Mental Health in Tech Survey

Guilherme Coelho ,NºEstudante: 2013136209
Samaritana Silva, NºEstudante: 2013151207
Tadeu Ferreira, NºEstudante: 2013151074

Mestrado Integrado em Engenharia Biomédica

Base de Dados e Análise de Informação

2017/2018

Índice

Lista de Tabelas

Lista de Figuras

1 Introdução

Os problemas mentais afetam cerca de 700 milhões de pessoas no mundo, representando cerca de 13% do total de doenças [1]. A depressão por exemplo é a segunda causa de invalidez, estando apenas atrás das dores nas costas. As principais doenças mentais são a depressão e a ansiedade e não a esquizofrenia.

Há evidências que mostram que pessoas com problemas mentais são recusadas num emprego devido à sua condição mental [2] ou não procuram emprego porque sabem que podem sofrer discriminação [3].

A divulgação de um problema mental no ambiente de trabalho pode conduzir a comportamentos discriminatórios por parte dos supervisores e dos colegas, como por exemplo exclusão social ou impedir que essas pessoas progridam na carreira [4]. Uma estrutura para a compreensão destes comportamentos conceitualiza o estigma como o conjunto de três problemas [5]:

- Connhecimento (ignorância ou má informação)
- Atitudes (preconceito)
- Comportamento (discriminação)

Num estudo ralizado por Manning e White [6] os fatores quase sempre considerados na contratação de uma pessoa são o padrão do trabalho anterior (89%), descrição do cargo (87%), se recebeu tratamento (69%), tempo que esteve doente ano anterior (68%) e diagnóstico (64%). Fenton et all [7] também concluiram que o registo do trabalho (78%), registo de sáude (69%), diagnóstico (36%), deteção nos termos da Lei de Saúde Mental (36%) e opnião médica (7%) são fatores importantes na contratação de uma pessoa. Krupa [8] destacou quatro pressupostos subjacentes ao estigma no local de trabalho:

1. pessoas com problemas de saúde mental não têm as competências necessárias para atender às exigências do trabalho;
2. pessoas com problemas de saúde mental são perigosas ou imprevisíveis;
3. trabalhar não é saudável para pessoas com problemas de saúde mental;
4. dar emprego a pessoas com problemas de sáude mental é um ato de caridade.

Esses pressupostos variam em intensidade com base numa série de fatores organizacionais, individuais e sociais.

É importante destacar a importância de um bom ambiente de trabalho para a melhoria da integração económica e social de pessoas com problemas mentais [9].

2 Estado de Arte

2.1 Seleção de Features

A seleção de características é um importante passo de pré-processamento em muitas tarefas de *machine learning*, é importante para acelerar a aprendizagem e melhorar a qualidade do conjunto de dados [10]. Reduzir a dimensionalidade dos dados reduz o tamanho do espaço da hipótese e permite que os algoritmos funcionem de forma mais rápida e eficaz. O objetivo é reduzir o número de características consideradas para o estudo, eliminando aquelas que são mais redundantes, irrelevantes ou ruidosas. Do ponto de vista de classificação, existem inúmeros benefícios associados com a seleção de características: (i) tempo de armazenamento, (ii) a redução da complexidade do classificador, (iii) aumento da precisão, (iv) redução dos tempos de teste, e (v) uma melhor compreensão e visualização de dados.

Os métodos de filtragem dependem das características gerais dos dados para selecionar um subconjunto de características sem envolver qualquer algoritmo de aprendizagem.

Recorreu-se ao software WEka [11] para analisar quais as características mais importantes para a nossa classificação. Mais precisamente utilizou-se o *InfoGainAttributeEval* em combinação com o *Ranker* e o *CorrelationAtrributeEval*.

- *InfoGainAttributeEval*, avalia o valor de uma característica medindo a informação do ganho em relação à classe. Este método baseia-se na seguinte equação: $InfoGain(Class, Attribute) = H(Class) - H(Class|Attribute)$, que basicamente diz como cada característica contribui para diminuir a entropia global [12, 13]. O *Ranker* classifica as características por ordem decresente de relevância de acordo com o ganho de informação.

- *CorrelationAttributeEval*, avalia a correlação de *Pearson* entre a característica e a classe, ou seja, devolve as características que estão mais correlacionadas com o classificação. Uma correlação geral para uma característica nominal é alcançada através de uma média ponderada [13].

2.2 Classificação

A classificação é uma técnica de mineração de dados usada para prever a associação de um grupo de dados. É uma forma de análise de dados que pode ser usada para extrair modelos, descrevendo classes de dados importantes ou para prever futuras tendências de dados [14, 15]. A classificação é tem como objetivo generalizar a estrutura conhecida para aplicar a novos dados.

Recorreu-se aos seguintes classificadores para analisar os nossos dados:

1. O SVM são métodos de classificação onde se projeta um conjunto de dados de treino que representam duas classes diferentes num espaço de alta dimensão por meio de um função kernel. Os dados não-lineares sofrem uma transformação algébrica de modo a que uma linha reta possa ser gerada (hiperplano de discriminação) para separar as classes,maximizando assim

a sua separação. Os pontos na margem dessa zona de separação são denominados de vetores de suporte. Os dados a testar serão projetados para o espaço de alta-dimensionalidade criado no treino, sendo então classificados com base na sua localização no que diz respeito ao hiperplano de discriminação. Uma kernel muito conhecida é a radial basis function (RBF), que tem como gama (γ) o parâmetro de liberdade da gaussianidade ou distribuição normal do RBF [16]. De acordo com J S Yu et al. [17], o SVM é menos sensível a outliers e os modelos oferecem alta sensibilidade e especificidade na previsão de sujeitos com transtorno depressivo maior

2. Utilizou-se também um classificador que tem em conta a regra do vizinho mais próximo, a *nearest neighbour rule*: *K-Nearest Neighbours (KNN)*. O objetivo do KNN é a utilização de uma base de dados na qual os novos pontos de dados são rotulados consoante a posição dos pontos de treino. A maneira, segundo a qual o algoritmo decide quais dos pontos do conjunto de treino serão utilizados para classificar um ponto de teste, é através da distância. São escolhidos os k pontos mais próximos da nova observação, tomando como posterior rótulo a classe mais comum entre estes [16].

3. Random Forest é um exemplo de um método de *ensemble*, o que significa que ele depende da agregação dos resultados de um conjunto de estimadores mais simples. É um método não linear que se baseia em árvores de decisão, e permite fazer correlações que não seriam possíveis com outros classificadores. O resultado surpreendente com tais métodos de *ensamble* é que a soma pode ser maior do que as partes: isto é, o voto maioritário entre uma série de estimadores pode acabar sendo melhor do que qualquer um dos estimadores individuais fazendo a votação. A regressão Random Forest tem apresenta um grande potencial como ferramenta exploratória, dada a sua capacidade de lidar com dados de alta dimensão para descobrir associações com stress e transtorno de déficit de atenção e hiperatividade e produzir medidas de importância. No entanto, a interpretabilidade dos seus resultados pode tornar-se difícil [18].

