

2018

Machine Learning Puzzles: How to select Use Cases, Algorithms and Technologies?

Ana Carolina Ribeiro

Universidade do Minho, pg3305@alunos.uminho.pt

Rui Frazão

Universidade de Aveiro, ruifilipefrazao@ua.pt

Jorge Oliveira e Sa

Universidade do Minho, Centro Algoritmi, jos@dsi.uminho.pt

Follow this and additional works at: <https://aisel.aisnet.org/capsi2018>

Recommended Citation

Ribeiro, Ana Carolina; Frazão, Rui; and Oliveira e Sa, Jorge, "Machine Learning Puzzles: How to select Use Cases, Algorithms and Technologies?" (2018). *2018 Proceedings*. 30.

<https://aisel.aisnet.org/capsi2018/30>

This material is brought to you by the Portugal (CAPSI) at AIS Electronic Library (AISeL). It has been accepted for inclusion in 2018 Proceedings by an authorized administrator of AIS Electronic Library (AISeL). For more information, please contact elibrary@aisnet.org.

Quebra-cabeças Machine Learning: Como selecionar Use Cases, Algoritmos e Tecnologias?

Machine Learning Puzzles: How to select Use Cases, Algorithms and Technologies?

Ana Carolina Ribeiro, Universidade do Minho, Portugal, pg3305@alunos.uminho.pt

Rui Frazão, Universidade de Aveiro, Portugal, ruifilipefrazao@ua.pt

Jorge Oliveira e Sá, Universidade do Minho, Centro Algoritmi, Portugal, jos@dsi.uminho.pt

Resumo

Atualmente, a aplicação de uma abordagem *Machine Learning* por investigadores ou empresas num determinado projeto, ainda é considerado um processo com um nível de dificuldade elevado. Isto acontece, porque é necessário conhecer com algum detalhe os algoritmos para entender o que na prática fazem e assim perceber se são os mais adequados para resolver o problema. Para além disto, é importante encontrar a tecnologia que satisfaça as necessidades computacionais do projeto, que suporte determinados algoritmos e que permita um bom desempenho com grandes quantidades de dados. Assim, este artigo, que é um trabalho em progresso, pretende apresentar um *framework* que ajude e, de certa forma, guie os investigadores num projeto de *Machine Learning* no que respeita à seleção de *Use Cases*, algoritmos e tecnologias de *Machine Learning*.

Palavras-chave: Machine Learning; Algoritmos; Tecnologias; Use Cases; Framework.

Abstract

Currently, the application of a Machine Learning approach by researchers or companies in a project is still considered a process with a high level of difficulty. This is because it is necessary to know in some detail the algorithms and how they can solve the problem. In addition, it is important to find the technology that meets the needs of computational project, which support certain algorithms and enabling a good performance with large amounts of data. So, this paper, a work in progress, intends to introduce a framework to help and, in a way, guide researchers on a project of Machine Learning to choose the Use Cases, algorithms and technologies.

Keywords: Machine Learning; Algorithms; Technologies; Use Cases; Framework.

1. INTRODUÇÃO

O presente artigo tem como objetivo apresentar um *framework* (conjunto de conceitos para resolver um problema de um domínio específico) que permita entender quais os *Use Cases*, algoritmos e tecnologias de *Machine Learning* que melhor se adequam a determinados *Business Cases*. Para isto, foram estudados vários algoritmos originando uma proposta de *framework* para utilização em projetos de *Machine Learning*.

O desenvolvimento deste *framework* tem como objetivo ajudar os investigadores desta área a utilizar e aplicar os algoritmos e as tecnologias de *Machine Learning* que melhor se adequam ao seu *Business Case*. Este *framework* deverá permitir aos investigadores que têm conhecimentos na área bem como aqueles que apenas tenham interesse ou estão a iniciar na área do *Machine Learning*, consigam escolher os *Use Cases*, algoritmos e as tecnologias que melhor se adequam ao problema a resolver.

A aplicação de métodos de *Machine Learning* nos últimos anos tornou-se recorrente no dia-a-dia de investigadores de todas as áreas e de muitas empresas como por exemplo, a Amazon, o Facebook e a Netflix. Desde as recomendações automáticas de filmes ao reconhecimento facial através de imagens, muitos *sites* de organizações e dispositivos já incorporaram os algoritmos de *Machine Learning* (Guido & Muller, 2016).

