dos dados, realizou-se a normalização das características (ver Figura 42). A normalização é uma etapa importante no pré-processamento de dados, onde os valores das características são ajustados para estarem dentro de uma faixa comum. Neste caso, estamos a utilizar o *MinMaxScaler* para dimensionar os valores das características para o intervalo entre 0 e 1, isso ajuda a garantir que as características tenham escalas compatíveis e facilita o treino da rede neuronal.

Os dados são reformulados a seguir para que possam ser usados como entrada em modelos de memória longa e curta (LSTM). Como as LSTMs esperam dados em um formato específico: (número de amostras, número de passos de tempo e número de características), a reformulação é necessária. Este caso, reformulou-se os dados para que cada amostra tenha a forma de (1, número de características), o que indica que temos uma sequência temporal única com várias características.

Por fim, definimos cinco modelos de redes neuronais LSTM com configurações variadas. Cada modelo é construído com a biblioteca *TensorFlow* (tf.keras) e possui uma variedade de arquiteturas. Essas arquiteturas incluem regularização L2, uso de dropout, número de camadas LSTM e até mesmo uma camada bidirecional. Esses modelos foram desenvolvidos para explorar diferentes métodos de previsão de resultados relacionados com a diabetes baseados nos dados. Esses modelos serão essenciais para o presente trabalho e análise do desempenho, pois permitirão investigar como várias configurações de LSTM podem afetar a precisão das previsões de diabetes.

Na Figura 43 ilustra-se este processo de definir os modelos LSTM para treino e avaliação.

```
# Define LSTM models to train and evaluate
models = [
    # Simple LSTM model with one LSTM layer and one dense output layer
    tf.keras.Sequential([
        tf.keras.layers.LSTM(32, input_shape=(1, 8)),
        tf.keras.layers.Dense(1, activation='sigmoid')
    ]),
    # LSTM model with two LSTM layers and one dense output layer
    tf.keras.Sequential([
        tf.keras.layers.LSTM(32, input_shape=(1, 8), return_sequences=True),
        tf.keras.layers.LSTM(32),
        tf.keras.layers.Dense(1, activation='sigmoid')
    ]),
    # Bidirectional LSTM model with one LSTM layer and one dense output layer
    tf.keras.Sequential([
        tf.keras.layers.Bidirectional(tf.keras.layers.LSTM(32), input_shape=(1, 8)),
        tf.keras.layers.Dense(1, activation='sigmoid')
    ]),
    # LSTM model with dropout
    tf.keras.Sequential([
        tf.keras.layers.LSTM(32, input_shape=(1, 8), dropout=0.2),
        tf.keras.layers.Dense(1, activation='sigmoid')
    ]),
    # LSTM model with L2 regularization
    tf.keras.Sequential([
        tf.keras.layers.LSTM(32, input_shape=(1, 8), kernel_regularizer=tf.keras.regularizers.l2(0.01)),
        tf.keras.layers.Dense(1, activation='sigmoid')
    ])
]
```

Figura 43 - Definir os modelos LSTM para treino e avaliação

5.3.2 AVALIAÇÃO DE DESEMPENHO

Neste trabalho, examinou-se o desempenho de modelos de *ML* usando um conjunto de dados relacionado ao diagnóstico de diabetes. A capacidade de prever resultados precisos e tomar decisões informadas com base em dados é crucial em vários campos, desde a medicina até a economia. Para garantir que as previsões sejam precisas, os modelos devem ser avaliados minuciosamente.

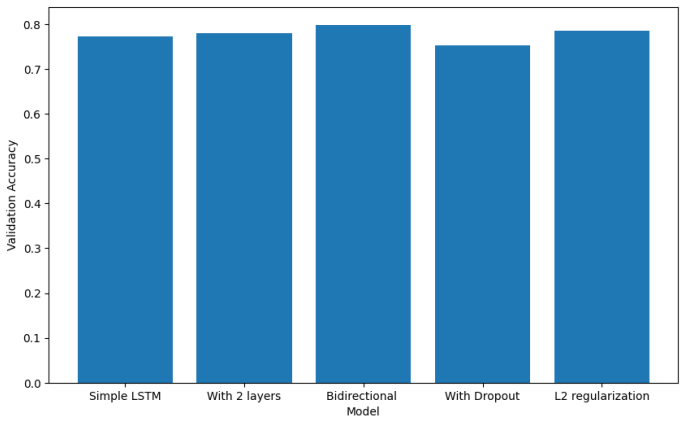


Figura 44 – Accuracy

Na Figura 44 pode se ver o gráfico de barras que representa a precisão dos cinco modelos LSTM, observa-se as diferenças no desempenho, com o modelo Bidirecional apresentando a mais alta precisão, enquanto o modelo com *Dropout* regista a menor precisão. Essa discrepância nos resultados pode ser explicada por algumas considerações específicas de cada modelo.

O Modelo Bidirecional utiliza uma camada LSTM que processa a sequência temporal tanto no sentido temporal quanto no inverso. Isso permite que o modelo capture informações contextuais de medições passadas e futuras, o que é particularmente valioso na previsão da diabetes. Essa abordagem abrange as tendências temporais de forma abrangente, tornando o modelo altamente eficaz na previsão. Assim, é compreensível que o Modelo Bidirecional tenha a maior precisão, uma vez que consegue capturar informações contextuais de forma mais abrangente.

O Modelo com *Dropout* incorpora uma técnica de regularização que desativa aleatoriamente unidades da rede durante o treino. Isso ajuda a evitar o sobre ajustamento do modelo aos dados de treino, tornando-o mais robusto e geral. No entanto, a introdução de *Dropout* pode também diminuir a precisão do modelo, uma vez que parte das unidades é desativada durante o treino. Isso pode levar a uma menor precisão no conjunto de teste, o que pode explicar a precisão inferior deste modelo em comparação com o Modelo Bidirecional.

Portanto, a diferença na precisão entre o Modelo Bidirecional e o Modelo com *Dropout* pode ser atribuída às estratégias de modelação e regularização empregadas. Enquanto o Modelo Bidirecional destaca-se na captura de tendências temporais, o Modelo com *Dropout* sacrifica um pouco de precisão em troca de maior robustez contra o sobre ajustamento

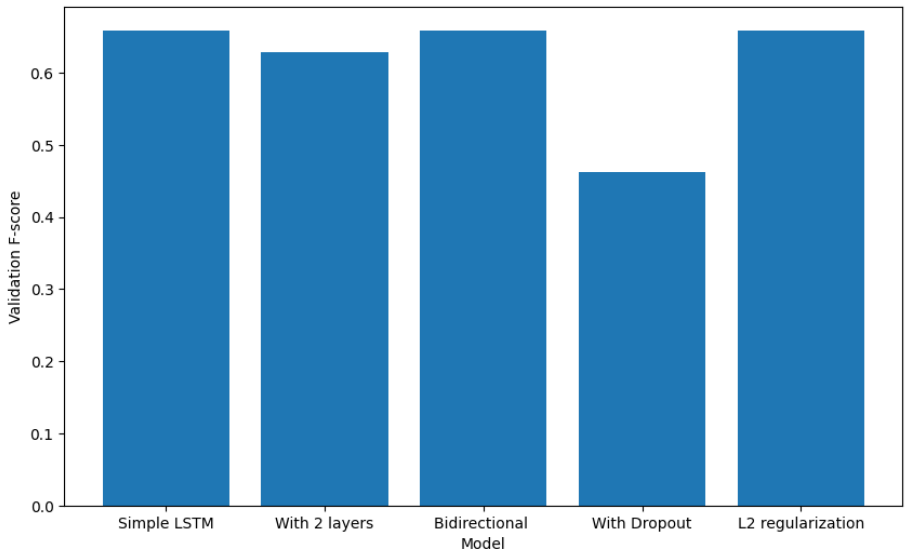


Figura 45 - F-score

Na Figura 45 pode se ver o gráfico de barras que representa a pontuação F1 dos cinco modelos LSTM, observamos diferenças notáveis no desempenho, com o Modelo com Dropout tendo a menor pontuação F1, e o Modelo Bidirecional, o Modelo LSTM Simples e o Modelo com

Regularização L2 apresentando pontuações F1 mais altas. Essas discrepâncias nos resultados podem ser explicadas pelas características e estratégias específicas de cada modelo. A alta pontuação F1 deste modelo sugere que é eficaz em equilibrar precisão e *recall*, capturando com sucesso os resultados positivos e negativos.

Embora mais simples do que algumas outras configurações, ele ainda consegue obter uma alta pontuação F1, o que é uma indicação da sua eficácia na deteção de resultados positivos e negativos.

O Modelo com Regularização L2 incorpora uma técnica de regularização para evitar modelos excessivamente complexos. Isso é importante na diabetes, onde a simplicidade do modelo pode ser vantajosa para a interpretação. A alta pontuação F1 deste modelo sugere que ele é capaz de manter um equilíbrio entre precisão e *recall*, mesmo com a regularização.

O Modelo com *Dropout* incorpora a técnica de desativar unidades aleatórias durante o treino para evitar o sobre ajustamento. Embora essa técnica seja benéfica para a generalização, ela pode ter um impacto negativo na pontuação F1, já que algumas unidades são desativadas. Isso pode explicar a menor pontuação F1 deste modelo em comparação com os outros.

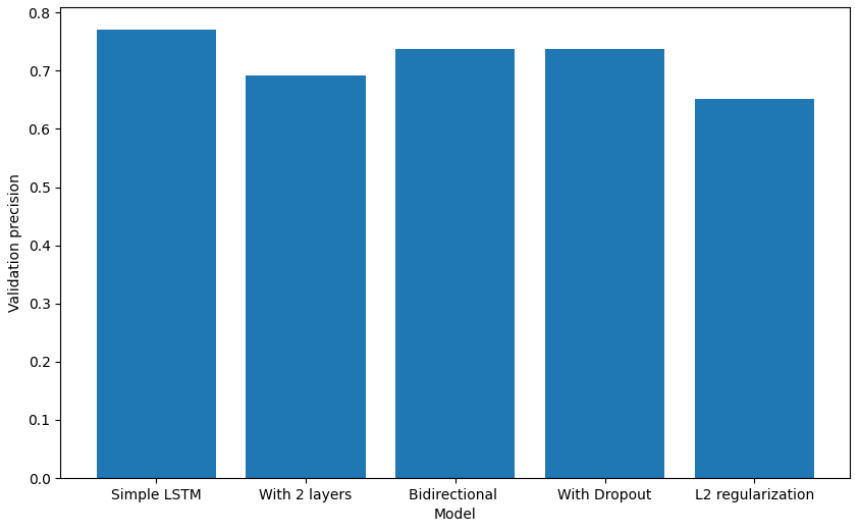


Figura 46 - Precision

Em síntese, as diferenças nas pontuações F1 entre os modelos podem ser atribuídas às estratégias de modelação e às técnicas específicas incorporadas em cada um. O Modelo *Bidirecional* destaca-se na previsão de tendências temporais, enquanto o Modelo LSTM Simples é eficaz e mais direto. O Modelo com Regularização L2 combina simplicidade com desempenho, e o Modelo com *Dropout* sacrifica um pouco da pontuação F1 em troca de generalização. Na Figura 46 pode se ver o gráfico de barras que representa a precisão dos cinco modelos LSTM, observamos notáveis diferenças no desempenho, com o Modelo com Regularização L2 registando a menor precisão e o Modelo LSTM Simples exibindo a maior precisão.

A Figura 47 apresenta o gráfico de barras que representa o *recall* (taxa de verdadeiros positivos) dos cinco modelos LSTM, notamos diferenças no desempenho, com o Modelo com 2 Camadas, o Modelo com *Dropout* e o Modelo com Regularização L2 exibindo diferenças significativas.

