Lisbon School of Economics & Management
Universidade de Lisboa

MESTRADO

MÉTODOS QUANTITATIVOS PARA A DECISÃO ECONÓMICA E EMPRESARIAL

TRABALHO FINAL DE MESTRADO

RELATÓRIO DE ESTÁGIO

PREVISÃO DO VALOR BRIX: APLICAÇÃO DE ALGORITMOS DE MACHINE LEARNING

CATARINA ANDRADE MIRA ANTUNES DA SILVA

ORIENTAÇÃO:

PROF. DOUTOR CARLOS COSTA
ENG. RUI LOPES

OUTUBRO – 2021

Aos meus avós.

ABREVIATURAS, ACRÓNIMOS E SIGLAS

CRISP-DM - *Cross-Industry Standard Process for Data Mining*

ML – *Machine Learning*

OLS – *Ordinary Least Squares*

RF – *Random Forest*

SVR – *Support Vector Regression*

TFM – Trabalho Final de Mestrado

RESUMO

O consumo sustentável é um tema cada vez mais debatido na atualidade. Com o aumento da população mundial e a diminuição de recursos naturais, é necessário aplicar técnicas que conduzam a uma produção controlada combatendo assim o desperdício, pelo que a previsão da qualidade de produtos agrícolas é um tópico crucial na tomada de decisão.

As áreas de *Machine Learning* e de *Remote Sensing* têm contribuído significativamente para responder a estas dificuldades, na medida em que o tempo de processamento desde a recolha de dados à previsão dos mesmos é relativamente curto.

Desta forma, o principal propósito deste trabalho é estudar o potencial das imagens Sentinel-2, em parceria com a empresa Forging Lab, para a análise e previsão da qualidade de produtos agrícolas, pelo valor Brix, para que, posteriormente, se possam mitigar os riscos de perda e consequentemente aumentar os lucros.

Ao longo do estudo utilizam-se várias abordagens de *Machine Learning* do ramo da aprendizagem supervisionada, nomeadamente, Regressão Linear (OLS), *Support Vector Regression*, Redes Neuronais, *Random Forest* e *LightGBM*.

Na comparação dos resultados de previsão obtidos pelas várias abordagens em estudo, verifica-se que os modelos em que se aplicou o algoritmo *Random Forest* geram maior precisão e menores erros de previsão. O melhor modelo, do algoritmo *Random Forest*, apresentou um coeficiente de determinação de 87,87%, com erro absoluto médio de 0,2985 e erro quadrático médio de 0,2741.

Palavras-chave: Valor Brix; *Machine Learning*; *Python*; Regressão; *Ordinary Least Squares*; *Support Vector Regression*; Redes Neuronais; *Random Forest*; *LightGBM*

ABSTRACT

Sustainable consumption is an increasingly debated topic these days. Due to the current increase in population and the depletion of natural resources, it is urgent that we implement production control techniques so that, in turn, we can effectively combat waste. Thus, agricultural product quality prediction is a crucial topic in decision-making.

Machine Learning and Remote Sensing have played a significant role in response to these challenges, as the processing time from data collection to data prediction is relatively short.

Bearing this in mind, with this thesis we aim to study, in partnership with the company Forging Lab, the potential of Sentinel-2 images in agricultural product quality analysis and prediction, according to the Brix value, so that the risk of loss can be later mitigated and profit can, consequently, increase.

Several approaches in the Machine Learning field are used in this research, namely Linear Regression (OLS), Support Vector Regression, Neural Networks, Random Forest, and LightGBM.

When we compare the predicted results obtained by the approaches used in this study, we verify that the models in which the Random Forest algorithm was used generate higher accuracy and smaller forecast errors. The best Random Forest algorithm model presented a coefficient of determination of 87,87%, with a mean absolute error of 0,2985 and a mean square error of 0,2741.

Keywords: Brix Value; Machine Learning; Python; Regression; Ordinary Least Squares; Support Vector Regression; Neural Network; Random Forest; LightGBM

ÍNDICE

ÍNDICE DE FIGURAS

ÍNDICE DE TABELAS

AGRADECIMENTOS

Aos meus pais, por serem o meu exemplo, por me apoiarem incondicionalmente e por acreditarem sempre que seria possível.

À minha irmã, por ser o meu maior pilar, me ensinar tudo o que sei hoje e tratar de mim como só ela sabe.

Ao meu orientador, Professor Doutor Carlos J. Costa, um agradecimento especial por toda a disponibilidade, dedicação e apoio em toda a realização deste projeto.

Agradeço também ao Dr. Rui Lopes e ao Dr. Fernando Bento pela oportunidade e confiança depositada em mim, por toda a partilha de ensinamentos e a disponibilidade.

Às minhas amigas Carolina, Rita e Patrícia por terem feito este caminho comigo, por toda a amizade, companheirismo e me apoiarem em todos os momentos.

1. INTRODUÇÃO

1.1. Enquadramento

A elaboração deste Trabalho Final de Mestrado (TFM) advém da realização de um estágio em parceria entre as empresas ACTON-IT e Forging Lab e o Lisbon School of Economics and Managment (ISEG), no âmbito do mestrado em Métodos Quantitativos para a Decisão Económica e Empresarial (MQDEE). O trabalho aqui reportado teve a duração de três meses, desde o início de janeiro de 2021 a abril do mesmo ano.

A Forging Lab, através do conhecimento científico, procura identificar como manter segura e como alimentar toda uma população cada vez maior, criando informações valiosas a partir de dados brutos para um melhor aproveitamento e gestão de recursos naturais. Para isso, oferece serviços de previsão fundamentais na mudança proveniente das técnicas modernas de gestão agrícola, resultante da incerteza e das ameaças impostas pelas alterações climáticas.

A sua principal missão é ser peça essencial na cadeia de valor entre a obtenção de dados, por intermédio de *Remote Sensing*, análise desses dados em conjunto com os dados da empresa e a oferta de informação relevante aos clientes de maneira que estes consigam obter a sustentabilidade financeira e económica desejada. Alguns dos serviços incluem a quantificação de áreas cortadas de eucalipto durante um intervalo de tempo desejado, diagnóstico de biomassa combustível para a prevenção de incêndios, identificação de espécies, tendências de seca, estimativa do valor Brix e tempo de colheita de frutas, entre outros.

A finalidade do estágio incidiu no desenvolvimento de um modelo preditivo, recorrendo a diversos algoritmos de *Machine Learning*, capaz de prever, através de observações de imagens de satélite, a qualidade dos produtos agrícolas.

1.2. Motivação

Sendo a escassez de recursos naturais um tema bastante atual, têm-se vindo a incluir na agricultura moderna procedimentos, processos, técnicas e sistemas integrados que visem mitigar estes problemas e evitar desperdícios.

Por outro lado, a questão económica cada vez mais presente, com o principal objetivo de otimizar a utilização, produção e o consumo de produtos agrícolas, é diariamente sujeita a práticas não só mais racionais, como também mais eficazes e eficientes.

Para compreender os contributos que as tecnologias, nomeadamente, *Remote Sensing* e o *Machine Learning*, oferecem ao setor agrícola, pretende-se analisar o modo como a utilização de observações de satélites prevê a evolução das colheitas relativamente à produtividade, qualidade e rentabilidade, em específico, o valor Brix, um aspeto pouco estudado. Pode definir-se o valor Brix como a quantidade de compostos solúveis numa solução de sacarose.

1.3. Questão de Investigação e Objetivos

Baseado no que foi apresentado anteriormente, resulta a proposta da seguinte questão ainda por responder: É possível fazer a previsão da qualidade de produtos agrícolas, mais concretamente, do valor Brix, tendo por base observações de imagens de satélite?

O principal objetivo deste Trabalho Final de Mestrado consiste em desenvolver um modelo de *Machine Learning*, baseado num conjunto de observações de satélite, que preveja o valor Brix e, consequentemente, forneça a informação necessária para a tomada de decisão a nível económico e financeiro de uma empresa, a fim de mitigar o risco de produção e aumentar o lucro. Para o desenvolvimento do referido modelo, estabelecem-se os seguintes objetivos: familiarização com o contexto dos dados, identificação dos indicadores obtidos por satélite, identificação da correlação dos indicadores com o valor Brix, e, finalmente, a estimação, avaliação e comparação da qualidade dos modelos.

1.4. Abordagem Metodológica

A metodologia de investigação adotada neste estudo de *Machine Learning*, é a CRISP-DM, *Cross-Industry Standard Process for Data Mining* (Costa & Aparicio 2020, 2021). Esta abordagem prevê seis fases, *Business Understanding, Data Understanding, Data Preparation, Modeling, Evaluation e Deployment*, que contribuem para o desenvolvimento da estimação dos vários modelos e posteriormente para a tomada de decisão.