3 Dataset

Este conjunto de dados é de uma pesquisa de 2014 que mede as atitudes em relação à saúde mental e à frequência de transtornos em saúde mental no local de trabalho tecnológico. Este inquérito foi feito a 1260 pessoas de diversos países, sendo que os 10 países que mais participaram na pesquisa podem ser vistos na *Figura*1

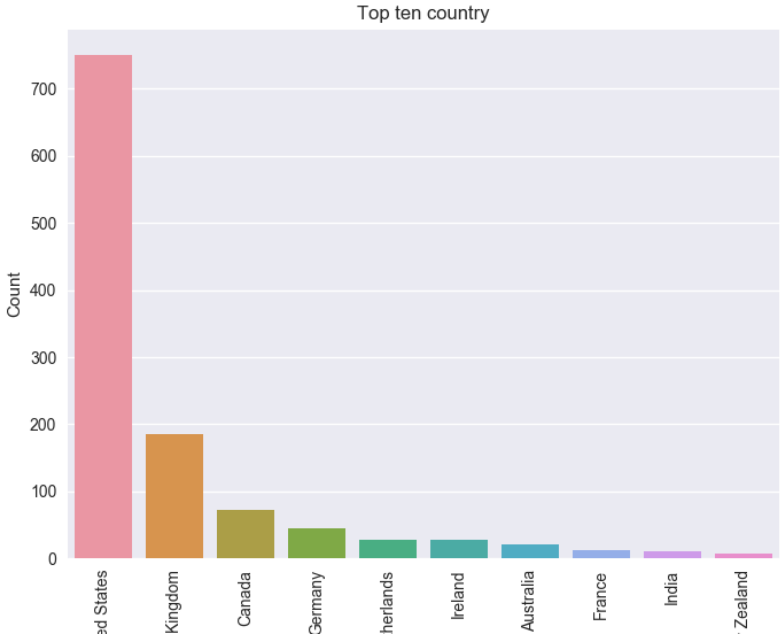

Figura 1: Top 10 dos países participantes na pesquisa

O conjunto de dados contém os seguintes dados:

- *Timestamp*
- *Age*
- *Gender*
- *Country*
- *state*: Se a pessoa mora nos Estados Unidos, em que estado ou território vive.
- *self_ employed*: Se a pessoa trabalha por conta própria.
- *family_ history*: Se pessoa tem algum histórico familiar de problemas mentais.

- *treatment*: Se a pessoa já precisou de algum tratamento para problemas de saúde mental.
- *work_ interfere*: Se a pessoa tem um problema de saúde mental, se acha que isso interfere com seu trabalho.
- *no_ employees*: Quantos empregados a empresa ou organização onde a pessoa trabalha tem.
- *remote_ work*: Se a pessoa trabalha fora do escritório pelo menos 50% do sue tempo.
- *tech_ company*: Se a empresa onde a pessoa trabalha é tecnológica.
- *benefits*: Se o patrão fornece benefícios para problemas da saúde mental.
- *care_ options*: Se a pessoa conhece as opções para cuidados de saúde mental que o seu patrão fornece.
- *wellness_ program*: Se o patrão já discutiu a saúde mental como parte de um programa de bem-estar dos empregados.
- *seek_ help*: Se o patrão fornece recursos para saber mais sobre problemas de saúde mental e como procurar ajuda.
- *anonymity*: Se o anonimato da pessoa é protegido caso ela opte por tirar proveito dos recursos de tratamento de saúde mental ou abuso de substâncias.
- *leave*: Quão fácil é para a pessoa tirar uma licença médica devido a uma condição de saúde mental.
- *mental_ health_ consequence*: Se a pessoa acha que discutir um problema de saúde mental com o patrão tem consequências negativas.
- *phys_ health_ consequence*: Se a pessoa acha que discutir um problema de saúde física com o patrão tem consequências negativas.
- *coworkers*: Se a pessoa estaria disposta a discutir um problema de saúde mental com os seus colegas de trabalho.
- *supervisor*: Se a pessoa estaria disposta a discutir um problema de saúde mental com os seus supervisores.
- *mental_ health_ interview*: Se a pessoa estaria disposta a falar de um problema de saúde mental numa entrevista de emprego.
- *phys_ health_ interview*: Se a pessoa estaria disposta a falar de um problema de saúde física numa entrevista de emprego.
- *mental_ vs_ physical*: Se a pessoa acha que o seu patrão leva a saúde mental tão a sério quanto a saúde física.
- *obs_ consequence*: Se a pessoa já ouviu ou observou consequências negativas para colegas de trabalho com problemas de saúde mental no seu local de trabalho.
- *comments*

Este Data Set é original do "Open Sourcing Mentla Illness"OSMI [19], e foi o maior inquérito sobre Saúde Mental no ambiente tecnológico, feito até à data. Além de ser reconhecido pela equipa do *Kaggle*, os dados brutos e resultados do inquérito podem ser encontrados em [20].

4 Pré-Processamento

4.1 Conhecimento dos Dados

Tal como já referido, o ponto de partida são os dados em bruto, ou seja as respostas diretas ao questionário. Foi então necessário inicialmente compreender os e rever todos os dados. Rapidamente se chegou à conclusão que havia respostas incoerentes, nulas, que os dados não estavam normalizados, entre outros problemas que não nos permitiriam avançar no projeto. Havia então a necessidade de limpar os dados.

```
RangeIndex: 1259 entries, 0 to 1258
Data columns (total 27 columns):
Timestamp                    1259 non-null object
Age                          1259 non-null int64
Gender                       1259 non-null object
Country                      1259 non-null object
state                        744 non-null object
self_employed                1241 non-null object
family_history               1259 non-null object
treatment                    1259 non-null object
work_interfere               995 non-null object
no_employees                 1259 non-null object
remote_work                  1259 non-null object
tech_company                 1259 non-null object
benefits                     1259 non-null object
care_options                 1259 non-null object
wellness_program             1259 non-null object
seek_help                    1259 non-null object
anonymity                    1259 non-null object
leave                        1259 non-null object
mental_health_consequence    1259 non-null object
phys_health_consequence      1259 non-null object
coworkers                    1259 non-null object
supervisor                   1259 non-null object
mental_health_interview      1259 non-null object
phys_health_interview        1259 non-null object
mental_vs_physical           1259 non-null object
obs_consequence              1259 non-null object
comments                     164 non-null object
dtypes: int64(1), object(26)
memory usage: 265.6+ KB
None
```

Figura 2: Vista geral dos dados

4.2 Limpeza dos Dados

Antes de entrar em maior detalhe na limpeza de dados, a abordagem que consideramos mais correta foi a de perceber que características do data set

poderiam ser eliminadas, sem comprometer a performance do algoritmo a construir. Ora, as features "Timestamp", "state"e "comments", foram consideradas como irrelevantes para o problema a ser abordado, e portanto foram eliminadas.