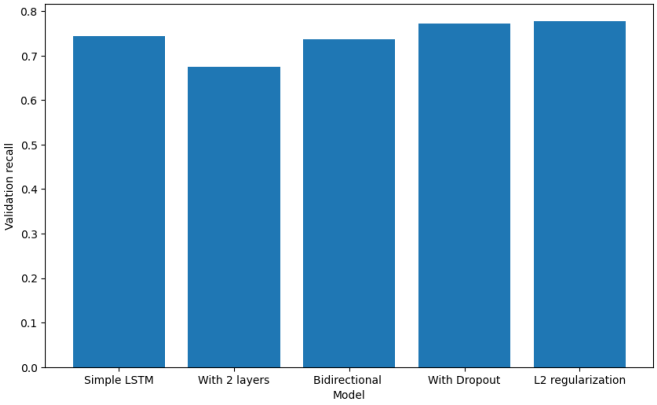


Figura 47 – Recall

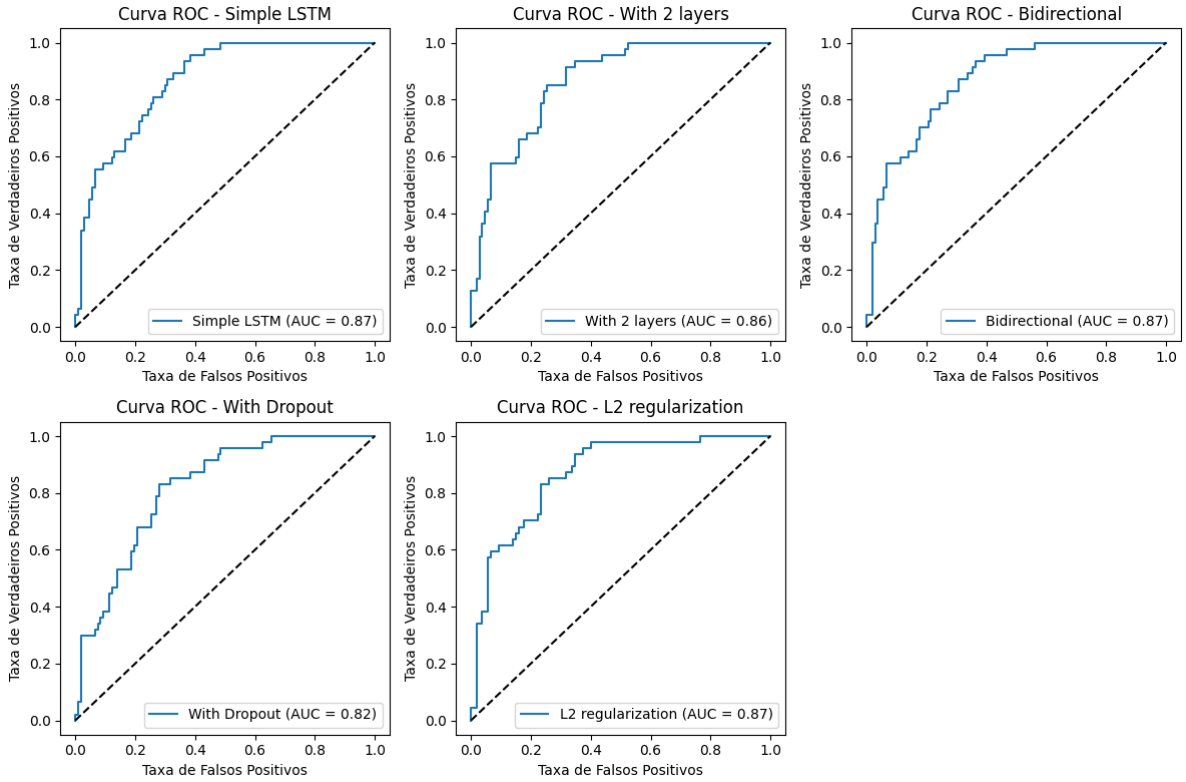


Figura 48 – Curva ROC

Na Figura 48 observa-se a curva ROC (*Receiver Operating Characteristic*) é uma ferramenta crucial no contexto do tratamento da diabetes, especialmente em modelos de classificação binária, como os modelos LSTM que foram treinados e avaliados. Ela descreve visualmente a

capacidade desses modelos em distinguir entre dois resultados possíveis: a presença ou ausência de diabetes.

Portanto, ao analisar a Figura 48, é possível determinar o poder discriminatório dos modelos LSTM no diagnóstico da diabetes. Isso é crucial para a tomada de decisões clínicas e aprimoramento de modelos de previsão no contexto do tratamento da diabetes.

Na Figura 48, os valores da Área sob a Curva (AUC) representam a habilidade de discriminação de diferentes modelos (‘Simple LSTM’, ‘With 2 layers’, ‘Bidirectional’, ‘With Dropout’ e ‘L2 regularization’) no contexto do tratamento da diabetes.

O modelo ‘Simple LSTM’ alcançou uma AUC de 0.87, indicando excelente capacidade de diferenciação entre pacientes com e sem diabetes, o modelo ‘With 2 layers’ obteve uma AUC de 0.86, também revelando boa capacidade discriminatória.

Já o modelo ‘Bidirectional’ atingiu uma AUC de 0.87, denotando forte poder de distinção, o modelo ‘With Dropout’ registou uma AUC de 0.82, um pouco menor, sugerindo uma capacidade de discriminação razoável. O modelo ‘L2 regularization’ apresentou uma AUC de 0.87, destacando sua habilidade de identificação precisa.

Em resumo, a maioria dos modelos demonstra um bom desempenho na distinção entre pacientes diabéticos e não diabéticos, com AUCs geralmente superiores a 0.8. Essas métricas são fundamentais para a seleção do modelo mais adequado às necessidades clínicas no tratamento da diabetes.

Na Figura 49 a curva *Precision-Recall* permite avaliar como cada modelo ('Simple LSTM', 'With 2 layers', 'Bidirectional', 'With Dropout' e 'L2 regularization') lida com a identificação de pacientes diabéticos. Em essência, ela representa visualmente como esses modelos distinguem entre dois resultados cruciais: identificar corretamente pacientes diabéticos (verdadeiros positivos) e evitar identificar erroneamente pacientes não diabéticos como diabéticos (falsos positivos).

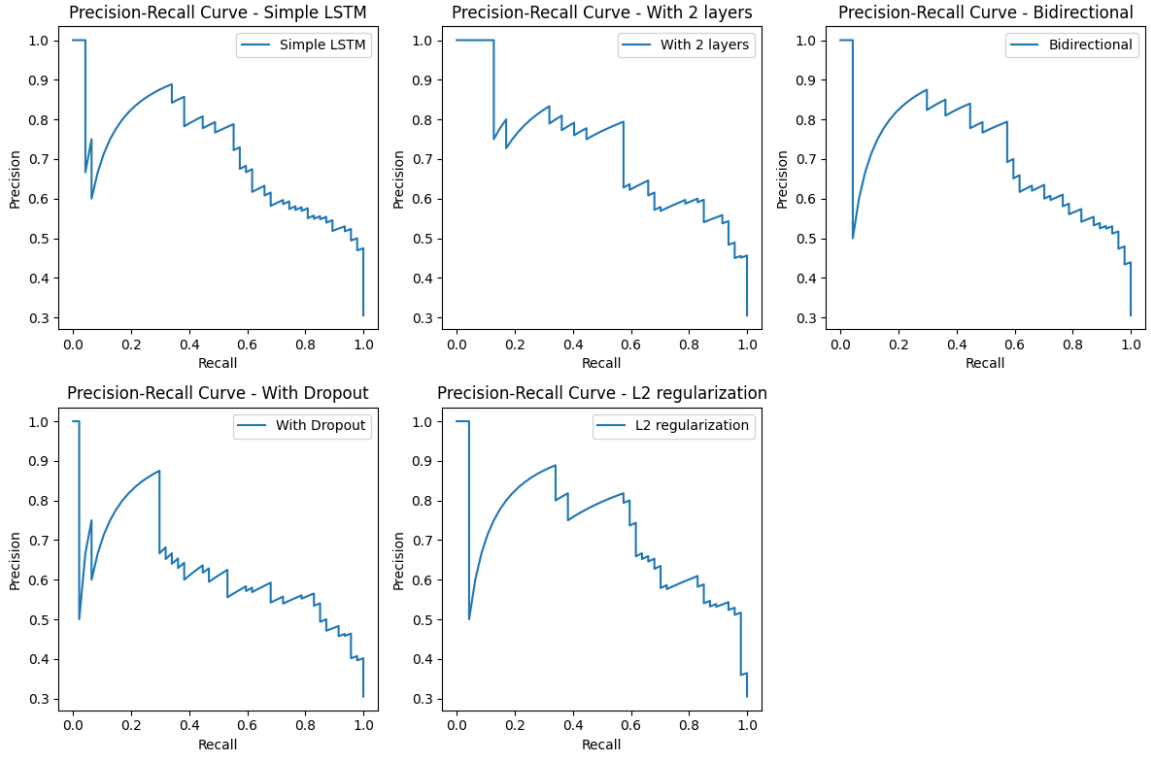


Figura 49 – Precision_ Recall

5.3.3 TESTE DOS MODELOS TREINADOS COM SAMPLE DATA

Após criados os modelos, passa-se ao teste dos mesmos com dados de teste de forma a fazer previsões acerca do nível de insulina no sangue de um paciente representado por dados aleatórios (ver Figura 50).

▾ Teste modelos

```
# Previsão usando os modelos
data_1 = [[6, 148, 72, 35, 0, 33.6, 0.627, 50]]
new_data = sc.transform(data_1)

for config in model_configs:
    model = tf.keras.models.load_model(f"diabetes_{config['name']}_model.h5")
    prediction = model.predict(new_data.reshape((1, new_data.shape[1], 1)))
    print(f"O modelo  {modelo} para estes dados : {data_1} prevé o seguinte resultado:")
    print(f" {result}")
```

```
1/1 [==============================] - 1s 912ms/step
O modelo  Simple LSTM para estes dados : [[6, 148, 72, 35, 0, 33.6, 0.627, 50]] prevé o seguinte resultado:
 É provável que em 30 minutos à 1 hora tenha uma variação que afetará os niveis de insulina no sangue
1/1 [==============================] - 1s 805ms/step
O modelo  With 2 layers para estes dados : [[6, 148, 72, 35, 0, 33.6, 0.627, 50]] prevé o seguinte resultado:
 É provável que em 30 minutos à 1 hora tenha uma variação que afetará os niveis de insulina no sangue
1/1 [==============================] - 2s 2s/step
O modelo  Bidirectional para estes dados : [[6, 148, 72, 35, 0, 33.6, 0.627, 50]] prevé o seguinte resultado:
 É provável que em 30 minutos à 1 hora tenha uma variação que afetará os niveis de insulina no sangue
1/1 [==============================] - 1s 639ms/step
O modelo  With Dropout para estes dados : [[6, 148, 72, 35, 0, 33.6, 0.627, 50]] prevé o seguinte resultado:
 É provável que em 30 minutos à 1 hora tenha uma variação que afetará os niveis de insulina no sangue
1/1 [==============================] - 1s 541ms/step
O modelo  L2 regularization para estes dados : [[6, 148, 72, 35, 0, 33.6, 0.627, 50]] prevé o seguinte resultado:
 É provável que em 30 minutos à 1 hora tenha uma variação que afetará os niveis de insulina no sangue
```

