

Universidade do Minho

Escola de Engenharia

Eduardo João Gomes Teixeira da Costa

**Algoritmos de Aprendizagem Automática para
Previsão de AVC**

Dissertação de Mestrado

Mestrado Integrado em Engenharia Informática

Trabalho efetuado sob a orientação do

Professor Doutor José Manuel Ferreira Machado

outubro de 2022

DECLARAÇÃO

Nome: Eduardo João Gomes Teixeira da Costa

Título da Dissertação: Algoritmos de Aprendizagem Automática para Previsão de AVC

Mentores: José Manuel Ferreira Machado

Ano de Conclusão: 2022

Designação do Mestrado: Mestrado Integrado em Engenharia Informática

Este é um trabalho académico que pode ser utilizado por terceiros desde que respeitadas as regras e boas práticas internacionalmente aceites, no que concerne aos direitos de autor e direitos conexos.

Assim, o presente trabalho pode ser utilizado nos termos previstos na licença abaixo indicada.

Caso o utilizador necessite de permissão para poder fazer um uso do trabalho em condições não previstas no licenciamento indicado, deverá contactar o autor, através do RepositóriUM da Universidade do Minho.



Atribuição-NãoComercial-SemDerivações

CC BY-NC-ND

<https://creativecommons.org/licenses/by-nc-nd/4.0/>

Universidade do Minho, 21 / 10 / 2022

Assinatura: _____

AGRADECIMENTOS

Gostaria de agradecer a todas as pessoas que acompanharam e que conheci durante o meu percurso académico, sem exceção, por todos os momentos ao longo deste caminho.

Gostaria de agradecer a todos os meus amigos, a eles, um muito obrigado.

Gostaria de agradecer fundamentalmente à minha família: aos meus pais, à minha irmã e aos meus avós por tudo o que me ensinaram, por fazerem de mim a pessoa que sou hoje. A eles, dedico esta dissertação.

DECLARAÇÃO DE INTEGRIDADE

Declaro ter atuado com integridade na elaboração do presente trabalho académico e confirmo que não recorri à prática de plágio nem a qualquer forma de utilização indevida ou falsificação de informações ou resultados em nenhuma das etapas conducente à sua elaboração. Mais declaro que conheço e que respeitei o Código de Conduta Ética da Universidade do Minho.

Universidade do Minho, 21 / 10 / 2022

Assinatura: _____

RESUMO

O Acidente Vascular Cerebral (AVC) foi, em 2020, a segunda principal causa de morte no mundo e primeira no que toca a incapacidade. Com a motivação de contribuir para ajudar a reduzir os números que são alarmantes e continuam a crescer, surge este projeto, do qual se pretende que resultem modelos que possam tentar prever se um indivíduo irá, ou não, ser vítima deste problema e descobrir quais as suas características ou dados clínicos que mais influenciam esta previsão, pois, segundo a Sociedade Portuguesa de Medicina Interna (SPMI), 80% dos casos podem ser prevenidos[1].

Para o efeito, o projeto a desenvolver incluirá uma recolha e tratamento de *datasets* que organizem dados clínicos de vários pacientes e a incidência desta problemática, um estudo acerca das técnicas e algoritmos de *Machine Learning* mais adequados aos modelos a desenvolver, sendo depois aplicados através de modelos de Data Mining (DM), dando uso a ferramentas como Weka e RapidMiner, para indução dos modelos de previsão, assim como algoritmos em linguagens como Python e R, conjugando, assim, os factos de que "o setor da saúde é rico em informação, e o *Data Mining* está a tornar-se uma necessidade"[2]. Finalmente, estes modelos serão testados, validados e comparados, do qual resulta esta dissertação.

Palavras-Chave: Mineração de Dados, Aprendizagem Automática, Acidente Vascular Cerebral, Previsão.

ABSTRACT

Stroke was, in 2020, the second leading cause of death in the world and the first in terms of disability. With the motivation of contributing to help reduce these alarming numbers, which continue to grow, this project arose, which aims to produce models that can try to predict whether or not an individual will be a victim of this problem and discover which characteristics or clinical data most influence this prediction, since, according to the Portuguese Society of Internal Medicine (SPMI), 80% of cases can be prevented[1].

To this end, the project to be developed will include the collection and processing of datasets that organize clinical data from several patients and the incidence of this problem, a study on the techniques and algorithms of Machine Learning more suitable for the models to be developed, These will then be applied through Data Mining (DM) models, using tools such as Weka and RapidMiner to induce the prediction models, as well as algorithms in languages such as Python, thus combining the facts that "because healthcare sector is rich with information, and data mining is becoming a necessity"[2]. Finally, these models will be tested, validated and compared, resulting in this dissertation.

Keywords: Data Mining, Machine Learning, Stroke, Prediction.

ÍNDICE

1.	INTRODUÇÃO.....	1
1.1	Contextualização e Motivação	1
1.2	Objetivos.....	2
1.3	Metodologia	2
1.4	Estrutura do Documento	4
2.	TÉCNICAS, TECNOLOGIAS E FERRAMENTAS	5
2.1	Técnicas	5
2.2	Tecnologias.....	7
2.2.1	Conjuntos de Dados.....	7
2.2.2	Algoritmos de <i>Machine Learning</i> e Modelos Preditivos	8
2.3	Ferramentas.....	10
2.3.1	WEKA	10
2.3.2	RapidMiner	10
2.3.3	Linguagens	11
3.	PROCESSO DE CRISP-DM.....	12
3.1	Compreensão do Negócio	12
3.2	Compreensão dos Dados	14
3.3	Preparação dos Dados	22
3.4	Modelação	25
3.4.1	WEKA	26
3.4.2	RapidMiner	32
3.4.3	Python	37
3.5	Avaliação dos Modelos.....	43
4.	CONCLUSÕES.....	48
5.	REFERÊNCIAS BIBLIOGRÁFICAS	50
6.	ANEXOS.....	52

ÍNDICE DE FIGURAS

Figura 1 - Modelo de processos CRISP-DM	6
Figura 2 - Amostra do dataset inicial	15
Figura 3 - Distribuição da incidência de AVC para cada atributo	18
Figura 4 - Distribuição da incidência de AVC por idade e sexo	19
Figura 5 - Relação da incidência de AVC com a saúde mental	19
Figura 6 - Relação da incidência de AVC com a saúde física	20
Figura 7 - Relação da incidência de AVC com a autoavaliação do estado de saúde geral	20
Figura 8 - Distribuição da incidência de diabetes por idade e sexo	21
Figura 9 - Contagem valores nulos em Python do Dataset	22
Figura 10 - Transformação do formato de dados (csv para arff)	24
Figura 11 - Transformação do atributo stroke para nominal no WEKA	26
Figura 12 - Resultados WEKA com Random Forest e CV 10-F no dataset original	27
Figura 13 - Resultados WEKA com Random Forest e Split 66% no dataset original	28
Figura 14 - Processo RapidMiner do modelo Random Forest e CV 10-F no dataset original	32
Figura 15 - Processo RapidMiner interno da CV 10-F no dataset original	32
Figura 16 - Resultados RapidMiner com Random Forest e CV 10-F no dataset original	33
Figura 17 - Processo RapidMiner do modelo Random Forest e Split 66% no dataset original	33
Figura 18 - Resultados RapidMiner com Random Forest e Split 66% no dataset original	33
Figura 19 - Resultados RapidMiner com NaiveBayes e CV 10-F no dataset original	33
Figura 20 - Resultados RapidMiner com NaiveBayes e Split 66% no dataset original	34
Figura 21 - Processo RapidMiner do modelo de Regressão Logística e Split 66% no dataset original ...	34
Figura 22 - Resultados RapidMiner com Regressão Logística e Split 66% no dataset original	34
Figura 23 - Resultados RapidMiner com Regressão Logística e CV 10-F no dataset original	35
Figura 24 - Código Python transformação do tipo de dados do atributo stroke	37
Figura 25 - Tipo de dados dos datasets	37
Figura 26 - Método de subdivisão em conjuntos de dados de treino e teste	38
Figura 27 - Classificadores/algoritmos utilizados para indução dos modelos em Python	38
Figura 28 - Resultados Python com Split de 0.5 no dataset original	39
Figura 29 - Resultados Python com Split de 0.66 no dataset original	39
Figura 30 - Resultados Python com Split de 0.8 no dataset original	39
Figura 31 - Resultados Python com Split de 0.5 no dataset equilibrado	40
Figura 32 - Resultados Python com Split de 0.66 no dataset equilibrado	40
Figura 33 - Resultados Python com Split de 0.8 no dataset equilibrado	40
Figura 34 - Alteração da seed de aleatoriedade no código Python	41
Figura 35 - Resultados Python com Split de 0.5 no dataset equilibrado (random_state = 42)	41
Figura 36 - Resultados Python com Split de 0.66 no dataset equilibrado (random_state = 42)	41
Figura 37 - Resultados Python com Split de 0.8 no dataset equilibrado (random_state = 42)	41
Figura 38 - Gráfico da distribuição do peso de cada atributo	42

ÍNDICE DE TABELAS

Tabela 1 – Descrição de cada atributo do dataset	15
Tabela 2 – Distinção de valores relativos ao atributo Diabetes_012	16
Tabela 3 - Distinção de valores relativos ao atributo Age	17
Tabela 4 - Distinção de valores relativos ao atributo Education.....	17
Tabela 5 – Distinção de valores relativos ao atributo Income	18
Tabela 6 – Tabela de comparação entre os datasets	23
Tabela 7 – Exemplo de matriz de confusão genérica	25
Tabela 8 – Modelos WEKA com Random Forest no dataset original	28
Tabela 9 - Modelos WEKA com NaiveBayes no dataset original	29
Tabela 10 - Modelos WEKA com BayesNet no dataset original	29
Tabela 11 - Modelos WEKA com Random Forest no dataset equilibrado	30
Tabela 12 - Modelos WEKA com NaiveBayes no dataset equilibrado	30
Tabela 13 - Modelos WEKA com BayesNet no dataset equilibrado	31
Tabela 14 - Resultados RapidMiner no dataset equilibrado	35
Tabela 15 - Resultados RapidMiner no dataset equilibrado (Random Forest, sem Shuffle)	36
Tabela 16 – Tabela comparação da precisão do Random Forest em WEKA e RapidMiner	44
Tabela 17 - Tabela comparação resultados Random Forest em RapidMiner com e sem Shuffle.....	44
Tabela 18 - Tabela comparação resultados Random Forest em WEKA, RapidMiner e Python.....	45

1. INTRODUÇÃO

1.1 Contextualização e Motivação

De acordo com os dados da Organização Mundial de Saúde (OMS) lançados no final do ano de 2020, o Acidente Vascular Cerebral (AVC) foi, nesse ano, a segunda maior causa de morte a nível mundial, responsável por cerca de 11% do total de mortes ocorridas, que corresponde a mais de 6 milhões de fatalidades anuais, número que tem apresentado uma tendência crescente desde o ano 2000 e que se traduz numa morte a cada 5 segundos. Anualmente, cerca de 15 milhões de pessoas em todo o mundo sofrem deste problema, das quais, para além dos casos fatais, 5 milhões ficam permanentemente incapacitadas, tornando o AVC a principal causa de incapacidade em adultos em todo o mundo. Segundo a mesma organização, apesar de ser incomum em pessoas com menos de 40 anos, quando ocorre, a sua causa principal é a pressão arterial elevada e estima-se que uma em cada seis pessoas sofrerá um AVC ao longo da vida. No entanto, este também ocorre em cerca de 8% das crianças com anemia falciforme, por exemplo [3]. Em Portugal, a Sociedade Portuguesa de Medicina Interna (SPMI) diz que os AVC são mesmo a principal causa de morte e incapacidade, mas refere que 80% dos casos podem ser prevenidos [2].

Prever este tipo de ocorrências, com base em alguns dados clínicos de cada paciente, pode permitir que os profissionais de saúde consigam tomar algumas medidas iniciais atempadamente, de forma a reduzir o risco de AVC e, caso isso não seja possível, agir de forma mais célere. Com esse objetivo, surgiu este projeto de dissertação, do qual foi pretendido que resultassem modelos que possam tentar prever se um indivíduo irá, ou não, ser vítima desta grave problemática e descobrir quais as suas características ou os fatores de maior risco que mais influenciam esta previsão. A solução passa pela recolha e tratamento de *datasets* que organizem dados clínicos de vários pacientes e a incidência, ou não, de Acidente Vascular Cerebral nas mesmas, de um estudo acerca das técnicas e algoritmos de *Machine Learning* mais adequados aos modelos a desenvolver, sendo depois aplicados através de modelos de *Data Mining* (DM), dando uso a ferramentas adequadas para indução dos modelos de previsão

1.2 Objetivos

O objetivo principal deste projeto e respectiva dissertação é um estudo acerca de qual o melhor modelo de previsão da ocorrência de AVC num paciente, com base em vários dados clínicos e informações, viável e que apresenta um grau de precisão tão alto quanto possível, utilizando algoritmos de *Machine Learning* e processos de *Data Mining* adequados, de forma a possibilitar que, num contexto real e com acesso a essa informação necessária e relevante, seja possível prever a incidência, ou não, desta problemática.

De forma a atingir esses objetivos, primeiramente, foi necessária a pesquisa e recolha de diversos *datasets* que reúnem dados clínicos relevantes de conjuntos de pacientes, com informação relativa à incidência, ou não, de AVC nos mesmos, passando, depois, à seleção daquele que mais se adequa ao trabalho a desenvolver, e, de seguida, um tratamento e uniformização dos dados nele presentes. Depois, um trabalho de investigação e estudo acerca de quais os algoritmos de *Machine Learning* e *Data Mining* mais adequados ao modelo a desenvolver, assim como das técnicas, ferramentas e metodologias mais adequadas ao problema em questão e à produção da presente dissertação nessa matéria. Finalmente, o desenvolvimento e aplicação desses algoritmos no *dataset* selecionado, da qual resultaram modelos de previsão, utilizando ferramentas como o WEKA, o RapidMiner e Python, que foram, finalmente, testados, validados e comparados em termos de precisão, de forma a encontrar o modelo mais viável e retirar relações relevantes.

1.3 Metodologia

O desenvolvimento deste projeto foi preparado e concretizado de acordo com a metodologia *Design Science Research* (DSR), pois esta é considerada no mundo da engenharia e ciência como paradigma para a resolução de problemas, através de soluções inovadoras para problemas do mundo real. Esta inclui várias fases distintas, começando pela identificação do problema e motivação, passando depois para a definição de objetivos claros para uma solução final, que se concretiza na sua respetiva conceção e desenvolvimento, terminando com a sua demonstração, avaliação e comunicação [4]. Assim, a adoção da metodologia DSR na implementação de um projeto, numa das variadíssimas áreas de conhecimento existentes, como é o caso concreto da ciência da computação e informática, pode impedir a reinvenção da roda, permitindo considerar a investigação relacionada como conhecimento adquirido válido.

Sendo assim, em primeiro lugar, identificou-se a problemática em estudo, além da sua contextualização e determinação da motivação para a resolução da mesma, foi feita uma investigação exhaustiva relativa às técnicas, ferramentas e tecnologias atualmente existentes e mais relevantes para a matéria em questão, comparando as suas vantagens e funcionalidades. Para tal, foi fundamental uma detalhada revisão de literatura. A informação recolhida no estado de arte permitiu agregar todo o conhecimento existente sobre os temas em estudo, como é o caso do *Machine Learning*, *Data Mining* e técnicas, ferramentas e metodologias a elas associadas, que serviram de suporte para o desenho do produto e definição clara dos objetivos do trabalho a desenvolver, relativos à segunda fase da metodologia DSR referida. Nesta fase, foi importante identificar quais as regras e passos a seguir na fase de desenvolvimento, antes de avançar para a mesma, com o objetivo de garantir que o artefacto a ser desenvolvido é, de facto, capaz de gerar novo conhecimento e/ou melhorar comprovadamente as soluções já existentes para o problema.

Na fase de desenvolvimento e concretização do desenho proposto, recorrendo ao artefacto inferido através da investigação e revisão literária da fase anterior, procurou-se seguir as regras e processos definidos, tentando cumprir os objetivos propostos para a solução final, de forma a resolver o problema em questão. Mas todo este processo é iterativo, pois, a cada tentativa de encontrar uma solução ideal, caso nas fases de demonstração e de avaliação não se encontre uma solução viável que cumpra os requisitos estipulados, deve-se analisar todo o processo anterior e voltar a inferir uma nova proposta de artefacto. Ou seja, no caso concreto do projeto em estudo, procurou-se encontrar um modelo preditivo da ocorrência de AVC em indivíduos, a partir de um conjunto das suas características e dados clínicos, através de algoritmos de *Machine Learning* e processos de *Data Mining*, para tal, foram concretizadas várias possíveis soluções seguindo o conhecimento adquirido na primeira fase da metodologia DSR, mas que, quando testadas, comparadas e avaliadas se revelaram mais ou menos viáveis, sendo necessário inferir novos modelos, conduzindo, assim, ao processo iterativo descrito. Nesta fase de avaliação, é importante notar que é sempre gerado conhecimento, quer seja pelo sucesso da avaliação do artefacto, quer seja pelo seu insucesso, pois esta metodologia parte do princípio de que, mesmo que a solução não seja ideal, todo o trabalho desenvolvido promove novo conhecimento e novos dados, para que, num novo ciclo, não se cometam os mesmos erros.

Finalmente, se o desenvolvimento deste artefacto se mostrar adequado e capaz de cumprir todos os requisitos estabelecidos na fase de demonstração e avaliação do mesmo, entra-se na última fase do processo DSR, a comunicação do trabalho realizado, onde é apresentado o resultado obtido. É ainda relevante perceber que este processo poderá nunca ter fim, pois a busca pelo conhecimento é um

processo constante, existindo ainda a possibilidade de repetir todo o ciclo e, com o novo conhecimento adquirido, completar falhas e lacunas que pudessem existir no conhecimento recolhido no ciclo anterior.

1.4 Estrutura do Documento

O presente documento retrata a aplicação da metodologia descrita e está dividido em seis capítulos. No primeiro, referente à Introdução, são contextualizados o problema e o trabalho a desenvolver e é descrita a motivação para a busca por uma solução para o mesmo, seguidos pela definição dos objetivos gerais do projeto de dissertação. Ainda nesta secção introdutória, é descrita a metodologia de trabalho mais adequada a seguir. Depois, o segundo capítulo foca-se em apresentar o resultado da pesquisa e do trabalho de investigação, em forma de estado da arte, dos temas fundamentais a este projeto, como *Machine Learning*, *Data Mining* e as técnicas, tecnologias e ferramentas mais utilizadas e mais adequadas ao contexto deste problema, permitindo a busca por uma solução o mais sólida e viável possível. No terceiro capítulo, é exposto todo o trabalho realizado e os resultados obtidos, naquilo que é o contexto mais prático do trabalho em questão, descrevendo-se todos os passos do processo de desenvolvimento concretizados. Depois, um capítulo de conclusão que reúne todas as elações retiradas do trabalho desenvolvido e no qual se centra o propósito da presente dissertação. Finalmente, no quinto e sexto capítulos, encontram-se, respetivamente, as referências bibliográficas consultadas e alguns anexos considerados pertinentes.

2. TÉCNICAS, TECNOLOGIAS E FERRAMENTAS

Conjugando os factos de que "o setor da saúde é rico em informação, e o *Data Mining* está a tornar-se uma necessidade"[2], têm surgido cada vez mais trabalhos nesta área no sentido de tentar prever a ocorrência de doenças com base em dados clínicos e outro qualquer tipo de informação relevante. O contexto pandémico pelo qual o mundo passou veio também reforçar ainda mais a necessidade de investimento e aumentar o estado de alerta para a saúde em geral, mostrando que os mecanismos existentes, resultado do investimento e do trabalho anterior, tornaram possível a gestão e o controlo ajustado da situação, mas também que este tem que ser extensível a todas as áreas do setor da saúde, permitindo melhorar os cuidados prestados e, acima de tudo, prever e prevenir [5].

Assim, em jeito de estado da arte, tendo em conta o referido contexto da problemática do Acidente Vascular Cerebral em Portugal e no mundo e de forma a procurar uma solução capaz de prever a sua ocorrência, foi necessária uma investigação exaustiva relativa às técnicas, ferramentas e tecnologias atualmente existentes e mais relevantes para o trabalho em questão, comparando as suas vantagens e funcionalidades.