De seguida, devido à falta de normalização das respostas, foi necessário analisar individualmente cada característica, de forma a verificar se os intervalos de respotas se enquadravam nos pretendidos. "Sex", "Age"e "self-employed"foram as features que tiveram de sofrer alterações.

No caso da característica "self-employed"o tratamento foi simples, na medida em que foi apenas necessário eliminar todos as resposta nulas. Enquanto, que para determinados atributos a resposta nula faça sentido, para este a gama de resposta só poderia ser sim ou não.

Em relação à idade, "Age", além da presença de valores negativos, existiam também valores excessivamente grande. Esta variável foi então balizadas entre 0 e 100, e as instâncias que se encontravam fora, forma apagadas.

Por fim, a variável "Sex"foi normalizada em apenas 3 respostas possíveis, "Male", "Female"e "Trans".

Desta forma, o nosso Data Set ficou reduzido a 1233 instâncias para 24 features. Este novo grupo de dados foi guardado, e foi sobre este que o restante trabalho incidiu.

4.3 Visualização dos Dados

Uma vez com o Data Set normalizado e pré-processado, partímos para a visualização do nosso conjunto de dados. Para tal, foi usada a ferramente WEKA, introduzida nas aulas da cadeira. Foi então necessário adaptar o formato dos dados, ao formato aceite pela ferramenta usada.

Usando o WEKA e definindo a nossa variável Classe, foi possível verificar a forma como os restantes atributos se comportam em relação à classe.

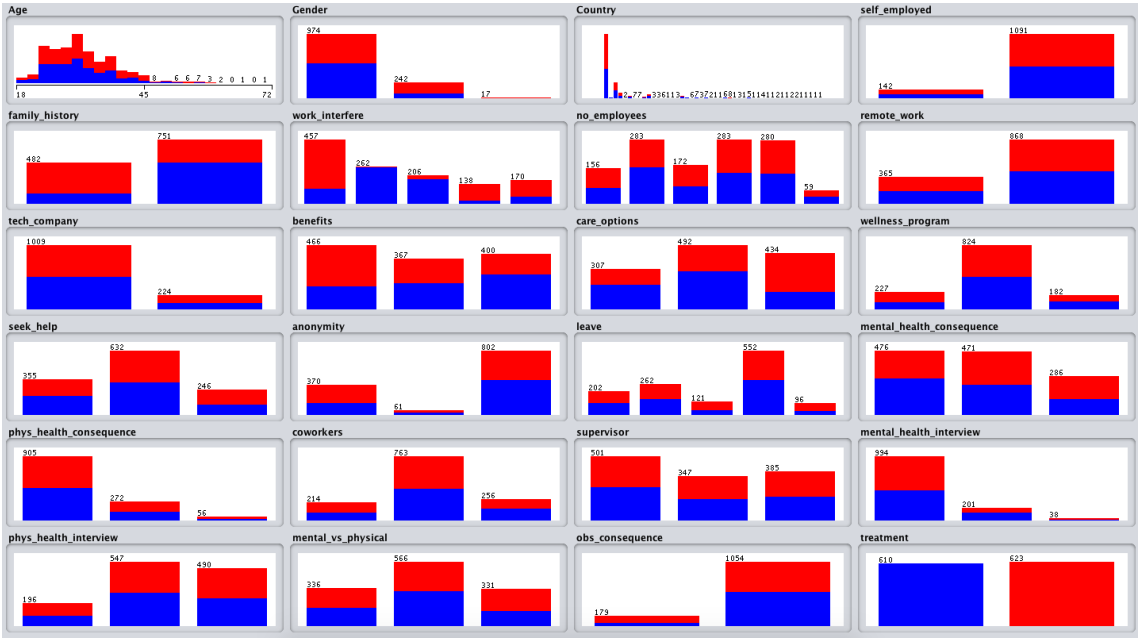

Figura 3: Visualização geral do Data Set

Com este primeiro passo é possível perceber logo quais são as features que poderão ter uma papel importante no momento da classificação. É o caso das variáveis "family_history", "work_interfere"e "care_options". Faz então sentido, focar a visualização nestes atributos.

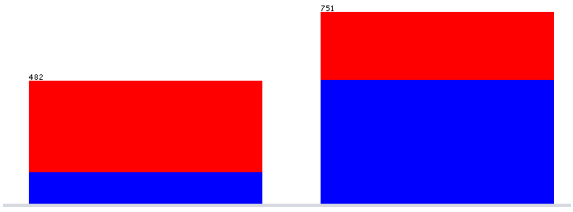

Figura 4: Visualização do atributo family_history

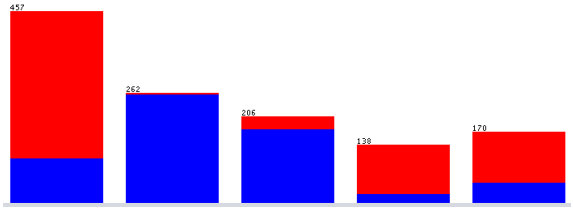

Figura 5: Visualização do atributo work_interfere

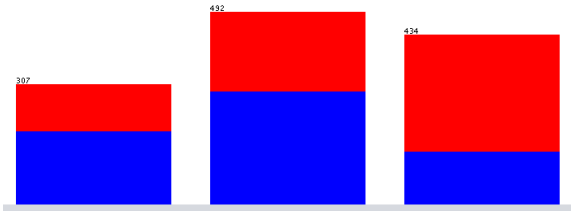

Figura 6: Visualização do atributo care_options

Como se pode constatar, estes atributos são bons discriminantes, na medida em que a distribuição da Classe do seu espectro de valores é heterogénea. Por exemplo, no atributo de "family_history", para a resposta negativa a predominância de classe negativa é obviamente superior, e vice-versa. Para os outros dois atributos, o comportamente é semelhante.

4.4 Relações entre Features

Nesta secção apresentamos algumas relações entre os atributos do conjunto de dados. Não explorámos relações dos atributos com a nossa classe alvo, pois esse trabalho já é realizado na secção "Seleção de Features". Todas as relações exploradas são as que mais nos suscitaram curiosidade e as que mais se relacionavam com o conteúdo apresentando na "Introdução".