Figura 50 - Teste Modelo Diabetics

▾ Teste modelos

```
# Previsão usando os modelos
data_2 = [[1,85,66,29,0,26.6,0.351,31]]
new_data = sc.transform(data_2)

for config in model_configs:
    model = tf.keras.models.load_model(f"diabetes_{config['name']}_model.h5")
    prediction = model.predict(new_data.reshape((1, new_data.shape[1], 1)))

    print(f"O modelo  {modelo} para estes dados : {data_2} prevé o seguinte resultado:")
    print(f" {result}")
```

```
1/1 [==============================] - 1s 507ms/step
O modelo  Simple LSTM para estes dados : [[1, 85, 66, 29, 0, 26.6, 0.351, 31]] prevé o seguinte resultado:
 Com base nos dados fornecidos dentro de 30 minutos à 1 hora os niveis de insulina no sangue continuarão estáveis
1/1 [==============================] - 1s 586ms/step
O modelo  With 2 layers para estes dados : [[1, 85, 66, 29, 0, 26.6, 0.351, 31]] prevé o seguinte resultado:
 Com base nos dados fornecidos dentro de 30 minutos à 1 hora os niveis de insulina no sangue continuarão estáveis
1/1 [==============================] - 1s 565ms/step
O modelo  Bidirectional para estes dados : [[1, 85, 66, 29, 0, 26.6, 0.351, 31]] prevé o seguinte resultado:
 Com base nos dados fornecidos dentro de 30 minutos à 1 hora os niveis de insulina no sangue continuarão estáveis
1/1 [==============================] - 0s 301ms/step
O modelo  With Dropout para estes dados : [[1, 85, 66, 29, 0, 26.6, 0.351, 31]] prevé o seguinte resultado:
 Com base nos dados fornecidos dentro de 30 minutos à 1 hora os niveis de insulina no sangue continuarão estáveis
1/1 [==============================] - 0s 304ms/step
O modelo  L2 regularization para estes dados : [[1, 85, 66, 29, 0, 26.6, 0.351, 31]] prevé o seguinte resultado:
 Com base nos dados fornecidos dentro de 30 minutos à 1 hora os niveis de insulina no sangue continuarão estáveis
```

Figura 51 - Teste Modelo no Diabetics

Como é possível observar nas figuras (Figura 50 e Figura 51), foram criados dois registos que representam dados de dois pacientes, o data_1 e data_2. De seguida, cada um dos modelos fez as suas previsões quanto ao nível de insulina do paciente. Todos os modelos tiveram as mesmas previsões, tanto para o data_1 como data_2. Para o conjunto de dados deste trabalho não houve contradição entre os modelos treinados.

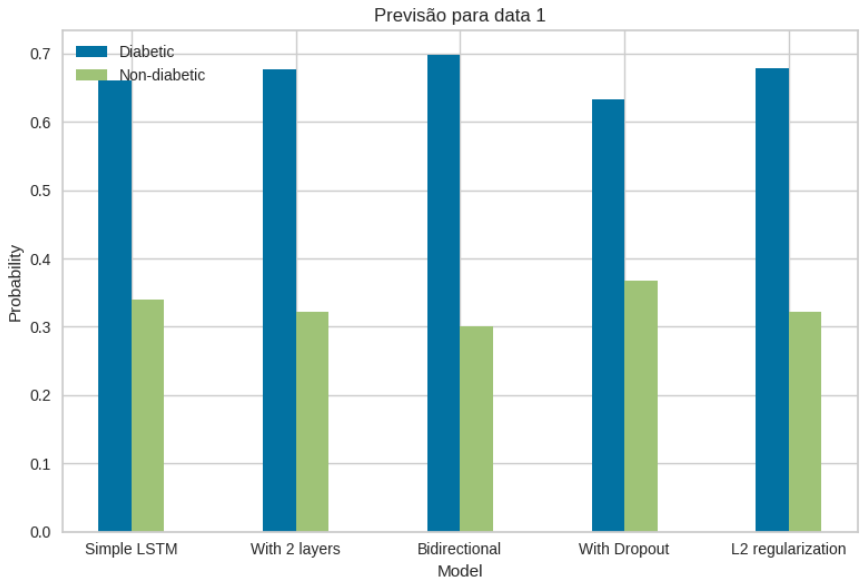


Figura 52 - Previsão data 1

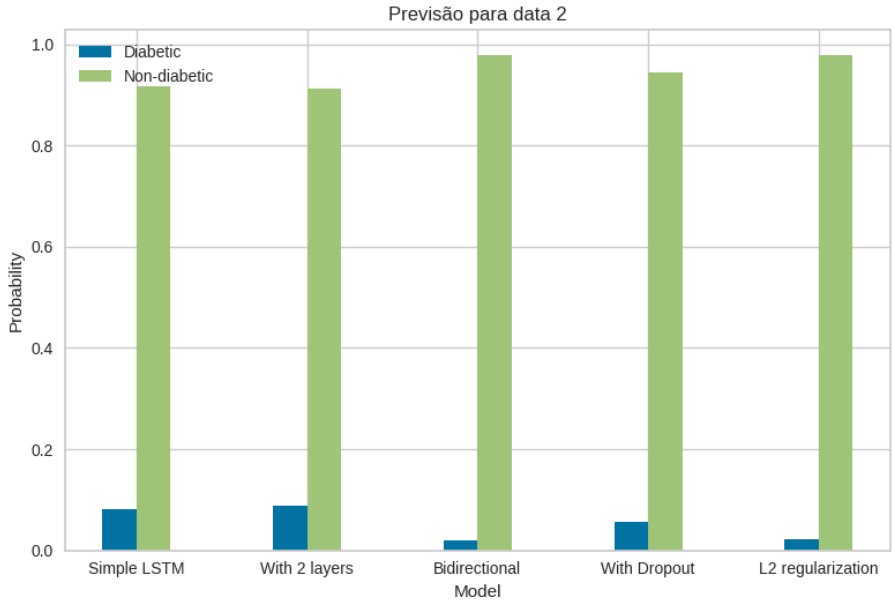


Figura 53 - Previsão data 2

Nas Figura 52 e Figura 53 pode se observar graficamente a previsão dos diferentes modelos LSTM treinados sobre os níveis de insulina no sangue com base em dados de pacientes. O modelo determina se, dentro de um período de 30 minutos a 1 hora, ocorrerá uma variação que afetará esses níveis ou se eles permanecerão estáveis. Para isso, utilizou-se modelos previamente treinados que foram guardados em ficheiros.

Pode-se observar o resultado da previsão para os dados do data_1 e data_2, a mesma ocorre com dependência de um limiar probabilístico definido para então o modelo tomar decisão.

Se o limiar for maior que 0.5, isso indica que, com base nos dados fornecidos, é provável que ocorra uma variação significativa nos níveis de insulina no sangue dentro de 30 minutos a 1 hora. Caso contrário, se o limiar for menor ou igual a 0.5, isso sugere que, com base nos dados atuais, os níveis de insulina no sangue provavelmente permanecerão estáveis no intervalo de 30 minutos a 1 hora.

O mesmo conjunto de dados usado para treinar o LSTM, foi usado dentro da variedade dos classificadores do *pyCaret* (uma biblioteca de código aberto em *python* que foi desenvolvida para simplificar e acelerar o processo de desenvolvimento de modelos de *ML, ver link)*. Em tarefas de classificação, é fundamental avaliar a capacidade de um modelo em distinguir entre classes distintas. Neste contexto, analisou-se o (*Area Under the Curve*)AUC de dois modelos, o *Random Forest* e a Análise Discriminante Linear (LDA), ambos aplicados a um conjunto de dados da diabetes. Para compreender o AUC, é útil considerar a curva ROC (*Receiver Operating Characteristic*).

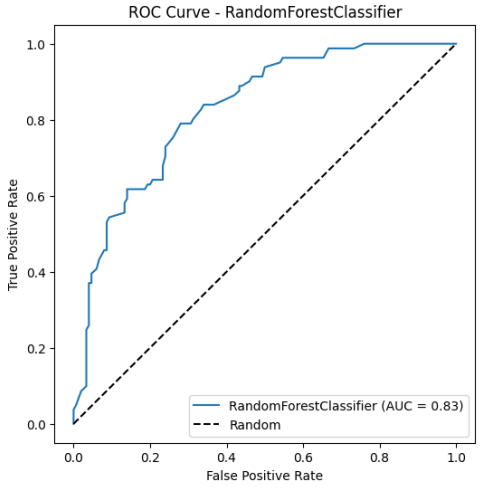


Figura 54 - Curva ROC Random Forest

- **Random Forest e AUC**

O AUC do modelo *Random Forest* é de 0.83 como se pode observar na Figura 54, isso significa que o modelo é capaz de distinguir entre as classes com uma boa capacidade discriminativa. Valores próximos a 1 sugerem que o modelo tem uma alta taxa de verdadeiros positivos e uma baixa taxa de falsos positivos, o que é desejável em muitas aplicações de classificação.

A alta AUC do *Random Forest* pode ser atribuída à sua natureza de ensemble, onde várias árvores de decisão são combinadas para produzir uma previsão final. Essa combinação de várias árvores ajuda a reduzir o *overfitting* e melhora a capacidade de generalização do modelo.

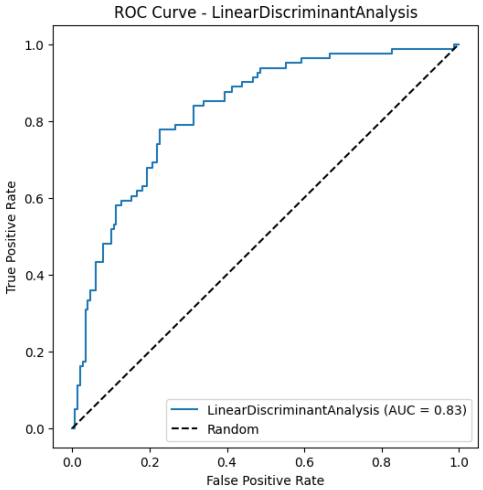


Figura 55 - Curva ROC Linear Discriminant Analysis

- **Análise Discriminante Linear (LDA) e AUC**

A Análise Discriminante Linear (LDA) é uma técnica de redução de dimensionalidade que visa encontrar projeções das características que maximizam a separação entre as classes. Quando o AUC de um modelo LDA é de 0. 83 como se pode ver na Figura 55, isso indica que as projeções geradas pelo LDA são capazes de discriminar eficazmente as classes. LDA é especialmente útil quando as classes são linearmente separáveis, pois ela busca encontrar a combinação linear das características que maximiza a distância entre as médias das classes e minimiza a variabilidade dentro das classes. Isso resulta em uma excelente capacidade de classificação, como indicado pelo alto valor de AUC.

Após treinar os modelos pode se observar que o melhor classificador é apresentado como o *Random Forest* com uma precisão de 0.77 e o que apresenta o pior resultado é o SVM com uma precisão de 0.55.