1.5. Estrutura do TFM

O Trabalho Final de Mestrado encontra-se dividido em seis capítulos. O primeiro apresenta o enquadramento e o tema abordado durante o estágio e todos os objetivos delineados. Segue-se o segundo capítulo que corresponde à revisão de literatura de vários temas, como a evolução da agricultura, a relevância do *Machine Learning* (ML) e a descrição dos algoritmos de ML utilizados na previsão dos dados trabalhados. O terceiro capítulo descreve a metodologia utilizada durante a elaboração do projeto. O quarto capítulo aborda todo o trabalho empírico e são mostrados todos os pontos fulcrais desenvolvidos durante o estágio que contribuíram para a criação dos modelos. É nesta fase que são descritas todas as etapas da abordagem metodológica. O capítulo cinco é exposta uma discussão de resultados. Por fim, no capítulo seis são apresentadas as conclusões e possíveis trabalhos futuros.

2. REVISÃO DE LITERATURA

Neste capítulo são abordados os temas fulcrais para o entendimento do problema apresentado e a resolução do mesmo. Neste caso, o objetivo passa por elaborar um modelo preditivo, tendo por base algoritmos de *Machine Learning*. Inicialmente, é abordada a agricultura e contribuições da tecnologia para a sua evolução. Posteriormente, é feita uma exposição do ML e são identificados os melhores algoritmos e mais congruentes com os dados trabalhados.

2.1. Agricultura de Precisão e valor Brix

A agricultura é uma das atividades económicas mais antigas e, ao longo dos anos, tem vindo a transformar-se e a adaptar-se às diferentes realidades, como, por exemplo, a mecanização e a automação por via de sensores. Nos últimos anos, na gestão e na produção agrícola, perante o contexto de escassez de recursos naturais, têm-se vindo a incluir um conjunto de ferramentas e tecnologias, com o objetivo de otimizar e utilizar de forma sustentável os recursos. Este processo denomina-se Agricultura de Precisão (Coelho & Silva, 2009; Molin et al., 2015).

Segundo Coelho & Silva (2009) a Agricultura de Precisão surge associada ao aumento do rendimento dos agricultores e à redução do impacto ambiental resultante da atividade agrícola. Referem que a Agricultura de Precisão envolve a aplicação diferenciada e medida de fatores de produção, tendo em conta a variação espacial e temporal e dos fatores de produção agrícola, de modo a aumentar a sua eficiência de utilização.

A Agricultura de Precisão é uma abordagem que se baseia em sistemas inovadores que combinam várias tecnologias, tais como, *Geographic Information System* (GIS), *Global Positioning System* (GPS), modelação computacional e *Remote Sensing* a partir de satélites (Liaghat & Balasundram, 2010). De referir que *Remote Sensing* é a ciência que extrai e interpreta informação à distância, por meio de sensores que fisicamente não estão em contacto com as observações (Jensen, 2000).

A utilização de satélites para a medição do valor Brix tem sido objeto de estudo de vários autores (Lee et al., 2012; Pronprasit & Natwichai, 2013; Saranwong et al., 2003; Silva et al., 2014) para testar a qualidade de plantas, frutos e vegetais.

A escala Brix foi desenvolvida por Adolf Brix em meados do séc. XIX e refere-se à percentagem de sólidos solúveis numa solução. No caso de frutas e vegetais, os sólidos solúveis são medidos em valores Brix e estes valores refletem o quão doces estes produtos podem ser e consequentemente a sua qualidade (Ball, 2006).

O valor Brix proporciona vantagens na sua utilização, porque, para além da sua importância como indicador qualitativo, assume grande relevância na tomada de decisão no momento da colheita dos produtos agrícolas (Kleinhenz, Matthew & Bumgarner, 2012).

2.2. *Machine Learning*

Machine Learning ou Aprendizagem Automática pode ser definido como um processo automatizado de resolução de problemas práticos onde, primeiro, se reúne um conjunto de dados, e segundo, se constroem algoritmicamente modelos estatísticos com base nesse conjunto de dados, isto é, é um processo que extrai padrões de dados (Burkov, 2019; Kelleher et al., 2015). Um algoritmo é uma sequência de instruções que transforma *input* em *outputs* (Alpaydin, 2014).

Muller & Guido (2007) e Aparicio et al (2019) referem *Machine Learning* como a interseção de várias áreas, tais como: estatística, inteligência artificial e *computer science*. Para estes autores, *Machine Learning* é ainda definido como análise preditiva e aprendizagem estatística.

Para explicar o *Machine Learning*, Alpaydin (2014) aborda a evolução da geração de dados e o seu aumento de dia para dia. O autor exemplifica com o caso de uma pequena transação, onde todos os detalhes são armazenados (data, id cliente, bens comprados e valor, etc.), e, por conseguinte, todos os dias muitos dados são guardados. Com estes dados, e por existirem certos padrões entre eles, o objetivo passa por poder prever as vendas e os lucros e, consequentemente, maximizá-los. Este processo referido é um nicho do *Machine Learning*. Segundo o autor, os padrões clarificam como todo o processo funciona e como fazer previsões.

Alpaydin (2014) refere ainda que *Machine Learning* não é um processo que envolve apenas base de dados, mas também inteligência artificial. Para um sistema ser inteligente, este deve ter a capacidade de aprender com as informações passadas disponíveis.

Aprender é o processo de converter experiência em experiência/conhecimento. Se o sistema aprender e conseguir adaptar-se às mudanças, não é necessário prever e "arranjar" soluções para todas as possíveis situações, ficando-se assim perante um sistema automático (Shalev-Shwartz & Ben-David, 2013).

2.3. Tipos de aprendizagem em *Machine Learning*

Em *Machine Learning* podem identificar-se vários tipos de aprendizagens, entre elas: a supervisionada; a não supervisionada; a semi-supervisionada e *reinforcement* (Burkov, 2019).

Usa-se a aprendizagem supervisionada para construir modelos utilizados na análise da previsão de dados, isto é, sempre que se quer prever um determinado resultado baseado num *input*, em que temos pares de *input/output.* São criados algoritmos a partir dos pares de *input/output* e da amostra treino da base de dados. O principal objetivo é fazer previsões precisas para dados novos, para além da amostra treino (Kelleher et al., 2015; Müller & Guido, 2017).

São vários os autores que têm vindo a utilizar diferentes abordagens com o intuito de solucionar diversos problemas associados a modelos, desde os mais básicos aos mais complexos, contribuindo com diversos métodos robustos, de modelação e previsão.

As técnicas da aprendizagem supervisionada muitas vezes necessitam de esforço humano para construir a amostra teste, mas depois estas aprendem automaticamente qual a relação entre um conjunto de características descritivas e uma característica alvo, com base em exemplos históricos, e aceleram o processo de previsão. Alguns exemplos de algoritmos de aprendizagem supervisionada são Regressão *Ordinary Least Squares* (OLS), *Support Vector Machine* (SVM), Classificação Naive Bayes e Árvores de Decisão.

Existem dois grandes tipos de problemas dentro da aprendizagem supervisionada: Classificação e Regressão. O principal objetivo da Classificação é prever uma classe dentro de uma lista de possibilidades, por exemplo, se se concede ou não um crédito a um cliente baseado no salário, montante do empréstimo, etc. Na Regressão a meta é prever o valor de uma ou mais variáveis contínuas, como por exemplo, fazer a previsão do salário anual baseado na educação, idade, etc. (Bishop, 1967) ou o preço de habitação (Samadani & Costa, 2021).

2.4. Modelo Regressão *OLS*

Modelos Lineares Clássicos e Método *Ordinary Least Squares* (OLS) foram introduzidos pela primeira vez por Gauss e Legendre no séc. XIX. Alguns exemplos de modelos lineares generalizados são a regressão linear e modelos de análise de variância, modelos *logit* e *probit*, log-lineares e modelos de regressão logística (McCullahg & Nelder, 1989). Pode definir-se o modelo de regressão linear simples como a relação entre X e Y, designado pela expressão:

$$Y = \beta_0 + \beta_1 X + u,$$

onde Y representa a variável dependente; X é a variável independente ou variável explicativa; β_0 e β_1 são dois parâmetros constantes, designados de coeficientes de regressão; u é o termo de erro.

Este modelo pode estender-se ao modelo regressão múltipla com objetivo de incluir k regressores ($X_1, X_2, \dots, X_k$), e pode definir-se como

$$Y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \cdots + \beta_k X_k + u.$$

Esta equação é mais adequada para uma análise *ceteris paribus*, permite controlar explicitamente vários fatores que afetam em simultâneo a variável dependente. Naturalmente, ao adicionar mais variáveis independentes ao modelo, a variável Y pode ser mais explicada. Logo, um modelo de regressão múltipla é utilizado para contruir melhores modelos de previsão da variável dependente.

Nos modelos de regressão linear, os coeficientes de regressão, $\beta_0, \beta_1, \beta_2, \dots, \beta_k$, são desconhecidos, por isso, para a estimação destes parâmetros ($\hat{\beta}_0, \hat{\beta}_{1,}, \hat{\beta}_2, \dots, \hat{\beta}_k$), adota-se o método dos mínimos quadrados (*Ordinary Least Squares* - OLS). Este baseia-se no critério da minimização da soma dos quadrados dos resíduos, isto é, minimiza-se o erro em explicar ou prever os valores de Y, a partir dos valores de X. Dadas n observações em $y, x_1, x_2, \dots, x_k$, $\{(x_1, x_2, \dots, x_k, y_i) : i = 1, 2, \dots, n\}$, os estimadores $\hat{\beta}_0, \hat{\beta}_{1,}, \hat{\beta}_2, \dots, \hat{\beta}_k$, são escolhidos pela minimização da seguinte expressão

$$\sum_{i=1}^{n} (y_i - \hat{\beta}_0 - \hat{\beta}_1 x_{i1} - \hat{\beta}_2 x_{i2} - \dots - \hat{\beta}_k x_{ik})^2$$

Para determinar se um modelo de regressão tem capacidade explicativa utiliza-se o R^2 também conhecido como coeficiente de determinação. Para garantir que um modelo é válido, é essencial testar todas as hipóteses do método dos mínimos quadrados (González-Rivera, 2012; Johnston & DiNardo, 1996; Wooldridge, 2012).