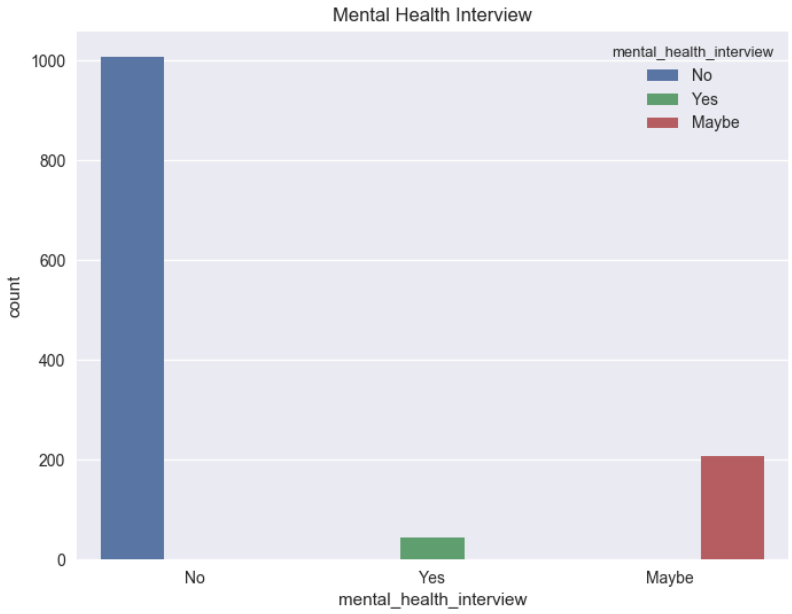

Figura 7: Doenças mentais numa entrevista

Como se pode concluir pela *Figura* 12, a maior parte dos inquiridos não diria que tinha um problema/ doença mental numa entrevista de emprego. O que está de acordo com Corrigan et al[2], pois as pessoas com doença mental são privadas das mesmas oportunidades que uma pessoa "normal", resultando numa qualidade de vida inferior. Isto acontece pois por um lado, estas pessoas lutam com os sintomas e deficiências que resultam da doença. E por outro lado, são desafiadas pelos estereótipos e preconceitos que resultam da má informação sobre doenças mentais.

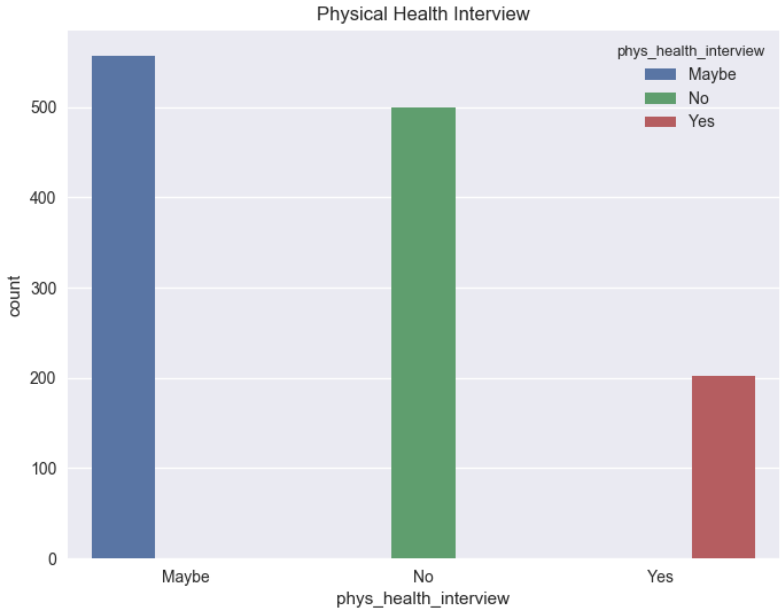

Figura 8: Problemas físicos numa entrevista

Pela análise da *Figura* 8 pode-se concluir que a maioria dos inquiridos talvez diria que tinha um problema físico numa entrevista de emprego. Isto prende-se com o facto de um problema físico não ter um impacto tão negativo como um problema mental. Atualmente ainda existe pouca informação sobre os problemas mentais, e de certo modo essas pessoas são postas de lado [8].

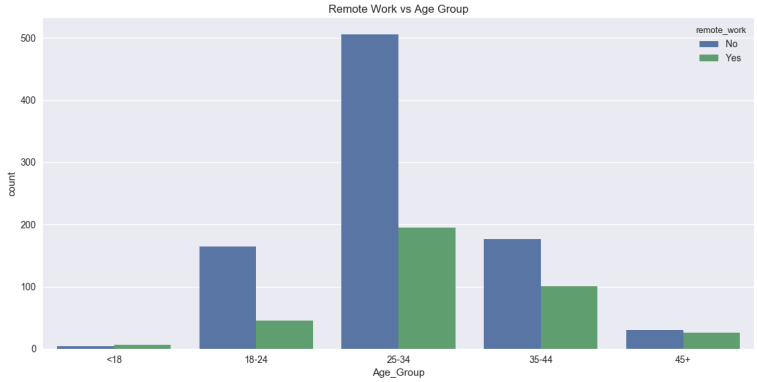

Figura 9: Trabalho remoto vs Grupo de Idades

Na *Figura* 9 está representada a relação entre a idade e trabalhar remotamente, ou seja, trabalhar fora do escritório. Como se pode ver a maior parte dos inquiridos trabalha num escritório. O grupo de pessoas com a idade compreendida entre os 25 e os 34 anos é o grupo que mais trabalha remotamente.

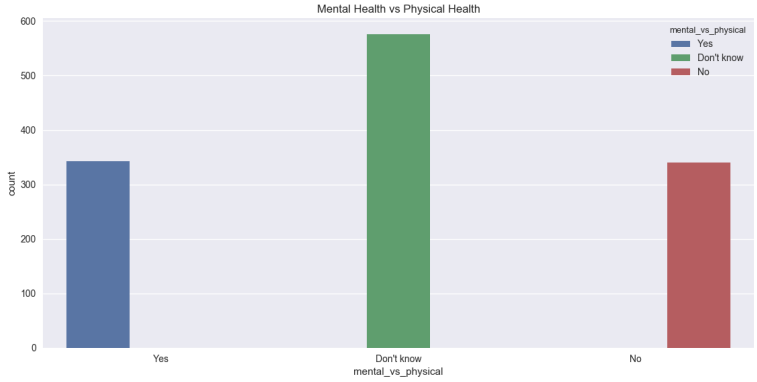

Figura 10: Saúde Mental vs Saúde Física

Na *Figura* 10 está representada a opinião das pessoas sobre se o patrão delas leva tão a sério a saúde mental quanto a saúde física. Como se pode concluir a maioria disse que não sabia, o que é grave pois a saúde mental tem que ter a mesma importância que a saúde física. A sáude física e a saúde mental estão relacionadas, para um indíviduo ser totalmente saudável tem que estar bem aos dois níveis. Pessoas que têm uma rotina muito stressante e que lidam com transtornos psicológicos ou problemas emocionais acabam por ter problemas físicos, como dores na cabeça e no corpo, cansaço, entre outros. Por outro lado, pessoas que têm problemas na saúde física, como dores frequentes ou outros

porblemas, acabam por ter a sua saúde mental prejudicada, com problemas como depressão, ansiedade, etc..[21].