Atualmente, as organizações recolhem uma enorme quantidade de dados e, para que os dados se tornem valiosos para a organização, é necessário que sejam analisados de forma a que sejam extraídas perceções entre as características descritivas (atributos ou características que descrevem os dados) e a característica denominada *target* (atributo ou característica alvo, ou seja, atributo que é utilizado como *label* em problemas de *supervised learning*) no conjunto de dados (Coelho & Richert, 2015). O objetivo da aplicação do *Machine Learning* é ensinar máquinas a realizar tarefas, fornecendo-lhes alguns exemplos. O processo de criar uma abordagem de *Machine Learning* é demorado e iterativo, isto é, volta-se à análise de dados experimentando diferentes versões dos dados de entrada em diversos conjuntos de algoritmos de *Machine Learning* (Coelho & Richert, 2015).

A estrutura deste artigo, que é um trabalho em progresso, é composta por: secção 2 é descrito o conceito de *Machine Learning* e apresenta vários tipos de algoritmos; secção 3 são descritos vários *Use Cases*; secção 4 são apresentadas várias tecnologias de *Machine Learning*; secção 5 é apresentado o *framework* de *Machine Learning* onde são combinados *Use Cases*, algoritmos e tecnologias de *Machine Learning*; finalmente na secção 6 são apresentadas as conclusões.

2. MACHINE LEARNING

Machine Learning, tem como objetivo principal a extração de conhecimentos a partir dos dados. Esta área de investigação apresenta uma forte interseção entre estatística, inteligência artificial e ciência da computação (Guido & Muller, 2016).

Tom Mitchell define que *Machine Learning* é uma técnica que permite que os sistemas de computação aprendam e melhorem automaticamente a partir da experiência (através dos dados disponíveis), o mesmo autor apresenta uma definição mais direcionada para a engenharia “Um programa aprende com a experiência E em relação a uma determinada tarefa T e uma medida de desempenho P, se o seu desempenho na tarefa T, medido por P, melhora com a experiência E” (Géron, 2017).

No desenvolvimento de um modelo de previsão utilizando *Machine Learning*, é utilizado um conjunto de dados de treino para fazer previsões baseadas nos dados. Um processo crucial na fase de treino dos dados é o *feature selection*, uma vez que é fundamental que as *features* escolhidas tenham um impacto significativo no modelo de previsão, elevando assim, a eficiência do modelo. Dependendo da forma como o modelo de previsão é treinado, os sistemas de *Machine Learning* apresentam as seguintes categoriais principais: *supervised learning*, *unsupervised learning* e *semisupervised* (Lampropoulos & Tsihrintzis, 2015).

Nos sistemas de *supervised learning*, os dados de treino dos algoritmos incluem as soluções desejadas, chamadas de *labels* (Géron, 2017). As técnicas *supervised learning* permitem o desenvolvimento de um modelo onde as relações entre as *features* descritivas e a *feature* designada *target*, são baseadas num histórico ou instâncias. A utilização deste tipo de modelo permite obter previsões para novas instâncias (Kelleher, Mac Namee, & D'Arcy, 2015). Um *Use Case* típico utilizado em *supervised learning* é a classificação e, um bom exemplo é o filtro de *spam*: o algoritmo é treinado com um conjunto variado de exemplos de *e-mails* agrupados em classe que, posteriormente, deve aprender como classificar novos *e-mails* (Géron, 2017). Outro *Use Case* muito utilizado é a regressão: quando se pretende prever um valor *target* numérico, como o rendimento de clientes de um banco, considerando um conjunto de *features* (rendimento mensal bruto e líquido, despesas mensais, créditos, etc.). Para treinar o modelo, é necessário fornecer vários exemplos de rendimentos, incluindo os preditores e as *labels* (ou seja, os rendimentos dos clientes). Os algoritmos considerados mais importantes e utilizados em problemas de *supervised learning* são: k-Nearest Neighbors (k-NN), Regressão Linear, Regressão logística, Máquinas de Vetores de Suporte (SVM), Árvores de decisão, Random Forests e Redes Neurais (Géron, 2017). É importante referir que alguns algoritmos de regressão podem ser usados para classificação e vice-versa.