	Model	Accuracy	AUC	Recall	Prec.	F1	Kappa	MCC	TT (Sec)
rf	Random Forest Classifier	0.7726	0.8242	0.6135	0.6940	0.6483	0.4820	0.4861	0.3600
lda	Linear Discriminant Analysis	0.7707	0.8288	0.5713	0.7261	0.6320	0.4698	0.4821	0.0230
lr	Logistic Regression	0.7689	0.8305	0.5553	0.7284	0.6235	0.4619	0.4752	0.0450
et	Extra Trees Classifier	0.7673	0.8117	0.5664	0.7124	0.6276	0.4621	0.4708	0.1670
ridge	Ridge Classifier	0.7671	0.0000	0.5556	0.7244	0.6227	0.4590	0.4716	0.0330
gbc	Gradient Boosting Classifier	0.7654	0.8192	0.6038	0.6916	0.6395	0.4675	0.4740	0.1370
catboost	CatBoost Classifier	0.7598	0.8209	0.5930	0.6759	0.6272	0.4525	0.4575	2.1910
xgboost	Extreme Gradient Boosting	0.7543	0.7953	0.5991	0.6660	0.6253	0.4449	0.4496	0.0690
lightgbm	Light Gradient Boosting Machine	0.7542	0.8044	0.6257	0.6560	0.6391	0.4532	0.4546	0.1480
nb	Naive Bayes	0.7485	0.8006	0.6038	0.6567	0.6251	0.4370	0.4410	0.0220
ada	Ada Boost Classifier	0.7411	0.7894	0.5658	0.6455	0.6005	0.4108	0.4143	0.1910
knn	K Neighbors Classifier	0.7335	0.7628	0.5813	0.6303	0.5974	0.4004	0.4063	0.0390
qda	Quadratic Discriminant Analysis	0.7223	0.8058	0.5383	0.6134	0.5702	0.3674	0.3708	0.0350
dt	Decision Tree Classifier	0.7021	0.6817	0.6149	0.5777	0.5909	0.3584	0.3626	0.0220
dummy	Dummy Classifier	0.6518	0.5000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0300
svm	SVM - Linear Kernel	0.5534	0.0000	0.4904	0.4127	0.3572	0.0722	0.1053	0.0210

Tabela 4- Classificadores

Num contexto de análise e exploração prática, a avaliação de desempenho de modelos de *ML* desempenha um papel crucial. Esta análise na Tabela 4 visa comparar o desempenho da precisão (*Accuracy*) de um modelo RNN com dois modelos clássicos de *ML*, o *Random Forest* e a Análise Discriminante Linear (LDA). Essa comparação é realizada com base em métricas de desempenho, que incluem precisão (*Precision*), revocação (*Recall*), pontuação F1 (F1 Score) e a precisão geral (*Accuracy*).

6. Análise de Resultados

No contexto do tratamento da diabetes, a precisão na previsão dos níveis de glicose desempenha um papel vital na gestão eficaz desta condição médica. A capacidade de prever com precisão as flutuações nos níveis de glicose permite aos profissionais de saúde e pacientes tomar decisões informadas, ajustar terapias e manter a glicose dentro de faixas-alvo.

Este estudo aborda a utilização de modelos de Aprendizagem Profundo, especificamente Redes Neuronais Recorrentes (*RNNs*) do tipo LSTM (*Long Short-Term Memory*), para criar sistemas de previsão de diabetes com base em dados. O objetivo é avaliar o desempenho de diferentes configurações de modelos LSTM no contexto da previsão de níveis de glicose em pacientes com diabetes.

Para tal, foram treinados cinco modelos LSTM com configurações variadas: *Simple LSTM, With 2 layers, Bidirectional, With Dropout e L2 Regularization*. Cada modelo foi avaliado com base em várias métricas de desempenho, incluindo *accuracy*, *F1 Score, precisão e recall*. Os resultados dessas avaliações fornecerão *insights* importantes sobre quais modelos são mais eficazes para a previsão de níveis de glicose em pacientes diabéticos, tendo em consideração as diferentes prioridades clínicas e os objetivos de modelação.

Neste contexto, a análise detalhada dos resultados é essencial para uma tomada de decisão informada sobre a escolha do modelo mais apropriado, considerando as necessidades específicas de um ambiente clínico no tratamento da diabetes. Este estudo visa contribuir para o avanço da utilização de técnicas de Aprendizagem Profundo na gestão da diabetes, fornecendo *insights* valiosos sobre a eficácia e adaptabilidade dos modelos LSTM.

Como se pode observar na Tabela 5 o resultado das métricas, o objetivo da seguinte analise é selecionar que classificadores apresenta o melhor desempenho. Neste trabalho, como já citado anteriormente, foram implementadas 5 arquiteturas LSTM e o método para a construção de característica escolhido foi o *MinMaxScaler* pois isso ajuda a garantir que as características tenham escalas compatíveis e facilita o treino da rede neuronal.

Model	Loss	Accuracy	F1 Score	Precision	Recall
Simple LSTM	0.43	0.80	0.64	0.71	0.57
With 2 layers	0.42	0.79	0.64	0.68	0.60
Bidirectional	**0.42**	**0.82**	0.66	0.77	0.57
With dropout	0.48	0.77	0.42	0.87	0.28
L2 regularization	0.46	0.80	0.66	0.68	0.64

Tabela 5 - Análise dos resultados das diferentes arquiteturas LSTM

No âmbito do tratamento da diabetes, é essencial atingir uma elevada precisão na previsão dos níveis de glicose para garantir uma gestão eficaz desta condição clínica. Neste trabalho, foram treinados e avaliados cinco modelos LSTM com diferentes configurações, com o objetivo de determinar a sua eficácia na previsão de níveis de glicose em pacientes diabéticos. Os resultados destas avaliações, apresentados na Tabela 5, são fundamentais para a seleção do modelo mais adequado à realidade clínica e aos objetivos de modelação.

Cada modelo foi submetido a um processo de avaliação e validação com base em várias métricas de desempenho. Estas métricas incluem a perda (*loss*), que representa o erro médio nas previsões do modelo, sendo desejável um valor próximo de 0. Além disso, foram analisadas métricas como a precisão, o *F1 Score*, a *precisão* (que representa a proporção de verdadeiros positivos em relação a todos os positivos previstos) e o *recall* (que representa a proporção de verdadeiros positivos em relação a todos os verdadeiros positivos reais).

Os resultados obtidos revelam valiosos *insights* para a escolha do modelo mais apropriado, tendo em conta a complexidade do tratamento da diabetes. O modelo *Bidirectional* sobressai com a mais elevada precisão (82%), indicando a sua capacidade notável de acertar as previsões. Os modelos *Simple LSTM* e *L2 Regularization* mantêm resultados sólidos, apresentando um equilíbrio entre precisão e recall. O modelo *With 2 layers* apresenta métricas semelhantes ao 'Simple LSTM', embora com ligeira variação. No entanto, o modelo *With Dropout*, apesar da precisão elevada, evidencia um recall significativamente reduzido, sugerindo a necessidade de otimizações para a melhoria da capacidade de detetar verdadeiros positivos.

Esta análise detalhada dos resultados contribui para a tomada de decisões informadas no contexto clínico do tratamento da diabetes. A escolha do modelo a adotar dependerá das prioridades clínicas específicas e dos objetivos de modelação. Assim, este estudo visa fornecer uma base sólida para a utilização de técnicas de Aprendizagem Profundo na gestão da diabetes, garantindo que as decisões clínicas sejam baseadas em resultados robustos e adaptados às necessidades individuais dos pacientes.

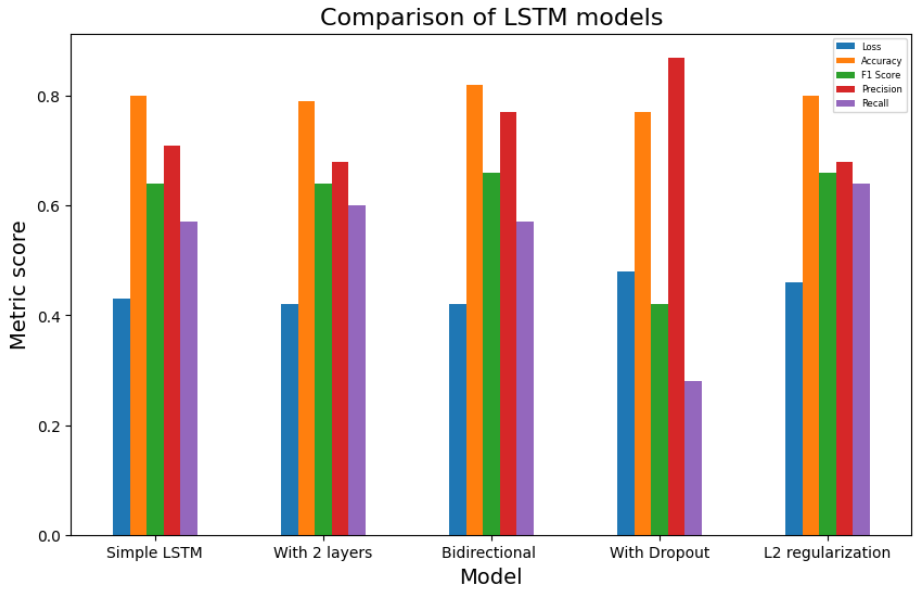


Figura 56 - Comparison of LSTM models

Na Figura 56, o gráfico compara os cinco modelos LSTM com base em várias métricas de desempenho, nomeadamente a perda (*Loss*), precisão (*Accuracy*), F1 Score, precisão (*Precision*) e *recall*. Cada modelo é representado no eixo horizontal, enquanto as pontuações das métricas são representadas no eixo vertical.

- **LSTM Simples;**

Apresenta uma perda de 0.43, indicando um erro médio moderado nas previsões, a sua precisão é de 0.80, o que significa que o modelo acerta cerca de 80% das previsões. Já o *F1 Score* é de 0.64, uma métrica que combina precisão e recall.

A precisão é de 0.71, representando a proporção de verdadeiros positivos em relação a todos os positivos previstos. O *recall* é de 0.57, representando a proporção de verdadeiros positivos em relação a todos os verdadeiros positivos reais.

- **Modelo With 2 layers;**

Evidencia uma perda ligeiramente inferior (0.42) em comparação com o *Simple LSTM*. A precisão é de 0.79 e o F1 Score é de 0.64, mantendo métricas semelhantes ao modelo anterior. A precisão é de 0.68, ligeiramente inferior à do *Simple LSTM*. O recall é de 0.60, também semelhante ao *Simple LSTM*.

- **Modelo Bidirectional:**

Apresenta uma perda de 0.42, semelhante aos modelos anteriores. A sua precisão é a mais alta entre os modelos, atingindo 0.82, o que significa que acerta aproximadamente 82% das previsões. O F1 Score é de 0.66, demonstrando um equilíbrio entre precisão e *recall*. A precisão é de 0.77, enquanto o *recall* é de 0.57.

- **Modelo With Dropout:**

Este modelo tem a perda mais alta (0.48), sugerindo um erro médio ligeiramente superior nas previsões. A precisão é de 0.77, indicando que acerta cerca de 77% das previsões. No entanto, o *F1 Score* é relativamente baixo (0.42), sugerindo que o equilíbrio entre precisão e recall pode não estar otimizado. A precisão é a mais alta entre os modelos (0.87), mas o *recall* é muito baixo (0.28).

- **Modelo L2 Regularization;**

Apresenta uma perda de 0.46, indicando um erro médio moderado nas previsões. A precisão é de 0.80, semelhante à do *Simple* LSTM. O F1 Score é de 0.66, demonstrando um equilíbrio entre precisão e recall. A precisão é de 0.68, enquanto o recall é de 0.64.

O gráfico na Figura 56 permite uma comparação visual das métricas de desempenho de cada modelo, facilitando a identificação dos modelos mais adequados com base nas prioridades clínicas e nos objetivos de modelação no contexto do tratamento da diabetes. Por exemplo, se a ênfase for na precisão, o *Bidirectional* é a escolha mais adequada. No entanto, se for necessário otimizar o equilíbrio entre precisão e recall, o Simple LSTM ou o L2 *Regularization* podem ser opções sólidas.