5 Seleção de Features

Apesar de, através da visualização dos atributos ser possível determinar quais os mais relevantes, há a necessidade de o comprovar com recurso a algoritmos próprios para o efeito. É então necessário aplicar métodos de Seleção/Extração de Features. Contudo, e sendo um problem muito nominal, a perda de informação e significado dos atributos não é desejada.

Dessa forma, limitámos esta secção à Seleção de Features. Ou seja, partindo das 24 características incluídas no Data Set, iremos selecionar apenas as mais relevantes para o processo de classificação.

Recorrendo à opção "Select Attributes"do Weka foi possível executar este processo. Os métodos usados forma InfoGainAttributeEval e CorrelationAttributeEval. Ambos os métodos desempenham funções de seleção de features, tal como já abordado anteriormente. Os resultados obtidos, para cada método, foram os seguintes:

```
Search Method:
        Attribute ranking.

Attribute Evaluator (supervised, Class (nominal): 24 treatment):
        Information Gain Ranking Filter

Ranked attributes:
 0.397725    6 work_interfere
 0.10595     5 family_history
 0.054045   11 care_options
 0.049512    3 Country
 0.036411   10 benefits
 0.026255    2 Gender
 0.016574   23 obs_consequence
 0.015829   15 leave
 0.014591   14 anonymity
 0.010575   16 mental_health_consequence
 0.009235   22 mental_vs_physical
 0.008329   20 mental_health_interview
 0.005861   12 wellness_program
 0.005766   13 seek_help
 0.005199    7 no_employees
 0.003844   18 coworkers
 0.002534   21 phys_health_interview
 0.001072   17 phys_health_consequence
 0.000826   19 supervisor
 0.000781    9 tech_company
 0.000523    8 remote_work
 0.000197    4 self_employed
 0           1 Age

Selected attributes: 6,5,11,3,10,2,23,15,14,16,22,20,12,13,7,18,21,17,19,9,8,4,1 : 23
```

Figura 11: Resultados do método InfoGainAttributeEval

```
=== Attribute Selection on all input data ===

Search Method:
        Attribute ranking.

Attribute Evaluator (supervised, Class (nominal): 24 treatment):
        Correlation Ranking Filter
Ranked attributes:
 0.3772    5 family_history
 0.3615    6 work_interfere
 0.1874   11 care_options
 0.1834    2 Gender
 0.1499   23 obs_consequence
 0.1464   10 benefits
 0.1328   14 anonymity
 0.0833   20 mental_health_interview
 0.0768   16 mental_health_consequence
 0.0743   22 mental_vs_physical
 0.0737    1 Age
 0.0668    3 Country
 0.0619   15 leave
 0.0415   13 seek_help
 0.0399   12 wellness_program
 0.0392   21 phys_health_interview
 0.0352   17 phys_health_consequence
 0.0333    7 no_employees
 0.0329    9 tech_company
 0.0269    8 remote_work
 0.0265   18 coworkers
 0.0244   19 supervisor
 0.0165    4 self_employed

Selected attributes: 5,6,11,2,23,10,14,20,16,22,1,3,15,13,12,21,17,7,9,8,18,19,4 : 23
```

Figura 12: Resultados do método CorrelationAttributeEval

Ambos os métodos devolvem um valor para cada atributo. Quanto maior for esse valor, maior é o impacto que essa característica tem, no momento de classificação. Como seria de esperar, os atributos que obtêm melhores resultados são aqueles previamente identicados, com recurso apenas à visualização da sua distribuição.

É possível agora, excluir atributos cujo o peso na classificação é insignificante. Optámos então por eliminar, "self_employed", "supervisor", "tech_company"e "remote_work". Avançamos então com apenas 19 features agora.

6 Clustering

Clustering é uma técnica de mineração de dados para fazer agrupamentos automáticos de dados segundo seu grau de semelhança. Neste trabalho utilizou-se o algoritmo *k means* que agrupa os dados tentando separar amostras em n grupos de variância igual, minimizando um critério conhecido como inércia ou soma de quadrados dentro do *cluster*. Neste algoritmo o número de *clusters* tem que ser difinido previamente[22].

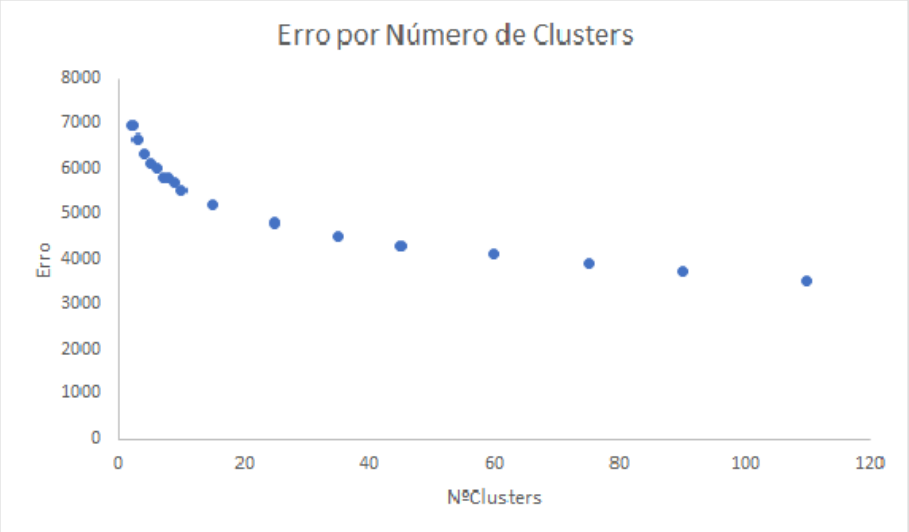

Figura 13: Gráfico do erro por número de clusters

Na *Figura* 13 está representado o erro por número de *clusters* para o conjunto de teste, que representa 34% de todo o conjunto de dados. Como se pode observar o erro começa a estabilizar a partir dos 20 *clusters*, ou seja, diminui menos. Mas decidimos escolher apenas 4 clusters para descrever o nosso problema. Isto porque, descrever 20 clusters para um problema de classificação binária com 23 features é muito complicado. Escolhendo 3 clusters é possível descrever qualitativamente cada cluster. Para fazer o gráfico do erro por número de clusters recorreu-se ao *Excel*.