2.5. *Support Vector Machine*

Support Vector Machine (SVM) é um algoritmo de *Machine Learning* que se tem tornado cada vez mais popular para resolver problemas de regressão, *Support Vector Regression* (SVR), bem como, de classificação, *Support Vector Classifier* (SVC) (Albon, 2018; Üstün et al., 2007).

Para perceber a ideia base por de trás do SVR é necessário perceber o conceito de hiperplanos. Um hiperplano é um subespaço n-1 num espaço n - dimensional que maximiza a margem entre conjuntos de dados. A margem é a largura máxima da área paralela ao hiperplano que não contem dados, por isso, para dividir um espaço tridimensional usamos um hiperplano bidimensional (Albon, 2018).

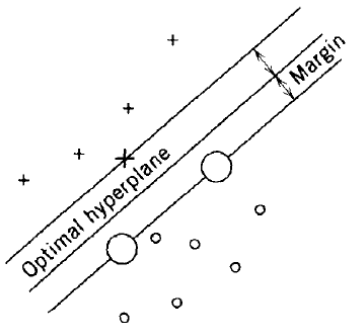

Figura 1 – Separação dos dados pelo hiperplano ótimo, com a margem máxima

Fonte: (Vapnik, 1998)

O algoritmo SVR é processado da seguinte forma. São definidos x *input vectors* num espaço de dimensão *Z*, a partir de um mapeamento não linear escolhido previamente. Neste espaço dimensional é construído um hiperplano ótimo, Figura 1, que separa os conjuntos que anteriormente não eram separáveis linearmente e dada uma amostra treino $\{(x_1, y_1), \dots, (x_l, y_l)\} \subseteq \{X \times Y\}^l$, define-se a função $f(x)$, que em termos de notação se assemelha com o modelo de regressão linear,

$$f(x) = \langle w, x \rangle + b,$$

onde $f(x)$ é o valor previsto que depende de uma inclinação $w \in R^n$ e uma interseção $b \in R$. Calcula-se o hiperplano ótimo pela minimização da função $\frac{1}{2}\|w\|^2$, sujeito a, $-\varepsilon \leq y_i - \langle w, x_i \rangle - b \leq \varepsilon$, onde *X* são os *inputs*, *Y* os *outputs*, *l* representa o número de

pontos do conjunto de treino, w é o vetor dos pesos, ε o desvio, $\langle\ ,\ \rangle$ o produto escalar e b o vetor de enviesamento *bias* (Smola & Schölkopf, 2004; Vapnik, 1999).

Considera-se que um dos benefícios deste algoritmo é a função *kernel trick*, pela possibilidade de as variáveis independentes não estarem linearmente relacionadas com a variável dependente, o que torna o algoritmo bastante flexível e adaptável à base de dados. Existem diversas funções *kernel*, as mais comuns são a linear $x_i^T x$, polinomial $(\gamma x_i^T x + r)^d$, função de base radial $e^{-\gamma(\|x_i - x\|^2)}$e *sigmoidal* $tanh(\gamma x_i^T x + r)$ (Sharma et al., 2020).

2.6. Redes Neuronais

Os primeiros estudos de Redes Neuronais, começaram com o desenvolvimento de um modelo matemático de um neurónio artificial e começaram a ser desenvolvidos em 1943 por Warren McCulloch, neurologista, e Walter Pitts, matemático. As redes neuronais são modelos computacionais inspirados no sistema nervoso humano e utilizados em problemas de reconhecimento de padrões que fornecem informação precisa.

Pode fazer-se a analogia de uma rede neuronal artificial à estrutura de um neurónio biológico. Entrada dos atributos/*inputs* $(x_j, j = 1, \dots, n)$ ou o Axónio transmite a informação de uns neurónios para os outros. Seguido dos pesos sinápticos ou as sinapses (w_{kj}), ligações entre os vários neurónios que permitem a transmissão da informação e estão associados ao peso e grau de importância que cada neurónio terá na camada seguinte. O corpo do neurónio *k* processa toda a informação recolhida e nas RN processa-se a combinação linear de todos os *inputs* recebidos por outros neurónios ou camadas anteriores, dada pela expressão:

$$u_k = \sum_{j=1}^{m} w_{kj} x_j.$$

O corpo inclui também um parâmetro de enviesamento b_k (*bias*) que tem a função de aumentar ou diminuir a entrada da função ativação, dependendo se é positiva ou negativa, respetivamente. A função de ativação $\big(\varphi(v)\big)$ limita a saída de cada neurónio dentro de uma amplitude de valores (Haykin, 2008; Mitchell, 1997).

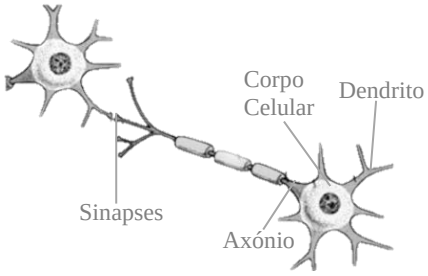

Figura 2 - Estrutura de um neurónio biológico

Fonte: Adaptação de (Maiese, 2021)

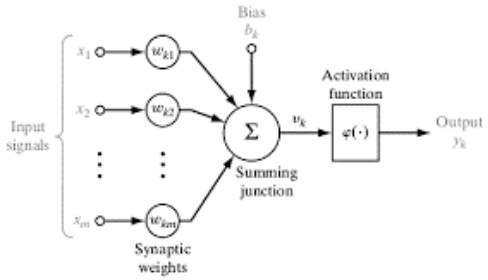

Figura 3 - Estrutura de uma rede neuronal

Fonte: (Haykin, 2008)

É possível identificar vários tipos de função de ativação para cada contexto: *Sigmoid Function, Relu Function*, *Than Function*, *Linear Function*, *Gaussian Function* (Sharma et al., 2020). Por fim, o *output* do neurónio gerado pelas várias camadas de neurónios dá a previsão baseada na aprendizagem ao longo da rede neuronal e é dado pela equação: $(y_k) = \varphi(u_k + b_k)$.

As redes neuronais com a estrutura mais simples, de apenas um neurónio, foram nomeadas por Rosenblatt em 1962, de *Perceptrons*. Estas redes são constituídas apenas por uma *input layer*, onde se dá a entrada dos atributos x_j linearmente combináveis, e uma *output layer*, onde os atributos são diretamente direcionados e geram o valor do *output*, neste caso, uma variável binária, $y_j \in \{-1; 1\}$.

As diversas limitações apontadas, de uma rede neuronal com um único *layer*, fazem com que esta não seja o mais precisa, e de modo a suprimir essas limitações, consideram-se sucessíveis transformações que originam as *multi-layer perceptron*, redes com várias camadas de pesos adaptativos ou, em alternativa, o algoritmo *backpropagation*.

O algoritmo *backpropagation* desenvolvido por Rumelhart, Hinton e Williams, é utilizado ao invés do *Perceptron*, porque os pesos sinápticos que interligam os nodos das várias *layers* são facilmente ajustáveis. É através deste algoritmo que se estimam os parâmetros do corpo do neurónio, pesos sinápticos e *bias*, pela aprendizagem da amostra treino, calculando diversas vezes os parâmetros até o erro estar minimizado (Chauvin & Rumelhart, 1995).

As redes *multi-layer perceptron* são facilmente analisadas e interpretáveis, e, na maioria das vezes, podem ser implementadas de forma mais eficiente num software. As *MultiPerceptron*, do tipo *feedfoward*, são constituídas por pelo menos três camadas (Bishop, 1995; Haykin, 2008).

Em termos de tipologia e estrutura, as redes neuronais dividem-se em duas secções, redes *Feedforward* e *Feedback*.

As redes do tipo *Feedfoward*, são genericamente constituídas por uma *input layer*, *hidden layer*, e *output layer*, e baseiam-se num conjunto de neurónios conectados e distribuídos por diversas camadas. A informação segue em sentido único, sem quaisquer ciclos, desde os nodos de entrada, passando pelos nodos ocultos, caso existam, e por fim chegam aos nodos de saída (Ham & Kostanic, 2001; Haykin, 2008). A arquitetura das redes *feedforward* é as mais utilizada pela vantagem de ser facilmente construída com um algoritmo de otimização, isto porque as redes neuronais são estáticas, isto é, os pesos são fixados só depois do treino. Por outro lado, algumas desvantagens apontadas são: as previsões pouco satisfatórias pelo tamanho reduzido dos dados de treino e a rigidez a grandes mudanças que não foram aprendidas no treino (Chiang et al., 2004).