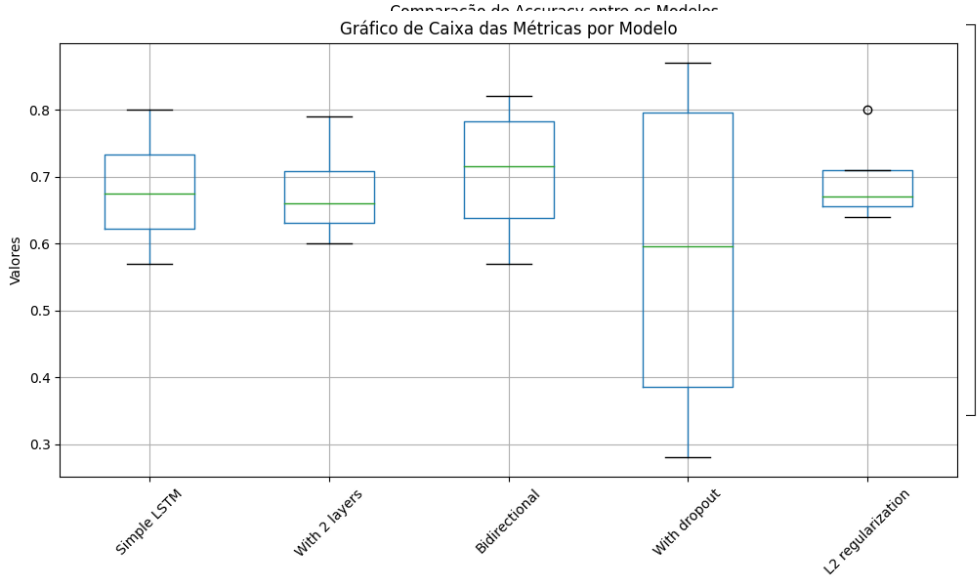


Figura 57 - Comparação de Accuracy entre os Modelos

Pode observar-se na Figura 57 o gráfico de barras que realça o desempenho do melhor modelo, evidenciando aquele que apresenta a maior precisão. A precisão é uma métrica fundamental na avaliação de modelos de *ML*, pois reflete a sua capacidade de fazer previsões precisas. No contexto da análise deste trabalho, a precisão mede a proporção de previsões corretas feitas por cada modelo.

O gráfico apresenta os diferentes modelos no eixo horizontal e os valores de precisão no eixo vertical, variando de 0 a 1. Cada barra representa um modelo específico, e a sua cor é significativa. As barras de modelos com precisão inferior à do melhor modelo são coloridas de cinzento-claro, enquanto o modelo com a melhor precisão é destacado em verde, para tornar clara a distinção.

Esta abordagem visual facilita a compreensão imediata das diferenças de desempenho entre os modelos. Ao destacar o melhor modelo em verde, é mais fácil identificar o modelo mais eficaz em termos de precisão. Esta técnica de apresentação gráfica é valiosa para tomadores de decisão que desejam escolher o modelo mais adequado com base na métrica de precisão.

Figura 58 - Gráfico de Caixa das Métricas por Modelo

No âmbito da investigação realizada neste estudo de mestrado, a avaliação de modelos de redes neuronais recorrentes (RNN) tornou-se uma componente crucial. Estas redes desempenham um papel significativo em inúmeras aplicações, desde processamento de linguagem natural até previsões temporais, e a sua eficácia é frequentemente medida através de diversas métricas de desempenho.

Como se observa na Figura 58 este gráfico de caixa apresenta as distribuições das métricas de avaliação de cinco modelos de RNN distintos. Cada modelo foi submetido a um conjunto de tarefas e, como resultado, foram registadas métricas como a Loss (Perda), Accuracy (Precisão),

F1 Score (Pontuação F1), Precision (Precisão) e Recall (Recall). Através desta análise, procurou-se compreender não apenas qual modelo obteve o melhor desempenho global, mas também como a variabilidade das métricas se reflete em cada modelo. Isso permite uma tomada de decisão informada na escolha da arquitetura de RNN mais adequada para aplicações futuras.

A presente análise destina-se a proporcionar uma visão detalhada do desempenho dos modelos, a fim de orientar a seleção de abordagens eficazes no domínio das redes neuronais recorrentes. O entendimento das diferenças nas métricas destes modelos é fundamental para a escolha da melhor estratégia de implementação e otimização, tendo em vista a obtenção dos resultados desejados nas aplicações práticas. A seguir, explica-se o gráfico de caixa que ilustra essas métricas em detalhe:

- **Simple LSTM (LSTM Simples):** Este modelo apresenta uma mediana de 0.80 na métrica Accuracy, indicando uma precisão geral de 80%. A distribuição evidencia uma variabilidade moderada nas outras métricas, incluindo uma mediana de 0.64 para F1 Score e 0.71 para Precision. No entanto, o valor de Recall é um pouco mais baixo, com uma mediana de 0.57.
- **With 2 layers (Com 2 camadas):** Este modelo apresenta uma precisão semelhante (mediana de 0.79) à do modelo Simple LSTM, mas com uma variabilidade um pouco maior, conforme indicado pelas caixas. A mediana do F1 Score é de 0.68 e a Precision é de 0.60. O Recall é de 0.64.
- **Bidirectional (Bidirecional):** O modelo Bidirectional destaca-se com uma mediana de 0.82 para a Accuracy, sugerindo um desempenho geral superior em comparação com os outros modelos. Além disso, possui uma mediana elevada para F1 Score (0.77) e Recall (0.57), indicando um bom equilíbrio entre precisão e capacidade de detetar positivos reais.
- **With dropout (Com dropout):** Este modelo exibe uma precisão mediana de 0.77, com uma variabilidade mais baixa em comparação com outros modelos. No entanto, a mediana do F1 Score (0.42) é notavelmente mais baixa, sugerindo uma capacidade inferior de equilibrar precisão e recall.
- **L2 regularization (Regularização L2):** Este modelo possui uma precisão mediana semelhante à do modelo Simple LSTM (0.80) e apresenta uma mediana consistente para F1 Score (0.66) e Precision (0.68). O Recall tem uma mediana de 0.64.
- Em resumo, o modelo Bidirecional (Bidirectional) destaca-se com a mais alta precisão média, enquanto o modelo com dropout (With dropout) apresenta uma precisão mediana inferior e um desequilíbrio mais acentuado entre precisão e recall. Os outros modelos, Simple LSTM e With 2 layers (Com 2 camadas), possuem desempenho intermediário, com resultados semelhantes em termos de precisão, F1 Score, Precision e Recall.

Modelo	Loss	Accuracy	F1 Score	Precision	Recall
Bidirectional	0.42	0.82	0.66	0.77	0.57
Random Forest	-	0.777	0.64	0.69	0.61
LDA	-	0.770	0.63	0.72	0.57

Tabela 6 - Comparação do desempenho entre LSTM e outros modelos

Após treinar os modelos com *pycaret (ver link)* apresentado na Tabela 4 e analisando os resultados da Tabela 6, observa-se que o modelo 'Bidirectional' alcançou uma precisão de 0.82 isso significa que o modelo foi capaz de classificar corretamente 82% das instâncias no conjunto de dados., superando o 'Random Forest' com uma precisão de 0.7772 e a Análise Discriminante Linear (LDA) com uma precisão de 0.770. A precisão mede a proporção de previsões corretas feitas pelo modelo e é um indicador importante do desempenho global.

A diferença de desempenho de precisão entre o modelo 'Bidirectional' e os modelos clássicos é notável. A precisão mais elevada do modelo 'Bidirectional' sugere que as Redes Neuronais Recorrentes (RNN) Bidirecionais foram mais eficazes no presente trabalho, superando os modelos tradicionais de *ML*.

A análise comparativa destacou que o modelo "Bidirectional LSTM" demonstrou um desempenho superior em termos de precisão em relação aos modelos "Random Forest" e "Análise Discriminante Linear." Isso sugere que, para tarefas de classificação que envolvem dependências sequenciais, como séries temporais ou dados de texto, o uso de redes neuronais recorrentes, como o LSTM, pode ser uma escolha mais adequada.

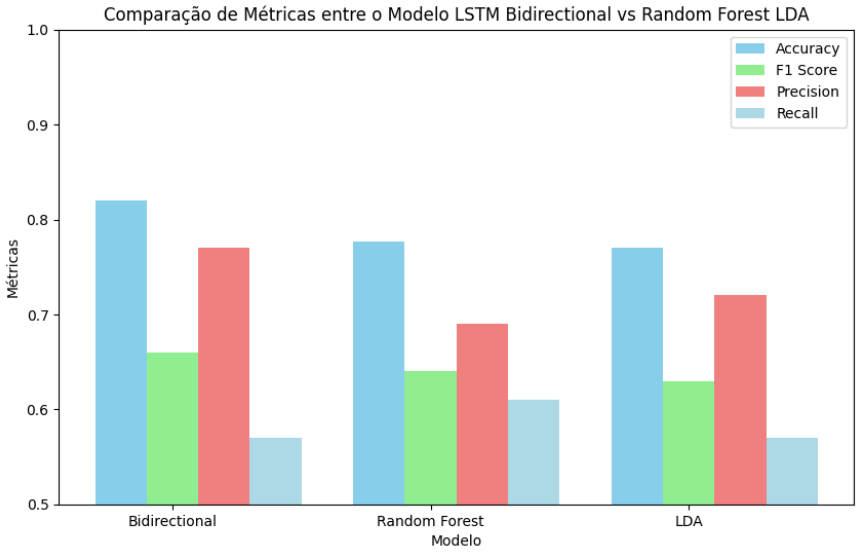


Figura 59 – Métricas: Modelo LSTM Bidirectional vs Random Forest LDA

O gráfico de barras apresentado na Figura 59 visa analisar e comparar as métricas de desempenho de diferentes modelos em um contexto de classificação. Este estudo é relevante para a tomada de decisões em projetos de *ML*, onde a escolha do modelo apropriado desempenha um papel crucial no alcance dos objetivos pretendidos. Os modelos considerados são o "Bidirectional", "Random Forest" e "LDA", e as métricas avaliadas são Accuracy, F1 Score, Precision e Recall. Através deste gráfico, pretende-se proporcionar uma visão clara das diferenças de desempenho entre os modelos, permitindo uma análise abrangente de sua capacidade de classificação.

Resultados e Análise - Conclusão:

O gráfico de barras empilhadas (ver Figura 56) revela informações essenciais sobre o desempenho dos modelos considerados. Cada modelo é representado por uma barra no gráfico, e as métricas são empilhadas verticalmente, permitindo uma comparação direta. Em seguida analisa-se as métricas individualmente:

Accuracy (Precisão):

- O modelo Bidirectional alcança a mais alta precisão, atingindo aproximadamente 0.82.
- O modelo "Random Forest" obtém a segunda maior precisão, com um valor próximo a 0.777.
- O modelo LDA segue de perto, com uma precisão de aproximadamente 0.770.

F1 Score:

- O modelo "Bidirectional" lidera novamente na métrica F1 Score, com um valor em torno de 0.66.
- O modelo "Random Forest" e o modelo "LDA" apresentam F1 Scores semelhantes, em torno de 0.64 e 0.63, respetivamente.

Precision (Precisão):

- O modelo Bidirectional também demonstra a maior precisão na métrica de Precision, aproximando-se de 0.77.
- Os modelos "Random Forest" e "LDA" seguem com valores de precisão de cerca de 0.69 e 0.72, respetivamente.

Recall (Revocação):

Mais uma vez, o modelo "Bidirectional" exibe o maior Recall, aproximando-se de 0.57. Random Forest e LDA registam valores de Recall próximos, em torno de 0.61 e 0.57, respetivamente.

Com base na análise do gráfico de barras empilhadas, é possível concluir que o modelo "Bidirectional" geralmente supera os outros modelos em termos de métricas de desempenho, incluindo Accuracy, F1 Score, Precision e Recall. No entanto, é importante considerar outros

fatores além das métricas, como complexidade do modelo e requisitos específicos do trabalho desenvolvido, ao tomar uma decisão sobre qual modelo utilizar. Este gráfico fornece uma visão abrangente que pode orientar a escolha do modelo mais apropriado com base nas métricas de desempenho analisadas.

7. *DEPLOYMENT* DO MODELO

Nesta seção, será explicado o serviço web desenvolvido como a ponte entre a tecnologia desenvolvida e o utilizador final. O *framework Flask* foi escolhido para o modelo treinado pela simplicidade e facilidade na implementação. O *Flask* é melhor para aplicações mais leves, como serviços web simples.