Attribute	Full Data (813.0)	Cluster# 0 (257.0)	1 (214.0)	2 (154.0)	3 (188.0)
Age	31.9176	30.3658	33.5327	32.5779	31.6596
Gender	Male	Male	Male	Male	Male
Country	United States	United States	United States	United States	United States
self_employed	No	No	No	No	No
family_history	No	No	Yes	No	Yes
work_interfere	Sometimes	NA	Sometimes	NA	Sometimes
no_employees	More than 1000	More than 1000	More than 1000	6-25	26-100
remote_work	No	No	No	Yes	No
tech_company	Yes	Yes	Yes	Yes	Yes
benefits	Yes	Dont know	Yes	No	No
care_options	No	No	Yes	No	No
wellness_program	No	No	Yes	No	No
seek_help	No	Dont know	Yes	No	No
anonymity	Dont know	Dont know	Yes	Dont know	Dont know
leave	Dont know	Dont know	Dont know	Dont know	Somewhat easy
mental_health_consequence	No	Maybe	No	Yes	Yes
phys_health_consequence	No	No	No	No	No
coworkers	Some of them	Some of them	Some of them	No	Some of them
supervisor	Yes	Yes	Yes	No	Some of them
mental_health_interview	No	No	No	No	No
phys_health_interview	Maybe	Maybe	No	Maybe	Maybe
mental_vs_physical	Dont know	Dont know	Yes	Dont know	No
obs_consequence	No	No	No	No	No
treatment	Yes	No	Yes	No	Yes

Figura 14: Distribuição das Features com 4 clusters

Na *Figura* 14 está represntada a distribuição das features utilizando com 4 *clusters*.

O cluster 1 e 3 dizem respeito a "yes" para o tratamento, a nossa classe alvo. As diferenças entres estes clusters prendem-se com os seguintes atributos:

- "no_employes", em que no cluster 1 o número de empregados que a empresa/organização tem é >1000 e no cluster 3 é entre 26-100;
- "benefits",em que no cluster 1 o patrão fornece benefícios para a saúde mental e no cluster 3 não há benefícios;
- "care_options", em que no cluster 1 a pessoa conhece as opções para cuidado da saúde mental e no cluster 3 não conhece;
- "weelness_program", em que no cluster 1 o patrão já falou com o empregado o programa de bem estar deste e no cluster 3 não falou;
- "seek_help", em que no cluster 1 o patrão fornece recursos para aprender mais sobre problemas de saúde mental e como procurar ajuda e no cluster 3 isto não acontece;
- "anonymity", em que no cluster 1 a anonimidade do trabalhador é protegida e no cluster 3 não sabem se é protegida;
- "leave", no cluster 1 o trabalhador não sabe se é fácil tirar uma licença médica e no cluster 3 existe essa facilidade;
- "mental_health_consequence", no cluster 1 a resposta à pergunta "acham que discutir a saúde mental com o patrão terá consequências negativas?" a resposta é não e no cluster 3 a resposta é sim;

- “supervisor”, no cluster 1 os trabalhadores estão dispostos a falar sobre o seu estado mental aos supervisores diretos e no cluster 3 a apenas alguns deles;
- “phys_health_interview”, no cluster 1 os trabalhadores não falam de um problema físico durante uma entrevista e no cluster 3 talvez o fizessem;
- “mentalvsphysical”, no cluster 1 o patrão leva tão a sério os problemas mentais como os físicos e no cluster 3 isso não se verifica.

O cluster 0 e 2 dizem respeito a “no” para o tratamento. As diferenças entres estes clusters prendem-se com os seguintes atributos:

- "no_employes", em que no cluster 0 o número de empregados que a empresa/organização tem é >1000 e no cluster 2 é entre 6-25;
- "remote_work", em que no cluster 0 as pessoas não costumam trabalhar fora do escritório pelo menos 50% do seu tempo e no cluster 2 costumam trabalhar fora do escritório;
- "benefits", no cluster 0 as pessoas não sabem se o patrão fornece benefícios para a saúde mental e no cluster 2 não há benefícios;
- "seek_help", no cluster 0 o patrão fornece recursos para aprender mais sobre problemas de saúde mental e como procurar ajuda e no cluster 2 isto não acontece;
- “mental_health_consequence”, no cluster0 a resposta à pergunta “acham que discutir a saúde mental com o patrão terá consequências negativas?” a resposta é talvez e no cluster 2 a resposta é sim;
- "coworkers", no cluster 0 os trabalhadores estão dispostos a falar sobre o seu estado mental a alguns colegas de trabalho e no cluster 2 não falam com nenhum colega de trabalho;
- "supervisor", no cluster 0 os trabalhadores estão dispostos a falar sobre o seu estado mental aos supervisores diretos e no cluster 0 não falam com nenhum supervisor.

7 Classificação

Numa primeira parte do trabalho recorreu-se ao software WEKA [11] para testar a performance de vários classificadores e assim saber qual o melhor. Utilizou-se validação cruzada [23] para a classificação dos dados. Para ver o desempenho de cada classificador, comparou-se o valor da precisão e da sensibidade de cada classificador. Os resultados podem ser vistos na *Tabela* 1

A precisão [24] é dada por $\frac{V_p}{V_p+F_p}$, ou seja, de todos os elementos que foram classificados como positivos, quais é que são realmente positivos. Quantos elementos selecionados são relevantes.

A sensibilidade [24] é dada por $\frac{V_p}{V_p+F_n}$,ou seja, a percentagem dos verdadeiros positivos dentro de todos os exemplos cuja classe esperada é a classe positiva.

A área sob a curva ROC é obtida pela representação taxa de verdadeiros positivos ($\frac{V_p}{P_t}$) versus a taxa de falsos positivos ($\frac{F_p}{N_t}$). Uma área com valor de 1 representa um teste perfeito, uma área com um valor de 0,5 representa um teste sem valor [25].

Classificador	**Precisão**	**Sensibilidade**	**Área ROC**
Naive Bayes	0,799	0,799	0,884
Bayes Net	0,807	0,807	0,885
LibSVM	0,846	0,827	0,826
Logistic	0,834	0,830	0,888
ZeroR	0,255	0,505	0,498
Multilayer Perceptron	0,794	0,794	0,865
SMO	0,844	0,828	0,827
OneR	0,848	0,830	0,828
J48	0,825	0,821	0,848
IBk	0,734	0,733	0,741
Random Forest	0,829	0,823	0,894
Random Tree	0,726	0,726	0,738

Tabela 1: Desempenho dos diferentes classificadores

O classificador que apresenta um melhor desempenho é o *OneR*, este algoritmo cria uma regra para cada atributo dos dados de treino e selecciona a regra com menor percentagem de erro como regra única [26]. Para criar uma regra para um atributo é necessário determinar a classe que aparece mais vezes para esse atributo. "Uma regra" é simplesmente um conjunto de valores dos atributos limitados pela sua classe maioritária.A percentagem de erro de uma regra é o número de instâncias de treino na qual a classe de um valor de um atributo não é concordante com a classificação desse atributo na regra.