Relativamente às redes *Feedback*, estas são facilmente distinguíveis das redes *Feedforward* por serem dinâmicas, isto é, têm pelo menos um ciclo ou *loop* entre a output *layer* ou *hidden layer* e a *input layer*. O fluxo de informação caracteriza-se por ser bidirecional, ou seja, a informação de um nodo de saída retoma ao nodos de entrada ou a um dos nodos ocultos (Ham & Kostanic, 2001; Haykin, 2008). Uma das principais vantagens consiste na possibilidade de ajuste dos pesos, o que gera uma redução bastante eficiente da dimensão dos *inputs* e, consequentemente, do tempo de treino. Porém, estas características levam a que a estabilidade da rede seja por vezes difícil de determinar (Chiang et al., 2004).

2.7. Árvores de Decisão

O algoritmo Árvores de Decisão é bastante utilizado na tomada de decisões e é aplicado em problemas de classificação assim como em casos de regressão, por ser facilmente interpretável. As árvores de decisão, como as diversas formulações e variações do algoritmo que se têm vindo a desenvolver, ID3, C5.0, C4.5, ASSISTANT, CART são algoritmos classificadores que indicam todas as possibilidades lógicas de uma série de decisões, apresentam todas as ações alternativas possíveis e, por fim, preveem, com base em dados antecedentes, em que classe os dados estão inseridos (Alpaydin, 2014; Burkov, 2019).

As árvores de decisão são um algoritmo hierárquico em que as primeiras ocorrências condicionam as seguintes. Do tronco principal, nodo raiz, saem os ramos, e cada ramo é uma estrutura hierárquica (Alpaydin, 2014).

Uma árvore de decisão é um grafo acíclico utilizado para classificação e é composto por três tipos de nodos que formam uma árvore enraizada. Primeiramente o nodo raiz, o nodo do topo da árvore, onde o atributo mais importante é apresentado; depois os *internal decision nodes* que contêm o teste sobre os restantes atributos, isto é, dado um *input*, a cada nodo, é aplicado um teste e um dos ramos é realizado consoante o resultado; e por fim, as *terminal leaves*, onde, após todo o processo de teste ser repetido, é possível contemplar o valor do *output* do problema nas mesmas (Alpaydin, 2014; Rokach & Maimon).

Este algoritmo tem o seguinte processo: começa por ser um problema complexo, que é decomposto em sub problemas mais acessíveis. Em cada *internal decion node*, uma característica *j* de um vetor de características é estudada, caso a condição seja satisfeita um dos ramos é escolhido, caso contrário é eleito o ramo contrário. Todo este processo é repetido até se chegar às *terminal leaves* onde é determinada a que classe o *input* pertence (Burkov, 2019; Mitchell, 1997).

Algumas das principais vantagens consistem na fácil interpretabilidade e flexibilidade de junção de atributos, tanto categóricos como numéricos, comparando com outros classificadores. A rigidez a pequenas alterações nos dados é uma desvantagem apontada por, possivelmente, afetar negativamente a construção da árvore (Kingsford & Salzberg, 2008).

Existem dois paradigmas *ensembles learning: Bagging* e *Boosting*. *Ensembles* são métodos de aprendizagem que combinam vários modelos de ML pouco eficientes e criam um único modelo mais poderoso que o modelo inicial (Müller & Guido, 2017).

Bagging, ou agregação *Boostrap*, é uma técnica sugerida por Breiman em 1996, utilizada em classificação bem como em métodos de regressão, e tem como objetivo, reduzir a variância associada à previsão e, consequentemente, melhorar o processo de previsão (Sutton, 2005). O processo *Bagging* consiste primeiramente em criar muitas cópias dos dados de treino, cada uma ligeiramente diferente da outra, e de seguida aplicar o *weak learner* (modelos preditivos mais fracos) a cada cópia com o objetivo de se obterem e combinarem variados *weak learners* O conceito de *Bagging* é baseado na

utilização de uma simples média dos resultados para a obtenção da previsão. Por de trás do algoritmo *Random Forest* temos o paradigma *Bagging* (Burkov, 2019).

Tal como o *Bagging*, o *Boosting* é uma abordagem que pode ser utilizada para melhorar a precisão dos métodos de classificação e regressão. Porém, o *Boosting*, ao contrário do *Bagging* utiliza uma média ponderada dos resultados, que são obtidos através da aplicação de um método de previsão de várias amostras. Alguns exemplos de algoritmos *Boosting* são: *AdaBoost, XGBoost, LightGBM e CatBoost.*

2.7.1. *Random Forest*

O algoritmo *Random Forest* (RF), introduzido por Breiman em 2001, foi desenvolvido com o objetivo de reduzir o *overfitting* (Shalev-Shwartz & Ben-David, 2013). Dada uma base de dados são geradas amostras *boostrap* (amostras aleatórias com reposição em que depois de treinadas e categorizadas podem ser selecionadas para nova avaliação) e depois de treinadas e construídas várias de árvores de decisão, onde cada árvore é ligeiramente diferente das outras, é construído o modelo final com base na média de todas as previsões das variadas árvores (*ensemble learning*) (Cutler et al., 2012; Müller & Guido, 2017).

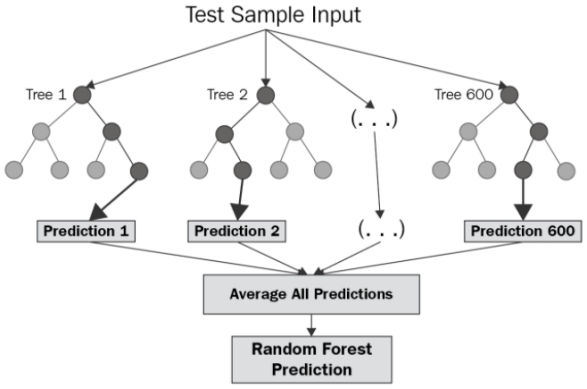

Figura 2 - Esquema Random Forest

Fonte: (Bakshi, 2020)

Mais concretamente, dado um vetor aleatório n-dimensional $X = (X_1, \dots, X_n)^T$, que representa os *inputs* e um variável aleatória Y, o objetivo consiste em encontrar a função previsão $f(x)$ para prever o valor Y. A função de previsão é determinada pela função de

perda dada pela equação, $L(Y, f(X))$ e é definida com o objetivo de minimizar o valor esperado da perda. Os *emsembles* constroem f com base em *weak learners* $h_1(x), \dots, h_J(x)$, que combinados geram o *ensemble* da previsão, $f(x)$. No caso de uma regressão, utiliza-se uma média dos *weak learners* para se obter a previsão, com base na equação

$$f(x) = \frac{1}{J}\sum_{j=1}^{J} h_j(x).$$

O algoritmo RF é considerado um modelo poderoso e preciso, e gerando bons resultados para problemas em que as variáveis não têm relação linear (Cutler et al., 2012).

2.7.2. *LightGBM*

Light Gradient Boosting Machine, *LightGBM*, introduzido pela Microsoft em 2017 é um algoritmo de *Machine Learning*, do tipo *Boosting*, popular por ser eficiente e ter uma excelente capacidade de suportar diversos meta-algoritmos como o GBDT, GBRT e GBM (Ju et al., 2019; Ke et al., 2016). O princípio por de trás destes algoritmos é a previsão com base em *weak leaners*, as árvores de decisão combinadas pela *ensemble learning* conduzem a melhores resultados (Chen et al., 2019).

Uma das principais características do *LightGBM* é a rapidez de processamento comparado com outros *grandient boosting trees*. Isto deve-se ao método de crescimento e construção das árvores baseado em folhas (*leaf-wise*) que escolhe sempre a folha que melhor divide a amostra de dados para ser expandida. A árvore cresce verticalmente.

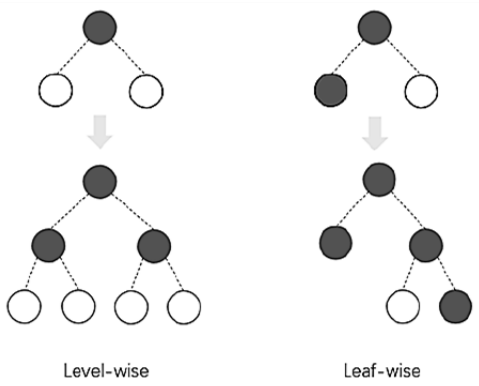

Figura 3 - Diagrama Light GBM

Fonte: (Ju et al., 2019)

Este algoritmo aplica técnicas para a seleção das variáveis independentes (*features)* dos nós das árvores, tais como, o *Gradient-Based One-Side Sampling* (GOSS) e *Exclusive Feature Bundling* (EFB). O primeiro, GOSS, seleciona apenas observações que têm maior gradiente, ou seja¸ as menos treinadas, e exclui as de menor gradiente aleatoriamente, tornando a estimativa mais precisa. Relativamente ao EFB, dado um grande número de *features*, esta metodologia elege apenas alguns *features* como vértices e adiciona arestas a cada dois, se não forem mutuamente exclusivos (Ke et al., 2017).