2.1 Técnicas

Data Mining refere-se a um passo preciso do processo global de Descoberta de Conhecimento e corresponde ao conjunto de métodos e técnicas para explorar e analisar grandes conjuntos de dados de forma automática com o objetivo de encontrar regras, associações ou padrões desconhecidos ou ocultos. É uma área cada vez mais popular, que recorre a processos estatísticos, *Machine Learning*, entre outras técnicas de manipulação de dados e extração de conhecimento. A análise fornecida pela Mineração de Dados, mais concretamente os modelos de previsão e os sistemas de suporte à decisão que dele resultam, podem permitir que as instituições, neste caso em específico, de saúde, melhorem a sua eficiência operacional, mantendo um elevado nível de qualidade dos serviços prestados [10].

Para colocar em prática este processo, existem várias metodologias que podem ser seguidas para garantir maior solidez e fiabilidade do mesmo. Os três modelos de processos de Mineração de Dados mais populares são *Knowledge Discovery Databases* (KDD), CRISP-DM e SEMMA [12]. Estes são os mais amplamente praticados por especialistas e investigadores na área e apresentam várias semelhanças

entre eles, até porque SEMMA e CRISP-DM podem ser vistos como implementações do processo KDD, que suportam a aplicação de *Data Mining* em sistemas reais [13].

Ao longo do projeto em questão, foi selecionada a adoção a metodologia CRISP-DM (*Cross Industry Standard Process for Data Mining*), que é a mais utilizada nesta área desde a sua criação e que tem como objetivo a uniformização dos modelos de processos de *Data Mining*. Esta engloba as fases de *Business Understanding*, *Data Understanding*, *Data Preparation*, *Modeling*, *Evaluation* e, eventualmente, *Deployment*. Estas várias fases têm como propósito uma compreensão necessária do negócio e área em estudo e, depois, do conjunto de dados a utilizar, assim como a preparação e modelação do mesmo, com a finalidade de, através de processos descritivos e preditivos, extrair conhecimento e antecipar nova informação com base nos factos neles presentes.

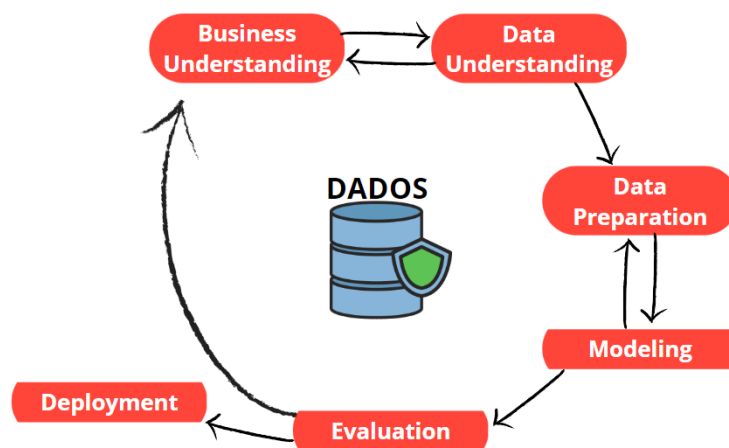


Figura 1 - Modelo de processos CRISP-DM

De forma mais detalhada, a primeira fase do processo, denominada *Business Understanding*, centra-se na compreensão dos objetivos e requisitos do projeto numa perspetiva de negócio (neste caso, o setor da saúde), convertendo este conhecimento numa definição de problema de Mineração de Dados e num plano preliminar para atingir esses objetivos. Depois, e começando pela recolha de um conjunto de dados adequado, a fase de *Data Understanding* consiste na compreensão dos mesmos, através da familiarização com o seu conteúdo e forma, identificação de problemas de qualidade, aferição de algum primeiro conhecimento ou deteção de subconjuntos relevantes. Segue-se, então, a fase de preparação de dados, que abrange todas as atividades para a construção do conjunto de dados homogêneos e uniformes a partir dos dados brutos iniciais. Este processo pode ser iterativo e poderá incluir a seleção de tabelas, registos e atributos, limpeza de dados, construção de novos atributos e transformação de

dados. Na fase de Modelação, o conjunto de dados resultante é introduzido na(s) ferramenta de modelação selecionadas e são aplicadas várias técnicas, sendo os seus parâmetros calibrados para valores ideais. A estreita ligação entre a Preparação de Dados e a Modelação deve-se à necessidade de tipos e tratamentos de dados específicos para técnicas e ferramentas, deparando, muitas vezes, com problemas nos dados durante a modelação ou a necessidade de nova modelação para a construção de novos dados para uma técnica ou ferramenta concreta. Depois, na etapa de Avaliação dos Modelos (*Evaluation*), constroem-se um ou mais modelos com a maior qualidade possível, numa perspetiva de análise de dados, mas, antes de avançar para a implementação final do modelo, é feita uma avaliação mais aprofundada dos modelos, revendo os passos executados, de forma a garantir que alcança corretamente os objetivos definidos inicialmente, pois, caso contrário, deve ser analisado se existe alguma problemática que não tenha sido suficientemente ponderada no plano inicial, recuando no processo. No final desta fase, deverá ser tomada uma decisão acerca da utilização, ou não, dos resultados do processo de Mineração de Dados. Finalmente, o *Deployment* implica que os conhecimentos adquiridos sejam organizados e apresentados de forma estruturada, prontos a ser utilizados, dependendo dos requisitos inicialmente definidos. Neste caso concreto, e tendo em conta o contexto em que se enquadra, esta fase é representada pela presente dissertação assente no trabalho realizado [12].

2.2 Tecnologias

Para a concretização do trabalho proposto, existem diferentes tipos de tecnologias que poderiam ser utilizadas de forma a cumprir com os objetivos definidos. Assim, e em conformidade com a técnica e a metodologia abordadas anteriormente, foi necessário, primeiramente, encontrar um *dataset* que reúna dados variados e relevantes ao tema em estudo acerca de múltiplos indivíduos para estudo. A esse foram, então, aplicados algoritmos de *Machine Learning* que possibilitam a construção dos modelos de previsão pretendidos.

2.2.1 Conjuntos de Dados

De forma geral, o universo de objetos é normalmente muito grande e um *dataset* representa apenas uma pequena parte dele. Normalmente, o objetivo é extrair informação dos dados disponíveis que se

espera que seja aplicável ao grande volume de dados que ainda é desconhecido. Cada objeto do conjunto de dados é descrito por uma série de variáveis que correspondem às suas propriedades e, em Mineração de Dados, são frequentemente chamados de atributos. Um conjunto de dados com o maior número de dados e atributos relevantes possível é a base do projeto a desenvolver [15].

Muito frequentemente, o *dataset* é representado por uma tabela, com cada linha a representar um caso. Por sua vez, cada coluna contém o valor de uma das variáveis (atributos) para cada uma das instâncias, que podem ser de vários tipos diferentes: nominais, binárias, ordinais, inteiras, discretas ou contínuas. Para um problema deste género, cuja solução se pretende que seja um modelo preditivo, é essencial que o conjunto de dados contenha um atributo que descreva a problemática em estudo, mais concretamente para este caso, a incidência, ou não, de Acidente Vascular Cerebral.

Por outro lado, existem outros formatos de representação dos conjuntos de dados próprios para algumas ferramentas, como, por exemplo, *arff* (*Attribute-Relation File Format*), que é um formato de ficheiro de texto ASCII que é, essencialmente, um ficheiro *csv* com um cabeçalho que descreve os metadados, onde, para dados categóricos, os valores possíveis são determinados a partir dos dados e apresentados para cada atributo.

2.2.2 Algoritmos de *Machine Learning* e Modelos Preditivos

Na implementação de um projeto de *Data Mining*, sobre um conjunto de dados final (após o processo de preparação dos mesmos) aplicam-se algoritmos de *Machine Learning* que irão permitir a criação dos modelos pretendidos, no caso do projeto em questão, de previsão. Nesse sentido, importa inferir quais são as abordagens mais utilizadas na Mineração de Dados médicos, pois os algoritmos selecionados precisam de estar em concordância com a estrutura do problema para obter informações úteis e um modelo mais preciso.

Os métodos de *Machine Learning* podem ser classificados em métodos simbólicos e sub-simbólicos. Por definição, os métodos simbólicos induzem representações simbólicas, visíveis e explicativas (como árvores de decisão) a partir dos dados, ao contrário dos sub-simbólicos, o que pode dificultar a sua explicação e dos seus resultados. No entanto, quando a precisão da classificação é o principal critério de aplicabilidade, os métodos sub-simbólicos podem revelar-se mais adequados, uma vez que usualmente conseguem precisões que são, pelo menos, tão boas quanto as dos classificadores simbólicos [10]. Os métodos simbólicos são métodos de indução de regras, tais como a aprendizagem

de regras de associação, árvores de decisão e regressão, programação de lógica indutiva e raciocínio baseado em casos. Por sua vez, métodos sub-simbólicos são métodos de aprendizagem baseados em instâncias (*instance-based learning*), redes neurais artificiais e classificação *Bayesiana*. Alguns exemplos de algoritmos mais utilizados para problemas de previsão são a Regressão Linear, Árvores de Regressão, *Naive Bayes* e *Random Forest*.

O resultado da aplicação destes algoritmos sobre os conjuntos de dados são modelos de previsão que, neste caso, tentam prever a ocorrência de Acidentes Vasculares Cerebrais. Atualmente, na resolução de problemas médicos é importante que um sistema de apoio à decisão seja capaz de explicar e justificar as decisões tomadas, daí que a interpretação do conhecimento induzido seja uma importante propriedade de sistemas que induzem soluções a partir de dados médicos relativos a casos passados [10].

Para que seja possível prever valores futuros para o conjunto de dados em estudo, é necessário que, de alguma forma, se possa captar e formular um modelo matemático capaz de representar o comportamento e as características desse conjunto. Essas informações são extraídas dos dados disponíveis e, através da modelação preditiva, que é o subcampo mais utilizado do *Data Mining* e se baseia em estatísticas, *Machine Learning*, técnicas de base de dados, reconhecimento de padrões e técnicas de otimização, permitem esclarecer sobre que tipo de funções podem ser aprendidas eficientemente dado um determinado conjunto de dados [17].

Apesar do estudo e investigação desta matéria apontarem para uma seleção prévia de algoritmos que possam, à partida, parecer mais apropriados, a procura pelos algoritmos mais adequados depende sempre da comparação e avaliação dos resultados dos mesmos, pois, tal como consta no teorema “No Free Lunch for Supervised Machine Learning” (Wolpert, 1996), nenhum algoritmo de aprendizagem automática funciona melhor para todos os problemas, e isto é especialmente relevante para os casos de aprendizagem supervisionada (ou seja, modelação preditiva) [19].

2.3 Ferramentas

Para o desenvolvimento do trabalho em questão, foi necessário utilizar ferramentas próprias para a implementação de problemas de Mineração de Dados, mais concretamente no que toca à aplicação dos algoritmos desejados sobre o conjunto de dados inicial. Então, com o objetivo de tomar uma decisão informada relativamente às ferramentas, importou investigar e analisar algumas das ferramentas mais utilizadas para o efeito num contexto real. São exemplo destas, os *softwares* RapidMiner e Weka, para além de algumas linguagens de programação.

2.3.1 WEKA

O *software Waikato Environment for Knowledge Analysis*, mais conhecido pela sua sigla WEKA, foi desenvolvido por estudantes da Universidade de Waikato, na Nova Zelândia no ano de 1999 e é um programa de distribuição e divulgação livre. O WEKA possui uma coleção de algoritmos que podem ser implementados em problemas de Mineração de Dados e, além disso, disponibiliza suporte para todo o processo, desde a preparação dos dados de entrada, avaliação estatística da aprendizagem e visualização dos dados de entrada e resultados [16]. O WEKA foi desenvolvido na linguagem de programação JAVA, mas a sua interface permite que sejam aplicados os algoritmos sem a necessidade de escrita de código, pois este possui vários métodos de regressão, classificação, regras de associação, agrupamento e seleção de atributos.

O objetivo no desenvolvimento do WEKA, ao fornecer um conjunto diversificado de métodos de *Machine Learning* disponíveis através de uma interface comum, foi permitir um máximo de flexibilidade ao experimentar diferentes tipos de métodos em conjuntos de dados, facilitando, assim, a comparação de diferentes estratégias de solução e identificar a que for mais adequada para o problema em questão [18].

2.3.2 RapidMiner

O RapidMiner, inicialmente conhecido como YALE (*Yet Another Learning Environment*), foi desenvolvido a partir de 2001 na Unidade de Inteligência Artificial da Universidade Técnica de Dortmund e é uma plataforma de *software* de ciência de dados desenvolvida agora pela empresa com o mesmo nome, que fornece um ambiente integrado para preparação de dados, *Machine Learning*, *Deep Learning*,

mineração de texto e análise preditiva. Este é utilizado em contextos empresariais, comerciais, investigação, educação, entre muitos outros e suporta todas as etapas do processo de *Data Mining*, desde a preparação de dados, visualização de resultados, validação e otimização de modelos. O *software* está escrito na linguagem de programação JAVA e fornece uma interface gráfica para conceber e executar fluxos de trabalho analíticos. Estes fluxos de trabalho são chamados "processos" no RapidMiner e consistem em conjugar múltiplos "operadores". Cada operador executa uma única tarefa dentro do processo, e a saída de cada operador forma a entrada do próximo, criando o fluxo. O RapidMiner fornece esquemas de aprendizagem, modelos e algoritmos e pode, ainda, ser estendido ao uso de *scripts* com as linguagens R e Python [20].

2.3.3 Linguagens

No que toca à mineração de dados e à implementação de algoritmos de *Machine Learning*, R e Python são duas das linguagens de programação mais populares, com ecossistemas de código aberto muito ricos e amplamente usadas por analistas e cientistas de dados.

Ambas têm os seus pontos fortes e fracos, mas com poder para seres usadas como ferramentas de tratamento, limpeza, manipulação, análise e visualização de dados [21]. R é um ambiente de *software* e linguagem de programação estatística construída para a computação estatística, manipulação de dados, análise estatística e visualização de dados, sendo extremamente capaz nestes campos. Por sua vez, Python é mais ampla e abrangente, geralmente conhecida como uma linguagem de programação de alto nível e muito versátil, conhecida pela sua sintaxe intuitiva que imita a linguagem natural. Esta pode ser utilizada para uma grande variedade de tarefas, como desenvolvimento de aplicações Web e automação/*scripting*, assim como, e mais relevantes para o caso em estudo, a ciência de dados e a análise de dados, através de bibliotecas desenvolvidas e disponibilizadas para o efeito.

3. PROCESSO DE CRISP-DM

Nesta fase, aplicaram-se as diversas fases do processo CRISP-DM explicitado anteriormente.

3.1 Compreensão do Negócio

Primeiramente, no processo de CRISP-DM, é relevante uma contextualização do problema, definição dos requisitos da solução final e elaboração do plano de implementação. Na secção de Introdução do presente documento foi contextualizado o problema no panorama nacional e mundial em termos da elevada ocorrência e mortalidade de AVC, assim como a tendência ascendente destes números, motivando, assim, a necessidade do desenvolvimento de modelos que ajudem a prever a ocorrência desta problemática. Requer-se, então, que a solução sejam modelos o mais fiáveis possível, que consigam prever a ocorrência, ou não, de um Acidente Vascular Cerebral num paciente, tendo em conta alguns dados relevantes acerca do mesmo.

O plano traçado e implementado inclui, seguindo sempre a metodologia CRISP-DM, o desenvolvimento dos modelos preditivos com algoritmos de *Machine Learning* através de diferentes ferramentas para garantir a sua validade e confiança, como abordado anteriormente. Os modelos resultantes devem permitir prever atempadamente e com a maior fiabilidade possível a ocorrência de um Acidente Vascular Cerebral, de forma a possibilitar uma rápida intervenção de forma a reduzir o risco de AVC ou, caso isso não seja possível, agir de forma mais célere e impedir ou diminuir os impactos do mesmo. Para o efeito e de forma a apoiar a decisão a tomar na escolha do *dataset* tendo em conta os atributos de cada instância do mesmo no passo seguinte da metodologia em utilização, é necessário compreender quais as principais causas e fatores de risco desta problemática. Assim, sabe-se por via de estudos prévios [22] que estes são:

Idade: é considerado o fator de risco mais relevante, tanto para o AVC isquémico como para a hemorragia intracerebral primária. Por exemplo, o risco em pessoas entre os 75 e os 84 anos é 25 vezes maior do que em pessoas entre os 45 e os 54 anos.

Género: o sexo masculino é um fator de risco, mas, no geral, devido à maior esperança de vida das mulheres e à grande importância da idade como fator de risco, estas sofrerão mais AVC durante a sua vida.

Pressão arterial: o aumento da pressão arterial é um grande fator de risco importante e está forte e independentemente associado a Acidentes Vasculares Cerebrais e hemorrágicos.

Tabagismo: segundo estudos, fumar pode duplicar o risco de acontecimento desta problemática.

Diabetes: a incidência desta doença, segundo estudos, também pode duplicar o risco de Acidente Vascular Cerebral.

Colesterol: o aumento do colesterol total e do colesterol da lipoproteína de baixa densidade são fatores de risco fortes para a doença cardíaca isquémica, mas a relação com o Acidente Vascular Cerebral parece mais fraca.

Índice de massa corporal e exercício físico: o aumento do índice de massa corporal e a falta de rotinas de exercício físico são fatores de alto risco para o AVC, embora isso se deva, em parte, à sua associação com outros fatores de risco, como hipertensão e diabetes.

Álcool: o consumo moderado de álcool pode proteger contra doenças cardíacas isquémicas e Acidentes Vasculares Cerebrais, mas o consumo exagerado é um fator de risco e, em particular, para a hemorragia intracerebral. Este facto, deve-se, por exemplo, ao aumento da pressão arterial por ele provocada.

Etnia: há evidências, por exemplo, de um aumento da incidência de AVC em afro-caribes e afro-americanos em comparação com os caucasianos.

Homocisteína: é um aminoácido presente no plasma do sangue que em níveis muito elevados está associada a um risco aumentado de Acidente Vascular Cerebral e outras trombozes arteriais em tenra idade.

Outros factos de risco: tanto a enxaqueca, como a pílula contraceptiva oral são outros exemplos de fatores de risco para o AVC. Vários estudos sugerem também que infeções agudas e a terapia de substituição hormonal, particularmente logo após o seu início, parecem aumentar o risco. Existe também uma forte associação entre o estatuto socioeconómico e o risco de Acidente Vascular Cerebral, embora isso possa ser amplamente confundido por outros fatores, como o tabagismo e a falta de exercício.

3.2 Compreensão dos Dados

Tal como visto anteriormente, a base para a implementação deste projeto são os dados, sob a forma de *dataset*, que serviram de ponto de partida para a obtenção dos modelos preditivos que visam tentar resolver o problema de *Data Mining* em questão.

Nesta fase do processo, iniciou-se a procura por um conjunto de dados o mais adequado possível ao problema em estudo, tendo em conta o conhecimento adquirido na fase de compreensão do negócio e na investigação prévia de conhecimento explicitada anteriormente, no sentido deste apresentar o maior número de atributos relevantes para uma amostra significativa de instâncias (neste caso, dados clínicos e outras informações relativas a cada pessoa), assim como todas as características definidas como essenciais.

Após pesquisa em plataformas de partilha e agregação de diversos conjuntos de dados e análise de dezenas de exemplares, foram pré selecionados quatro *datasets*, mas, devido ao número de instâncias e/ou qualidade dos atributos dos mesmos, restou apenas um, que foi considerado mais adequado.

O conjunto de dados selecionado foi encontrado na plataforma Kaggle e foi construído a partir de uma quantidade enorme de informação recolhida através do questionário BRFSS de 2015, levado a cabo pelos CDC (*Centers for Disease Control and Prevention*) dos Estados Unidos da América [27]. Neste, são agregados alguns indicadores de saúde, informações pessoais e dados clínicos de múltiplos pacientes residentes em todos os territórios dos Estados Unidos da América e Porto Rico. O *Behavioral Risk Factor Surveillance System* (BRFSS) é o principal sistema de inquéritos telefónicos relacionados com a saúde nos EUA, que recolhem dados dos residentes no país acerca dos seus comportamentos de risco relacionados com a saúde, condições de saúde crónicas, utilização de serviços preventivos, entre muitas outras informações. Ao recolher dados de comportamentos de risco de saúde a nível estatal e local, o BRFSS tornou-se uma ferramenta poderosa para direccionar e realizar atividades de promoção da saúde.