Relativamente aos sistemas de *unsupervised learning*, ao contrário do caso anterior, os dados de treino não apresentam *labels*. Isto significa que, apenas são recebidas as entradas sem incluir os resultados desejados (o que acontece em *supervised learning* com a *target*), mas o modelo visa encontrar padrões ocultos nos dados baseados em propriedades estatísticas dos conjuntos de dados de treino. Este tipo de sistema de *Machine Learning* é considerado menos definido em relação ao sistema de *supervised learning*, uma vez que não é explícito que tipo de padrões procurar e não há nenhuma métrica de erro óbvia a ser utilizada (ao contrário da abordagem *supervised learning*, onde é possível comparar a previsão y para um dado valor x observado) (Murphy, 2012). Alguns dos mais importantes algoritmos de *unsupervised learning* são: Clustering, K-means, Visualização e Redução de dimensionalidade, Análise de Componentes Principais, Apriori, Regras de Associação. Um algoritmo muito utilizado nesta abordagem é o Clustering. Por exemplo, para problemas de agrupamento e dados em vários domínios (por exemplo, análise de sequência, pesquisas de mercado, reconhecimento de objetos, análise de redes sociais, etc.), os algoritmos de *clustering* permitem

subdividir cada grupo em grupos menores, ajudando a segmentar o problema (Géron, 2017). Os algoritmos de visualização são também bons exemplos de algoritmos de *unsupervised learning*. Estes algoritmos tentam preservar o máximo possível da estrutura dos dados, para que seja possível entender como é que os dados estão organizados e, talvez, identificar padrões suspeitos (Géron, 2017).

Considerando agora os sistemas *semi-supervised*, os algoritmos desta abordagem conseguem lidar com dados de treino parcialmente rotulados, um exemplo deste tipo de abordagem, é o serviço de fotografias da Google, por exemplo depois de todas as fotografias serem carregadas para esta aplicação, ela reconhece automaticamente que a mesma pessoa aparece nas fotos 1, 5 e 11, enquanto outra pessoa aparece nas fotos 2, 5 e 7 (Géron, 2017). Esta é considerada a parte *unsupervised* do algoritmo (*clustering*). Posteriormente o algoritmo tenta prever quem são essas pessoas, que diz respeito à parte *supervised* do algoritmo. Com este exemplo, é explícito que a maioria dos algoritmos *semisupervised* são combinações de algoritmos *unsupervised* e *supervised*. Por exemplo, o algoritmo Deep Belief Networks, baseia-se em componentes *unsupervised* chamados Restricted Boltzmann Machines que são treinados sequencialmente de forma *unsupervised* e, então, todo o sistema é aperfeiçoado utilizando técnicas de *supervised learning* (Géron, 2017).

3. USE CASES

De uma forma geral, para iniciar um projeto, é apresentado aos investigadores um *Use Case* e, a partir daqui os investigadores iniciam o processo de estudo do problema. No âmbito do *Machine Learning*, existem vários *Business Cases* desde a segurança dos dados, marketing, recomendações, saúde, entre outros que permitem aos investigadores entender o âmbito do seu problema e aplicar *Machine Learning*.

Apesar disto, os investigadores antes de escolherem o *Business Case* deverão responder à seguinte pergunta: Qual é o tipo de problema a resolver? É nesta altura que os investigadores são confrontados com os *Use Cases* gerais de *Machine Learning*, nomeadamente: regressão, classificação (*multi-class* e *two-class*), *clustering* e deteção de anomalias.

A regressão é uma técnica de modelação preditiva que investiga a relação entre uma variável dependente (alvo) e uma ou várias variáveis independentes (preditores). Esta técnica é utilizada para previsão, modelação de séries temporais e para encontrar a relação de efeito causal entre as variáveis.

Relativamente à classificação, podemos estar perante dois *Use Cases*:

- Classificação de duas classes ou classificação binária – Neste tipo de classificação existem apenas dois resultados: sim ou não, sucesso ou insucesso, 0 ou 1. Um exemplo de classificação binária é a identificação do género (masculino ou feminino). Neste tipo de

classificação, estamos perante duas classes para prever, por exemplo, como identificar se uma pessoa é do sexo masculino ou feminino. O *output* de muitos algoritmos de classificação binária é uma pontuação de previsão. Um problema de classificação binária tem, possivelmente, apenas dois resultados: sim ou não, sucesso ou insucesso, 0 ou 1.