3. METODOLOGIA

É possível aplicar diferentes metodologias na elaboração de projetos de *Machine Learning*, tais como, CRISP-DM, KDD e SEMMA (Costa & Aparicio, 2020). A metodologia CRISP-DM, *Cross-Industry Standard Process for Data Mining*, apresenta diversas vantagens, como, a capacidade de resposta a qualquer tipo de projeto, pois não depende de uma só ferramenta para ser realizada e a sua ciclicidade, que torne possível avançar e retroceder no processo a qualquer momento. No contexto deste projeto, foram aplicadas as quatro primeiras das seis fases da metodologia CRISP-DM, apresentadas da Figura 6, *Business Understanding, Data Understanding, Data Preparation, Modeling, Evaluation e Deployment.*

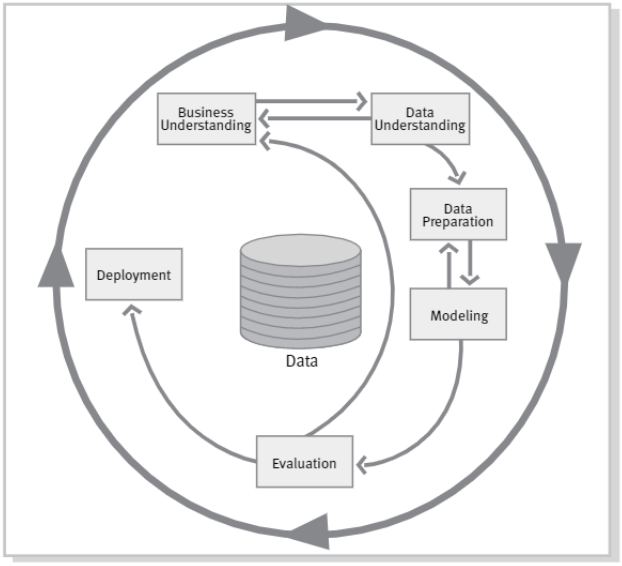

Figura 4 - Fases da metodologia de referência CRISP – DM

Fonte: (Chapman et al., 2000)

A primeira etapa, *Business Understanding*, diz respeito a toda a compreensão do negócio, missão, visão e valores da empresa, e definição de todos os objetivos e resultados expectáveis, bem como, a determinação das ferramentas a utilizar. Durante o estágio foram usadas as ferramentas Amazon Sage Maker e Jupyter.

Na segunda e terceira fases, *Data Understanding* e *Data Preparation*, selecionam-se, extraem-se e verifica-se a qualidade dos dados necessários e que serão utilizados na

modelação dos algoritmos. Nestas fases é feita a limpeza de dados e a transformação e seleção de atributos.

A quarta fase, *Modeling*, é a mais importante de todo o processo, por se tratar da fase onde são aplicados os algoritmos, previamente selecionados, aos dados em questão. Neste projeto foram selecionados: Regressão OLS, *Support Vector Machine*, Redes Neuronais, *Random Forest* e *LightGBM*. É também nesta fase que os dados são repartidos em duas amostras, amostra treino, da qual são criados os modelos, e a amostra teste, onde se avaliam os modelos.

Depois da modelação e estimação dos modelos, vem a quinta fase, *Evaluation*, onde é feita a avaliação e comparação dos resultados obtidos, confrontando com objetivos delineados para o negócio na primeira etapa.

A última fase, *Deployment*, baseia-se na implementação e disponibilização dos recursos e resultados ao utilizador final, como por exemplos a elaboração de um relatório e o desenvolvimento de um *software*.

4. TRABALHO EMPÍRICO

4.1. Compreensão do Negócio

Pode definir-se o valor Brix como a quantidade de compostos solúveis numa solução de sacarose. O valor Brix tem um papel relevante na tomada de decisão, não só na seleção do momento da colheita, mas também como um indicador qualitativo da produção. De facto, quanto maior o valor Brix, mais saudáveis, mais saborosas, mais nutritivas as frutas são e maior é a sua duração de vida. Na agricultura convencional, o Brix pode ser medido através de um refratómetro diretamente no campo, já na agricultura de precisão, pode utilizar-se um espectrómetro. E é desse espectrómetro que se tenta perceber se efetivamente existe alguma relação entre o que se observa nas imagens de satélite e o valor Brix.

Este é um tema que tem vindo a ser estudado nos últimos anos por diferentes autores. No caso deste estudo, os objetivos passam por tentar, primeiramente, perceber a importância do valor Brix a nível económico e financeiro para uma empresa para, posteriormente, mitigar o risco de produção e consequentemente aumentar o lucro. Para isso é necessário encontrar um modelo que explique os valores passados e preveja valores futuros com as mesmas características.

4.2. Compreensão e Preparação dos Dados

As imagens multiespectrais, que são a base dos dados do projeto, foram retiradas do satélite Sentinel-2 em parceria com a Forging Lab. A missão Sentinel-2, da European Space Agency (ESA) é composta por dois satélites multiespectrais que fornecem dados e imagens de alta resolução a cada cinco dias.

O Sentinel-2 é composto por 13 bandas espectrais, quatro delas com 10 metros de resolução espacial (B2, B3, B4 e B8), seis com 20 metros (B5, B6, B7, B8A, B11 e B12) e três com 60 metros (B1, B9 e B10) (Copernicus; The European Space Agency). Estes satélites cedem informações de monitoramento de terras, mais concretamente informações relacionadas com os seguintes temas, planeamento espacial, gestão hídrica, segurança e cobertura do solo tanto em práticas florestais como em práticas agrícolas (The European Space Agency).

Os valores referentes ao valor Brix, recolhidos pelo satélite Sentinel-2, no dia anterior ao da colheita dos produtos agrícolas, têm informação alusiva a 21 quintas. Note-se que cada quinta tem uma dimensão diferente, o que, consequentemente, implica que cada amostra tenha uma dimensão distinta. Numa imagem de satélite, um elemento tem a designação de pixel, que guarda todas as coordenadas, e a cada um está associado um valor de atributo, isto é, foi associado um valor Brix a cada observação.

Depois da recolha, todos os dados foram agregados num único ficheiro PKL para que a utilização dos mesmos seja mais intuitiva no Jupyter.

As variáveis que foram recolhidas para a realização deste trabalho podem ser lidas na Tabela 1, apresentada abaixo.

Tabela 1 - Descrição das variáveis recolhidas

Nome da Variável	**Descrição da Variável**
B1	banda espectral aerossol
B2	banda espectral azul
B3	banda espectral verde
B4	banda espectral vermelho
B5	banda espectral *red-edge* 1
B6	banda espectral *red-edge* 2
B7	banda espectral *red-edge* 3
B8	banda espectral NIR – infravermelho próximo
B8A	banda espectral *red-edge* 4
B9	banda espectral vapor de água
B11	banda espectral infravermelho de ondas curtas 1
B12	banda espectral infravermelho de ondas curtas 2
$Layer_j$	índice de vegetação j (j=1, …,11)

Depois da recolha dos dados, passou-se ao processo de preparação, tratamento e limpeza dos mesmos. Começou-se por identificar todos os valores nulos e valores que não estavam disponíveis em certas observações, *missing values* (NaN), e, posteriormente, foram eliminados com o intuito de não influenciar negativamente os resultados.

Seguiu-se a análise da correlação entre as variáveis para perceber a qualidade dos dados em duas vertentes distintas, inferência estatística e previsão. A primeira perspetiva,

da área estatística, utiliza os modelos para a aprendizagem do processo de geração de dados, porém requer uma série de testes e ajustes adequados, como a validação de ausência de colinearidade perfeita. A segunda, virada para o *Machine Learning*, aplica os modelos com o objetivo de prever os resultados para novos dados; o modelo é válido determinando empiricamente a perda na amostra teste e é importante perceber como é que cada variável independente se relaciona positiva ou negativamente com a variável dependente (Matthias, 2018).

A correlação é a medida que avalia se existe algum grau de dependência, causal ou não, entre as variáveis e é analisada através de coeficiente de Pearson. Este coeficiente, assume valores do intervalo [-1;1] e quanto mais perto dos extremos mais forte é a correlação. Através da função $.corr()$ verificou-se a existência de casos de colinearidade, isto é, algumas variáveis independentes revelaram ter correlação entre si.

Para resolver o problema de correlação, que pode vir a gerar *overfitting* nos modelos, aplicaram-se dois procedimentos: primeiro testou-se remover uma ou várias das variáveis com correlação substancialmente elevada, o que não altera significativamente o coeficiente de determinação, pois a informação das variáveis é redundante; segundo foi testado o método *Principal Components Analysis*, que consiste na conversão de um conjunto de variáveis com correlação entre si em *Principal Components*, conjunto de variáveis linearmente não correlacionadas, preservando a maior informação possível.

Em ambas as situações, acabou por se verificar uma grande redução de precisão e qualidade de ajuste do modelo aos dados iniciais. Por isso, decidiu-se manter a base de dados original com as 23 variáveis independentes e testar a modelação de três maneiras diferentes, a primeira com todas as variáveis, outra com apenas as primeiras 12 variáveis (B's), por fim, a terceira com as restantes variáveis (*Layers*).