Passando, então, para a exploração e compreensão do conjunto de dados selecionado, podemos verificar que o mesmo apresenta 22 atributos distintos acerca de cada um dos 253.680 indivíduos neste descritos, como é exemplificado na figura seguinte.

A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V
Diabetes_01	HighBP	HighChol	CholCheck	BMI	Smoker	Stroke	HeartDiseaseorAttack	PhysActivity	Fruits	Veggies	AlcoholConsumption	HealthcareCost	NoDocbcCost	GenHlth	MentHlth	PhysHlth	DiffWalk	Sex	Age	Education	Income
0	1	1	1	40	1	0	0	0	0	1	0	1	0	5	18	15	1	0	9	4	3
0	0	0	0	25	1	0	0	1	0	0	0	0	1	3	0	0	0	0	7	6	1
0	1	1	1	28	0	0	0	0	1	0	0	1	1	5	30	30	1	0	9	4	8
0	1	0	1	27	0	0	0	1	1	1	0	1	0	2	0	0	0	0	11	3	6
0	1	1	1	24	0	0	0	1	1	1	0	1	0	2	3	0	0	0	11	5	4
0	1	1	1	25	1	0	0	1	1	1	0	1	0	2	0	2	0	1	10	6	8
0	1	0	1	30	1	0	0	0	0	0	0	1	0	3	0	14	0	0	9	6	7
0	1	1	1	25	1	0	0	1	0	1	0	1	0	3	0	0	1	0	11	4	4
2	1	1	1	30	1	0	1	0	1	1	0	1	0	5	30	30	1	0	9	5	1
0	0	0	1	24	0	0	0	0	0	1	0	1	0	2	0	0	0	1	8	4	3
2	0	0	1	25	1	0	0	1	1	1	0	1	0	3	0	0	0	1	13	6	8
0	1	1	1	34	1	0	0	0	1	1	0	1	0	3	0	30	1	0	10	5	1
0	0	0	1	26	1	0	0	0	0	1	0	1	0	3	0	15	0	0	7	5	7
2	1	1	1	28	0	0	0	0	0	1	0	1	0	4	0	0	1	0	11	4	6
0	0	1	1	33	1	1	0	1	0	1	0	1	1	4	30	28	0	0	4	6	2
0	1	0	1	33	0	0	0	1	0	0	0	1	0	2	5	0	0	0	6	6	8
0	1	1	1	21	0	0	0	1	1	1	0	1	0	3	0	0	0	0	10	4	3
2	0	0	1	23	1	0	0	1	0	0	0	1	0	2	0	0	0	1	7	5	6
0	0	0	0	23	0	0	0	0	0	1	0	1	0	2	15	0	0	0	2	6	7
0	0	1	1	28	0	0	0	0	0	0	1	1	0	2	10	0	0	1	4	6	8
0	1	1	1	22	0	1	1	0	1	0	0	1	0	3	30	0	1	0	12	4	4

Figura 2 - Amostra do dataset inicial

Cada atributo presente no *dataset*, representado em cada coluna, especifica um dado de cada indivíduo, representado ao longo de cada linha do mesmo. Esses atributos são:

Diabetes_012	Incidência e tipo da diabetes
HighBP	Pressão arterial alta
HighChol	Colesterol alto
CholCheck	Colesterol analisado nos últimos 5 anos
BMI	Índice de massa corporal
Smoker	Tabagismo
Stroke	Acidente Vascular Cerebral
HeartDiseaseorAttack	Problemas ou ataque cardíaco
PhysActivity	Atividade física regular
Fruits	Ingestão regular de fruta (1 vez por dia ou mais)
Veggies	Ingestão regular de legumes (1 vez por dia ou mais)
HvyAlcoholConsump	Ingestão excessiva de álcool
AnyHealthcare	Seguro de saúde
NoDocbcCost	Incapacidade de acesso a cuidados médicos por falta de dinheiro
GenHlth	Auto classificação de saúde em geral
MentHlth	Problemas no que toca à saúde mental
PhysHlth	Problemas no que toca à saúde física ou lesões
DiffWalk	Dificuldades sérias em caminhar ou subir escadas
Sex	Sexo
Age	Idade
Education	Nível de educação
Income	Nível de rendimento familiar anual

Tabela 1 – Descrição de cada atributo do dataset

Como é possível verificar pela imagem anterior, todos os dados são numéricos inteiros, embora muitos deles sejam representados por valores binários (0 ou 1). Entre eles, temos:

- o principal atributo do *dataset*, assinalado a cor diferente, que permitirá a aprendizagem por parte dos algoritmos e consequente previsão e que representa o valor da incidência, ou não, de Acidente Vascular Cerebral num indivíduo (*Stroke*). É ainda de relevar que o número de ocorrências de AVC no conjunto de dados em análise é de 10.292, apresentando, então, uma proporção de cerca de 1 caso em cada 25 instâncias nesta amostra;
- os dados relativos ao colesterol (*HighChol*, *CholCheck*) que mostram se um indivíduo tem, ou não, um valor de colesterol considerado alto e se o mesmo foi analisado nos últimos 5 anos;
- os dados relativos ao tabagismo (*Smoke*), indicando se fuma, ou não;
- também a existência, ou não, de pressão arterial alta (*HighBP*) e de problemas ou ataques cardíacos (*HeartDiseaseorAttack*);
- a prática de exercício físico (*PhysActivity*) e ingestão de frutas (*Fruits*) e legumes (*Veggies*) regularmente, ou não, assim como o consumo excessivo de álcool (*HvyAlcoholConsump*);
- a posse de seguro de saúde (*AnyHealthcare*) e incapacidade de acesso a cuidados médicos por falta de dinheiro (*NoDocbcCost*), ou não;
- a dificuldade, ou não, em caminhar ou subir escadas (*DiffWalk*);
- e o sexo, sendo o 0 o sexo feminino e o 1 o sexo masculino.

No conjunto de dados, existem três valores relativos à presença da doença da diabetes num paciente:

Valor	Diabetes
0	Não ocorrência
1	Pré diabetes
2	Ocorrência

Tabela 2 – Distinção de valores relativos ao atributo Diabetes_012

Quanto ao índice de massa corporal (*BMI*), este encontra-se presente no *dataset* em valores compreendidos entre 12 e 98. Este parâmetro é utilizado para saber se o peso está de acordo com a altura, através de uma fórmula simples que relaciona essas duas características, podendo, a partir do seu valor, inferir-se a se a pessoa se encontra abaixo (<18,5), dentro (18,5 a 24,9) ou acima (24,9 a 30)

dos limites consideramos normais de peso para a sua altura, ou, até, se se encontra com obesidade (>30).

Outro atributo que apresenta valores específicos para a sua classificação é a auto classificação da saúde em geral (*GenHlth*) feita por cada inquirido, entre valores de 1 (excelente) a 5 (muito fraca).

Quanto à medição dos problemas no que toca à saúde mental (*MentHlth*) e à saúde física ou lesões (*PhysHlth*), esta é feita através do número de dias em que estes foram sentidos nos últimos 30 dias, o que significa que estes atributos se encontram descritos em valores compreendidos entre 0 e 30.

Relativamente à sua idade (*Age*), cada indivíduo é classificado com um dos seguintes catorze níveis, como apresentado na tabela abaixo:

Valor	Idade
1	18 a 24
2	25 a 29
3	30 a 34
4	35 a 39
5	40 a 44
6	45 a 49
7	50 a 54
8	55 a 59
9	60 a 64
10	65 a 69
11	70 a 74
12	75 a 79
13	80 ou mais

Tabela 3 - Distinção de valores relativos ao atributo Age

Também o nível de educação mais elevado completo (*Education*) por cada indivíduo do *dataset* se encontra dividido em diferentes níveis, como apresentado na tabela:

Valor	Nível mais elevado completo
1	Nunca frequentou a escola ou apenas jardim de infância
2	Ensino Básico
3	Ensino Secundário, mas incompleto
4	Ensino Secundário
5	Ensino Superior (1 a 3 anos)
6	Ensino Superior (4 anos ou mais)

Tabela 4 - Distinção de valores relativos ao atributo Education

Finalmente, quanto ao rendimento anual do agregado familiar de cada paciente, os valores existentes representam o seguinte:

Valor	Rendimento Anual
1	Menos de \$10.000
2	\$10.000 a \$15.000
3	\$15.000 a \$20.000
4	\$20.000 a \$25.000
5	\$25.000 a \$35.000
6	\$35.000 a \$50.000
7	\$50.000 a \$75.000
8	\$75.000 ou mais

Tabela 5 – Distinção de valores relativos ao atributo Income

Já após a transformação do formato de dados, abordado na secção seguinte, foi possível, ainda, através de uma ferramenta de visualização de dados disponibilizada pelo software WEKA, consultá-los graficamente. Abaixo está representada a distribuição de cada atributo na população da amostra do *dataset* em estudo.

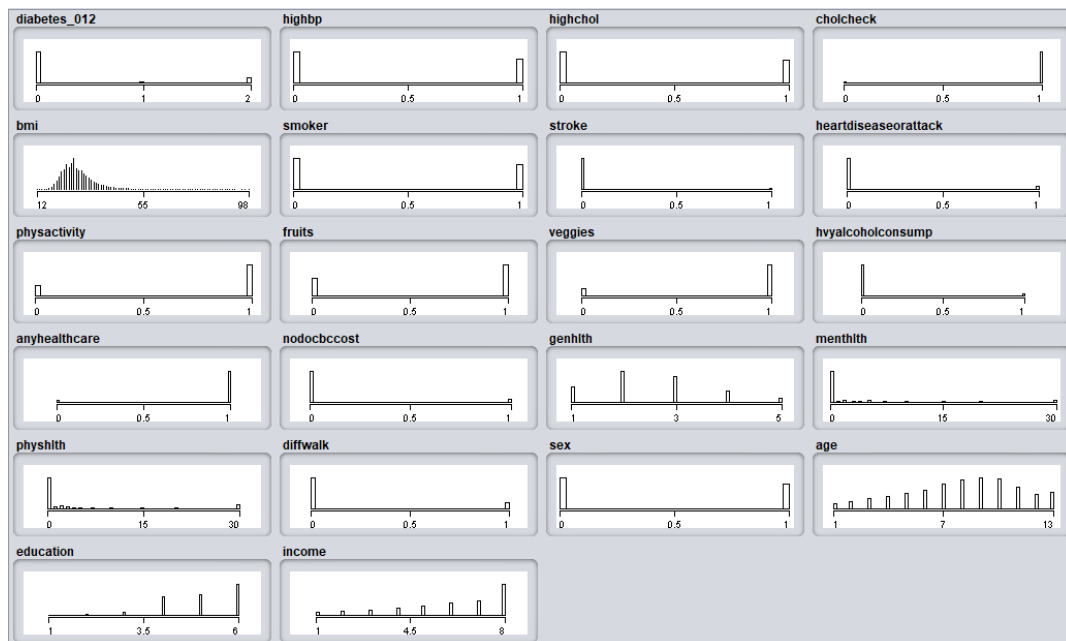


Figura 3 - Distribuição da incidência de AVC para cada atributo

Esta ferramenta permite ainda estabelecer algumas relações entre os diferentes atributos e chegar a algumas conclusões a partir das mesmas. Estas conclusões podem ser retiradas a partir de relações estabelecidas graficamente entre os diferentes atributos e permitem conhecer a um nível mais profundo o conjunto de dados em estudo, assim como a distribuição dos mesmos pelas muitas instâncias nele presentes.

Na figura seguinte, pode-se ver, por exemplo, a distribuição da ocorrência de AVC (eixo dos YY) por idades (eixo dos XX) e sexo (cor) na amostra da população representada pelo conjunto de dados. É possível verificar que a incidência desta problemática é mais frequente à medida que a idade aumenta e que atinge mais as mulheres dos que os homens.

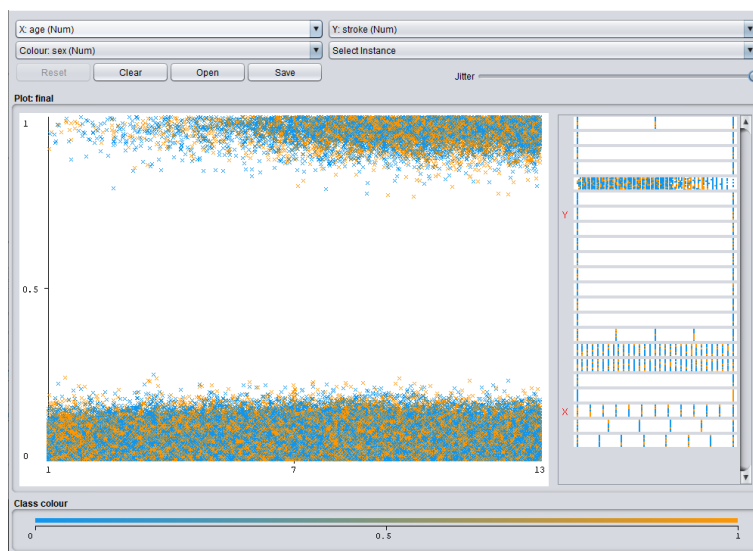


Figura 4 – Distribuição da incidência de AVC por idade e sexo

É visível também nas imagens abaixo, que recentes sintomas de problemas físicos e mentais (e, portanto, a possibilidade de maior regularidade da sua ocorrência) e uma auto classificação baixa do nível geral de saúde parecem refletir uma maior incidência de Acidente Vascular Cerebral.

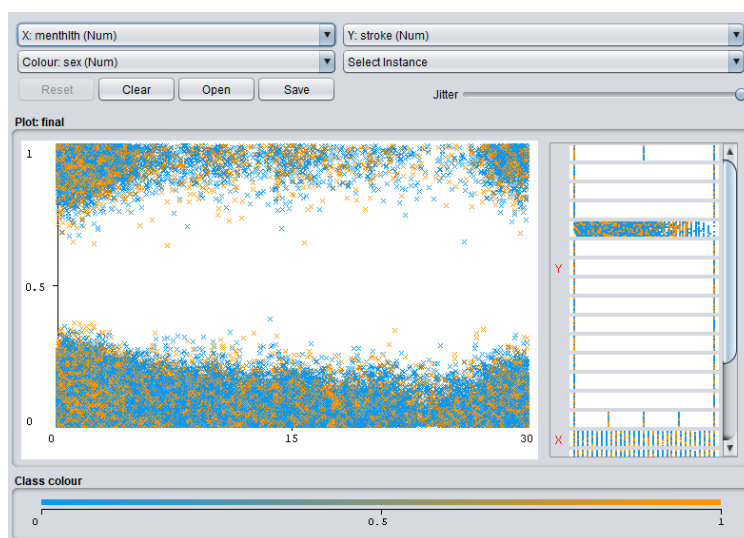


Figura 5 - Relação da incidência de AVC com a saúde mental

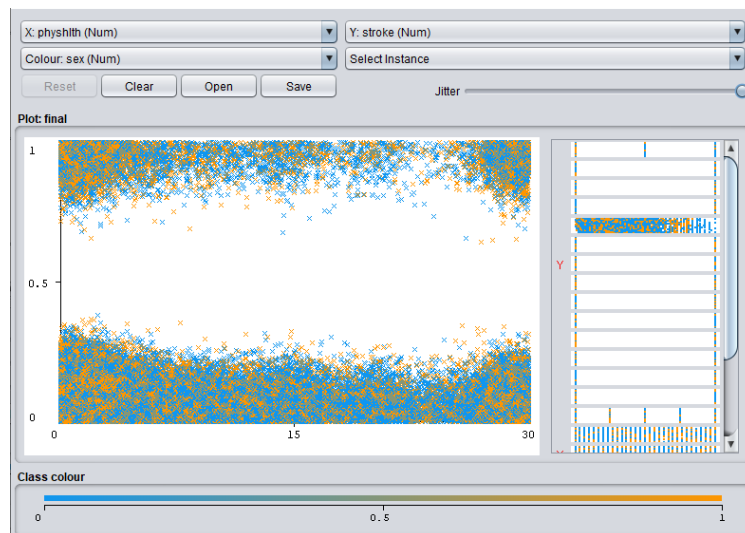


Figura 6 - Relação da incidência de AVC com a saúde física

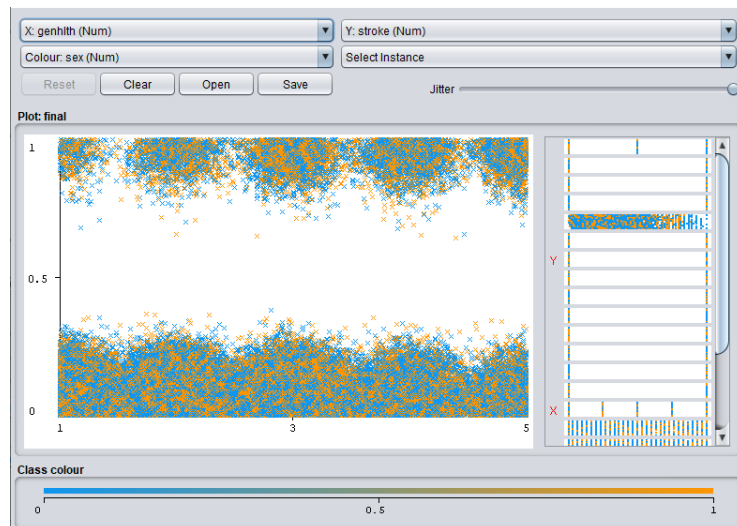


Figura 7 - Relação da incidência de AVC com a autoavaliação do estado de saúde geral

Estas relações podem também ajudar a perceber, à partida, alguns dos possíveis fatores que, por si só, possam ser de mais ou menos relevância e impacto para este estudo, observando a distribuição da incidência de AVC consoante os diferentes atributos, mas que apenas poderão ser confirmados com mais certeza, mais à frente, na fase de modelação.

De forma semelhante, podem-se também estabelecer relações entre atributos que não incluam o atributo principal do estudo (a ocorrência, ou não, de Acidente Vascular Cerebral), como mostra a figura abaixo, e da qual se conclui que a ocorrência da doença de diabetes e pré diabetes também aumenta com a idade e é mais frequente nas mulheres, por exemplo. Poderá, por esse motivo, não existir relação direta de uma problemática com a outra, mas sim semelhanças nos fatores de risco que as causam, sem que sejam diretamente um par causa-efeito ou, por outro lado, a incidência destas doenças estar relacionado e, dessa forma, os grupos nos quais é mais frequente ser semelhante.

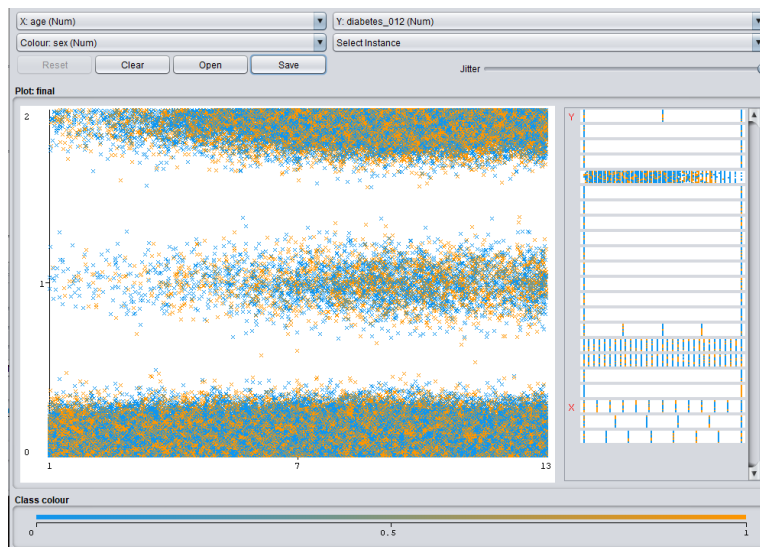


Figura 8 - Distribuição da incidência de diabetes por idade e sexo

3.3 Preparação dos Dados

Após uma melhor percepção dos dados, ainda antes da fase de Modelação, é necessária, então, a sua preparação, que poderá incluir vários tipos de atividades, como a seleção de tabelas, registos e atributos, limpeza de dados, construção de novos atributos e transformação de dados, para a construção de um conjunto de dados homogêneos e uniformes a partir dos dados brutos iniciais.