Este *framework* também é extremamente flexível, o que permite uma abordagem mais simples, o que era ideal para o objetivo específico de servir o modelo treinado. É possível construir *APIs* de forma rápida e direta com ele, o que torna a integração do modelo mais ágil e fácil de implementar.

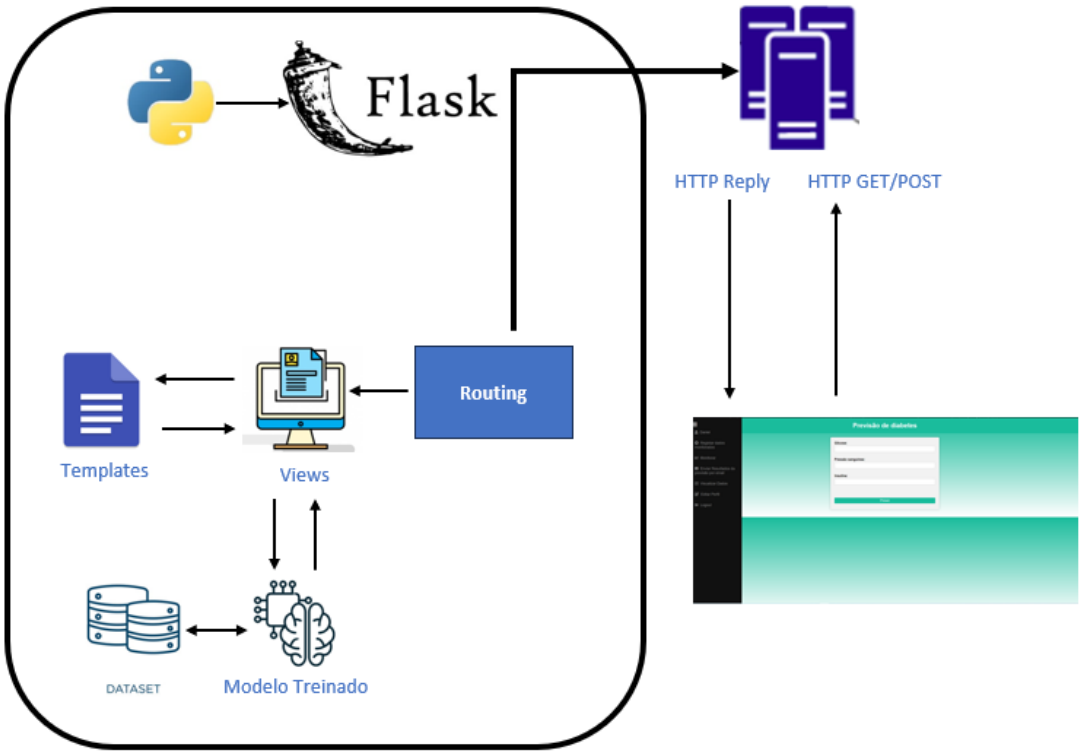


Figura 60 - Arquitetura do Modelo consumido

A Figura 60 apresenta a estrutura da aplicação desenvolvida, onde o modelo treinado é consumido por meio do framework *flask*, trazendo assim uma experiência ao utilizador final de como se comporta o modelo usando novos dados de um paciente. Para a invocação do modelo, pode-se visitar o seguinte *link:*

https://1drv.ms/u/s!At17n0ZiD_EIhHXDKTq7XabJb038?e=HNN9I3

Neste *link* existe um passo a passo para o *deployment* e a aplicação completa, desde a análise do conjunto de dados, o treino do modelo e o consumo do modelo em *Flask*

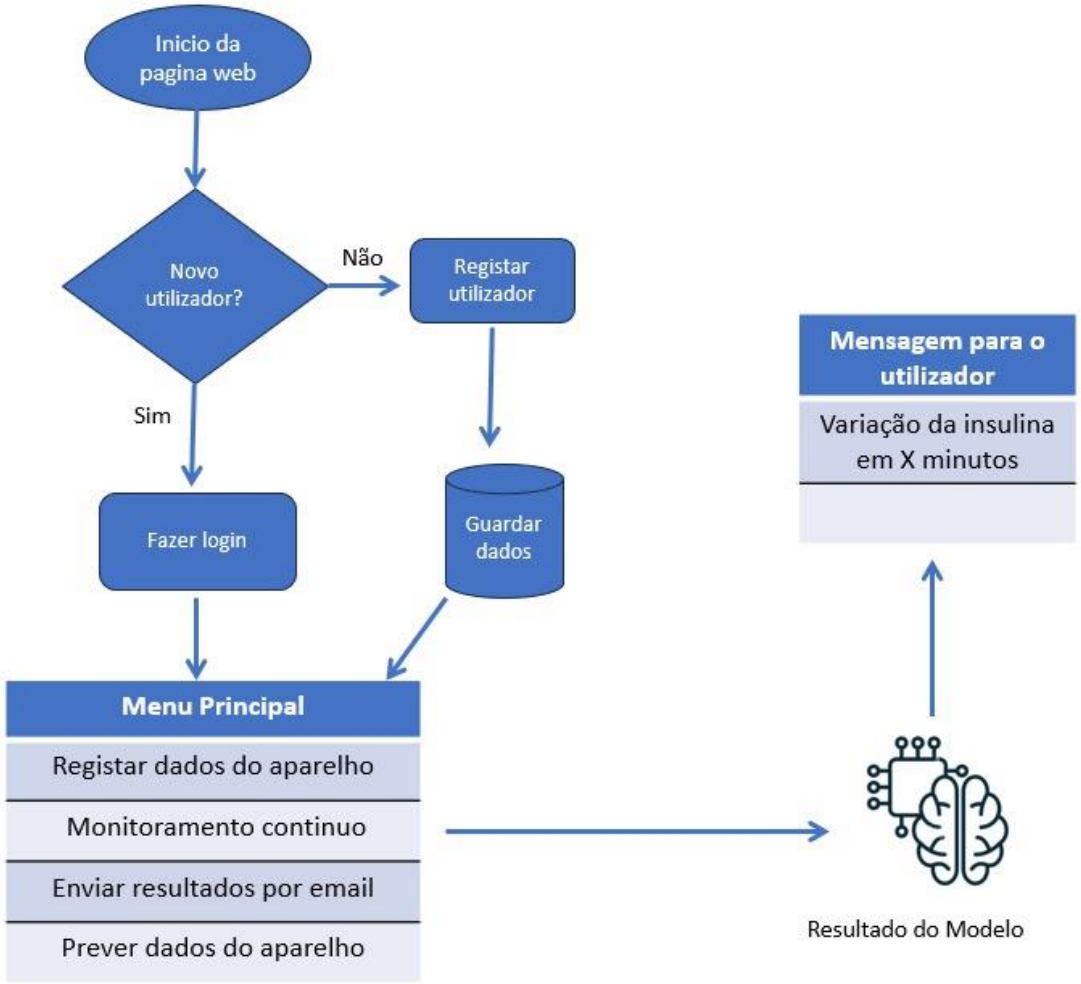


Figura 61 – Fluxo de execução da plataforma web

Analisando o fluxograma na Figura 61, quando se está na página web deve-se iniciar a sessão ou criar um utilizador. Em caso de que se vai criar um utilizador, faz-se um clique no botão Registar, seguidamente procede-se ao preenchimento dos campos requeridos.

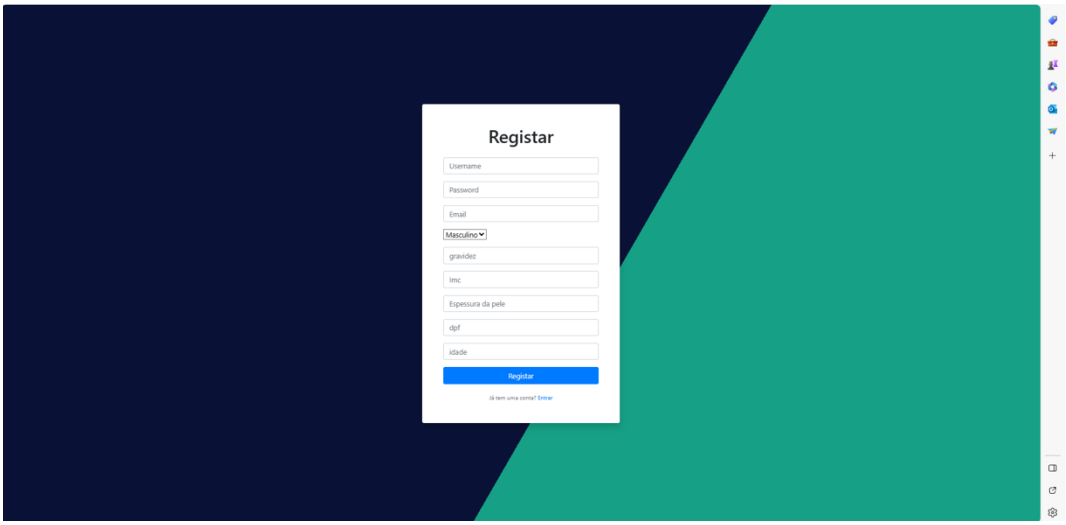


Figura 62 - Registar utilizador

Na Figura 62 coloca-se os dados do paciente e as credenciais para poder usar a aplicação.

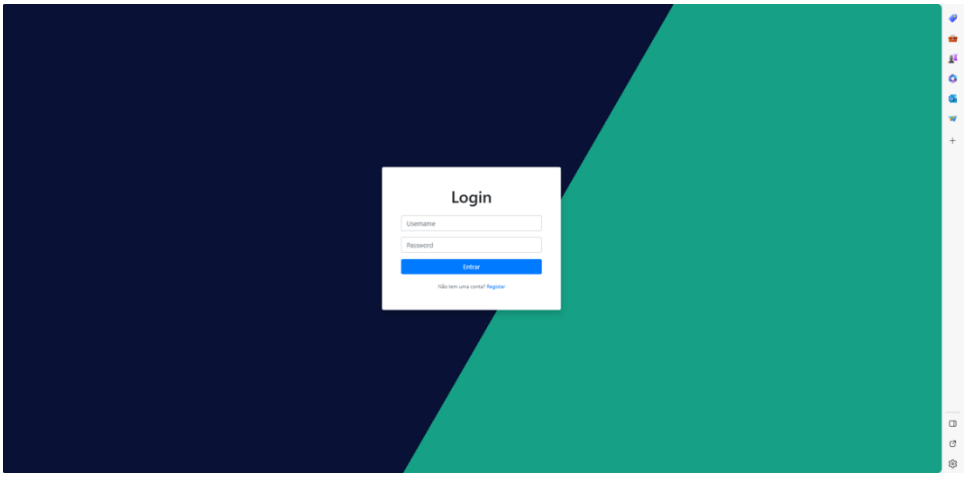


Figura 63 - Login utilizador

Após a criação do utilizador pode-se entrar no menu principal exibido na Figura 63.