É importante referir que de um ponto de vista de inferência seria necessário alterar o modelo e aplicar medidas e métodos que fossem ao encontro dos testes e ajustes adequados. Contudo, na perspetiva mais virada para o *Machine Learning*, apesar de ser importante verificar e solucionar o problema de colinearidade, dá-se mais importância ao facto de o modelo conseguir prever com qualidade ou não.

4.3. Modelação

Terminada a recolha e tratamento de dados, passou-se à fase de modelação dos variados algoritmos, sobre os dados selecionados. A fase de modelação divide-se em três etapas: invocação de bibliotecas, preparação dos dados e modelação do algoritmo. Todo o processo foi executado em linguagem de programação Python, criada por Guido van Rossum em 1991.

Neste trabalho foram utilizadas algumas das bibliotecas e módulos disponibilizados, apresentados na Tabela 2:

Tabela 2 - Descrição das bibliotecas utilizadas em Python

Nome biblioteca	Módulos	Descrição
pandas		Construção, manipulação e análise de grande volume de dados
statsmodels.api		Análise e estimação de dados e modelos estatísticos
sklearn.linear_model	LinearRegression	Implementação de modelos lineares
sklearn.SVM	SVR	Implementação de modelos baseados em SVR
sklearn.neural_network	MLPRegressor	Implementação de modelos baseados em redes neuronais
sklearn.ensemble	RandomForestRegressor	Implementação de modelos baseados em algoritmos *emsemble learning*
lightgbm		Implementações altamente otimizadas do Gradient Boosting
numpy		Processamento de grandes matrizes e matrizes multidimensionais
sklearn.preprocessing	StandardScaler	Estandardização dos dados
sklearn.model_selection	train_test_split	Divisão em amostra treino e teste
sklearn.metrics	mean_absolute_error, mean_squared_error, r2_score	Quantificar a qualidade das previsões

Seguiu-se a preparação dos dados. Neste processo a amostra, com 32749 observações, foi dividida em amostra treino, que conta com cerca de 80% das observações e em amostra de teste, que contém as restantes observações. Realizou-se igualmente a estandardização dos dados, com o comando *StandardScaler*, isto é, transformaram-se os dados, com

dimensões e amplitudes diferentes, num conjunto de dados que passou a ter média igual a zero e a variância unitária. Este processo torna possível a comparação de variáveis que têm medidas diferentes e é calculado pela seguinte fórmula: $z = \frac{x-u}{s}$, onde z é o valor estandardização da amostra x, u é a média da amostra de treino e s o desvio padrão da amostra de treino.

Posteriormente, vem a etapa de aplicação dos variados algoritmos de *Machine Learning*, onde foram aplicados diferentes parâmetros correspondentes a cada algoritmo. É de referir que nesta fase foram testadas as implementações com os três diferentes conjuntos de variáveis independentes. Como a estrutura de código em Python é bastante semelhante, optou-se por apresentar apenas o código para os modelos com as 23 variáveis.

Na Figura 7 é possível ver a estrutura de código referente ao algoritmo Regressão OLS.

```
# Import the model we are using
from sklearn.linear_model import LinearRegression
lr = LinearRegression()
# Train the model on training data
lr.fit(X_treino, y_treino);
#Prediction
predictions = lr.predict(X_teste)
```

Figura 5 - Código – Aplicação Regressão OLS

No caso Redes Neuronais (Figura 8) foi fixado 1000 como número máximo de interações (max_inter) e definiu-se na *hidden_layer_sizes* (Stathakis, 2009) que a rede teria, nas primeiras *hidden-layer*, $\sqrt{(m+n)N} + 2\sqrt{N/(m+2)}$ neurónios e a segunda camada $m\sqrt{N/(m+2)}$ neurónios, onde m é número de neurónios do *output*, neste caso um, e N é o número de neurónios na *input layer*, isto é, o número de variáveis que são 23.

```
#12 neurons -> 10 (1ºlayer) -> 2 (2º layer) -> 1 (output layer)
regressor=MLPRegressor(max_iter=1000, hidden_layer_sizes=(10,2))
regressor.fit(X_treino,y_treino.ravel())
```

```
MLPRegressor(hidden_layer_sizes=(10, 2), max_iter=1000)
```

Figura 6 - Código – Aplicação Redes Neuronais em Python

Quando aplicado o algoritmo *Support Vector Regression* (Figura 9), o *kernel*, função usada em SVR para auxílio de resolução dos problemas, escolhido foi o RBF (*Gaussian Radial Basis Function*), uma vez que este é habitualmente escolhido para dados não lineares.

```
#support vector regression
regressor = SVR(kernel="rbf")
regressor.fit (X_treino, y_treino.ravel())
```

Figura 7 - Código – Aplicação SVR em Python

No algoritmo *Random Forest* (Figura 10), fixou-se o *n_estimators* igual a 200, isto é, número de estimadores (*Decision Trees*) que são utilizados na *Random Forest*. Quanto maior este parâmetro mais preciso o modelo é. Em contrapartida, a complexidade de previsão aumenta e pode também gerar-se um problema de *overfitting*.

```
# Import the model we are using
from sklearn.ensemble import RandomForestRegressor
# Instantiate model with 1000 decision trees
rf = RandomForestRegressor(n_estimators = 200, random_state = 42)
# Train the model on training data
rf.fit(X_treino, y_treino);
```

Figura 8 - Código – Aplicação *Random Forest* em Python

Para o algoritmo *LightGBM* (Figura 11), foram definidos como parâmetros a taxa de aprendizagem (*learning rate*), inicialmente igual a 0,05, adaptou-se posteriormente para 0,1 quando ajustados os hiperparâmetros. Hiperparâmetros são parâmetros que se vão ajustando ao modelo e são usados para aperfeiçoar a previsão do modelo e/ou torná-lo mais rápido. Em relação ao *boosting_type* definiu-se o tradicional *Gradient Boosting Decision Tree*, *gbdt*. Como se trata de um caso de Regressão estipulou-se o *objective* como *regression*. A *sub_feature* seleciona aleatoriamente um subconjunto de *features* em cada iteração/árvore, neste caso, estabeleceu-se 0,5, isto é, o algoritmo *LightGBM* irá selecionar 50% das *features* antes do treino de cada árvore. De forma a evitar o problema

de *overfitting*, estipulou-se o *min_data*, número mínimo de dados numa *leaf*, como sete e *max_depth*, profunidade máxima da árvore, igual 10.

```
import lightgbm as lgb
train_data = lgb.Dataset(X_treino, label=y_treino)

params = {}
params['learning_rate'] = 0.1
params['boosting_type'] = 'gbdt'
params['objective'] = 'regression'
params['metric'] = ''
params['sub_feature'] = 0.5
params['num_leaves'] = 10
params['min_data'] = 7
params['max_depth'] = 10
params['force_col_wise'] = 'true'
num_round = 10
lgb.LGBMRegressor(verbose=-1)
bst = lgb.train(params,train_data ,num_round)
```

Figura 9 - Código – Aplicação *Light GBM* em Python

4.4. Comparação de modelos

Após a modelação e estimação dos modelos, segue-se a avaliação do desempenho de cada um, com o intuito de fazer uma análise comparativa e eleger o modelo que faz a melhor previsão dos dados em questão. Esta pode ser considerada uma das etapas mais complexas do processo, dado que existem inúmeras formas de avaliar e fatores externos que influenciam a tomada de decisão.

Para esta avaliação foram utilizados o coeficiente de determinação (R^2), o erro absoluto médio (EAM) e o erro quadrático médio (EQM).

O coeficiente de determinação é designado como a percentagem da variação da variável dependente que é explicada em função da variabilidade da variável independente. É dado pela seguinte razão:

$$R^2 = 1 - \frac{RSS}{TSS} = 1 - \frac{\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}{\sum_{i=1}^{n}(y_i - \bar{y})^2},$$

onde RSS e TSS correspondem à soma dos quadrados dos resíduos e soma total dos quadrados, respetivamente. Tratando-se de um problema de regressão, quanto mais perto R^2 está de um, mais próxima a nuvem de pontos prevista está da reta da regressão, quando

o R^2 se aproxima de zero, não se contempla um modelo linear, ou se está na presença de *outliers* (Martins, 2018; Renaud & Victoria-Feser, 2010).

O EAM é outra medida regularmente utilizada em estudos de avaliação de modelos de previsão. Este erro baseia-se na soma dos valores absolutos dos erros para obter um erro total (e_i) divido pela dimensão da amostra (n), admitindo que o peso de cada amostra (w_i) é igual a um, é calculado pela seguinte equação*:* $\text{EAM} = \frac{1}{n}\sum_{i=1}^{n}|e_i|$. São vários os autores, como Willmott & Matsuura (2005), que defendem que o desempenho de um modelo deve ser avaliado tendo por base o EAM.

O Erro quadrático médio, $\text{EQM} = \frac{1}{n}\sum_{i=1}^{n}(\hat{\theta}_i - \theta_i)^2$, é também utilizado para determinar o quão bem um modelo faz a previsão. Esta medida é facilmente adaptada a modelos com características diferentes. Um EQM mais próximo de zero representa uma menor variância dos dados, logo, um bom estimador.