Neste caso concreto, o conjunto de dados escolhido não apresentava valores em falta e todos os atributos, tal como descrito acima, se encontravam classificados numericamente por valores binários ou por níveis, não sendo necessário qualquer tipo de processamento nesse sentido, nem de limpeza de dados.

```
dataset.isnull().sum()
✓ 0.9s
Diabetes_012      0
HighBP            0
HighChol          0
CholCheck         0
BMI               0
Smoker            0
Stroke            0
HeartDiseaseorAttack 0
PhysActivity      0
Fruits            0
Veggies           0
HvyAlcoholConsump 0
AnyHealthcare     0
NoDocbcCost       0
GenHlth           0
MentHlth          0
PhysHlth          0
DiffWalk          0
Sex               0
Age               0
Education         0
Income            0
dtype: int64
```

Figura 9 - Contagem valores nulos em Python do Dataset

Mas, após uma análise do *dataset*, foi verificado que existia um desequilíbrio entre o número de instâncias onde o atributo *stroke* tinha valor 1 e o total de instâncias. Isto poderia levar a que os modelos de previsão induzidos a partir do mesmo não fossem tão realistas quanto possível, dada a grande diferença na quantidade de instâncias para cada valor deste atributo, devido à dificuldade no treino dos modelos a partir de uma amostra tão pouco significativa. No conjunto de dados inicial existiam cerca de 10.000 instâncias com o atributo a 1 e as restantes cerca de 240.000 com o atributo *stroke* com o valor

0, que reflete uma relação de 1 caso de ocorrência de AVC em cada 25 pessoas, como havia sido referido anteriormente. Do ponto de vista dos algoritmos, uma proporção tão desequilibrada entre os dois casos poderia levar a que o treino dos mesmos não fosse suficiente para os casos de ocorrência de Acidente Vascular Cerebral em relação à não ocorrência.

De forma a combater esta desproporcionalidade apresentada pelo *dataset* em estudo, foi criada uma variante do conjunto de dados original onde estas diferenças são atenuadas. Assim, e recorrendo à informação recolhida, foram quintuplicadas as instâncias com valor 1 do atributo *stroke*, de forma a aproximar a razão de casos de incidência para 1 em cada 6 indivíduos, segundo as previsões dos estudos da OMS descritas anteriormente. Deste processamento, resultou um novo *dataset* que irá, ao longo deste documento, ser referido como «*dataset* equilibrado» em relação ao inicial, que será o «*dataset* original».

<i>Dataset</i>	Número total de instâncias	Número de instâncias com atributo <i>stroke</i> = 1
<i>Dataset</i> original	253.680	10.292
<i>Dataset</i> equilibrado	294.848	51.460

Tabela 6 – Tabela de comparação entre os datasets

Por outro lado, e como o *software* WEKA dá preferência à importação de dados no formato *arff* explicitado previamente, foi necessária a transformação do formato dos dados, inicialmente em *csv*. Para o efeito, foi usada uma de muitas ferramentas *online* do género, que efetua essa alteração automaticamente, visível na figura seguinte.

Diabetes_012;HighBP;HighChol;CholCheck;BMI;Smoker;Stroke; HeartDiseaseorAttack;PhysActivity;Fruits;Veggies;HvyAlcoholConsump; AnyHealthcare;NoDocbcCost;GenHlth;MenthHlth;PhysHlth;DiffWalk;Sex;Age;Education;Income 0;1;1;1;40;1;0;0;0;1;0;1;0;5;18;15;1;0;9;4;3 0;0;0;0;25;1;0;0;1;0;0;0;0;1;3;0;0;0;0;7;6;1 0;1;1;1;28;0;0;0;0;1;0;0;1;5;30;30;1;0;9;4;8 0;1;0;1;27;0;0;0;1;1;1;0;1;0;2;0;0;0;0;11;3;6 0;1;1;1;24;0;0;0;1;1;1;0;1;0;2;3;0;0;0;0;11;5;4 0;1;1;1;25;1;0;0;1;1;1;0;1;0;2;0;2;0;1;10;6;8 0;1;0;1;30;1;0;0;0;0;0;1;0;3;0;14;0;0;9;6;7 0;1;1;1;25;1;0;0;1;0;1;0;1;0;3;0;0;1;0;11;4;4 2;1;1;1;30;1;0;1;0;1;1;0;1;0;5;30;30;1;0;9;5;1 0;0;0;1;24;0;0;0;0;0;1;0;1;0;2;0;0;0;1;8;4;3 2;0;0;1;25;1;0;0;1;1;1;0;1;0;3;0;0;0;1;13;6;8 0;1;1;1;34;1;0;0;0;1;1;0;1;0;3;0;30;1;0;10;5;1 0;0;0;1;26;1;0;0;0;0;1;0;1;0;3;0;15;0;0;7;5;7 2;1;1;1;28;0;0;0;0;0;1;0;1;0;4;0;0;1;0;11;4;6 0;0;1;1;33;1;1;0;1;0;1;0;1;4;30;28;0;0;4;6;2 0;1;0;1;33;0;0;0;1;0;0;0;1;0;2;5;0;0;0;6;6;8 0;1;1;1;21;0;0;0;1;1;1;0;1;0;3;0;0;0;0;10;4;3 2;0;0;1;23;1;0;0;1;0;0;0;1;0;2;0;0;0;1;7;5;6 0;0;0;0;23;0;0;0;0;0;1;0;1;0;2;15;0;0;0;2;6;7 0;0;1;1;28;0;0;0;0;0;0;1;1;0;2;10;0;0;1;4;6;8 0;1;1;1;22;0;1;1;0;1;0;0;1;0;3;30;0;1;0;12;4;4 0;1;1;1;38;1;0;0;0;1;1;0;1;0;5;15;30;1;0;13;2;3 0;0;0;1;28;1;0;0;0;0;1;0;1;0;3;0;7;0;1;5;5;5 2;1;0;1;27;0;0;0;1;1;1;0;1;0;1;0;0;0;0;13;5;4 0;1;1;1;28;1;0;0;0;1;1;0;1;0;3;6;0;1;0;9;4;6 0;0;0;1;32;0;0;0;1;1;1;0;1;0;2;0;0;0;0;5;6;8 2;1;1;1;37;1;1;1;0;0;1;0;1;0;5;0;0;1;1;10;6;5 2;1;1;1;28;1;0;1;0;0;1;0;1;0;4;0;0;0;1;12;2;4		@RELATION final @ATTRIBUTE diabetes_012 REAL @ATTRIBUTE highbp REAL @ATTRIBUTE highchol REAL @ATTRIBUTE cholcheck REAL @ATTRIBUTE bmi REAL @ATTRIBUTE smoker REAL @ATTRIBUTE stroke REAL @ATTRIBUTE heartdiseaseorattack REAL @ATTRIBUTE physactivity REAL @ATTRIBUTE fruits REAL @ATTRIBUTE veggies REAL @ATTRIBUTE hvyalcoholconsump REAL @ATTRIBUTE anyhealthcare REAL @ATTRIBUTE nodocbccost REAL @ATTRIBUTE genhlth REAL @ATTRIBUTE menthlth REAL @ATTRIBUTE physhlth REAL @ATTRIBUTE diffwalk REAL @ATTRIBUTE sex REAL @ATTRIBUTE age REAL @ATTRIBUTE education REAL @ATTRIBUTE income REAL @DATA 0,1,1,1,40,1,0,0,0,0,1,0,1,0,5,18,15,1,0,9,4,3 0,0,0,0,25,1,0,0,1,0,0,0,0,1,3,0,0,0,0,7,6,1 0,1,1,1,28,0,0,0,0,1,0,0,1,5,30,30,1,0,9,4,8 0,1,0,1,27,0,0,0,1,1,1,0,1,0,2,0,0,0,0,11,3,6 0,1,1,1,24,0,0,0,1,1,1,0,1,0,2,3,0,0,0,0,11,5,4 0,1,0,1,25,1,0,0,0,0,0,1,0,3,0,14,0,0,9,6,7 0,1,1,1,25,1,0,0,1,0,1,0,1,0,3,0,0,1,0,11,4,4 2,1,1,1,30,1,0,1,0,1,1,0,1,0,5,30,30,1,0,9,5,1 0,0,0,1,24,0,0,0,0,0,1,0,1,0,2,0,0,0,1,8,4,3 2,0,0,1,25,1,0,0,1,1,1,0,1,0,3,0,0,0,1,13,6,8 0,1,1,1,34,1,0,0,0,1,1,0,1,0,3,0,30,1,0,10,5,1 0,0,0,1,26,1,0,0,0,0,1,0,1,0,3,0,15,0,0,7,5,7 2,1,1,1,28,0,0,0,0,0,1,0,1,0,4,0,0,1,0,11,4,6 0,0,1,1,33,1,1,0,1,0,1,0,1,4,30,28,0,0,4,6,2 0,1,0,1,33,0,0,0,1,0,0,0,1,0,2,5,0,0,0,6,6,8 0,1,1,1,21,0,0,0,1,1,1,0,1,0,3,0,0,0,0,10,4,3 2,0,0,1,23,1,0,0,1,0,0,0,1,0,2,0,0,0,1,7,5,6 0,0,0,0,23,0,0,0,0,0,1,0,1,0,2,15,0,0,0,2,6,7 0,0,1,1,28,0,0,0,0,0,0,1,1,0,2,10,0,0,1,4,6,8 0,1,1,1,22,0,1,1,0,1,0,0,1,0,3,30,0,1,0,12,4,4 0,1,1,1,38,1,0,0,0,1,1,0,1,0,5,15,30,1,0,13,2,3 0,0,0,1,28,1,0,0,0,0,1,0,1,0,3,0,7,0,1,5,5,5 2,1,0,1,27,0,0,0,1,1,1,0,1,0,1,0,0,0,0,13,5,4 0,1,1,1,28,1,0,0,0,1,1,0,1,0,3,6,0,1,0,9,4,6 0,0,0,1,32,0,0,0,1,1,1,0,1,0,2,0,0,0,0,5,6,8 2,1,1,1,37,1,1,1,0,0,1,0,1,0,5,0,0,1,1,10,6,5 2,1,1,1,28,1,0,1,0,0,1,0,1,0,4,0,0,0,1,12,2,4
---	---	--

Figura 10 - Transformação do formato de dados (csv para arff)

3.4 Modelação

Nesta fase do processo de *Data Mining*, foram aplicados aos conjuntos de dados resultantes das anteriores fases deste processo vários tipos de algoritmos, com diferentes ferramentas, complementando as suas diferenças e pontos fortes, possibilitaram a indução de múltiplos modelos de previsão, para que os resultados fossem comparados posteriormente, possibilitando a sua validação. Para efeitos de comparação e normalização da terminologia usada na representação dos resultados obtidos dos testes aos modelos induzidos nesta fase, consideraram-se três tipos de métricas: precisão (*precision*), acuidade (*accuracy*) e *recall*.

Segundo a tabela abaixo, que exemplifica a matriz de confusão de um modelo, tem-se que a razão entre verdadeiros reais e todos os exemplos previstos como verdadeiros é chamada de precisão e é calculada como $d/(c+d)$, a acuidade é a relação de exemplos corretamente classificados em comparação com o número de todos os exemplos e é calculada como $(a+d)/(a+b+c+d)$ e que, por fim, a razão entre verdadeiros positivos e todos os exemplos realmente positivos é chamada de *recall* e é calculada como $d/(b+d)$.

	FALSE	TRUE
Previsto FALSE	a	b
Previsto TRUE	c	d

Tabela 7 – Exemplo de matriz de confusão genérica

É ainda de notar que a diversidade de métricas obtidas para cada modelo induzido nesta fase está dependente do *software* ou ferramenta utilizado em cada momento, pois cada tecnologia apresenta-as de forma distinta. No entanto, a terminologia usada é homogênea ao longo de todo o processo de *Data Mining*.

3.4.1 WEKA

Após o carregamento dos dados para o WEKA, foi feita a transformação do atributo fundamental do conjunto de dados em questão (*stroke*) de numérico para nominal. Este processo é feito automaticamente através de uma ferramenta de filtragem disponibilizada pelo próprio *software* que permite escolher a transformação pretendida e a que atributos a aplicar. Esta transformação permite a previsão do atributo com valores absolutos, 0 ou 1, e que assim sejam feitas medições válidas aos modelos na precisão da previsão do mesmo como, por exemplo, no que toca à matriz de confusão [23].

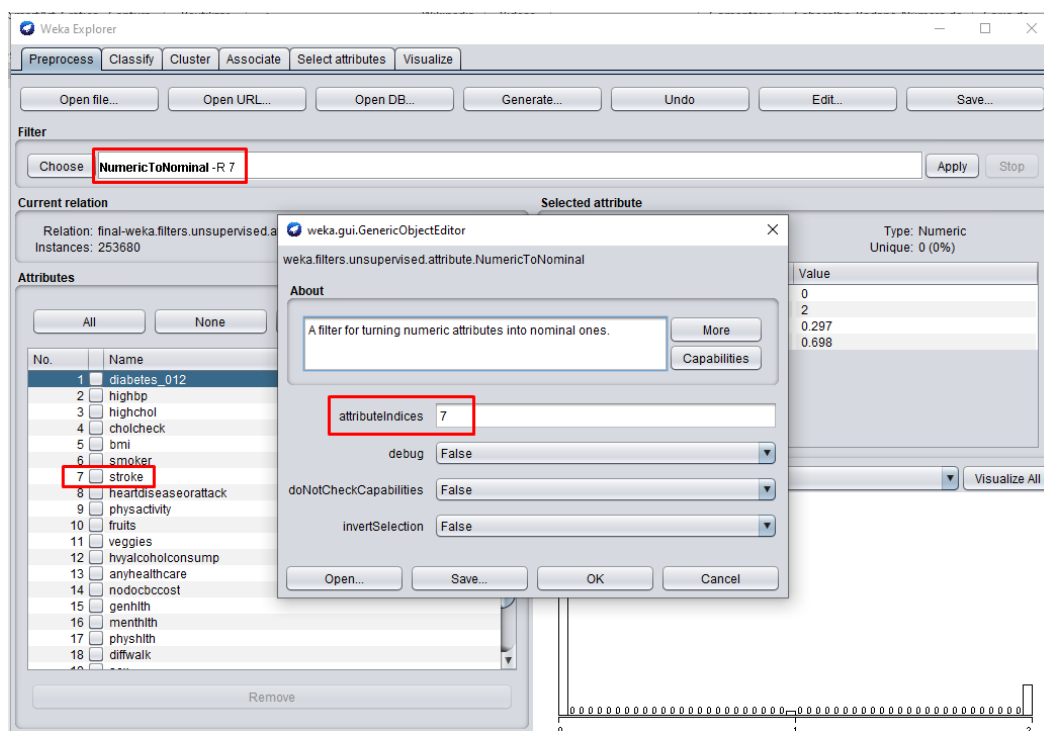


Figura 11 - Transformação do atributo *stroke* para nominal no WEKA

De seguida, foram aplicados vários algoritmos considerados adequados segundo o estudo feito nas fases anteriores deste estudo e para o problema de previsão em questão. Para cada um deles foram feitos induzidos vários modelos com diferentes configurações de treino, de forma a obter informação suficiente a partir dos seus testes para retirar conclusões acerca de quais as melhores abordagens com cada algoritmo, permitir uma comparação entre os melhores modelos possíveis para cada um deles e a validar os resultados obtidos [24].

Ao *dataset* original, foi aplicado o algoritmo *Random Forest* com Validação Cruzada (*Cross-Validation*) com 10 subdivisões (*folds*). Esta divide aleatoriamente o conjunto de amostras numa série de 10 subdivisões igualmente dimensionadas (grupos). Neste caso, 9 das divisórias são utilizadas para

dados de treino, enquanto as restantes são utilizadas para como dados de teste. O treino é repetido 10 vezes, usando uma partição diferente como conjunto de teste em cada uma delas e as restantes 9 divisórias como dados de treino, sendo a média dos resultados, então, reportada [25].

```
weka.classifiers.trees.RandomTree -K 0 -M 1.0 -V 0.001 -S 1 -do-not-check-capabilities

Time taken to build model: 193.67 seconds

=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances      243102          95.8302 %
Incorrectly Classified Instances    10578           4.1698 %
Kappa statistic                    0.0082
Mean absolute error                 0.0741
Root mean squared error             0.194
Relative absolute error             95.2216 %
Root relative squared error         98.3142 %
Total Number of Instances          253680

=== Detailed Accuracy By Class ===
               TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
               0,999    0,994    0,960     0,999    0,979      0,022    0,785    0,986     0
               0,006    0,001    0,148     0,006    0,011      0,022    0,785    0,136     1
Weighted Avg.   0,958    0,954    0,927     0,958    0,939      0,022    0,785    0,951

=== Confusion Matrix ===
      a      b  <-- classified as
243042  346 |      a = 0
 10232   60 |      b = 1
```

Figura 12 - Resultados WEKA com Random Forest e CV 10-F no dataset original

Como é possível constatar pelos resultados na figura anterior, nesta situação, apesar da acuidade geral ser de cerca de 96%, este é um fraco modelo de previsão, visto que permite uma alta precisão e exatidão na previsão de casos de não ocorrência de Acidente Vascular Cerebral, mas uma taxa de acuidade muito baixa para os casos contrários. Por esse motivo, foi testada a aplicação do mesmo algoritmo ao conjunto de dados, mas usando uma divisão do mesmo em 66% para treino e os restantes 34% para teste.

Nos resultados seguintes, podemos confirmar uma leve melhoria da precisão e acuidade ao prever casos de ocorrência de AVC, mas continuando muito baixos.

```

Time taken to build model: 197.81 seconds

=== Evaluation on test split ===

Time taken to test model on test split: 7.26 seconds

=== Summary ===

Correctly Classified Instances      82612          95.7809 %
Incorrectly Classified Instances    3639           4.2191 %
Kappa statistic                    0.0102
Mean absolute error                 0.0745
Root mean squared error             0.1949
Relative absolute error              95.3372 %
Root relative squared error          97.9569 %
Total Number of Instances          86251

=== Detailed Accuracy By Class ===

                TP Rate  FP Rate  Precision  Recall  F-Measure  MCC      ROC Area  PRC Area  Class
                0,999    0,993    0,959      0,999    0,978      0,029    0,789    0,986    0
                0,007    0,001    0,189      0,007    0,013      0,029    0,789    0,142    1
Weighted Avg.   0,958    0,952    0,927      0,958    0,939      0,029    0,789    0,951

=== Confusion Matrix ===

      a    b  <-- classified as
82588  103 |    a = 0
 3536   24 |    b = 1

```

Figura 13 - Resultados WEKA com Random Forest e Split 66% no dataset original

Assim sendo, foi realizado o mesmo processo, mas com diferentes percentagens na divisão do conjunto de dados de treino e teste. Destes testes, o melhor resultado obtido foi para uma divisão de 30% do conjunto para treino, onde a precisão se manteve praticamente igual e se verificou uma leve melhoria da taxa de acuidade e da precisão de casos de ocorrência de AVC, mas continuando muito baixos, na ordem de 1% e 22.9%, respetivamente.

Os modelos resultantes da aplicação deste algoritmo para treino a partir do *dataset* original encontram-se representados na tabela abaixo, na forma de precisão e taxa de acuidade do seu teste.