- Classificação multi-classe – Neste tipo de classificação, significa que existem mais do que duas classes para prever, por exemplo, classificar um conjunto de imagens de frutas que podem ser laranjas, maçãs ou peras. A classificação multi-classe supõe que cada amostra é atribuída a uma e apenas uma classe, ou seja, um fruto pode ser uma maçã ou uma pera, mas não pode ser os dois ao mesmo tempo.

O *clustering*, em *Machine Learning*, também conhecido por segmentação, é um método de agrupamento dos dados em *clusters* semelhantes e permite a localização de uma estrutura numa coleção de dados sem *labels* (técnica de *unsupervised learning*). O *cluster* consegue encontrar a relação entre pontos de dados para que estes possam ser segmentados.

A deteção de anomalias, também conhecida como deteção de valores discrepantes (*outliers*), é um processo de exploração de dados utilizado para detetar *outliers* presentes num determinado conjunto de dados e para determinar detalhes sobre as suas ocorrências. Inicialmente, a deteção de anomalias encontra valores que são procurados e o seu padrão e, através disso, encontra os *outliers*. Posteriormente, um modelo de classificação é construído para atribuir novos valores aos *outliers* identificados.

Como as anomalias são eventos raros por definição, pode ser difícil recolher uma amostra representativa de dados para usar na modelação. Os algoritmos incluídos nesta categoria foram especialmente projetados para abordar os principais desafios de construir e treinar modelos utilizando conjuntos de dados não balanceados.

4. TECNOLOGIAS MACHINE LEARNING

Atualmente, as empresas têm acesso a uma grande quantidade de dados (*Big Data*) a partir de várias fontes, como sensores, *media*, análises, clientes, etc. Grandes empresas como a Google, IBM, Amazon e Microsoft ajudam as outras empresas a processar grandes quantidades de dados e constroem *Application Programming Interfaces* (APIs) de *Machine Learning* para que consigam, da melhor forma, utilizar as tecnologias de *Machine Learning*.

As APIs de *Machine Learning* dispensam as complexidades envolvidas na criação e na implementação de modelos de *Machine Learning* para que os investigadores apenas se concentrem nos dados, no *design*, na experimentação e na capacidade de extração de informações a partir dos dados. As APIs de *Machine Learning* facilitam aos investigadores aplicar *Machine Learning* a um conjunto de dados para adicionar recursos preditivos às suas aplicações. Estas APIs fornecem às

empresas a capacidade de reunir análises preditivas para que estas conheçam melhor os seus clientes, compreendam os seus requisitos e forneçam produtos ou serviços com base nas tendências de dados, iniciando assim o processo de venda. Desta forma, os desenvolvedores de APIs de *Machine Learning*, procuram apresentar novos e inovadores recursos que podem ajudar a economizar tempo.

Pode-se verificar então que, para além da compreensão de todos os algoritmos *Machine Learning* e os seus *Use Cases*, a escolha da tecnologia é um fator determinante no projeto a desenvolver. Algumas características a ter em consideração são os algoritmos que a tecnologia suporta, as linguagens que apresentam como requisito e a popularidade que têm na comunidade.

Segundo o quadrante da Gartner, as tecnologias com melhor posicionamento são, por exemplo, o H2O, o KNIME e o RapidMiner.

5. FRAMEWORK DE MACHINE LEARNING

Neste artigo é proposto um *framework* de *Machine Learning* que tem como objetivo a compilação do estudo de *Use Cases*, algoritmos e tecnologias de *Machine Learning*. A construção deste *framework* envolve a interligação e a permutabilidade entre as 3 dimensões estudadas que permite a qualquer investigador, perante um problema de *Machine Learning*, ter um ponto de início de investigação. A interligação e permutabilidade entre as 3 dimensões são fatores importantes no *framework* porque, a partir de uma das dimensões é possível saber o que melhor se adequa às outras 2 dimensões do problema. Esta ligação das 3 dimensões é a base deste *framework*, facilitando as escolhas de algoritmos e tecnologias de *Machine Learning* que nem sempre são fáceis de entender a sua aplicabilidade.