5. DISCUSSÃO

O desempenho dos cinco algoritmos de *Machine Learning*, presentes neste estudo, foi avaliado pelas três métricas mencionadas anteriormente. De acordo com os resultados do presente trabalho, plasmados na Tabela 3, o modelo de regressão do algoritmo *Random Forest* apresentou ter alta precisão e melhores previsões do valor Brix para este conjunto de dados, com o coeficiente de determinação elevado ($R^2 = 0{,}8797$) e os erros absoluto médio e quadrático médio baixos (EAM = 0,2985, EQM = 0,2741).

Tabela 3 - Avaliação do desempenho dos algoritmos

	R^2	EAM	EQM
Regressão OLS	0,4767	0,8692	1,1923
SVR	0,6714	0,6122	0,7486
Redes Neuronais	0,7788	0,5309	0,5039
Random Forest	**0,8797**	**0,2985**	**0,2741**
LightGBM	0,4524	0,5514	0,5476

Verifica-se que o segundo melhor algoritmo desenvolvido, baseado nas três métricas, é o algoritmo Redes Neuronais. De facto, estudos como o de Abdullah Al-Sanabani et al., (2019) e o de Gunaratne et al. (2019), mostram que o algoritmo Redes Neuronais gerou melhores resultados para a previsão da qualidade de produtos agrícolas, tendo por base imagens de satélite, comparativamente com as abordagens lineares (Regressão OLS).

Com base nos resultados apresentados, pode dizer-se que o algoritmo *Support Vector Regression,* apresentou resultados medianos com o R^2= 0,6714. Porém se comparado com a Regressão OLS, considera-se um modelo melhor. Para corroborar esta ideia Liu et al. (2010), apresentam uma comparação direta entre os algoritmos SVM (LS-SVM) e *Multiple Linear Regression* (MLR), em que o primeiro mostrou ter uma capacidade de previsão superior à do segundo.

A investigação de Sexton et al. (2017), tem como objetivo encontrar um modelo de *Machine Learning* que preveja a qualidade da cana do açúcar através da medida *commercial cane sugar*, recorrendo a dados de satélite. Foram comparados alguns modelos de *Machine Learning*, tais como, SVR e Redes Neuronais, e, tal como no

presente estudo, o algoritmo Redes Neuronais apresentou melhores resultados com um coeficiente de determinação superior e um erro inferior.

6. CONCLUSÕES E TRABALHOS FUTUROS

O consumo excessivo de recursos naturais, consequente do aumento diário da população, é uma preocupação cada vez maior da sociedade, o que tem vindo a provocar alterações no modo de estar, tanto individualmente como a nível empresarial. Cada vez mais, as empresas pensam como podem combater o desperdício de forma a minorar a escassez de recursos e dar resposta às necessidades do mercado.

Atualmente, são desenvolvidas tecnologias e áreas como o *Machine Learning* e *Remote Sensing* que tentam dar resposta a estas dificuldades.

Ao longo do estudo utilizam-se várias abordagens de *Machine Learning* do ramo da aprendizagem automática, nomeadamente, Regressão Linear (OLS), *Support Vector Regression*, Redes Neuronais, *Random Forest* e *LightGBM*.

As observações de satélite e os algoritmos de *Machine Learning* podem ser utilizados para desenvolver diversos modelos de previsão para avaliar, designadamente, a qualidade de frutas e legumes.

Os resultados obtidos neste projeto contribuíram para o desenvolvimento de alternativas mais céleres na avaliação da qualidade agrícola, na medida em que o tempo de processamento desde a recolha de dados à previsão dos modelos é relativamente curto. Demonstrou ainda que a utilização do valor Brix para a avaliação da qualidade de produtos agrícolas é válida, sendo um bom indicador na tomada de decisão.

A utilização de imagens de satélite é uma das maiores vantagens deste projeto, uma vez que se falam de produtos agrícolas, expostos a condições climatéricas imprevisíveis a longo prazo. Estas imagens podem ser atualizadas em tempo real, o que permite ajustar as decisões em tempo útil.

Na comparação dos resultados de previsão obtidos pelas várias abordagens em estudo, verifica-se que os modelos em que se aplicou o algoritmo *Random Forest* geram maior precisão e menores erros de previsão. O melhor modelo, do algoritmo *Random Forest*, apresentou um coeficiente de determinação de 87,97%, com erro absoluto médio de 0,2985 e erro quadrático médio de 0,2741.

No decorrer do projeto, surgiram algumas limitações, das quais se salienta a presença de colinearidade entre as variáveis, o que, consequentemente, tornou difícil a seleção das variáveis a remover.

Sugere-se, como continuação e complemento deste trabalho: a análise e recolha de outro tipo de variáveis em que se verifique uma correlação menos evidente; uma extração de informação, do ponto de vista qualitativo, isto é, variáveis que são separadas por várias categorias e que representam uma classificação da amostra, com o intuito de se criarem modelos de Classificação em vez de Regressão; a análise de bases de dados constituídas por um elevado histórico de dados dos momentos importantes da evolução da qualidade dos produtos agrícolas e não apenas um único momento temporal.

REFERÊNCIAS BIBLIOGRÁFICAS

Abdullah Al-Sanabani, D. G., Solihin, M. I., Pui, L. P., Astuti, W., Ang, C. K., & Hong, L. W. (2019). Development of non-destructive mango assessment using Handheld Spectroscopy and Machine Learning Regression. *Journal of Physics: Conference Series*, *1367*(1). https://doi.org/10.1088/1742-6596/1367/1/012030

Albon, C. (2018). Machine Learning with Python Cookbook. In *Angewandte Chemie International Edition, 6(11), 951–952.* O'Reilly Media, Inc.

Alpaydin, E. (2014). *Introduction to Machine Learning* (3rd ed.). The MIT Press.

Aparicio, S., Aparicio, J. T., & Costa, C. J. (2019). Data Science and AI: trends analysis. In *2019 14th Iberian Conf on Information Systems and Technologies (CISTI)* (pp. 1-6). IEEE. https://doi.org/10.23919/CISTI.2019.8760820

Bakshi, C. (2020). *Random Forest Regression*. https://levelup.gitconnected.com/random-forest-regression-209c0f354c84

Ball, D. W. (2006) Concentration Scales for Sugar Solutions Journal of Chemical Education 83 (10) 1489 October, 1. https://doi.org/10.1021/ed083p1489

Bishop, C. M. (1967). Pattern Recognition and Machine Learning. In *Angewandte Chemie International Edition, 6(11), 951–952.*

Bishop, C. M. (1995). *Neural Networks for Pattern Recognition*. Oxford University Press.

Burkov, A. (2019). The Hundred-Page Machine Learning. (Vol. 1). Quebec City, QC, Canada: Andriy Burkov.

Chapman, P., Clinton, J., Kerber, R., Khabaza, T., Reinartz, T., Shearer, C., & Rudiger, Wirth. (2000). Crisp-DM 1.0 - Step-by-step data mining guide. In *CRISP-DM Consortium*.

Chauvin, Y., Rumelhart, D. E. (1995). *Backpropagation: Theory, architectures, and applications.* Lawrence Erlbaum Associates, Inc.

Chen, C., Zhang, Q., Ma, Q., & Yu, B. (2019). LightGBM-PPI: Predicting protein-protein interactions through LightGBM with multi-information fusion. *Chemometrics and Intelligent Laboratory Systems*, *191*(June), 54–64. https://doi.org/10.1016/j.chemolab.2019.06.003

Chiang, Y.-M., Chang, L.-C., & Chang, F.-J. (2004). Comparison of static-feedforward and dynamic-feedback neural networks for rainfall-runoff modeling. *Journal of Hydrology*, *290*(3–4), 297–311. https://doi.org/10.1016/j.jhydrol.2003.12.033

Coelho, J. P. C., & da Silva, J. R. M. (2009). *Agricultura de Precisão*, AJAP; 1st ed.

Copernicus. *O Copernicus em resumo*. https://www.copernicus.eu/pt-pt/acerca-do-copernicus/o-copernicus-em-resumo

Costa, C. & Aparicio, J. (2020) "POST-DS: A Methodology to Boost Data Science", 15th Iberian Conference on Information Systems and Technologies (CISTI), pp. 1-6, https://doi.org/10.23919/CISTI49556.2020.9140932

Costa, C., & Aparicio, J. (2021). A Methodology to Boost Data Science in the Context of COVID-19. In *Advances in Parallel & Distributed Processing, and Applications* (pp. 65-75). Springer, Cham.

Cutler, A., Cutler, D. R., & Stevens, J. R. (2012). Random Forest. In Z. Cha & Y. Ma (Eds.), *Ensemble Machine Learning*. Springer. https://doi.org/10.1007/978-1-4419-9326-7

González-Rivera, G. (2012). Forecasting for Economics and Business. In *Forecasting for Economics and Business*. Pearson Education, Inc. https://doi.org/10.4324/9781315510415

Gunaratne, T. M., Viejo, C. G., Gunaratne, N. M., Torrico, D. D., Dunshea, F. R., & Fuentes, S. (2019). Chocolate quality assessment based on chemical fingerprinting using near infra-red and machine learning modeling. *Foods*, *8*(10). https://doi.org/10.3390/foods8100426

Ham, F. M., & Kostanic, I. (2001). *Principles of neurocomputing for science and*

engineering. McGraw-Hill.