	Precisão Geral	Precisão <i>stroke</i> = 0	Precisão <i>stroke</i> = 1	Recall <i>stroke</i> = 0	Recall <i>stroke</i> = 1
Random Forest (CV 10-F)	95.8%	96%	14.8%	99.9%	0.6%
Random Forest (CV 15-F)	95.8%	96%	16.1%	99.9%	0.7%
Random Forest (Split 66%)	95.8%	95.9%	18.9%	99.9%	0.7%
Random Forest (Split 80%)	95.8%	95.9%	15.2%	99.9%	0.6%
Random Forest (Split 50%)	95.8%	95.9%	19.4%	99.9%	0.6%
Random Forest (Split 30%)	95.8%	95.9%	22.9%	99.9%	0.9%

Tabela 8 – Modelos WEKA com Random Forest no dataset original

Depois, seguindo os mesmos princípios, foram realizados vários testes com diferentes configurações para o algoritmo NaiveBayes e, de entre os algoritmos disponíveis no WEKA, fez também sentido aplicar o algoritmo BayesNet. Os modelos resultantes estão representados nas tabelas abaixo pelas suas medições de precisão e taxa de acuidade.

	Precisão Geral	Precisão <i>stroke</i> = 0	Precisão <i>stroke</i> = 1	Recall <i>stroke</i> = 0	Recall <i>stroke</i> = 1
NaiveBayes (CV 10-F)	94%	97.4%	15.1%	89.8%	43%
NaiveBayes (CV 15-F)	94%	97.4%	15.1%	89.8%	43%
NaiveBayes (Split 66%)	94%	97.4%	15.4%	89.7%	43.5%
NaiveBayes (Split 50%)	94%	97.4%	15.4%	89.7%	43.5%
NaiveBayes (Split 75%)	94.1%	97.4%	15.4%	89.7%	44.2%

Tabela 9 - Modelos WEKA com NaiveBayes no dataset original

	Precisão Geral	Precisão <i>stroke</i> = 0	Precisão <i>stroke</i> = 1	Recall <i>stroke</i> = 0	Recall <i>stroke</i> = 1
BayesNet (CV 10-F)	94.1%	97.4%	16.2%	90.6%	43%
BayesNet (CV 15-F)	94.1%	97.4%	16.2%	90.6%	43%
BayesNet (Split 66%)	94.1%	97.4%	16.6%	90.6%	43.2%
BayesNet (Split 50%)	94.1%	97.4%	16.5%	90.7%	43%
BayesNet (Split 75%)	94.2%	97.5%	16.6%	90.7%	44%

Tabela 10 - Modelos WEKA com BayesNet no dataset original

Tal como mencionado anteriormente, foi concebida uma variante do *dataset* original com mais representatividade dos casos de ocorrência de AVC, de forma a equilibrar e aproximar o conjunto de dados da realidade. Assim, foi-lhe seguidamente aplicado o mesmo processo, começando pelo algoritmo *Random Forest*.

	Precisão Geral	Precisão stroke = 0	Precisão stroke = 1	Recall stroke = 0	Recall stroke = 1
Random Forest (CV 10-F)	99.3%	99.9%	96.6%	99.3%	99.6%
Random Forest (CV 15-F)	99.4%	99.9%	96.7%	99.3%	99.6%
Random Forest (Split 66%)	98.8%	99.6%	94.6%	98.8%	98.2%
Random Forest (Split 80%)	99.1%	99.8%	95.8%	99.1%	99.2%
Random Forest (Split 50%)	97.7%	98.7%	92.6%	98.4%	93.9%

Tabela 11 - Modelos WEKA com Random Forest no dataset equilibrado

Ao contrário do que sucedeu para o *dataset* original, neste caso os modelos resultantes da aplicação deste algoritmo imediatamente se mostraram extremamente precisos e com taxas de acuidade muito altas na previsão de casos de ocorrência de Acidente Vascular Cerebral (atributo *stroke* com valor 1).

De seguida, prosseguiu-se, então, também com a modelação utilizando os algoritmos *Bayesianos* (*NaiveBayes* e *BayesNet*), cujos modelos resultantes estão representados nas tabelas seguintes pelas suas medições de precisão e taxa de acuidade na previsão em causa.

	Precisão Geral	Precisão stroke = 0	Precisão stroke = 1	Recall stroke = 0	Recall stroke = 1
NaiveBayes (CV 10-F)	81.7%	89.9%	43.4%	84.9%	54.7%
NaiveBayes (CV 15-F)	81.7%	89.9%	43.4%	84.9%	54.7%
NaiveBayes (Split 66%)	81.9%	89.9%	43.4%	85.1%	54.6%
NaiveBayes (Split 50%)	81.8%	89.9%	43.4%	85.0%	54.6%
NaiveBayes (Split 75%)	81.8%	89.8%	43.6%	85.2%	54.3%

Tabela 12 - Modelos WEKA com NaiveBayes no dataset equilibrado

	Precisão Geral	Precisão <i>stroke</i> = 0	Precisão <i>stroke</i> = 1	Recall <i>stroke</i> = 0	Recall <i>stroke</i> = 1
BayesNet (CV 10-F)	82.4%	90.5%	43.8%	84.2%	58.4%
BayesNet (CV 15-F)	82.4%	90.5%	43.9%	84.2%	58.4%
BayesNet (Split 66%)	82.6%	90.6%	43.9%	84.3%	58.5%
BayesNet (Split 50%)	82.4%	90.6%	43.9%	84.3%	58.4%
BayesNet (Split 75%)	82.4%	90.5%	44.2%	84.5%	58.0%

Tabela 13 - Modelos WEKA com BayesNet no dataset equilibrado

3.4.2 RapidMiner

Ainda na fase de modelação, de forma a conseguir obter o máximo de variabilidade nos modelos em termos de algoritmos e configurações relativas aos conjuntos de dados de entrada, foram implementados vários processos no *software* RapidMiner. O fator diferenciador é que este disponibiliza, para além de diferentes operadores de aprendizagem referentes a diversos algoritmos de *Machine Learning* para criação de modelos preditivos, também um importante operador de *Shuffle* que reordena aleatoriamente as instâncias do conjunto de dados de entrada. A sua importância está na possibilidade de impedir, neste caso, o agrupamento de muitos casos de ocorrência ou não ocorrência de AVC na subdivisão do *dataset* para Validação Cruzada ou na divisão em subconjuntos de dados de treino e teste, pois, aquando da elaboração desta segunda variante do conjunto de dados, todas as novas instâncias foram adicionadas no final do mesmo, não havendo, assim, uma distribuição homogênea das instâncias no *dataset*. Este operador permitiu não só uma validação dos resultados obtidos no WEKA, como uma análise da diferença dos resultados.

Começou-se, então, pela modelação com o algoritmo *Random Forest* com Validação Cruzada. Foi elaborado um Processo onde é feita a transformação do atributo *stroke* de numérico para nominal, é sinalizado como atributo a prever e depois é feita a referida reordenação das instâncias do conjunto de dados, de forma automática.

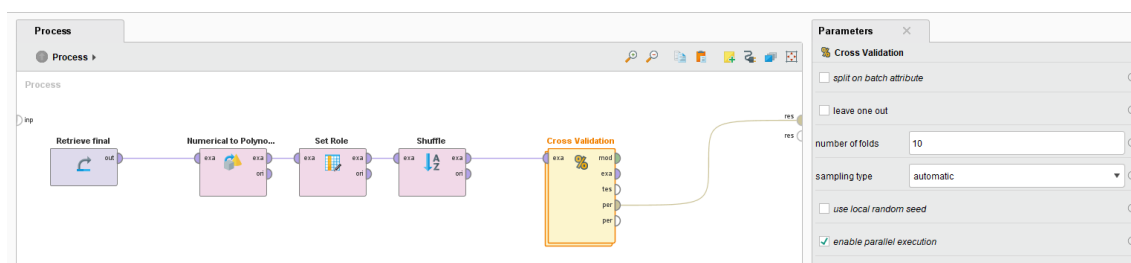


Figura 14 - Processo RapidMiner do modelo Random Forest e CV 10-F no dataset original

O operador de Validação Cruzada, configurado, neste caso, com 10 subdivisões, contém o algoritmo a aplicar e os operadores básicos de aplicação do modelo para teste e medição da sua performance.

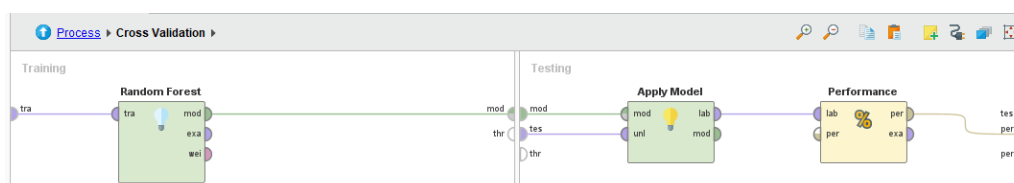


Figura 15 – Processo RapidMiner interno da CV 10-F no dataset original

Tal como esperado, os resultados foram semelhantes à aplicação no WEKA: apesar de alta precisão geral e taxa de acuidade nos casos onde não ocorrem AVC, é quase sempre previsto pelo modelo que o valor do atributo *stroke* seja 0, pelo que para o caso contrário a taxa de acuidade é nula (0%).

accuracy: 95.94% +/- 0.00% (micro average: 95.94%)

	true 0	true 1	class precision
pred. 0	243387	10292	95.94%
pred. 1	1	0	0.00%
class recall	100.00%	0.00%	

Figura 16 - Resultados RapidMiner com Random Forest e CV 10-F no dataset original

Da mesma forma, procedeu-se à modelação com o mesmo algoritmo mas com divisão do *dataset* em dados de treino (66%) e dados de teste (34%).

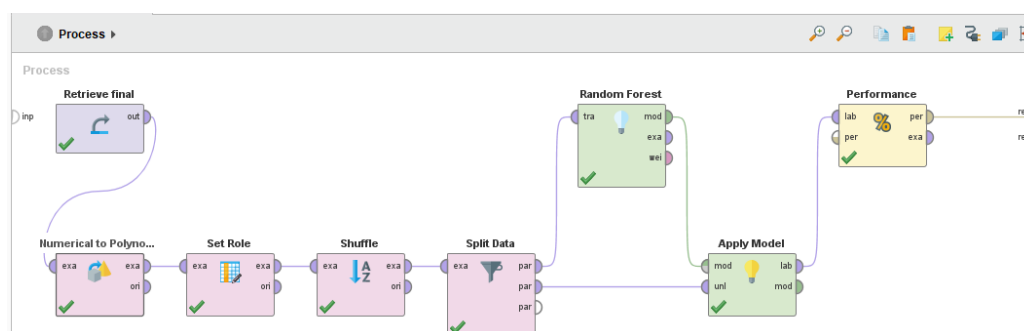


Figura 17 – Processo RapidMiner do modelo Random Forest e Split 66% no dataset original

Os resultados foram semelhantes, como é possível constatar na imagem abaixo.

accuracy: 95.94%

	true 0	true 1	class precision
pred. 0	82752	3499	95.94%
pred. 1	0	0	0.00%
class recall	100.00%	0.00%	

Figura 18 - Resultados RapidMiner com Random Forest e Split 66% no dataset original

Também para o algoritmo *NaiveBayes*, com configurações dos processos semelhantes, foram obtidos resultados idênticos àqueles que haviam sido alcançados no WEKA.

accuracy: 84.18% +/- 0.22% (micro average: 84.18%)

	true 0	true 1	class precision
pred. 0	207881	4636	97.82%
pred. 1	35507	5656	13.74%
class recall	85.41%	54.96%	

Figura 19 - Resultados RapidMiner com NaiveBayes e CV 10-F no dataset original

accuracy: 84.26%			
	true 0	true 1	class precision
pred. 0	70736	1564	97.84%
pred. 1	12016	1935	13.87%
class recall	85.48%	55.30%	

Figura 20 - Resultados RapidMiner com NaiveBayes e Split 66% no dataset original

Como os modelos induzidos através dos algoritmos *NaiveBayes* e *BayesNet* foram anteriormente muito semelhantes em termos de valores de precisão e taxa de acuidade, foi então desconsiderada a indução com o segundo e foram, desta feita, construídos processos semelhantes para modelação com o algoritmo de Regressão Logística, quer com divisão em dados de treino e teste, como é exemplo a figura seguinte, como também com Validação Cruzada.

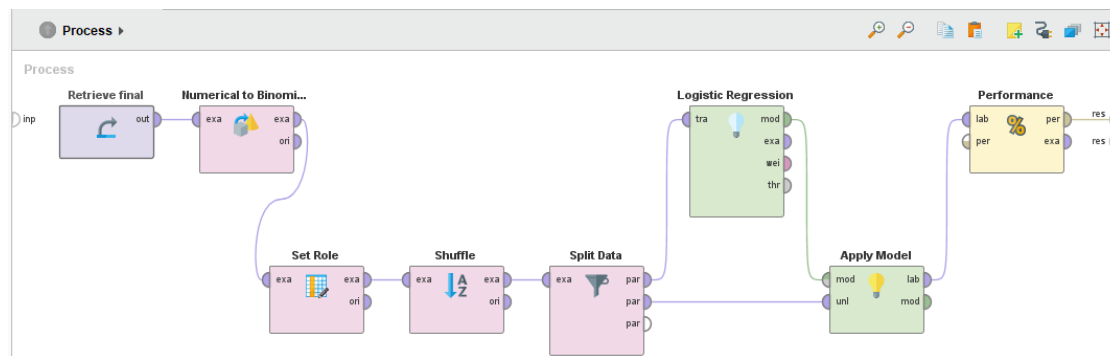


Figura 21 - Processo RapidMiner do modelo de Regressão Logística e Split 66% no dataset original

Para estes casos, o operador de transformação do atributo *stroke* foi alterado para binominal, mas, como existem apenas dois possíveis valores para o mesmo (0 ou 1), o efeito é o mesmo que o operador anteriormente utilizado, pelo que a alteração foi realizada devido apenas a restrições do *software* para este algoritmo.

accuracy: 95.92%			
	true false	true true	class precision
pred. false	82716	3480	95.96%
pred. true	36	19	34.55%
class recall	99.96%	0.54%	

Figura 22 - Resultados RapidMiner com Regressão Logística e Split 66% no dataset original

accuracy: 95.93% +/- 0.02% (micro average: 95.93%)

	true false	true true	class precision
pred. false	243307	10233	95.96%
pred. true	81	59	42.14%
class recall	99.97%	0.57%	

Figura 23 - Resultados RapidMiner com Regressão Logística e CV 10-F no dataset original

Os modelos obtidos no RapidMiner até este ponto com os vários algoritmos utilizados sugeriram que o operador de *Shuffle* aplicado ao conjunto de dados original não surtia uma melhoria significativa em termos de precisão e acuidade na previsão de ocorrência de AVC, provavelmente devido à pouca significância de instâncias com o atributo *stroke* com valor a 1, mostrando-se ser um forte indicador de que a criação da variante equilibrado do conjunto de dados teria sido acertada e de que seria sobre o mesmo que deveriam ser aplicados os algoritmos e inferidos os modelos de previsão. Por esse motivo, foram, então, desenvolvidos processos semelhantes aos anteriores, mas a executar sobre o conjunto de dados mais equilibrado.

Para cada algoritmo aplicado foram, para efeitos de comparação, usadas as mesmas configurações para a indução dos modelos em WEKA. Os resultados são os que se apresentam na tabela abaixo, retratados pelas medições de precisão, acuidade e *recall* disponíveis.

Algoritmo	Configurações	Acuidade Geral	Precisão stroke = 0	Precisão stroke = 1	Recall stroke = 0	Recall stroke = 1
Random Forest	CV 10-F	84.32%	85.75%	63.61%	97.14%	23.67%
	CV 15-F	84.33%	85.76%	63.70%	97.15%	23.69%
	Split 66%	84.33%	85.74%	63.80%	97.17%	23.58%
	Split 80%	84.25%	85.64%	63.52%	97.22%	22.89%
	Split 50%	84.31%	85.76%	63.54%	97.12%	23.73%
NaiveBayes	CV 10-F	77.60%	91.37%	40.94%	80.46%	64.06%
	CV 15-F	77.60%	91.37%	40.94%	80.46%	64.06%
	Split 66%	77.87%	91.46%	41.38%	80.73%	64.33%
	Split 80%	77.76%	91.34%	41.15%	80.70%	64.83%
	Split 50%	77.80%	91.36%	41.22%	80.74%	63.87%
Regressão Logística	CV 10-F	84.08%	86.33%	59.24%	95.90%	28.15%
	CV 15-F	84.08%	86.33%	59.24%	95.90%	28.15%
	Split 66%	84.16%	86.39%	59.71%	95.93%	28.50%
	Split 80%	84.04%	86.31%	58.97%	95.87%	28.07%
	Split 50%	84.08%	86.34%	59.22%	95.89%	28.23%

Tabela 14 - Resultados RapidMiner no dataset equilibrado

Os modelos de previsão induzidos com o conjunto de dados equilibrado apresentam taxas de acuidade e previsão bastante inferiores às obtidas em WEKA, onde as diferenças se centram na utilização do operador de *Shuffle* disponibilizado pelo RapidMiner. Por esse motivo, de forma a estabelecer um termo de comparação e de controlo dos modelos induzidos, foram treinados com o algoritmo *Random Forest*, que se havia mostrado, de um modo geral, mais preciso e eficaz na previsão de AVC até este ponto, e recorrendo a processos RapidMiner semelhantes aos explicitados acima, novos modelos preditivos, mas sem a utilização do operador *Shuffle*. Os valores das métricas obtidas do teste de cada modelo estão representados na tabela seguinte.

Algoritmo	Configurações	Acuidade Geral	Precisão <i>stroke</i> = 0	Precisão <i>stroke</i> = 1	<i>Recall</i> <i>stroke</i> = 0	<i>Recall</i> <i>stroke</i> = 1
<i>Random Forest</i> (sem <i>Shuffle</i>)	CV 10-F	84.34%	85.75%	63.86%	97.17%	23.63%
	CV 15-F	84.34%	85.76%	63.83%	97.16%	23.68%
	Split 66%	84.23%	85.83%	62.40%	95.90%	24.31%
	Split 80%	84.13%	85.75%	61.72%	96.87%	23.84%
	Split 50%	84.32%	85.71%	63.90%	97.21%	23.24%

Tabela 15 - Resultados RapidMiner no dataset equilibrado (Random Forest, sem Shuffle)

3.4.3 Python


Finalmente, recorrendo à linguagem de programação Python, foram aplicados, para além dos algoritmos de *Machine Learning* utilizados anteriormente, outros adequados ao problema de classificação em questão. Deste modo, foi possível não só constatar se outros modelos induzidos com outros algoritmos seriam mais precisos na previsão de ocorrência de AVC em relação aos induzidos até aqui, como validar os resultados anteriormente obtidos.

Para isso, foi utilizada a distribuição Anaconda da linguagem de programação Python, específica para computação científica e que visa simplificar a gestão de pacotes e implementação. Esta inclui pacotes para ciência de dados, aprendizagem automática, análise preditiva, entre outros, como é o caso da biblioteca *scikit-learn*, utilizada neste projeto.