Como referido anteriormente, quando um investigador está perante um problema de *Machine Learning*, existe frequentemente alguma dificuldade em saber quais os algoritmos e as tecnologias mais apropriadas a aplicar ao *Use Case* em questão. Para facilitar este processo, a construção deste *framework* permitirá que o investigador a partir de um *Use Case* consiga perceber quais os algoritmos mais apropriados, bem como as tecnologias que existem e que deve selecionar e utilizar. É importante referir que não é obrigatório iniciar a pesquisa no *framework* pelo *Use Case*, é também possível a partir da escolha de um algoritmo ou algoritmos, saber qual o *Use Case* que melhor se adequa, tal como a tecnologia. A Figura 1 pretende esclarecer esta situação e demonstrar que é possível partir de qualquer um dos fatores: *Use Cases*, Algoritmos ou Tecnologias determinar os outros dois fatores.

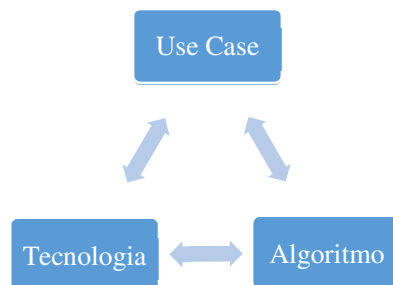


Figura 1 – Ciclo de interação entre os algoritmos de Machine Learning, Use Cases e Tecnologias.

Para a construção deste *framework* foram estudados vários algoritmos de *Machine Learning*: Regressão Linear, Regressão Logística, Árvores de Classificação e Regressão, C4.5, Random Decision Forest, Naive Bayes, k-NN, Perceptron, Adaboost, K-Means, Apriori, entre outros. O objetivo é abranger o maior número de algoritmos, nomeadamente os mais utilizados. Para iniciar o desenvolvimento do *framework* foram analisadas as seguintes características dos algoritmos:

- **Tipo de Algoritmo** – Nesta característica os algoritmos estão classificados como *Supervised Learning* ou *Unsupervised Learning*.
- **Acuidade do Algoritmo** – A característica que avalia a acuidade, tal como maioria, depende sempre do tipo de dados do investigador e a sua qualidade. Os algoritmos podem ser classificados como algoritmos de “Baixa”, “Alta” ou “Média” acuidade.
- **Tempo de Treino** – Esta característica pretende que os investigadores consigam saber antecipadamente como é que os algoritmos se comportam na fase de treino. Os algoritmos podem ser classificados como: “Rápido”, “Lento” ou “Mediano”.
- **Linear** – Neste campo de análise, os algoritmos são classificados como lineares ou não lineares. Os algoritmos de classificação linear partem do princípio de que as classes podem ser separadas por uma linha reta e que as tendências dos dados seguem esta linha, por exemplo, a Regressão Logística e as Máquinas de Suporte Vetorial (SVM).
- **Custo Computacional** – Nesta característica é apresentada a fórmula do custo computacional de cada algoritmo de *Machine Learning*. É importante ter em conta que, com os algoritmos de *Machine Learning*, quanto maior o volume de dados e a sua complexidade, maior custo computacional é exigido.
- **Big Data** – Esta característica permite avaliar se os algoritmos têm a capacidade de escalabilidade dos dados, ou seja, se estão aptos para um aumento exponencial de dados.

Todas estas características foram estudadas para todos os algoritmos acima mencionados. No entanto, na Tabela 1, será apenas apresentado um exemplo demonstrativo de três algoritmos de *Machine Learning*. Para além das características, serão também apresentados para cada algoritmo, os *Use Cases* e as tecnologias que suportam de forma a ser mais fácil para o investigador perceber qual o algoritmo que deve usar tendo em conta o seu *Use Case* e as tecnologias que poderá utilizar.

Tal como para os algoritmos, algumas tecnologias de *Machine Learning* foram estudadas com algum detalhe, com o objetivo de perceber quais são as mais utilizadas na comunidade científica. As tecnologias analisadas foram as seguintes: Spark MLlib, TensorFlow, Scikit-Learn, Weka, Dask, Caffe, Microsoft Cognitive Toolkit, MXNet, Accord.NET, Theano, PaddlePaddle, DeepLearning4j, Apache Singa, Apache Mahout, KNIME, Vowpal Wabbit, R, Keras, XGBoost Pytorch, RapidMiner, OpenCV e Chainer (Brownlee, 2016; Géron, 2017).