Figura 64 - Menu principal

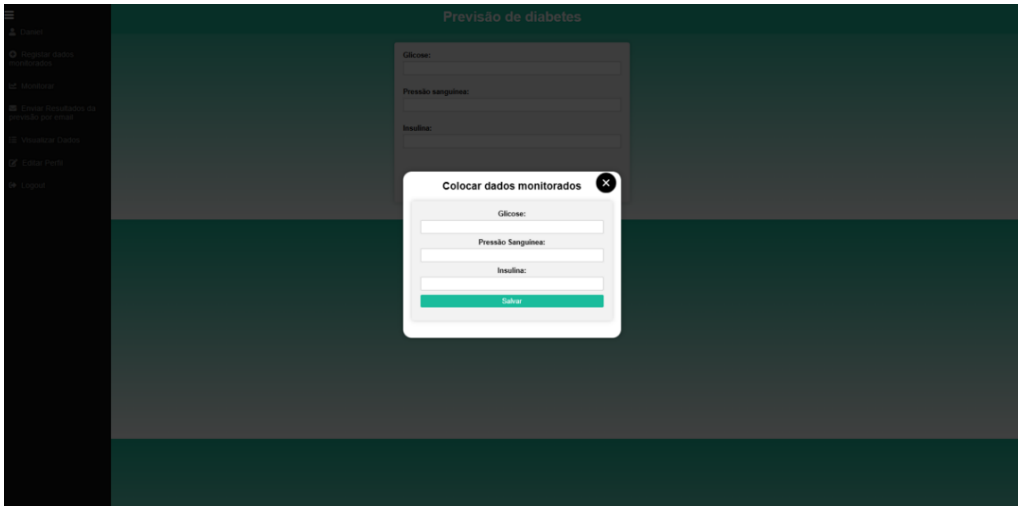


Figura 65 - Inserir dados monitorados

Neste formulário, na Figura 65 coloca-se os dados do paciente que são de estrema importância para a previsão, tais como: glicose, pressão sanguínea e insulina.

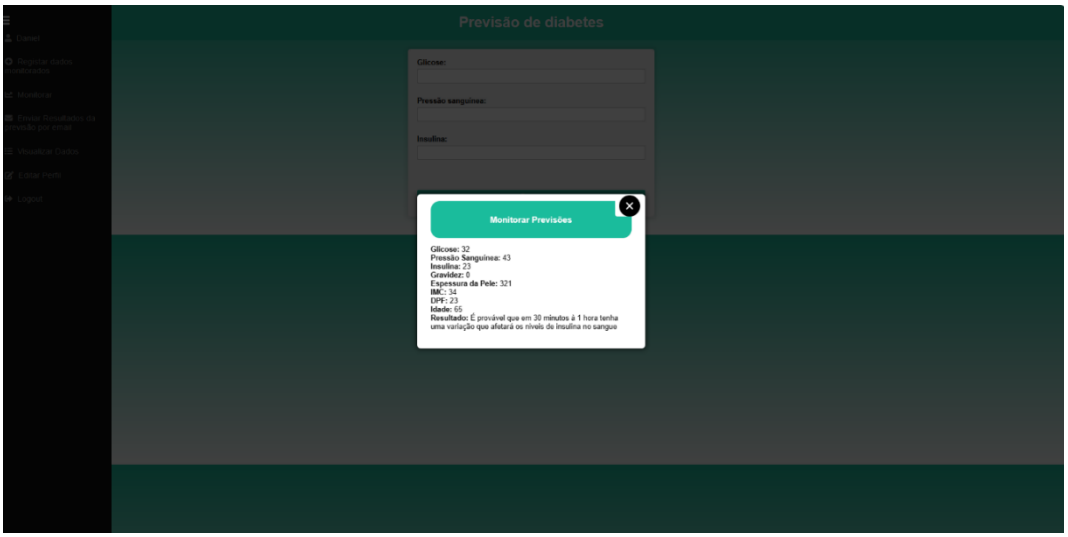


Figura 66 - Monitorar previsão

Observa-se na Figura 66 a monitorização continua dos dados inseridos. Tendo opção de enviar os resultados por email do paciente registado na aplicação como se pode ver na Figura 67.

Figura 67-Resultado por email

8. CONCLUSÕES E TRABALHO FUTURO

Este trabalho de conclusão de mestrado apresentou uma investigação sobre os algoritmos de Inteligência Artificial, *ML* e *DL* e previsão de curto prazo de níveis de insulina ativa e de glicose no sangue, com base em series temporais.

A abordagem proposta, se aplicada na prática, pode melhorar a previsão de diabetes usando técnicas de *DL*, em particular as *Long Short Term Memory* (LSTM). Este documento examina várias abordagens de classificação no conjunto de dados PIMA (é um conjunto de dados de *ML* que contém informações sobre mulheres indígenas Pima) [80]. As abordagens *ML* existentes são testadas no conjunto de dados PIMA. Na Tabela 5 pode se observar o resultado da exatidão alcançado pelo modelo *Bidirectional* é maior do que outras abordagens LSTM tais como: *Simple LSTM, Dropout, Two Layers e L2 Regularization*. Comparando o modelo *Bidirectional* com outras abordagens tradicionais como: *Random Forest, Decision Tree, Linear Discriminant Analysis and Logistic Regression* ainda assim o modelo *Bidirectional teve o melhor desempenho ao fazer a previsão.*

Com base no conjunto de dados utilizados no presente trabalho que totalizam aproximadamente 770 pacientes, as métricas reportadas pelos modelos variam entre 60% a 82% de exatidão sendo o valor mais alto representado pelo modelo *Bidirectional,* revelando uma vantagem significativa a seus competidores, isso pode provar a eficácia e eficiência do *LSTM* para prever dados de series temporais.

O modelo *LSTM Bidirecional* destacou-se com uma precisão de aproximadamente 82%, superando modelos comparativos como *Random Forest* 77% e Análise Discriminante Linear 77%. *F1 Score do LSTM bidirecional atingiu cerca de 66%, enquanto Precision* e *Recall* atingiram 77% e 57%*,* respetivamente.

Após treinar os modelos e obter o modelo com o melhor desempenho desenvolveu-se uma aplicação web que possibilita testar o modelo treinado para que o utilizador final possa então assim prever a sua situação de saúde num período de 30 minutos a 1 hora, após a previsão além do utilizador saber o resultado no momento também é notificado por email com os dados utilizados para a previsão.

Quanto aos desenvolvimentos futuros neste trabalho, pretende-se testar o modelo numa aplicação para smartphones no sentido de facilitar uma melhor interação e usabilidade com o utilizador. Por outro lado, considera-se que também pode valer a pena explorar a substituição da camada *Bidirectional* por alguns outros tipos de redes neuronais, como a *GRU* testar o desempenho para o conjunto de dados utlizado fazendo uma comparação das métricas com os classificadores de tradicionais de *ML*.

Para estudos futuros seria muito interessante realizar mais melhorias e aplicá-lo a um maior conjunto de dados de pacientes, de preferência dados de pacientes reais. A partir deste estudo,

seria importante realizar a comparação deste modelo mais robusto com outros modelos similares ao desenvolvido no começo deste trabalho, que necessitam da entrada de carboidrato.

Finalmente, deve acrescentar-se que embora este trabalho tente de uma forma simples abordar fielmente os resultados, são necessários muito mais recursos e um estudo mais exaustivo dos modelos de *DL*.

9. REFERÊNCIAS

[1] infopedia, «ilhéus de Langerhans». Acedido: 27 de Dezembro de 2023. [Em linha]. Disponível em: https://www.infopedia.pt/dicionarios/termos-medicos/ilh%C3%A9us%20de%20Langerhans

[2] M. E. Cox e D. Edelman, «Tests for Screening and Diagnosis of Type 2 Diabetes», *Clinical Diabetes*, vol. 27, n. 4, pp. 132–138, Jan. 2009, doi: 10.2337/diaclin.27.4.132.

[3] F. dos Santos Nunes, «Previsão de número de dias de internamento em doentes diabéticos-Uma abordagem de *Machine Learning*», 2020.

[4] G. del Carmen Gómez-Encino, R. Zapata-Vázquez, F. Morales-Ramón, e A. Cruz-León, «Nivel de conocimiento que tienen los pacientes con Diabetes Mellitus tipo 2 en relación a su enfermedad», *Salud en tabasco*, vol. 21, n. 1, pp. 17–25, 2015.

[5] J. Inchauspé, « *A revolução da glicose: Equilibre os níveis de açúcar no sangue e mude sua vida*. » Objetiva, 2022.

[6] A. Z. Woldaregay *et al.*, «Data-driven modeling and prediction of blood glucose dynamics: *Machine Learning* applications in type 1 diabetes», *Artif Intell Med*, vol. 98, pp. 109–134, 2019.

[7] E. F. Cruz, A. M. R. da Cruz, e R. Gomes, «Analysis of a traceability and quality monitoring platform for the fishery and aquaculture value chain», em *2019 14th Iberian conference on information systems and technologies (CISTI)*, IEEE, 2019, pp. 1–6.

[8] R. Baskerville, «What design science is not», *European Journal of Information Systems*, vol. 17, n. 5. Taylor & Francis, pp. 441–443, 2008.

[9] M. T. Hernández e N. C. Estrada, «Cetoacidosis diabética», *Anales Médicos de la Asociación Médica del Centro Médico ABC*, vol. 51, n. 4, pp. 180–187, 2006.

[10] L. Pérez-García, M. J. Goñi-Iriarte, e M. García-Mouriz, «Comparison of treatment with continuous subcutaneous insulin infusion versus multiple daily insulin injections with bolus calculator in type 1 diabetes», *Endocrinología y Nutrición (English Edition)*, vol. 62, n. 7, pp. 331–337, 2015.

[11] N. Sneha e T. Gangil, «Analysis of diabetes mellitus for early prediction using optimal features selection», *J Big Data*, vol. 6, n. 1, pp. 1–19, 2019.

[12] P. M. Priya, «Disease Prediction and Classification using Intelligent Algorithms», em *2023 Third International Conference on Artificial Intelligence and Smart Energy (ICAIS)*, IEEE, 2023, pp. 494–498.

[13] I. Contreras, M. Muñoz-Organero, A. Beneyto, e J. Vehi, «Active Labeling Correction of Mealtimes and the Appearance of Types of Carbohydrates in Type 1 Diabetes Information Records», *Mathematics*, vol. 11, n. 19, p. 4050, 2023.

[14] American Diabetes Association, «TREATMENT & CARE Hyperglycemia (High Blood Glucose)». Acedido: 7 de Outubro de 2023. [Em linha]. Disponível em: https://diabetes.org/living-with-diabetes/treatment-care/hyperglycemia

[15] American Diabetes Association, «TREATMENT & CARE Hypoglycemia (Low Blood Glucose)». Acedido: 7 de Outubro de 2023. [Em linha]. Disponível em: https://diabetes.org/living-with-diabetes/treatment-care/hypoglycemia

[16] C. F. G. de Andrade, «Monotorização continua da glicose em grávidas com diabetes», 2018.

[17] R. A. Vigersky, «The benefits, limitations, and cost-effectiveness of advanced technologies in the management of patients with diabetes mellitus», *J Diabetes Sci Technol*, vol. 9, n. 2, pp. 320–330, 2015.

[18] A. Thakur e A. Konde, «Fundamentals of neural networks», *Int J Res Appl Sci Eng Technol*, vol. 9, pp. 407–426, 2021.

[19] A. Y. Zakiyyah, «An application of Graf Neural Networks in Diabetes Mellitus Protein Interactions», *Studies on Scientific Developments in Geometry, Algebra, and Applied Mathematics Adnan Tercan Aydin Gezer*, p. 123, 2022.

[20] C. Chatfield e H. Xing, «*The analysis of time series: an introduction with R*.» CRC press, 2019.

[21] Shipra Saxena, «What is LSTM? Introduction to *Long Short-Term Memory*». Acedido: 3 de Janeiro de 2024. [Em linha]. Disponível em: https://www.analyticsvidhya.com/blog/2021/03/introduction-to-long-short-term-memory-lstm/

[22] I. Goodfellow *et al.*, «Generative adversarial networks», *Commun ACM*, vol. 63, n. 11, pp. 139–144, 2020.

[23] I. Stanko, «The Architectures of Geoffrey Hinton», em *Guide to Deep Learning Basics: Logical, Historical and Philosophical Perspectives*, Springer, 2020, pp. 79–92.

[24] Q. E. McCallum e K. Gleason, *Business models for the data economy*. « O'Reilly Media, Inc.», 2013.