Inicialmente, tal como aconteceu na modelação em WEKA e RapidMiner, foi necessário transformar o atributo *stroke* de cada conjunto de dados (original e variante equilibrada) de numérico para nominal (*category*, neste contexto), de forma que este só pudesse ter valores absolutos (0 ou 1) correspondentes à ocorrência, ou não, de Acidente Vascular Cerebral para um indivíduo.

```
#convert stroke to category
dataset['Stroke'] = dataset['Stroke'].astype('category',copy=False)
```

Figura 24 – Código Python transformação do tipo de dados do atributo stroke



Data columns (total 22 columns):				
#	Column	Non-Null Count		Dtype
0	Diabetes_012	294848 non-null		int64
1	HighBP	294848 non-null		int64
2	HighChol	294848 non-null		int64
3	CholCheck	294848 non-null		int64
4	BMI	294848 non-null		int64
5	Smoker	294848 non-null		int64
6	Stroke	294848 non-null		int64
7	HeartDiseaseorAttack	294848 non-null		int64
8	PhysActivity	294848 non-null		int64
9	Fruits	294848 non-null		int64
10	Veggies	294848 non-null		int64
11	HvyAlcoholConsump	294848 non-null		int64
12	AnyHealthcare	294848 non-null		int64
13	NoDocbcCost	294848 non-null		int64
14	GenHlth	294848 non-null		int64
15	MentHlth	294848 non-null		int64
16	PhysHlth	294848 non-null		int64
17	DiffWalk	294848 non-null		int64
18	Sex	294848 non-null		int64
19	Age	294848 non-null		int64
...				
20	Education	294848 non-null		int64
21	Income	294848 non-null		int64
				dtypes: int64(22)

Data columns (total 22 columns):				
#	Column	Non-Null Count		Dtype
0	Diabetes_012	294848 non-null		int64
1	HighBP	294848 non-null		int64
2	HighChol	294848 non-null		int64
3	CholCheck	294848 non-null		int64
4	BMI	294848 non-null		int64
5	Smoker	294848 non-null		int64
6	Stroke	294848 non-null		category
7	HeartDiseaseorAttack	294848 non-null		int64
8	PhysActivity	294848 non-null		int64
9	Fruits	294848 non-null		int64
10	Veggies	294848 non-null		int64
11	HvyAlcoholConsump	294848 non-null		int64
12	AnyHealthcare	294848 non-null		int64
13	NoDocbcCost	294848 non-null		int64
14	GenHlth	294848 non-null		int64
15	MentHlth	294848 non-null		int64
16	PhysHlth	294848 non-null		int64
17	DiffWalk	294848 non-null		int64
18	Sex	294848 non-null		int64
19	Age	294848 non-null		int64
...				
20	Education	294848 non-null		int64
21	Income	294848 non-null		int64
				dtypes: category(1), int64(21)

Figura 25 - Tipo de dados dos datasets

Depois, foram definidos os conjuntos de dados de treino e teste a partir dos *datasets*, através do método de subdivisão da imagem abaixo. Neste, é possível sinalizar o atributo a prever e em que percentagem dividir o conjunto de dados inicial nos subconjuntos de treino e teste. É relevante mencionar que em todos os testes foi utilizada aleatoriedade na ordenação das instâncias, desta vez através do valor de *random_state* definido neste método. Caso seja um número, como foi neste caso concreto, esse valor é usado pelo gerador de números aleatórios (*seed*). Em todos os testes realizados abaixo foi-lhe atribuído o mesmo valor, neste caso 0, para que os testes e seus resultados fossem consistentes, independentes dessa *seed* de aleatoriedade e replicáveis futuramente.

```
# Labels
y = dataset['Stroke']

# Predictor variables
X = dataset.drop(['Stroke'], axis=1)

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=.2, random_state=0)

sc = StandardScaler()
X_train = sc.fit_transform(X_train)
X_test = sc.transform(X_test)
```

Figura 26 – Método de subdivisão em conjuntos de dados de treino e teste

Foram, então, escolhidos os seguintes classificadores/algoritmos considerados adequados ao problema em estudo e a aplicar a cada conjunto de dados pela ordem apresentada na figura. Em cada execução do código foram alteradas apenas as percentagens de divisão dos subconjuntos de treino e teste referidas acima, para ambos os *datasets*, de forma a manter a consistência dos modelos e ser possível e válida a sua comparação.

```
# Logistic Regression
from sklearn.linear_model import LogisticRegression
models['Logistic Regression'] = LogisticRegression()

# Support Vector Machines
from sklearn.svm import LinearSVC
models['Support Vector Machines'] = LinearSVC()

# Decision Trees
from sklearn.tree import DecisionTreeClassifier
models['Decision Trees'] = DecisionTreeClassifier()

# Random Forest
from sklearn.ensemble import RandomForestClassifier
models['Random Forest'] = RandomForestClassifier()

# Naive Bayes
from sklearn.naive_bayes import GaussianNB
models['Naive Bayes'] = GaussianNB()

# K-Nearest Neighbors
from sklearn.neighbors import KNeighborsClassifier
models['K-Nearest Neighbor'] = KNeighborsClassifier()
```

Figura 27 - Classificadores/algoritmos utilizados para indução dos modelos em Python

Para cada modelo resultante induzido a partir dos algoritmos referidos sobre o conjunto de dados original, foram obtidos os seguintes valores de precisão, acuidade e *recall* na previsão de ocorrência de AVC.

	Accuracy	Precision	Recall
Logistic Regression	0.959256	0.004657	0.387097
Support Vector Machines	0.959366	0.000000	0.000000
Decision Trees	0.918590	0.155025	0.118021
Random Forest	0.958728	0.005433	0.204380
Naive Bayes	0.841635	0.557043	0.138863
K-Nearest Neighbor	0.955992	0.032984	0.221354

Figura 28 - Resultados Python com Split de 0.5 no dataset original

	Accuracy	Precision	Recall
Logistic Regression	0.958842	0.005079	0.428571
Support Vector Machines	0.958911	0.000000	0.000000
Decision Trees	0.917567	0.151806	0.115898
Random Forest	0.958088	0.006490	0.196581
Naive Bayes	0.842299	0.562359	0.141911
K-Nearest Neighbor	0.955920	0.036117	0.249027

Figura 29 - Resultados Python com Split de 0.66 no dataset original

	Accuracy	Precision	Recall
Logistic Regression	0.958885	0.004317	0.473684
Support Vector Machines	0.958905	0.000000	0.000000
Decision Trees	0.919387	0.168345	0.129664
Random Forest	0.957880	0.005276	0.148649
Naive Bayes	0.843090	0.574580	0.144826
K-Nearest Neighbor	0.956086	0.038849	0.265574

Figura 30 - Resultados Python com Split de 0.8 no dataset original

Da mesma forma que aconteceu anteriormente, os modelos resultantes não se demonstraram precisos para nenhum dos algoritmos sobre o conjunto de dados original, pelo que também se procedeu à modelação sobre a variante mais equilibrada do mesmo, de forma a validar os resultados obtidos anteriormente e identificar possíveis melhores algoritmos/classificadores. Do teste dos modelos resultantes deste processo foram obtidas as seguintes métricas na previsão de ocorrência de Acidente Vascular Cerebral.

	Accuracy	Precision	Recall
Logistic Regression	0.841322	0.281616	0.602183
Support Vector Machines	0.842536	0.246124	0.631485
Decision Trees	0.926817	0.945989	0.722623
Random Forest	0.977575	0.940421	0.932347
Naive Bayes	0.776671	0.643418	0.412513
K-Nearest Neighbor	0.844327	0.599033	0.551918

Figura 31 - Resultados Python com Split de 0.5 no dataset equilibrado

	Accuracy	Precision	Recall
Logistic Regression	0.841704	0.285070	0.599592
Support Vector Machines	0.842891	0.250499	0.627017
Decision Trees	0.945376	0.983513	0.768716
Random Forest	0.988219	0.983114	0.951154
Naive Bayes	0.776985	0.641965	0.411685
K-Nearest Neighbor	0.861934	0.742256	0.582565

Figura 32 - Resultados Python com Split de 0.66 no dataset equilibrado

	Accuracy	Precision	Recall
Logistic Regression	0.843158	0.287605	0.605366
Support Vector Machines	0.843921	0.253551	0.629773
Decision Trees	0.952857	0.994260	0.789722
Random Forest	0.992522	0.994941	0.963354
Naive Bayes	0.780142	0.647013	0.415963
K-Nearest Neighbor	0.889452	0.882662	0.630657

Figura 33 - Resultados Python com Split de 0.8 no dataset equilibrado

Tal como havia sido registado para os modelos induzidos anteriormente, também o algoritmo *Random Forest* foi o mais preciso e acertado na previsão em estudo, embora os valores para estas métricas acima dos 95% sejam dissonantes com os valores obtidos no RapidMiner, onde, tal como aqui, é feita a reordenação aleatória das instâncias dos conjuntos de dados antes da sua subdivisão em dados de treino e teste e esses valores foram mais baixos, enquanto que no WEKA, onde isso não acontece, a precisão e taxa de acuidade foram mais próximas dos valores agora obtidos. Sendo assim, e havendo suspeitas de uma possível alta influência da *seed* de aleatoriedade dessa referida reordenação nos resultados finais dos modelos induzidos, foram realizados os mesmos testes, mas com esse valor do gerador de números aleatórios acima explicado com valor 42 (sem qualquer significado, apenas para teste diferencial).

```
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=.2, random_state=42)
```

Figura 34 – Alteração da seed de aleatoriedade no código Python

Para cada modelo induzido com as novas configurações mencionadas sobre o conjunto de dados equilibrado, foram obtidos os seguintes valores de precisão e taxa de acuidade na previsão de ocorrência de AVC.

	Accuracy	Precision	Recall
Logistic Regression	0.841505	0.282917	0.594973
Support Vector Machines	0.842339	0.244180	0.620929
Decision Trees	0.926884	0.947948	0.720576
Random Forest	0.977962	0.940396	0.933599
Naive Bayes	0.776224	0.636845	0.408746
K-Nearest Neighbor	0.843879	0.599237	0.547485

Figura 35 - Resultados Python com Split de 0.5 no dataset equilibrado (random_state = 42)

	Accuracy	Precision	Recall
Logistic Regression	0.842043	0.280257	0.599485
Support Vector Machines	0.839809	0.249384	0.595785
Decision Trees	0.944807	0.983561	0.765992
Random Forest	0.988838	0.982817	0.954444
Naive Bayes	0.777155	0.638639	0.410213
K-Nearest Neighbor	0.874024	0.733948	0.616117

Figura 36 - Resultados Python com Split de 0.66 no dataset equilibrado (random_state = 42)

	Accuracy	Precision	Recall
Logistic Regression	0.840597	0.285228	0.589357
Support Vector Machines	0.825827	0.002624	0.771429
Decision Trees	0.953451	0.993197	0.792555
Random Forest	0.992233	0.993100	0.963511
Naive Bayes	0.775428	0.639456	0.408366
K-Nearest Neighbor	0.898949	0.872983	0.658819

Figura 37 - Resultados Python com Split de 0.8 no dataset equilibrado (random_state = 42)

Os modelos resultantes continuaram a apresentar uma diferença considerável nos valores de precisão, acuidade e *recall* em relação aos obtidos em RapidMiner e bastante semelhantes entre si e aos resultados anteriormente alcançados em WEKA e Python, independentemente da aleatoriedade a que se recorre em cada momento para a reordenação das instâncias do conjunto de dados antes da indução dos modelos.

Outra importante métrica possível de obter na indução destes modelos com o algoritmo *Random Forest* retrata quais os atributos que mais pesam na previsão da ocorrência de AVC. Cada árvore da “floresta” resultante do treino feito a partir dos dados pode calcular a importância de uma característica de acordo com a sua capacidade de aumentar a pureza das folhas. Quanto maior o incremento na pureza das folhas, maior é a importância dessa característica. Isto é feito para cada árvore, depois é calculada a média entre todas as árvores e, finalmente, normalizado para 1. Assim, a soma das importâncias calculadas é 1 e está distribuída, neste caso, da seguinte forma:

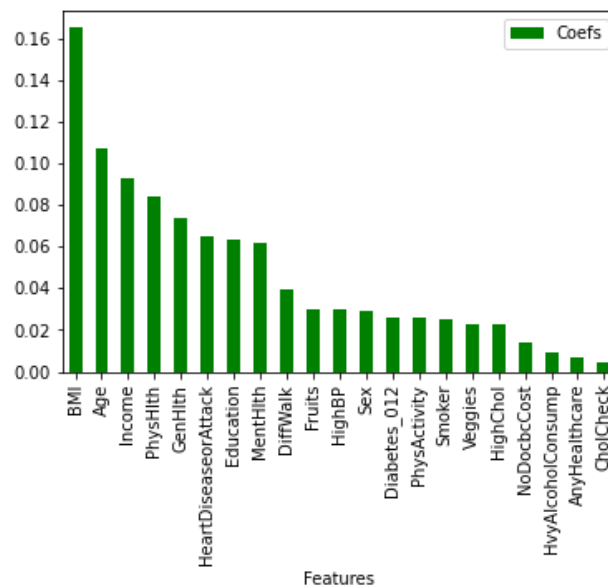


Figura 38 – Gráfico da distribuição do peso de cada atributo

3.5 Avaliação dos Modelos

Nesta fase do processo de *Data Mining*, foi feita uma avaliação mais aprofundada dos modelos induzidos na fase anterior, dos passos executados e dos resultados obtidos nos testes de cada um deles, de forma a garantir que alcança corretamente os objetivos definidos inicialmente. Essa avaliação passa pela validação e pela comparação dos resultados obtidos nos testes dos modelos treinados com as várias ferramentas utilizadas, aplicando diferentes algoritmos aos dois conjuntos de dados referidos anteriormente, em termos de precisão, acuidade e *recall* na previsão da ocorrência de Acidente Vascular Cerebral.

Primeiramente, todos os modelos induzidos em WEKA apresentaram valores gerais de precisão e acuidade altos, embora essas métricas tenham sido bastante baixas na previsão dos casos específicos de incidência de AVC para os modelos obtidos a partir do *dataset* original com todos os algoritmos. Isto deveu-se, como se veio a confirmar, ao facto da pouca representatividade das instâncias com valor do atributo *stroke* a 1, levando a pouco treino para esses casos em relação aos casos contrários.

Por outro lado, os resultados dos modelos induzidos a partir do *dataset* equilibrado apresentaram para todos os algoritmos uma melhoria significativa, devido, desta feita, à maior quantidade de dados de treino com valores a 1 do atributo principal. Já nesta fase, foi possível identificar uma superioridade do algoritmo *Random Forest* em relação aos classificadores *Bayesianos*, mas valores tão altos nas métricas de teste levantaram suspeitas acerca da sua validade, pois, tal como foi mencionado anteriormente, aquando da elaboração da variante equilibrada do conjunto de dados a partir do original, a replicação das instâncias nas quais há incidência de AVC foi realizada colocando-as no final do conjunto de dados, não sendo a sua distribuição uniforme. Isto poderia ter influência nestes resultados, devido às subdivisões feitas em dados de treino e teste e à validação cruzada, explicitadas anteriormente, não serem balanceadas em termos de razão entre instâncias com valor do atributo principal a 0 e a 1. Por esse motivo, fez sentido recorrer ao RapidMiner e ao seu operador de *Shuffle* das instâncias, que poderia, recorrendo à aleatoriedade, validar a importância da ordem das instâncias no conjunto de dados, ou não.

Neste *software*, utilizando o operador de *Shuffle*, foram treinados modelos com várias configurações e com dois dos algoritmos usados anteriormente (*NaiveBayes* e *Random Forest*) e, ainda, com Regressão Logística. É possível constatar que o algoritmo *Random Forest* se mostrou, mais uma vez, o melhor em termos das métricas resultantes dos testes com modelos induzidos, mas os valores das mesmas mostraram-se mais baixos do que aqueles que haviam sido obtidos nos testes com os modelos de previsão induzidos em WEKA a partir, também, do *dataset* equilibrado.

	WEKA		RapidMiner	
	Precisão <i>stroke</i> = 0	Precisão <i>stroke</i> = 1	Precisão s <i>stroke</i> = 0	Precisão <i>stroke</i> = 1
<i>Random Forest</i> (CV 10-F)	99.9%	96.6%	85.75%	63.61%
<i>Random Forest</i> (CV 15-F)	99.9%	96.7%	85.76%	63.70%
<i>Random Forest</i> (Split 66%)	99.6%	94.6%	85.74%	63.80%
<i>Random Forest</i> (Split 80%)	99.8%	95.8%	85.64%	63.52%
<i>Random Forest</i> (Split 50%)	98.7%	92.6%	85.76%	63.54%

Tabela 16 – Tabela comparação da precisão do *Random Forest* em WEKA e RapidMiner

À partida, parecia que o operador Shuffle trazia algum realismo ao treino e teste dos modelos, no sentido de homogeneizar a dispersão das instâncias com o atributo *stroke* com valor a 1. De forma a confirmar esta teoria, e tal como foi mencionado na fase de modelação, foram também induzidos modelos de previsão sem esse operador de reordenação aleatória em RapidMiner, apenas para o algoritmo *Random Forest*, por se mostrar, de um modo geral, o mais eficaz de entre os outros que haviam sido utilizados.

Algoritmo	Configurações		Acerto Geral	Precisão <i>stroke</i> = 0	Precisão <i>stroke</i> = 1	<i>Recall</i> <i>stroke</i> = 0	<i>Recall</i> <i>stroke</i> = 1
<i>Random Forest</i>	Com <i>Shuffle</i>	CV 10-F	84.32%	85.75%	63.61%	97.14%	23.67%
		CV 15-F	84.33%	85.76%	63.70%	97.15%	23.69%
		Split 66%	84.33%	85.74%	63.80%	97.17%	23.58%
		Split 80%	84.25%	85.64%	63.52%	97.22%	22.89%
		Split 50%	84.31%	85.76%	63.54%	97.12%	23.73%
	Sem <i>Shuffle</i>	CV 10-F	84.34%	85.75%	63.86%	97.17%	23.63%
		CV 15-F	84.34%	85.76%	63.83%	97.16%	23.68%
		Split 66%	84.23%	85.83%	62.40%	95.90%	24.31%
		Split 80%	84.13%	85.75%	61.72%	96.87%	23.84%
		Split 50%	84.32%	85.71%	63.90%	97.21%	23.24%

Tabela 17 - Tabela comparação resultados *Random Forest* em RapidMiner com e sem *Shuffle*

Nestes testes, os resultados obtidos mostraram o contrário: a aleatoriedade na reordenação das instâncias do conjunto de dados antes da divisão do mesmo em dados de treino e teste em nada influencia a precisão, acuidade e *recall* dos modelos induzidos. Se essa ordem e consequente distribuição das instâncias para treino não foi a causa da diferença entre os valores supramencionados, então a ponto

diferencial poderia estar na implementação do algoritmo. Para comprovar esta possibilidade, foi desenvolvido um *script* em Python no qual, para cada conjunto de dados (original e equilibrado) e com diferentes percentagens de divisão do dataset em dados de treino e teste, foram induzidos modelos com vários algoritmos, alguns já aplicados anteriormente e outros diferentes, considerados também adequados.

Tal como explicitado na fase de modelação, a biblioteca de Python utilizada para o desenvolvimento deste problema de previsão utiliza, também ela, aleatoriedade para a reordenação das instâncias do conjunto de dados antes da divisão em subconjuntos de treino e teste, sendo possível, através da alteração da *seed* de geração aleatória de números utilizada pela mesma, o controlo da influência dessa aleatoriedade na indução de modelos nas mesmas condições. Para além disso e nesse mesmo sentido, a indução de modelos nestas condições permite a direta comparação dos mesmos com os obtidos em RapidMiner com o operador *Shuffle*, de forma a verificar, mais uma vez, o impacto, ou não, dessa reordenação nos resultados.

Dos resultados obtidos em termos de métricas na previsão da ocorrência de Acidente Vascular Cerebral com os modelos induzidos em Python obteve-se, desde logo, a confirmação daquilo que haviam sido as conclusões iniciais em relação aos dois conjuntos de dados: a variante equilibrada do conjunto de dados original mostrou-se sempre superior em termos de acuidade, precisão e *recall*.

Por esse motivo, o treino dos modelos a ter em conta e usados para comparação em termos de métricas na tabela abaixo foi feito a partir do mesmo. Da mesma forma, também aqui se confirmou que o algoritmo *Random Forest* é o melhor algoritmo para o treino dos modelos em estudo e para todos os casos de teste, tal como havia acontecido na indução dos modelos anteriores, motivo pelo qual a referida tabela retrata apenas a sua performance para todas as ferramentas utilizadas.

	WEKA		RapidMiner		Python	
	Precisão <i>stroke</i> = 0	Precisão <i>stroke</i> = 1	Precisão <i>stroke</i> = 0	Precisão <i>stroke</i> = 1	Precisão geral (RS = 0)	Precisão geral (RS = 42)
<i>Random Forest</i> (CV 10-F)	99.9%	96.6%	85.75%	63.61%	NA	NA
<i>Random Forest</i> (CV 15-F)	99.9%	96.7%	85.76%	63.70%	NA	NA
<i>Random Forest</i> (Split 66%)	99.6%	94.6%	85.74%	63.80%	98.31%	98.28%
<i>Random Forest</i> (Split 80%)	99.8%	95.8%	85.64%	63.52%	99.49%	99.31%
<i>Random Forest</i> (Split 50%)	98.7%	92.6%	85.76%	63.54%	94.04%	94.03%

Tabela 18 - Tabela comparação resultados *Random Forest* em WEKA, RapidMiner e Python

Das comparações possíveis de retirar da análise da tabela anterior, é fácil constatar que, tal como tinha acontecido em RapidMiner, onde se havia concluído que a ordenação não influenciava os resultados em termos de métricas nos testes de previsão, também em Python foi possível comprovar este facto, pois a precisão foi tão alta como a dos modelos induzidos em WEKA com o *dataset* equilibrado, onde não havia ocorrido essa ordenação aleatória das instâncias do *dataset*, assim como se pode confirmar, por observação das duas últimas colunas, que essa aleatoriedade na reordenação das instâncias tem uma influência praticamente nula na performance do modelo treinado.