O classificador que apresenta um pior desempenho é o *ZeroR*, este algoritmo baseia-se no alvo e ignora todos os preditores. Simplesmente prevê a classe maioritária [27].

Na segunda parte do trabalho recorreu-se à linguagem de programação *Python* [28].

Começou-se esta parte com 19 features sendo que 18 são nominais e 1 numérica. Deste modo foi necessário prodeder à transformação das variáveis nominais para variáveis numéricas. Este passo é crucial para a classificação porque muitos classificadores só aceitam variáveis numéricas e alguns apenas binários, como p.e. no algoritmo Random Forest onde o vetor *Target* tem que ser binário. Mas antes desta transformação, retirou-se as amostras com valor NaN da variável *work_ interfere*. A sua remoção é importante pois, caso se avançasse com esta variável, o classificador poderia considerar um NaN um valor importante.

Uma vez alterado os valores da variável procedeu-se ao estudo se o conjunto de dados era ou não balanceado de modo a que classificação seja equilibrada e justa. Com a remoção das amostras onde a variável *work_ interfere* tinha valores NaN, o conjunto de dados ficou desequilibrado visto que a maioria das amostras com NaN eram de sujeitos que nunca tinham recorrido a um tratamento. Deste modo, retirou-se 260 amostras de sujeitos que não recorreram a tratamento ficando com 352 amostras cada classe. Por fim baralhou-se o conjunto de dados.

Com o conjunto de dados pronto a trabalhar começou-se por dividir em treino e teste. Decidiu-se dividir 70% em treino e 30% teste, estando o nosso conjunto pronto para a classificação.

Nesta secção foram utilizados os seguintes classificadores: Support Vector Machine (SVM), Random Forest, K-Nearest Neighbours (KNN), sendo que no Random Forest e no KNN precedeu-se um estudo de parâmetros com o objetivo de maximizar a performance.

No classificador Random Forest tentou-se variar o número de ramos no classificador e o número mínimo de amostras necessárias para estar num nó. Inicialmente o classificador apresentou *accuracy*, sensibilidade e especificidade 1.0. Com o objetivo de evitar o overfit, procedeu-se ao aumento de ramos no classficador e do número mínimo de amostras num nó, acabando por ser em vão.

No classificador KNN, o número de vizinhos é o fator mais importante a otimizar. Através de um ciclo, classificou-se começando com 1 vizinho até ao mesmo número de amostras do grupo treino. Através da *Figura* 15 pode-se perceber o número de vizinhos que potencializa a performance do classificador é 42, atingindo uma *acurracy* de 75,3%.

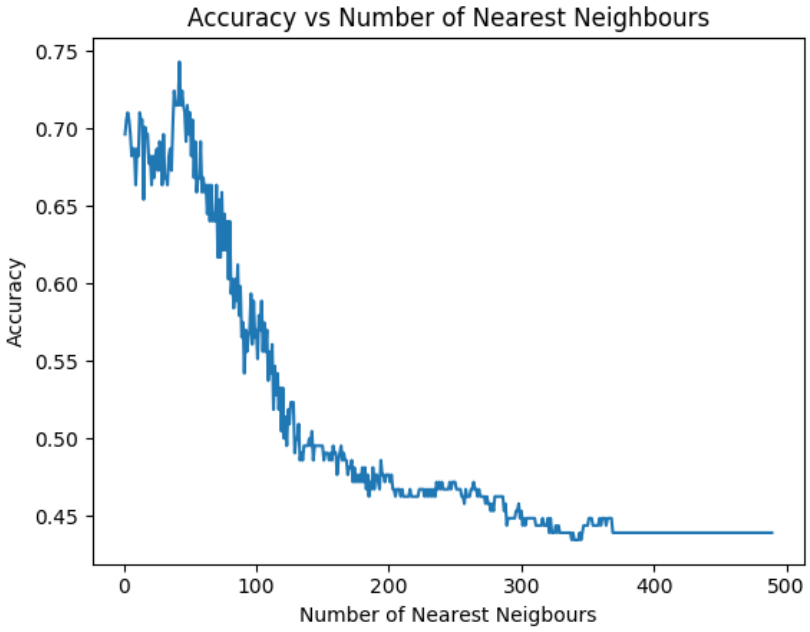

Figura 15: Influência do número de vizinhos na performance de KNN

7.1 Resultados

Para termos comparativos classificou-se ainda todas as features obtidas, sem qualquer seleção, desta forma podemos verificar as alterações que a otimização de features provoca. Os resultados podem ser observados na tabela 5

Classificador	***Accuracy***	**Sensibilidade**	**Especificidade**
SVM	91,1%	89,4%	92,5%
Random Forest	100%	100%	100%
KNN	75,3%	81,9%	68,3%

Tabela 2: Valores de *Accuracy*, Sensibilidade e Especificidade para cada classificador.

De modo a ajudar a análise de resultados, apresenta-se as matrizes confusão para cada classificador.

		Previsto	
		Positivo	**Negativo**
Verdadeiro	**Positivo**	111	9
	Negativo	10	84

Tabela 3: Matriz Confusão do Classificador SVM

		Previsto	
		Positivo	**Negativo**
Verdadeiro	**Positivo**	120	0
	Negativo	0	94

Tabela 4: Matriz Confusão do Classificador Random Forest

		Previsto	
		Positivo	**Negativo**
Verdadeiro	**Positivo**	82	38
	Negativo	17	77

Tabela 5: Matriz Confusão do Classificador Knn

Analisando os resultados podemos tirar umas breves conclusões:

1. SVM é de longe o melhor classificador apresentando *accuracy*, sensibilidade e especificidade na casa dos 90%.
2. Random Forest está claramente overfit. Não é um classificador de confiança.
3. KNN apresenta uma sensibilidade elevada em comparação aos outros discritores.

8 Discussão e Conclusão

O trabalho foi baseado num desafio recente do Kaggle. Optámos por escolher um desafio, para o qual ainda não havia grande tentavias de abordagem. Este facto fez com que o nosso trabalho se torna-se em algo mais prático e exploratório, ao invés de algo mais teórico e de melhoria de trabalhos já realizados.