Tabela 1 - Algoritmos Machine Learning

USE CASE	TIPO DE ALGORITMO	ALGORITMO	TECNOLOGIAS	ACUIDADE	TEMPO DE TREINO	LINEAR	CUSTO COMPUTACIONAL	BIG DATA
Deteção de Anomalias; Classificação Binária; Classificação Multi-classe.	Supervised Learning	Regressão Logística	Scikit-Learn; R; OpenCV; Spark MLlib; TensorFlow; H2O; Weka; Theano; PaddlePaddle RapidMiner.	Baixa	Rápido	Sim	$O(m)$, onde m é o número de entradas diferentes de zero na matriz do sistema linear.	Sim
Classificação binária; Classificação Multi-classe; Deteção de Anomalias.	Supervised Learning	k-NN	Spark; Scikit-Learn; DeepLearning4j;	Alta	Médio	Não	$O(n_{\text{dimensões}} * n_{\text{amostras}}^2)$. n_{amostras} : número de amostras; $n_{\text{dimensões}}$: número de dimensões.	Sim
Classificação binária; Classificação Multi-classe; Deteção de Anomalias; Clustering	Unsupervised Learning	K-means	Scikit-learn; Weka; OpenCV; R; Spark MLlib; TensorFlow; H2O;	Médio	Médio	Sim	$O(n_{\text{dimensões}} * n_{\text{amostras}}^2)$. n_{amostras} : número de amostras; $n_{\text{dimensões}}$: número de dimensões.	Sim

As características analisadas das tecnologias foram as seguintes:

- **Popularidade** – Para característica foi analisada a popularidade de cada tecnologia com base nos registos do GitHub (*Stars*), em comunidades e fóruns.

- **Compatibilidade com GPU** – Esta característica avalia a compatibilidade da tecnologia com a aceleração de *hardware*.
- **Paralelização** – Nesta característica pretende-se entender se as tecnologias têm capacidade de executar processos em múltiplas unidades de processamento.
- **Big Data** – Esta característica permite avaliar se as tecnologias têm a capacidade de escalabilidade dos dados, ou seja, se estão aptas para um aumento exponencial de dados.
- **Linguagens de programação** – Nesta característica foram registadas as linguagens de programação consideradas como requisito para conseguir utilizar as tecnologias.
- **Vantagens e Desvantagens** – Nesta característica foram registadas algumas vantagens e desvantagens de cada tecnologia.
- **Algoritmos Suportados** – Nesta característica estão identificados todos os algoritmos que cada tecnologia suporta.

Todas estas características foram estudadas para as tecnologias acima mencionadas. No entanto, na Tabela 2, será apenas apresentado um exemplo demonstrativo de duas tecnologias de *Machine Learning* (TensorFlow, 2018; Python, 2018; GitHub, 2018; Scikit-Learn, 2018).

Tabela 2 - Tecnologias Machine Learning

TECNOLOGIA	POPULARIDADE	COMPATIBILIDADE COM GPU	PARALELIZAÇÃO	BIG DATA	LING.	VANTAGENS E DESVANTAGENS	ALGORITMOS SUPORTADOS
ScikitLearn	Stack Overflow:10,3K GitHub stars: 26.2K GitHub Fork: 13.2K	Não	Sim	Sim	Python	Vantagens: documentação detalhada; <i>Machine Learning cheat sheet</i> ; Bom desempenho; soluções para grandes <i>datasets</i> ; Desvantagens: não suporta algoritmos de <i>deep e reinforcement Learning</i> .	Classificação Regressão <i>Clustering</i> .
TensorFlow	Stack Overflow:23,5K GitHub stars: 91.3K GitHub Fork: 59K	Sim	Sim	Sim	Python	Vantagens: muita documentação; Bom desempenho; Escalável; Suposta <i>deep Learning</i> ; Plataforma para visualização; Desvantagens: Mais lento que outras frameworks (MXNet); Difícil de aprender e de usar; Não tem muitos modelos pré-tratados.	Deep Learning; K-Means; Random Forests; Gaussian Mixture Clustering; <i>Linear/Logistic Regression</i> .