Comprovou-se, então, também a teoria inicial de que a implementação do algoritmo nas diferentes ferramentas utilizadas pode influenciar a eficácia dos modelos induzidos na previsão. É possível verificar que a diferença de performance nos testes entre os modelos induzidos em WEKA e Python em comparação com os obtidos em RapidMiner é significativa. A única explicação para este facto será algum ou mais pontos diferenciais entre as implementações do algoritmo em cada uma das ferramentas utilizada, causando resultados distintos do treino realizado pela aplicação dos mesmos sobre o mesmo conjunto de dados, com as mesmas configurações.

Pode-se, então, concluir nesta fase do processo de Mineração de Dados que *Random Forest* é o melhor algoritmo para treino de modelos de previsão da ocorrência de Acidente Vascular Cerebral para os atributos presentes nos *datasets* em estudo, em todos os casos, mas que a proporção de instâncias em que o atributo representativo desta incidência tem valor 1 em relação aos casos contrários tem que ser significativa, visível na comparação dos resultados obtidos dos testes feitos com os modelos induzidos com os dois conjuntos de dados utilizados. A amostra usada para treino do modelo deverá ser equilibrada em termos de proporção de casos positivos e negativos, verificada pela larga diferença em termos de precisão, acuidade e *recall* entre os modelos induzidos a partir do conjunto original e do conjunto de dados equilibrado e que contém uma representação mais real de volume de ocorrências. Também o facto de a aleatoriedade não influenciar a performance de previsão dos modelos induzidos a partir do conjunto de dados equilibrados, mostra que este tem proporção suficiente entre instâncias com atributo *stroke* com valor a 0 e a 1 para que o treino permita uma precisão de quase 100% na previsão de ocorrência de AVC.

Outra importante conclusão a retirar é que as várias configurações de treino em termos de Validação Cruzada e divisão dos dados em subconjuntos para treino e teste utilizadas mostram que, regra geral, não há muita preponderância por parte da quantidade de dados usados para treino nos resultados dos modelos induzidos, para um conjunto de dados com número de instâncias desta magnitude, pois todas

as métricas utilizadas como meio de comparação neste estudo foram muito altas, perto dos 100%, com o referido algoritmo, a partir do conjunto de dados equilibrado para todas as diferentes configurações de treino e teste.

Também desta análise, é possível reiterar que cada atributo do conjunto de dados tem uma importância/peso distintos na previsão de ocorrência de AVC nos modelos induzidos considerados ideais (com o algoritmo *Random Forest*), significando que há características de um indivíduo que têm mais relevância na incidência deste problema e que podem implicar, direta ou indiretamente, maior risco. Aquele que se apresentou como principal fator de risco de entre os considerados neste estudo foi o índice de massa corporal, mas outros indicadores também se mostraram preponderantes como, por exemplo e por ordem decrescente de influência, a idade, o nível de rendimento familiar anual, problemas no que toca à saúde física ou lesões, a autoavaliação do estado de saúde geral, problemas ou ataque cardíacos, o nível de educação e, ainda, problemas no que toca à saúde mental.

4. CONCLUSÕES

O objetivo principal deste projeto passa pela indução de um ou mais modelos de previsão da ocorrência de Acidente Vascular Cerebral num indivíduo, através de um conjunto de características que o definam, que sejam o mais precisos possível.

Para tal, todo o desenvolvimento do projeto seguiu a metodologia *Design Science Research*, cuja fase de conceção e desenvolvimento se define como um projeto de *Data Mining* que, por sua vez, seguiu a metodologia CRISP-DM. Um projeto deste género pode, desde logo, revelar diversas complicações, nomeadamente, nas opções tomadas inicialmente no que toca à seleção do conjunto de dados a utilizar, aos algoritmos a aplicar e às ferramentas escolhidas para o executar, que podem condicionar o sucesso do mesmo. Assim, as metodologias referidas, mantendo sempre presentes os objetivos traçados, foram um importante guia para o desenvolvimento do trabalho retratado no presente documento.

Neste explicitam-se, em jeito de introdução, a contextualização e motivação do trabalho a desenvolver, definem-se os objetivos e explica-se a metodologia a seguir, como ponto de partida para o desenvolvimento do mesmo. Depois, relata-se todo o trabalho de investigação, mais concretamente no que toca às técnicas, tecnologias e ferramentas mais utilizadas no mercado, de forma a justificar as opções tomadas adiante. Finalmente, mostra-se todo o trabalho desenvolvido em termos de modelos induzidos e a avaliação feita aos mesmos.

Desta, concluem-se vários pontos importantes para uma futura implementação do trabalho desenvolvido num contexto real. Primeiramente, será sempre de extrema relevância que o modelo treine sobre um conjunto de dados cujos atributos sejam o mais próximos possível dos utilizados neste estudo e que a proporção de casos de ocorrência de AVC em relação aos casos opostos seja, pelo menos, de cerca de um para seis. Assim, para além de retratar um contexto mais próximo do real por estar em concordância com as previsões da OMS, esta é suficiente para um treino adequado dos modelos, como foi mostrado pelos resultados obtidos na indução de modelos a partir do conjunto de dados equilibrado por replicação de instâncias. O modelo deverá ser treinado com um algoritmo *Random Forest*, verificado neste estudo como aquele que apresenta melhor performance na previsão da ocorrência de Acidente Vascular Cerebral num indivíduo, em termos das métricas inferidas, para todas as ferramentas usadas. Nesse treino e respetivo teste dos modelos foram ainda empregues várias configurações, de forma a controlar e comparar os resultados, mostrando, ainda, que a ordem das instâncias no *dataset* não afeta

a performance dos modelos treinados para qualquer configuração usada, validando os resultados obtidos.

Um dos pontos mais relevantes a retirar de todo o trabalho desenvolvido e a ter em conta numa possível implementação futura de um modelo deste tipo, é a forma como está implementado o algoritmo de *Random Forest* a aplicar, uma vez que a divergência da precisão, acuidade e *recall* dos testes de previsão entre os modelos induzidos em WEKA e Python em relação aos obtidos em RapidMiner mostra que a mesma pode influenciar os resultados em larga escala. É fundamental que a implementação do algoritmo seja o mais próxima possível da utilizada pelo *software* WEKA e pela biblioteca *scikit-learn* de Python, que se apresentaram mais eficazes na indução de modelos deste género e para o tipo de dados em questão.

Finalmente, é também significativo destacar que a importância relativa de cada atributo do conjunto de dados referida anteriormente está, quase na totalidade, em consonância com a recolha de informação e estudo feitos na fase de compreensão do negócio, na medida em que, por exemplo, o índice de massa corporal e a idade são fatores muito relacionados com a incidência esta problemática, mas há também outras conclusões que podem ser daqui retiradas. Fatores como o nível de rendimento familiar anual e o nível de educação podem estar relacionados com o nível da qualidade de vida do indivíduo e com o seu estatuto socioeconómico, que são considerados fatores de risco e que, por consequência, influenciam o acesso a cuidados de saúde e impelem diferentes estilos de vida. É ainda importante realçar a enorme relevância da autoavaliação do estado geral de saúde do paciente e das condições de saúde física e mental em que este se encontra para um diagnóstico da ocorrência de Acidente Vascular Cerebral, segundo os modelos de previsão elaborados. Isto mostra não só que o controlo regular do estado global da saúde de um paciente poderá ser de extrema importância na previsão deste tipo de problemática “silenciosa”, mas que, em concordância com estudos prévios, também a saúde mental poderá ser um outro fator de risco significativo, muitas vezes negligenciada nesta matéria [26].

5. REFERÊNCIAS BIBLIOGRÁFICAS

- [1] Campos, L. (2017). *Dê Mais Saúde à Sua Vida! Recomendações da Sociedade Portuguesa de Medicina Interna sobre Prevenção (SPMI)*. Medicina Interna, 24 (3), 176–177.
<https://doi.org/10.24950/rspmi/PP/2017>
- [2] Milovic, B. & Milovic, M. (2012). *Prediction and decision making in health care using data mining*. International Journal of Public Health Science (IJPHS).
- [3] *World health statistics 2020: monitoring health for the SDGs, sustainable development goals*. Geneva: World Health Organization; 2020. Licence: CC BY-NC-SA 3.0 IGO.
- [4] vom Brocke, J., Hevner, A. & Maedche, A. (Eds.). (2020). *Design Science Research. Cases*. Springer International Publishing. <https://doi.org/10.1007/978-3-030-46781-4>.
- [5] Reid, M., Abdool-Karim, Q., Geng, E. & Goosby, E. (2021) *How will COVID-19 transform global health post-pandemic? Defining research and investment opportunities and priorities*. PLoS Med 18(3): e1003564. <https://doi.org/10.1371/journal.pmed.1003564>.
- [6] Wolfe, C. D. A. (2000). *The impact of stroke*. British Medical Bulletin, 56(2), 275–286.
<https://doi.org/10.1258/0007142001903120>.
- [7] Santosh A. Shinde, M. & P. Raja Rajeswari, D. (2018). *Intelligent health risk prediction systems using machine learning: a review*. International Journal of Engineering & Technology, 7 (3).
<https://doi.org/10.14419/ijet.v7i3.12654>.
- [8] Singh, B. S. (2014). *International Journal of Innovation and Scientific Research* ISSN 2351-8014 Vol. 3 No. 2 Jun. 2014, pp. 213-217© 2014 Innovative Space of Scientific Research Journals.
- [9] Schröer, C., Kruse, F. & Gómez, J. M. (2021). *A Systematic Literature Review on Applying CRISP-DM Process Model*. Procedia Computer Science, 181, 526–534.
<https://doi.org/10.1016/j.procs.2021.01.199>.
- [10] Lavrač, N. (1999). *Machine Learning for Data Mining in Medicine* (pp. 47–62).
https://doi.org/10.1007/3-540-48720-4_4.
- [11] Heo, J., Yoon, J. G., Park, H., Kim, Y. D., Nam, H. S. & Heo, J. H. (2019). *Machine Learning-Based Model for Prediction of Outcomes in Acute Stroke*. Stroke, 50 (5).
<https://doi.org/10.1161/STROKEAHA.118.024293>.
- [12] Wirth, R. & Hipp, J. (2000, April). *CRISP-DM: Towards a standard process model for data mining*. In *Proceedings of the 4th international conference on the practical applications of knowledge discovery and data mining* (Vol. 1). London, UK: Springer-Verlag.
- [13] Azevedo, Ana & Santos, M.F. (2008) *KDD, SEMMA and CRISP-DM: A parallel overview*.
- [14] Shafique, Umair & Qaiser, Haseeb (2014) *International Journal of Innovation and Scientific Research*, Vol.12, No. 1, (pp. 217-222).

- [15] Bramer, M. (2020). *Principles of Data Mining*. Springer London. <https://doi.org/10.1007/978-1-4471-7493-6>.
- [16] Furlan, Matheus Batista (2018) *Algoritmos e técnicas para mineração de dados*.
- [17] Hong, S. J. & Weiss, S. M. (2001). *Advances in predictive models for data mining. Pattern Recognition Letters*, 22(1), (pp. 55-61). [https://doi.org/10.1016/S0167-8655\(00\)00099-4](https://doi.org/10.1016/S0167-8655(00)00099-4).
- [18] Frank, E., Hall, M., Trigg, L., Holmes, G. & Witten, I. H. (2004). *Data mining in bioinformatics using Weka. Bioinformatics*, 20(15), (pp. 2479–2481). <https://doi.org/10.1093/bioinformatics/bth261>.
- [19] Magdon-Ismail, M. (2000). *No Free Lunch for Noise Prediction. Neural Computation*, 12 (3), (pp. 547–564). <https://doi.org/10.1162/089976600300015709>.
- [20] Markus Hofmann, & Ralf Klinkenberg. (2014). *RapidMiner Rapid Miner Data Mining Use Cases and Business Analytics Applications*.
- [21] Mr. R. M. Huddar, & Dr. R.V. Kulkarni. (2018). *Role of R and Python in Data Science*. International Multidisciplinary E- Research Journal, (pp. 32-35).
- [22] Markus, H. (2008). Stroke: causes and clinical features. *Medicine*, 36(11), (pp. 586–591). <https://doi.org/10.1016/j.mpmed.2008.08.009>.
- [23] Bharati, S., Rahman, M. A., & Podder, P. (2018). Breast Cancer Prediction Applying Different Classification Algorithm with Comparative Analysis using WEKA. 2018 4th International Conference on Electrical Engineering and Information & Communication Technology (ICEEICT), 581–584. <https://doi.org/10.1109/CEEICT.2018.8628084>.
- [24] Holmes, G., Donkin, A., & Witten, I. H. (n.d.). WEKA: a machine learning workbench. Proceedings of ANZIS '94 - Australian New Zealand Intelligent Information Systems Conference, 357–361. <https://doi.org/10.1109/ANZIS.1994.396988>.
- [25] A. Ramezan, C., A. Warner, T., & E. Maxwell, A. (2019). Evaluation of Sampling and Cross-Validation Tuning Strategies for Regional-Scale Machine Learning Classification. *Remote Sensing*, 11(2), 185. <https://doi.org/10.3390/rs11020185>.
- [26] Zuflacht, J. P., Shao, Y., Kronish, I. M., Edmondson, D., Elkind, M. S. V., Kamel, H., Boehme, A. K., & Willey, J. Z. (2017). Psychiatric Hospitalization Increases Short-Term Risk of Stroke. *Stroke*, 48(7), 1795–1801. <https://doi.org/10.1161/STROKEAHA.116.016371>
- [27] Diabetes Health Indicators Dataset. Kaggle, 2021. <https://www.kaggle.com/datasets/alexteboul/diabetes-health-indicators-dataset>

6. ANEXOS

Tabelas de Apoio

<i>Dataset original</i>	WEKA				
	Precisão Geral	Precisão stroke = 0	Precisão stroke = 1	Recall stroke = 0	Recall stroke = 1
Random Forest (CV 10-F)	95.8%	96%	14.8%	99.9%	0.6%
Random Forest (CV 15-F)	95.8%	96%	16.1%	99.9%	0.7%
Random Forest (Split 66%)	95.8%	95.9%	18.9%	99.9%	0.7%
Random Forest (Split 80%)	95.8%	95.9%	15.2%	99.9%	0.6%
Random Forest (Split 50%)	95.8%	95.9%	19.4%	99.9%	0.6%
Random Forest (Split 30%)	95.8%	95.9%	22.9%	99.9%	0.9%
NaiveBayes (CV 10-F)	94%	97.4%	15.1%	89.8%	43%
NaiveBayes (CV 15-F)	94%	97.4%	15.1%	89.8%	43%
NaiveBayes (Split 66%)	94%	97.4%	15.4%	89.7%	43.5%
NaiveBayes (Split 50%)	94%	97.4%	15.4%	89.7%	43.5%
NaiveBayes (Split 75%)	94.1%	97.4%	15.4%	89.7%	44.2%
BayesNet (CV 10-F)	94.1%	97.4%	16.2%	90.6%	43%
BayesNet (CV 15-F)	94.1%	97.4%	16.2%	90.6%	43%
BayesNet (Split 66%)	94.1%	97.4%	16.6%	90.6%	43.2%
BayesNet (Split 50%)	94.1%	97.4%	16.5%	90.7%	43%
BayesNet (Split 75%)	94.2%	97.5%	16.6%	90.7%	44%

<i>Dataset equilibrado</i>	WEKA				
	Precisão Geral	Precisão stroke = 0	Precisão stroke = 1	Recall stroke = 0	Recall stroke = 1
Random Forest (CV 10-F)	99.3%	99.9%	96.6%	99.3%	99.6%
Random Forest (CV 15-F)	99.4%	99.9%	96.7%	99.3%	99.6%
Random Forest (Split 66%)	98.8%	99.6%	94.6%	98.8%	98.2%
Random Forest (Split 80%)	99.1%	99.8%	95.8%	99.1%	99.2%
Random Forest (Split 50%)	97.7%	98.7%	92.6%	98.4%	93.9%
NaiveBayes (CV 10-F)	81.7%	89.9%	43.4%	84.9%	54.7%
NaiveBayes (CV 15-F)	81.7%	89.9%	43.4%	84.9%	54.7%
NaiveBayes (Split 66%)	81.9%	89.9%	43.4%	85.1%	54.6%
NaiveBayes (Split 50%)	81.8%	89.9%	43.4%	85.0%	54.6%
NaiveBayes (Split 75%)	81.8%	89.8%	43.6%	85.2%	54.3%
BayesNet (CV 10-F)	82.4%	90.5%	43.8%	84.2%	58.4%
BayesNet (CV 15-F)	82.4%	90.5%	43.9%	84.2%	58.4%
BayesNet (Split 66%)	82.6%	90.6%	43.9%	84.3%	58.5%
BayesNet (Split 50%)	82.4%	90.6%	43.9%	84.3%	58.4%
BayesNet (Split 75%)	82.4%	90.5%	44.2%	84.5%	58.0%

Dataset equilibrado	RapidMiner				
	Acerto Geral	Precisão stroke = 0	Precisão stroke = 1	Recall stroke = 0	Recall stroke = 1
Random Forest (CV 10-F)	84.32%	85.75%	63.61%	97.14%	23.67%
Random Forest (CV 15-F)	84.33%	85.76%	63.70%	97.15%	23.69%
Random Forest (Split 66%)	84.33%	85.74%	63.80%	97.17%	23.58%
Random Forest (Split 80%)	84.25%	85.64%	63.52%	97.22%	22.89%
Random Forest (Split 50%)	84.31%	85.76%	63.54%	97.12%	23.73%
NaiveBayes (CV 10-F)	77.60%	91.37%	40.94%	80.46%	64.06%
NaiveBayes (CV 15-F)	77.60%	91.37%	40.94%	80.46%	64.06%
NaiveBayes (Split 66%)	77.87%	91.46%	41.38%	80.73%	64.33%
NaiveBayes (Split 80%)	77.76%	91.34%	41.15%	80.70%	64.83%
NaiveBayes (Split 50%)	77.80%	91.36%	41.22%	80.74%	63.87%
Regressão Logística (CV 10-F)	84.08%	86.33%	59.24%	95.90%	28.15%
Regressão Logística (CV 15-F)	84.08%	86.33%	59.24%	95.90%	28.15%
Regressão Logística (Split 66%)	84.16%	86.39%	59.71%	95.93%	28.50%
Regressão Logística (Split 80%)	84.04%	86.31%	58.97%	95.87%	28.07%
Regressão Logística (Split 50%)	84.08%	86.34%	59.22%	95.89%	28.23%

Dataset equilibrado (Sem Shuffle)	RapidMiner				
	Acerto Geral	Precisão stroke = 0	Precisão stroke = 1	Recall stroke = 0	Recall stroke = 1
Random Forest (CV 10-F)	84.34%	85.75%	63.86%	97.17%	23.63%
Random Forest (CV 15-F)	84.34%	85.76%	63.83%	97.16%	23.68%
Random Forest (Split 66%)	84.23%	85.83%	62.40%	95.90%	24.31%
Random Forest (Split 80%)	84.13%	85.75%	61.72%	96.87%	23.84%
Random Forest (Split 50%)	84.32%	85.71%	63.90%	97.21%	23.24%

WEKA

Weka Explorer

Preprocess | **Classify** | Cluster | Associate | Select attributes | Visualize

Classifier

Choose: RandomForest -P 100 -I 100 -num-slots 1 -K 0 -M 1.0 -V 0.001 -S 1

Test options

- ☐ Use training set
- ☐ Supplied test set
- ☒ Cross-validation Folds: 10
- ☐ Percentage split % 66

More options...