Haykin, S. (2008). Neural Networks and Learning Machines. In *Biophysics* (3rd ed.). Pearson. https://doi.org/10.1007/978-3-030-44146-3_14

Jensen, J. (2000). *Remote Sensing of the Environment: An Earth Resource Perspective*. Pearson

Johnston, J., & DiNardo, J. (1996). *Econometric Methods* (4th ed.). McGraw-Hill.

Ju, Y., Sun, G., Chen, Q., Zhang, M., Zhu, H., & Rehman, M. U. (2019). A model combining convolutional neural network and lightgbm algorithm for ultra-short-term wind power forecasting. *IEEE Access*, *7*, 28309–28318. https://doi.org/10.1109/ACCESS.2019.2901920

Ke, G., Meng, Q., Finley, T., Wang, T., Chen, W., Ma, W., Ye, Q., & Liu, T. Y. (2017). LightGBM: A highly efficient gradient boosting decision tree. *Advances in Neural Information Processing Systems*, *2017-Decem*(Nips), 3147–3155.

Ke, G., Ye, Q., Chen, W., & Liu, T. Y. (2016). *LightGBM*. https://www.microsoft.com/en-us/research/project/lightgbm/

Kelleher, J. D., Mac Namee, B., & D'Arcy, A. (2015). *Fundamentals of Machine Learning for Predictive Data Analytics*. MIT Press.

Kingsford, C., & Salzberg, S.L. (2008). What are decision trees? *Nature Biotechnology, 26*, 1011-1013.

Kleinhenz, Matthew D.; Bumgarner, N. R. (2012). Using Brix as an indicator of vegetable quality. *The Ohio State University Extension*, HYG-1651-12.

Lee, J. S., Kim, S. C., Seong, K. C., Kim, C. H., Um, Y. C., & Lee, S. K. (2012). Quality prediction of kiwifruit based on near infrared spectroscopy. *Korean Journal of Horticultural Science and Technology*, *31*(1), 709–717. https://doi.org/10.7235/hort.2012.12139

Liaghat, S., & Balasundram, S. K. (2010). A review: The role of remote sensing in

precision agriculture. *American Journal of Agricultural and Biological Science*, *5*(1), 50–55. https://doi.org/10.3844/ajabssp.2010.50.55

Liu, Y. De, Gao, R. J., Sun, X. D., OuYang, A. G., Pan, Y. Y., & Dong, X. (2010). Predicting brix of intact pears by a portable NIR spectrometry with LS-SVM. *Proceedings - 2010 6th International Conference on Natural Computation, ICNC 2010*, *2*(Icnc), 909–913. https://doi.org/10.1109/ICNC.2010.5583908

Maiese, K. (2021). *Nervos*. https://www.msdmanuals.com/pt-pt/casa/distúrbios-cerebrais,-da-medula-espinal-e-dos-nervos/biologia-do-sistema-nervoso/nervos#

Martins, M. E. G. (2018). Coeficiente de determinação. *Revista de Ciência Elementar*, (01). https://doi.org/10.24927/rce2018.024

Matthias, D. (2018). *Inference vs Prediction - Generative modeling or predictive modeling?*

https://www.datascienceblog.net/post/commentary/inference-vs-prediction/

McCullahg, P., & Nelder, J. A. (1989). *Generalized Linear Models* (2nd ed.). Chapman & Hall.

Mitchell, T. M. (1997). *Machine Learning* (1st ed.). McGraw-Hill

Molin, J. P., Amaral, L. R. do, & Colaço, A. F. (2015). *Agricultura de Precisão* (1st ed). oficinas tecnicas

Müller, A. C., & Guido, S. (2017). *Introduction to Machine Learning with Python*. O'Reilly Media, Inc. https://doi.org/10.1007/978-3-030-36826-5_10

Pronprasit, R., & Natwichai, J. (2013). Prediction of Mango Fruit Quality from NIR Spectroscopy using an Ensemble Classification. *International Journal of Computer Applications*, *83*(14), 25–30. https://doi.org/10.5120/14517-2903

Renaud, O., & Victoria-Feser, M.-P. (2010). A robust coefficient of determination for regression. *Journal of Statistical Planning and Inference*, *140*(7), 1852–1862. https://doi.org/10.1016/j.jspi.2010.01.008

Rokach, L., & Maimon, O. Classification Trees. In O. Maimon & L. Rokach (Eds.), *Data Mining and Knowledge Discovery Handbook* (2nd ed.). Springer.

Samadani, S., & Costa, C. J. (2021). Forecasting real estate prices in Portugal: A data science approach. In 2021 16th Iberian Conf. on Inf. Systems and Technologies (CISTI) (pp.1-6). IEEE. https://doi.org/ 10.23919/CISTI52073.2021.9476447

Saranwong, S. I., Sornsrivichai, J., & Kawano, S. (2003). Performance of a Portable near Infrared Instrument for Brix Value Determination of Intact Mango Fruit. *Journal of Near Infrared Spectroscopy*, *11*(3), 175–181. https://doi.org/10.1255/jnirs.364

Sexton, J., Everingham, Y., & Donald, D. (2017). *Comparison of Data Mining Algorithms for On-line NIR Models of CCS in Australia*. *June*, 16–17. https://doi.org/10.13140/RG.2.2.33779.53289

Shalev-Shwartz, S., & Ben-David, S. (2013). Understanding machine learning: From theory to algorithms. In *Understanding Machine Learning: From Theory to Algorithms*. Cambridge University Press. https://doi.org/10.1017/CBO9781107298019

Sharma, S., Sharma, S., & Athaiya, A. (2020). Activation Functions in Neural Networks. *International Journal of Engineering Applied Sciences and Technology*, *04*(12), 310–316. https://doi.org/10.33564/ijeast.2020.v04i12.054

Silva, S. de A., Queiroz, D. M. de, Pinto, F. de A. C., & Santos, N. T. (2014). Coffee quality and its relationship with Brix degree and colorimetric information of coffee cherries. Precision Agriculture, 15, 543–554. https://doi.org/10.1007/s11119-014-9352-y

Smola, A. J., & Schölkopf, B. (2004). A tutorial on support vector regression. *Statistics and Computing*, *14*(3), 199–222. https://doi.org/10.1023/B:STCO.0000035301.49549.88

Stathakis, D. (2009). How many hidden layers and nodes? *International Journal of Remote Sensing*, *30*(8), 2133–2147. https://doi.org/10.1080/01431160802549278

Sutton, C. D. (2005). Classification and Regression Trees, Bagging, and Boosting. *Handbook of Statistics*, *24*(04), 303–329. https://doi.org/10.1016/S0169-7161(04)24011-1

The European Space Agency. (Acedido a 04/11/2021). *Introducing Sentinel-2*. https://www.esa.int/Applications/Observing_the_Earth/Copernicus/Sentinel-2/Introducing_Sentinel-2

The European Space Agency. (Acedido a 20/11/2021). *Spatial Resolution*. https://sentinels.copernicus.eu/web/sentinel/user-guides/sentinel-2-msi/resolutions/spatial

Üstün, B., Melssen, W. J., & Buydens, L. M. C. (2007). Visualisation and interpretation of Support Vector Regression models. *Analytica Chimica Acta*, *595*(1-2 SPEC. ISS.), 299–309. https://doi.org/10.1016/j.aca.2007.03.023

Vapnik, V. N. (1998). *Pattern Recognition-Statistical Learning Theory*.

Vapnik, V. N. (1999). *The Nature of Statistical Learning Theory* (2nd ed.). Springer.

Willmott, C. J., & Matsuura, K. (2005). Advantages of the mean absolute error (MAE) over the root mean square error (RMSE) in assessing average model performance. *Climate Research*, *30*, 79–82. https://doi.org/10.3354/CR030079

Wooldridge, J. M. (2012). Introductory Econometrics - A Modern Approach. In *Introductory Econometrics* (5th ed.). South-Western Cengage Learning. https://doi.org/10.4324/9780203157688

ANEXO

Tabela 4 - Métricas de Avaliação para todos os modelos

		R^2	EAM	EQM
Regressão OLS	B's	0,408	0,9151	1,3503
	Layers	0,262	1,026	1,7185
	Completo	0,487	43,4923	2821,1432
SVR	B's	0,6764	0,5895	0,7372
	Layers	0,4464	0,8213	1,2612
	Completo	0,6714	0,6122	0,7486
Redes Neuronais	B's	0,6320	0,7026	0,8384
	Layers	0,4795	0,8385	1,1858
	Completo	0,7788	0,5309	0,5039
RF	B's	0,8767	0,2989	0,2809
	Layers	0,6691	0,5721	0,7540
	Completo	0,8797	0,2985	0,2741
LightGBM	B's	0,4096	0,5872	0,5904
	Layers	0,2238	0,6769	0,7762
	Completo	0,4524	0,5514	0,5476