[25] A. Zollanvari, «*Machine Learning with Python: Theory and Implementation*.» Springer Nature, 2023.

[26] K. Relan e K. Relan, «Beginning with flask», *Building REST APIs with Flask: Create Python Web Services with MySQL*, pp. 1–26, 2019.

[27] M. Grinberg, *Flask web development: developing web applications with python*. « O'Reilly Media, Inc.», 2018.

[28] TreinaWeb Tecnologia, «TreinaWeb». Acedido: 27 de Dezembro de 2023. [Em linha]. Disponível em: https://www.treinaweb.com.br/blog/o-que-e-o-slim-framework

[29] V. Cerqueira, L. Torgo, e C. Soares, «A case study comparing *Machine Learning* with statistical methods for time series forecasting: size matters», *J Intell Inf Syst*, vol. 59, n. 2, pp. 415–433, 2022.

[30] C. Chatfield, «*Time-series forecasting*. » CRC press, 2000.

[31] A. Pulver e S. Lyu, «LSTM with working memory», em *2017 International Joint Conference on Neural Networks (IJCNN)*, IEEE, 2017, pp. 845–851.

[32] J. Omana e M. Moorthi, «Predictive analysis and prognostic approach of diabetes prediction with *Machine Learning* techniques», *Wirel Pers Commun*, pp. 1–14, 2021.

[33] H. Naz e S. Ahuja, «Deep Learning approach for diabetes prediction using PIMA Indian dataset», *J Diabetes Metab Disord*, vol. 19, pp. 391–403, 2020.

[34] M. F. Rabby, Y. Tu, M. I. Hossen, I. Lee, A. S. Maida, e X. Hei, «Stacked LSTM based deep recurrent neural network with kalman smoothing for blood glucose prediction», *BMC Med Inform Decis Mak*, vol. 21, pp. 1–15, 2021.

[35] I. A. Francisco, «Inteligência artificial no local de trabalho: dimensoes da sua intervençao», 2019.

[36] C. Bartneck, C. Lütge, A. Wagner, e S. Welsh, « *An introduction to ethics in robotics and AI»*. Springer Nature, 2021.

[37] A. Kaplan e M. Haenlein, «Siri, Siri, in my hand: Who's the fairest in the land? On the interpretations, illustrations, and implications of artificial intelligence», *Bus Horiz*, vol. 62, n. 1, pp. 15–25, 2019.

[38] P. Malik, M. Pathania, e V. K. Rathaur, «Overview of artificial intelligence in medicine», *J Family Med Prim Care*, vol. 8, n. 7, p. 2328, 2019.

[39] T. M. B. R. Louro, «A parabiose humano-máquina-uma investigação sobre os limites da mente humana», 2022.

[40] F. M. Torres, «Framework de Apoio ao desenvolvimento da utilização de redes SDN com a integração de IA nos munic\'\ipios portugueses», 2023.

[41] Shivangi, «Artificial Intelligence(AI)». Acedido: 27 de Dezembro de 2023. [Em linha]. Disponível em: https://shivangi-160997.medium.com/artificial-intelligence-ai-990b4dda2f2

[42] P. P. Shinde e S. Shah, «A Review of *Machine Learning* and Deep Learning Applications», em *2018 Fourth International Conference on Computing Communication Control and Automation (ICCUBEA)*, 2018, pp. 1–6. doi: 10.1109/ICCUBEA.2018.8697857.

[43] M. Mohammed, M. B. Khan, e E. B. M. Bashier, «*Machine Learning: algorithms and applications*. » Crc Press, 2016.

[44] I. Sarker, «*Machine Learning*: Algorithms, Real-World Applications and Research Directions», *SN Comput Sci*, vol. 2, Mar. 2021, doi: 10.1007/s42979-021-00592-x.

[45] J. Han, J. Pei, e H. Tong, « *Data mining: concepts and techniques*.» Morgan kaufmann, 2022.

[46] I. H. Sarker, A. S. M. Kayes, S. Badsha, H. Alqahtani, P. Watters, e A. Ng, «Cybersecurity data science: an overview from Machine Learning perspective», *J Big Data*, vol. 7, pp. 1–29, 2020.

[47] L. P. Kaelbling, M. L. Littman, e A. W. Moore, «Reinforcement learning: A survey», *Journal of artificial intelligence research*, vol. 4, pp. 237–285, 1996.

[48] K. P. Murphy, « *Probabilistic Machine Learning: an introduction*. » MIT press, 2022.

[49] J. M. Johnson e T. M. Khoshgoftaar, «Survey on Deep Learning with class imbalance», *J Big Data*, vol. 6, n. 1, pp. 1–54, 2019.

[50] A. M. Antoniadi *et al.*, «Current challenges and future opportunities for XAI in Machine Learning-based clinical decision support systems: a systematic review», *Applied Sciences*, vol. 11, n. 11, p. 5088, 2021.

[51] N. A. Trayanova, D. M. Popescu, e J. K. Shade, «Machine Learning in arrhythmia and electrophysiology», *Circ Res*, vol. 128, n. 4, pp. 544–566, 2021.

[52] P. P. Shinde e S. Shah, «A review of Machine Learning and Deep Learning applications», em *2018 Fourth international conference on computing communication control and automation (ICCUBEA)*, IEEE, 2018, pp. 1–6.

[53] A. Z. Woldaregay *et al.*, «Data-driven modeling and prediction of blood glucose dynamics: Machine Learning applications in type 1 diabetes», *Artif Intell Med*, vol. 98, pp. 109–134, 2019, doi: https://doi.org/10.1016/j.artmed.2019.07.007.

[54] I. Kavakiotis, O. Tsave, A. Salifoglou, N. Maglaveras, I. Vlahavas, e I. Chouvarda, «Machine Learning and Data Mining Methods in Diabetes Research», *Comput Struct Biotechnol J*, vol. 15, pp. 104–116, 2017, doi: https://doi.org/10.1016/j.csbj.2016.12.005.

[55] S. Basu, K. T. Johnson, e S. A. Berkowitz, «Use of Machine Learning approaches in clinical epidemiological research of diabetes», *Curr Diab Rep*, vol. 20, pp. 1–19, 2020.

[56] E. Rodriguez e R. Villamizar, «Artificial Pancreas: A Review of Meal Detection and Carbohydrates Counting Techniques», *Review of Diabetic Studies*, vol. 18, n. 4, pp. 171–180, 2022.

[57] T. Wang, W. Li, e D. Lewis, «Blood glucose forecasting using lstm variants under the context of open source artificial pancreas system», 2020.

[58] C. J. Zuñiga-Aguilar *et al.*, «Blood glucose prediction with a fractional order neural network», *Diabetes Technol Ther*, vol. 22, p. 82, 2020.

[59] J. Xia *et al.*, «A *Long Short-Term Memory* ensemble approach for improving the outcome prediction in intensive care unit», *Comput Math Methods Med*, vol. 2019, 2019.

[60] N. Y. Gómez-Castillo *et al.*, «A Machine Learning approach for blood glucose level prediction using a LSTM network», em *International Conference on Smart Technologies, Systems and Applications*, Springer, 2021, pp. 99–113.

[61] B. Lindemann, T. Müller, H. Vietz, N. Jazdi, e M. Weyrich, «A survey on *Long Short-Term Memory Networks* for time series prediction», *Procedia CIRP*, vol. 99, pp. 650–655, 2021.

[62] A. Bhimireddy, P. Sinha, B. Oluwalade, J. W. Gichoya, e S. Purkayastha, «Blood glucose level prediction as time-series modeling using sequence-to-sequence neural networks», CEUR Workshop Proceedings, 2020.

[63] A. Chatterjee, A. Prinz, M. A. Riegler, e J. Das, «A systematic review and knowledge mapping on ICT-based remote and automatic COVID-19 patient monitoring and care», *BMC Health Serv Res*, vol. 23, n. 1, pp. 1–17, 2023.

[64] X. Wu *et al.*, «*Long Short-Term Memory* model–a Deep Learning approach for medical data with irregularity in cancer predication with tumor markers», *Comput Biol Med*, vol. 144, p. 105362, 2022.

[65] S. Zhang, D. Zheng, X. Hu, e M. Yang, «Bidirectional *Long Short-Term Memory Networks* for relation classification», em *Proceedings of the 29th Pacific Asia conference on language, information and computation*, 2015, pp. 73–78.

[66] T. Hastie, R. Tibshirani, J. H. Friedman, e J. H. Friedman, « *The elements of statistical learning: data mining, inference, and prediction*, vol. 2. » Springer, 2009.

[67] I. Dexcom, «Dexcom G6 CGM». Acedido: 5 de Novembro de 2023. [Em linha]. Disponível em: https://www.dexcom.com/pt-pt

[68] medtronic, «medtronic». Acedido: 5 de Novembro de 2023. [Em linha]. Disponível em: https://www.medtronicdiabetes.com/home

[69] G. L. Peralta, «Hypoglycaemia Prediction on Type 1 Diabetes Patients using Continuous Glucose Monitoring and Health Record Data», 2022.

[70] VINCENT LUGAT, «Pima Indians Diabetes», *UCI MACHINE LEARNING*. Acedido: 21 de Janeiro de 2024. [Em linha]. Disponível em: https://www.kaggle.com/code/vincentlugat/pima-indians-diabetes-eda-prediction-0-906/input?select=diabetes.csv

[71] V. Chang, J. Bailey, Q. A. Xu, e Z. Sun, «Pima Indians diabetes mellitus classification based on Machine Learning (*ML*) algorithms», *Neural Comput Appl*, vol. 35, n. 22, pp. 16157–16173, 2023.

[72] B. A. Bushman, «Body Composition: Measurement Techniques to Increase Accuracy», *ACSMs Health Fit J*, vol. 26, n. 2, 2022, [Em linha]. Disponível em: https://journals.lww.com/acsm-healthfitness/fulltext/2022/03000/body_composition__measurement_techniques_to.4.aspx

[73] K. Ogurtsova *et al.*, «IDF diabetes Atlas: Global estimates of undiagnosed diabetes in adults for 2021», *Diabetes Res Clin Pract*, vol. 183, p. 109118, 2022, doi: https://doi.org/10.1016/j.diabres.2021.109118.

[74] S. Van Buuren, «*Flexible imputation of missing data*. » CRC press, 2018.

[75] S. Esteban, M. Tablado, F. Peper, S. Terrasa, e K. Kopitowski, «*Deep Bidirectional Recurrent Neural Networks as End-To-End Models for Smoking Status Extraction from Clinical Notes in Spanish*. » 2018. doi: 10.1101/320846.

[76] C. Che, C. Xiao, J. Liang, B. Jin, J. Zho, e F. Wang, «An rnn architecture with dynamic temporal matching for personalized predictions of parkinson's disease», em *Proceedings of the 2017 SIAM international conference on data mining*, SIAM, 2017, pp. 198–206.

[77] R. Yasrab, W. Jiang, e A. Riaz, «*Fighting Deepfakes Using Body Language Analysis*. » 2021. doi: 10.20944/preprints202103.0412.v1.

[78] S. AKDAĞ, F. Kuncan, e Y. Kaya, «A new approach for congestive heart failure and arrhythmia classification using downsampling local binary patterns with LSTM», *Turkish Journal of Electrical Engineering and Computer Sciences*, vol. 30, n. 6, pp. 2145–2164, 2022.

[79] Ki-wook Lee, «*Long Short-Term Memory*», Acedido: 4 de Janeiro de 2024. [Em linha]. Disponível em: https://bi*ML*kw.medium.com/3-long-short-term-memory-f7d703f461a3

[80] Kaggle Contributor, «Pima Indians Diabetes Database», Acedido: 4 de Janeiro de 2024. [Em linha]. Disponível em: https://www.kaggle.com/datasets/uci*ML*/pima-indians-diabetes-database