(Nom) stroke

Start Stop

Result list (right-click for options)

18:38:28 - trees.RandomForest

Classifier output

```

Time taken to build model: 223.93 seconds

=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances      292809          99.3085 %
Incorrectly Classified Instances    2039           0.6915 %
Kappa statistic                    0.9763
Mean absolute error                 0.0582
Root mean squared error             0.1147
Relative absolute error             20.2053 %
Root relative squared error         30.2188 %
Total Number of Instances          294848

=== Detailed Accuracy By Class ===

          TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Cla
0,993    0,004    0,999    0,993    0,996    0,976    0,999    1,000    0
0,996    0,007    0,966    0,996    0,980    0,976    0,999    0,994    1
Weighted Avg.   0,993    0,005    0,993    0,993    0,993    0,976    0,999    0,999

=== Confusion Matrix ===
      a      b  <-- classified as
241576  1812 |      a = 0
 227    51233 |      b = 1

```

Status

OK Log x0

Weka Explorer

Preprocess Classify Cluster Associate Select attributes Visualize

Classifier

Choose RandomForest -P 100 -I 100 -num-slots 1 -K 0 -M 1.0 -V 0.001 -S 1

Test options

☐ Use training set
☐ Supplied test set Set...
☒ Cross-validation Folds 15
☐ Percentage split % 66
 More options...

(Nom) stroke

Start Stop

Result list (right-click for options)

18:38:28 - trees.RandomForest
19:34:14 - trees.RandomForest

Classifier output

Time taken to build model: 235.53 seconds

=== Stratified cross-validation ===

=== Summary ===

Correctly Classified Instances	292902	99.34 %
Incorrectly Classified Instances	1946	0.66 %
Kappa statistic	0.9774	
Mean absolute error	0.056	
Root mean squared error	0.1121	
Relative absolute error	19.4489 %	
Root relative squared error	29.5416 %	
Total Number of Instances	294848	

=== Detailed Accuracy By Class ===

	TP Rate	FP Rate	Precision	Recall	F-Measure	MCC	ROC Area	PRC Area	Class
0,993	0,004	0,999	0,993	0,996	0,978	0,999	1,000	0	
0,996	0,007	0,967	0,996	0,981	0,978	0,999	0,994	1	
Weighted Avg.	0,993	0,004	0,994	0,993	0,993	0,978	0,999	0,999	

=== Confusion Matrix ===

a	b	<-- classified as	
241638	1750	a = 0	
196	51264	b = 1	

Status

OK Log x0

Weka Explorer

Preprocess Classify Cluster Associate Select attributes Visualize

Classifier

Choose RandomForest -P 100 -I 100 -num-slots 1 -K 0 -M 1.0 -V 0.001 -S 1

Test options

☐ Use training set
☐ Supplied test set Set...
☐ Cross-validation Folds 15
☒ Percentage split % 66
 More options...

(Nom) stroke

Start Stop

Result list (right-click for options)

18:38:28 - trees.RandomForest
19:34:14 - trees.RandomForest
20:43:15 - trees.RandomForest

Classifier output

=== Evaluation on test split ===

Time taken to test model on test split: 11.07 seconds

=== Summary ===

Correctly Classified Instances	98962	98.7172 %
Incorrectly Classified Instances	1286	1.2828 %
Kappa statistic	0.9559	
Mean absolute error	0.0798	
Root mean squared error	0.1436	
Relative absolute error	27.7371 %	
Root relative squared error	37.9389 %	
Total Number of Instances	100248	

=== Detailed Accuracy By Class ===

	TP Rate	FP Rate	Precision	Recall	F-Measure	MCC	ROC Area	PRC Area	Class
0,988	0,018	0,996	0,988	0,992	0,956	0,996	0,999	0	
0,982	0,012	0,946	0,982	0,964	0,956	0,996	0,986	1	
Weighted Avg.	0,987	0,017	0,988	0,987	0,987	0,956	0,996	0,997	

=== Confusion Matrix ===

a	b	<-- classified as	
81896	976	a = 0	
310	17066	b = 1	

Status

OK Log x0

Weka Explorer

Preprocess Classify Cluster Associate Select attributes Visualize

Classifier

Choose RandomForest -P 100 -I 100 -num-slots 1 -K 0 -M 1.0 -V 0.001 -S 1

Test options

☐ Use training set
☐ Supplied test set Set...
☐ Cross-validation Folds 10
☒ Percentage split % 80
 More options...

(Nom) stroke

Start Stop

Result list (right-click for options)

21:20:39 - trees.RandomForest

Status

OK Log x0

Classifier output

```

=== Evaluation on test split ===

Time taken to test model on test split: 6.44 seconds

=== Summary ===

Correctly Classified Instances      58442           99.1046 %
Incorrectly Classified Instances    528             0.8954 %
Kappa statistic                    0.9692
Mean absolute error                0.0658
Root mean squared error            0.1251
Relative absolute error             22.8779 %
Root relative squared error        33.0298 %
Total Number of Instances          58970

=== Detailed Accuracy By Class ===

          TP Rate  FP Rate  Precision  Recall  F-Measure  MCC   ROC Area  PRC Area  Cla
0,991    0,008    0,998    0,991    0,995    0,969    0,998    1,000    0
0,992    0,009    0,958    0,992    0,975    0,969    0,998    0,992    1
Weighted Avg.    0,991    0,008    0,991    0,991    0,991    0,969    0,998    0,998

=== Confusion Matrix ===

      a    b  <-- classified as
48286  446 |      a = 0
 82 10156 |      b = 1
  
```

Weka Explorer

Preprocess Classify Cluster Associate Select attributes Visualize

Classifier

Choose RandomForest -P 100 -I 100 -num-slots 1 -K 0 -M 1.0 -V 0.001 -S 1

Test options

☐ Use training set
☐ Supplied test set Set...
☐ Cross-validation Folds 10
☒ Percentage split % 50
 More options...

(Nom) stroke

Start Stop

Result list (right-click for options)

17:11:55 - trees.RandomForest
 17:20:50 - trees.RandomForest
 17:30:54 - trees.RandomForest

Status

OK Log x0

Classifier output

```

=== Evaluation on test split ===

Time taken to test model on test split: 16.48 seconds

=== Summary ===

Correctly Classified Instances      143944           97.6395 %
Incorrectly Classified Instances    3480             2.3605 %
Kappa statistic                    0.9183
Mean absolute error                0.1022
Root mean squared error            0.1756
Relative absolute error             35.467 %
Root relative squared error        46.3327 %
Total Number of Instances          147424

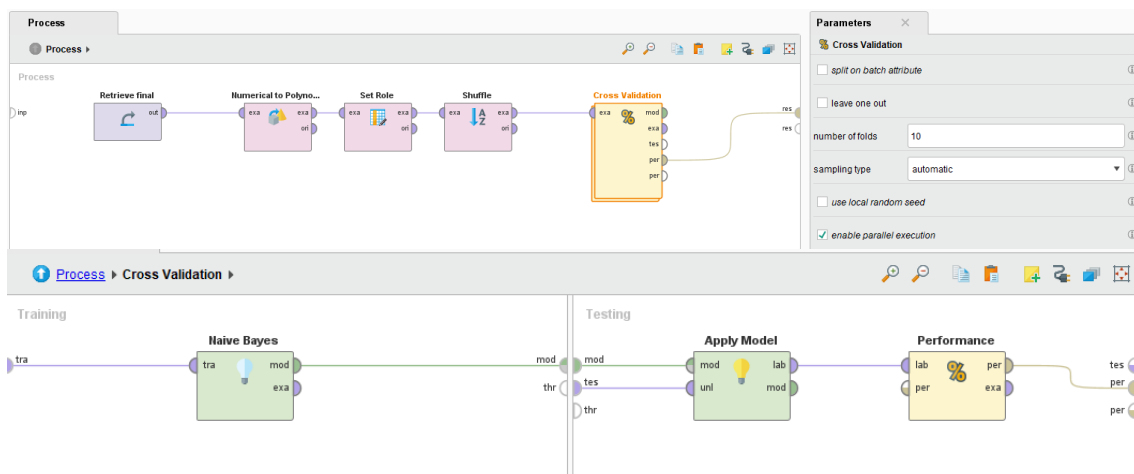
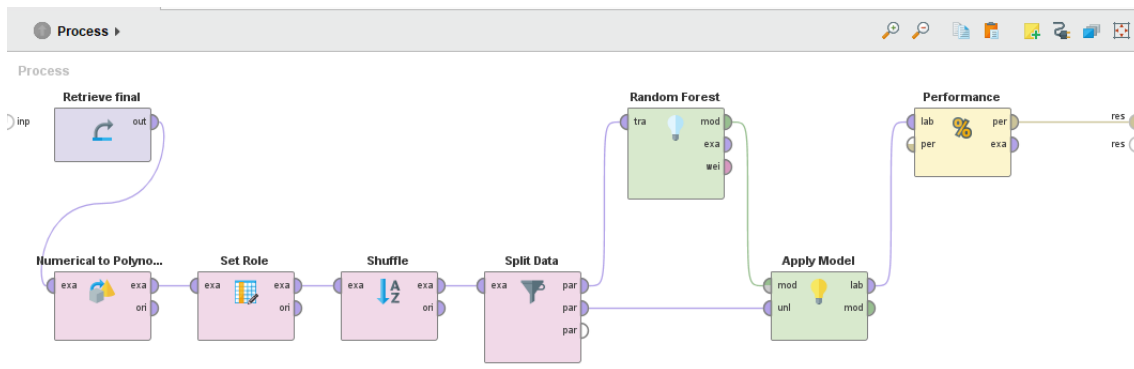
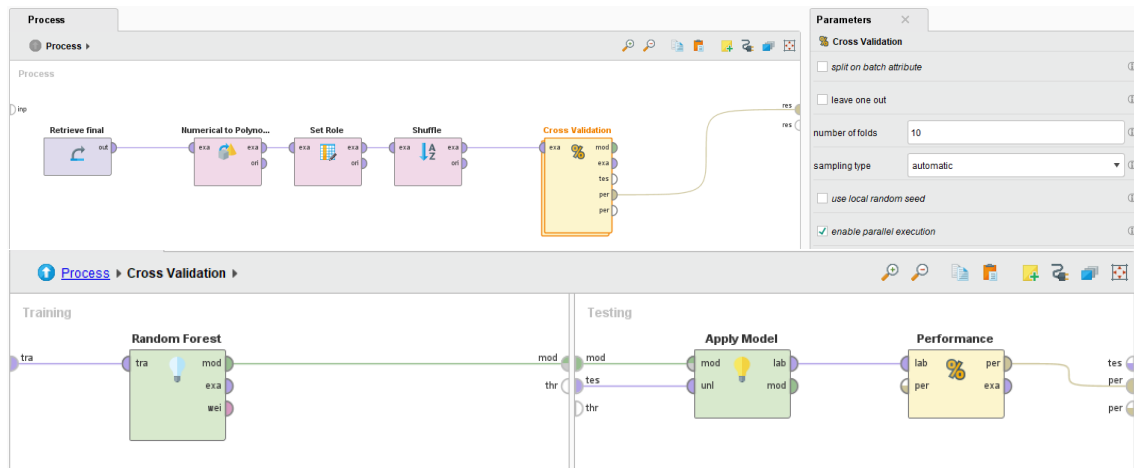
=== Detailed Accuracy By Class ===

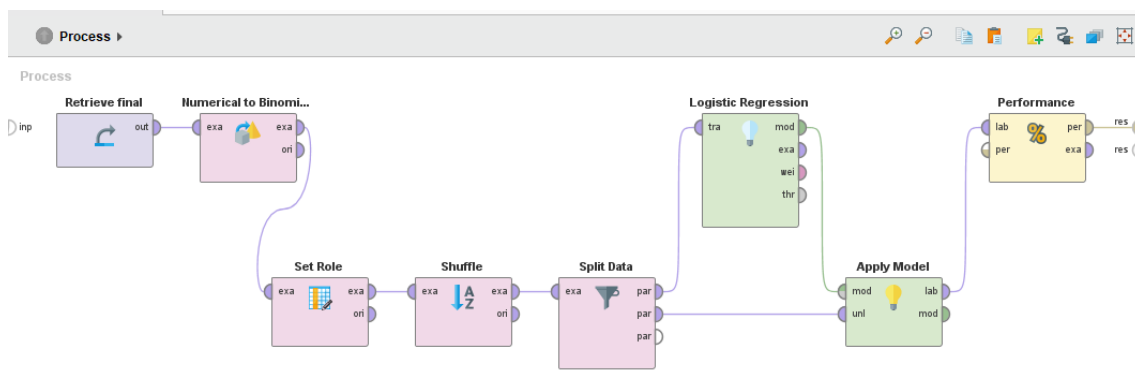
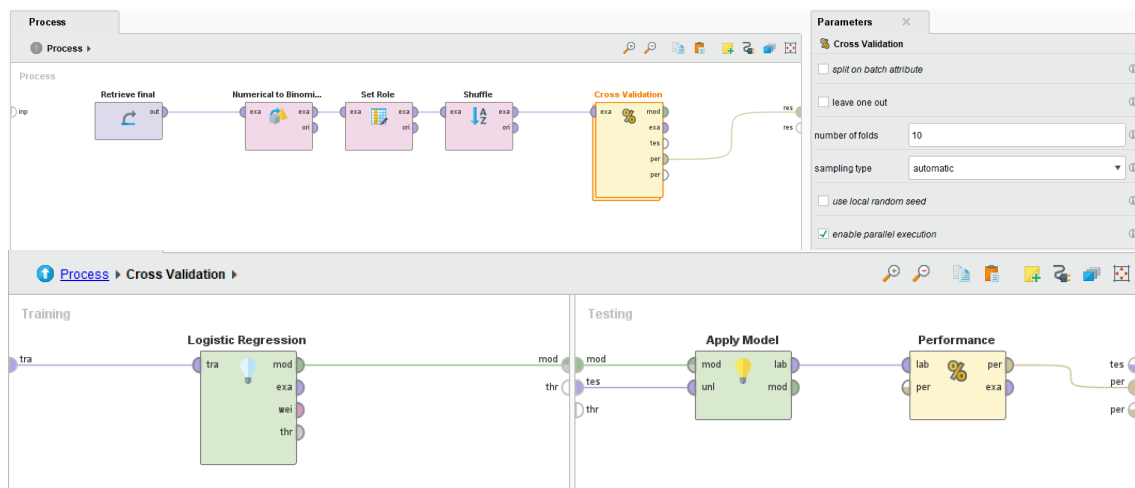
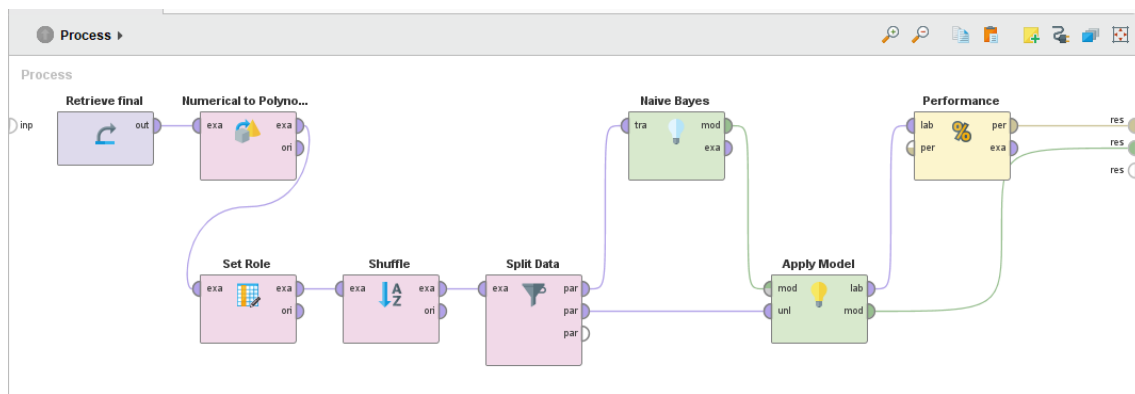
          TP Rate  FP Rate  Precision  Recall  F-Measure  MCC   ROC Area  PRC Area  Cla
0,984    0,061    0,987    0,984    0,986    0,918    0,987    0,996    0
0,939    0,016    0,926    0,939    0,933    0,918    0,987    0,967    1
Weighted Avg.    0,976    0,053    0,977    0,976    0,976    0,918    0,987    0,991

=== Confusion Matrix ===

      a    b  <-- classified as
119862  1919 |      a = 0
 1561  24082 |      b = 1
  
```

RapidMiner





Random Forest

accuracy: 84.32% +/- 0.11% (micro average: 84.32%)

	true 0	true 1	class precision
pred. 0	236420	39278	85.75%
pred. 1	6968	12182	63.61%
class recall	97.14%	23.67%	

accuracy: 84.33% +/- 0.15% (micro average: 84.33%)

	true 0	true 1	class precision
pred. 0	236440	39268	85.76%
pred. 1	6948	12192	63.70%
class recall	97.15%	23.69%	

accuracy: 84.33%

	true 0	true 1	class precision
pred. 0	80411	13371	85.74%
pred. 1	2341	4125	63.80%
class recall	97.17%	23.58%	

accuracy: 84.25%

	true 0	true 1	class precision
pred. 0	47325	7936	85.64%
pred. 1	1353	2356	63.52%
class recall	97.22%	22.89%	

accuracy: 84.31%

	true 0	true 1	class precision
pred. 0	118189	19623	85.76%
pred. 1	3505	6107	63.54%
class recall	97.12%	23.73%	

NaiveBayes

accuracy: 77.60% +/- 0.23% (micro average: 77.60%)

	true 0	true 1	class precision
pred. 0	195841	18496	91.37%
pred. 1	47547	32964	40.94%
class recall	80.46%	64.06%	

accuracy: 77.60% +/- 0.26% (micro average: 77.60%)

	true 0	true 1	class precision
pred. 0	195837	18496	91.37%
pred. 1	47551	32964	40.94%
class recall	80.46%	64.06%	

accuracy: 77.87%

	true 0	true 1	class precision
pred. 0	66808	6240	91.46%
pred. 1	15944	11256	41.38%
class recall	80.73%	64.33%	

accuracy: 77.76%

	true 0	true 1	class precision
pred. 0	39284	3723	91.34%
pred. 1	9394	6569	41.15%
class recall	80.70%	63.83%	

accuracy: 77.80%

	true 0	true 1	class precision
pred. 0	98259	9297	91.36%
pred. 1	23435	16433	41.22%
class recall	80.74%	63.87%	

Regressão Logística

accuracy: 84.08% +/- 0.15% (micro average: 84.08%)

	true false	true true	class precision
pred. false	233420	36972	86.33%
pred. true	9968	14488	59.24%
class recall	95.90%	28.15%	

accuracy: 84.08% +/- 0.17% (micro average: 84.08%)

	true false	true true	class precision
pred. false	233420	36973	86.33%
pred. true	9968	14487	59.24%
class recall	95.90%	28.15%	

accuracy: 84.16%

	true false	true true	class precision
pred. false	79387	12510	86.39%
pred. true	3365	4986	59.71%
class recall	95.93%	28.50%	

accuracy: 84.04%

	true false	true true	class precision
pred. false	46668	7403	86.31%
pred. true	2010	2889	58.97%
class recall	95.87%	28.07%	

accuracy: 84.08%

	true false	true true	class precision
pred. false	116692	18466	86.34%
pred. true	5002	7264	59.22%
class recall	95.89%	28.23%	

Random Forest (sem Shuffle)

accuracy: 84.34% +/- 0.17% (micro average: 84.34%)

	true 0	true 1	class precision
pred. 0	236506	39302	85.75%
pred. 1	6882	12158	63.86%
class recall	97.17%	23.63%	

accuracy: 84.34% +/- 0.23% (micro average: 84.34%)

	true 0	true 1	class precision
pred. 0	236483	39276	85.76%
pred. 1	6905	12184	63.83%
class recall	97.16%	23.68%	

accuracy: 84.23%

	true 0	true 1	class precision
pred. 0	80189	13243	85.83%
pred. 1	2563	4253	62.40%
class recall	96.90%	24.31%	

accuracy: 84.13%

	true 0	true 1	class precision
pred. 0	47156	7838	85.75%
pred. 1	1522	2454	61.72%
class recall	96.87%	23.84%	

accuracy: 84.32%

	true 0	true 1	class precision
pred. 0	118301	19724	85.71%
pred. 1	3393	6006	63.90%
class recall	97.21%	23.34%	