6. DISCUSSÃO

Relativamente às 3 dimensões utilizadas, todas demonstraram ter uma importância relevante num projeto que envolve a utilização de *Machine Learning*. A seleção de uma tecnologia envolve a análise e a comparação dos fatores que são essenciais para o projeto com os fatores que as

tecnologias efetivamente têm, como as linguagens de programação que exigem, os algoritmos que apresentam e também, por exemplo, a possibilidade de paralelização. Os algoritmos devem ser selecionados com base na sua adequabilidade ao problema, a compatibilidade com a tecnologia e com o *Use Case*. Os *Use Cases*, de uma forma geral, são a dimensão pela qual se inicia a pesquisa na *framework* porque conhecem o problema que estão a resolver e, a partir daí, sabem qual o *Use Case* mais adequado. Depois é mais fácil a seleção dos algoritmos e tecnologias.

A construção deste *framework* permite melhorar o desenvolvimento de projetos de *Machine Learning*, no sentido em que dá acesso a um conjunto de fatores e características de forma resumida das 3 dimensões. Isto permite que, quem pretenda utilizar este tipo de tecnologia, não tenha um trabalho de pesquisa exaustivo antes de saber o que realmente deve utilizar e, este *framework*, tem exatamente essa função.

De uma forma geral, a construção deste *framework* é pertinente porque, atualmente, a utilização de *Machine Learning* e inteligência artificial está em voga, mas existem poucos investigadores que tenham conhecimento suficiente e que saibam aplicar devidamente as tecnologias de *Machine Learning*. Este *framework* permite também que iniciantes desta área consigam uma integração mais prática nas 3 principais dimensões que devem ser analisadas antes da aplicação de *Machine Learning* num problema.

7. CONCLUSÃO

Após a construção do *framework* é perceptível que, apesar de todas as pesquisas efetuadas, não existe um algoritmo específico para um tipo de problema, tudo dependerá dos dados e dos objetivos pretendidos. Este *framework* permite que os investigadores consigam apresentar uma possível solução de uma forma rápida e compreendam onde, quando e como podem utilizar algoritmos, os *Use Cases* e as tecnologias disponíveis.

Em termos de desenvolvimento futuro, pretende-se garantir que o *framework* está atualizado, de forma a garantir o aparecimento de novas tecnologias ou atualizações das existentes. Pretende-se ainda enriquecer o *framework* e torná-lo cada vez mais sucinto e direto. Para além disto, o *framework* poderá ser implementado numa plataforma *online* para utilização de toda a comunidade científica.

AGRADECIMENTOS

This work has been supported by COMPETE: POCI-01-0145-FEDER-007043 and FCT – Fundação para a Ciência e Tecnologia within the Project Scope: UID/CEC/00319/2013.

REFERÊNCIAS

- Brownlee, J. (2016). *Master Machine Learning Algorithms - Discover How they work and implement them from scratch*.
- Coelho, L., & Richert, W. (2015). *Building Machine Learning Systems with Python*. Packt Publishing Ltd.
- Géron, A. (2017). *Hands-On Machine Learning with Scikit-Learn and TensorFlow*. United States of America: O'Reilly Media, Inc., 1005 Gravenstein Highway North, Sebastopol.
- GitHub. (2018). *Computation using data flow graphs for scalable machine learning*. Obtido de <https://github.com/tensorflow/tensorflow>
- Guido, S., & Muller, A. (2016). *Introduction to Machine Learning with Python*. United States of America: O'Reilly Media, Inc., 1005 Gravenstein Highway North, Sebastopol.
- Kelleher, J., Mac Namee, B., & D'Arcy, A. (2015). *Fundamentals of Machine Learning for Predictive Data Analytics*. States of America: Massachusetts Institute of Technology.
- Lampropoulos, A., & Tsihrintzis, G. (2015). *Machine Learning Paradigms - Applications in Recommender Systems*. Springer International Publishing AG Switzerland.
- Murphy, K. (2012). *Machine Learning - A Probabilistic Perspective*. Cambridge, Massachusetts: Massachusetts Institute of Technology.
- Python. (2018). *scikit-learn: machine learning in Python*. Obtido de GitHub: <https://github.com/scikit-learn/scikit-learn>
- Scikit-Learn. (2018). *Documentation of scikit-learn 0.19.1*. Obtido de scikit-learn: <http://scikitlearn.org/stable/>
- TensorFlow. (2018). Obtido de <https://www.tensorflow.org/install/>