O trabalho foi então transversal à diversas componentes de Data Mining. Pegando em dados brutos, que tiveram de ser devidamente trabalhados, percorremos todos os passos até à construção de classificador com resultados de performance satisfatórios. Efetivamente o melhor classificador obtido, SVM, apresenta resultados muito bons, que permitem, perante novos dados, classificar corretamente a sua classe. Importante também referir que um elevado valor de sensibilidade significa que o classificador identifica as pessoas que realmente já precisou de tratamento, ou recorreu a um tratamento. O classificador Random Forest apresenta valores de um classificador *overfit*. O desempenho preditivo dos modelos de *data mining*, o *overfitting* também tende a construir falsas expectativas a entre aqueles que dependem dos "insights"desses modelos para orientar as decisões de engenharia. Pode-se inadvertidamente basear uma decisão crítica sobre algum cenário futuro previsto por um modelo superado, expondo-nos assim a todos os riscos concomitantes. Por fim, referir que tentou-se usar uma rede neuronal feedforward na classificação, no entanto, problemas de configuração não permitiram observar os resultados, provavelmente por um *bug* do package keras ou um conflito entre o Python e o Anaconda. Uma rede neuronal teria sido uma excelente opção visto que já se provou ser muito adaptável ao tipo de variáveis e, apresentando em regra valores elevados de performance.

Além, das componentes de Machine Learning, durante todo o processo foi-nos possível retirar algumas conclusões mais teóricas, relativas ao tema escolhido para estudar.

Uma das perguntas a que nos proposemos responder era como é que a frequência de doenças e atitudes de saúde mental varia de acordo com a localização geográfica. Apesar de, a grande maioria das instâncias ter origem nos Estados Unidos, é possível fazer uma distribuição da amostra a nível mundial [29] como se pode ver na *Figura* 16.

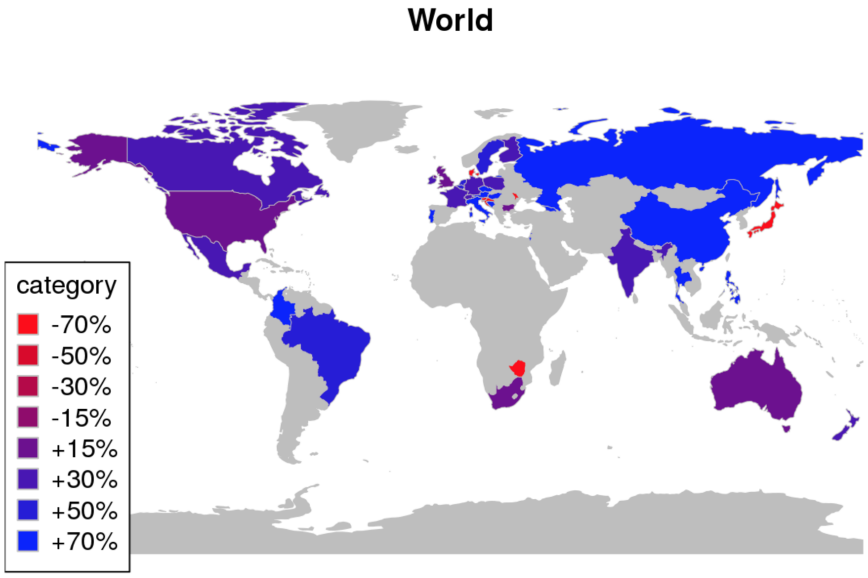

Figura 16: Distribuição da amostra a nível mundial.

É ainda possível concluir quais são os melhores fatores preditivos de uma doença mental. A resposta a esta pergunta surge no momente de Feature Selection, onde efetivamente o historial familiar é, sem dúvida o fator com um maior poder preditivo.

Em suma, consideramos que o trabalho cumpre os requisitos impostos. A componente de aprendizagem esteve bastante presente, assim como o autodidatismo. Tivemos a oportunidade de estudar e aplicar conhecimentos de uma área que está numa fase de explosão a nível mundial.

Bibliografia

[1] "Doenças Mentais." https://noticias.uol.com.br/saude/ultimas-noticias/redacao/2013/11/11/transtornos-mentais-afetam-cerca-de-700-mi-no-mundo-veja-mitos-e-verdades.htm. Acedido: 2017-14-12.

[2] P. W. Corrigan and A. C. Watson, "Understanding the impact of stigma on people with mental illness," *World psychiatry*, vol. 1, no. 1, p. 16, 2002.

[3] A. Üçok, E. Brohan, D. Rose, N. Sartorius, M. Leese, C. Yoon, A. Plooy, B. Ertekin, R. Milev, and G. Thornicroft, "Anticipated discrimination among people with schizophrenia," *Acta Psychiatrica Scandinavica*, vol. 125, no. 1, pp. 77–83, 2012.

[4] P. W. Corrigan, A. Green, R. Lundin, M. A. Kubiak, and D. L. Penn, "Familiarity with and social distance from people who have serious mental illness," *Psychiatric services*, vol. 52, no. 7, pp. 953–958, 2001.

[5] G. Thornicroft, D. Rose, A. Kassam, and N. Sartorius, "Stigma: ignorance, prejudice or discrimination?," 2007.

[6] C. Manning and P. D. White, "Attitudes of employers to the mentally ill," *The Psychiatrist*, vol. 19, no. 9, pp. 541–543, 1995.

[7] J. Fenton, D. O'Hanlon, and D. Allen, "Does having been on a 'section'reduce your chances of getting a job?," *The Psychiatrist*, vol. 27, no. 5, pp. 177–178, 2003.

[8] T. Krupa, B. Kirsh, L. Cockburn, and R. Gewurtz, "Understanding the stigma of mental illness in employment," *Work*, vol. 33, no. 4, pp. 413–425, 2009.

[9] P. Gabriel, M.-R. Liimatainen, *et al.*, "Mental health in the workplace: Introduction, executive summaries," 2000.

[10] K. Kira and L. A. Rendell, "A practical approach to feature selection," in *Proceedings of the ninth international workshop on Machine learning*, pp. 249–256, 1992.

[11] "WEKA." https://www.cs.waikato.ac.nz/ml/weka/index.html. Acedido: 2017-14-12.

[12] "Infogainattributeeval."

[13] E. Frank, M. Hall, G. Holmes, R. Kirkby, B. Pfahringer, I. H. Witten, and L. Trigg, "Weka-a machine learning workbench for data mining," in *Data mining and knowledge discovery handbook*, pp. 1269–1277, Springer, 2009.

[14] J. M. D. Balakrishnan *et al.*, "Significance of classification techniques in prediction of learning disabilities," *arXiv preprint arXiv:1011.0628*, 2010.

[15] W. I. D. Mining, "Data mining: Concepts and techniques," *Morgan Kaufinann*, 2006.

[16] X. Wu, V. Kumar, J. R. Quinlan, J. Ghosh, Q. Yang, H. Motoda, G. J. McLachlan, A. Ng, B. Liu, S. Y. Philip, *et al.*, "Top 10 algorithms in data mining," *Knowledge and information systems*, vol. 14, no. 1, pp. 1–37, 2008.

[17] J. Yu, A. Xue, E. Redei, and N. Bagheri, "A support vector machine model provides an accurate transcript-level-based diagnostic for major depressive disorder," *Translational psychiatry*, vol. 6, no. 10, p. e931, 2016.

[18] D. Meer, P. Hoekstra, M. van Donkelaar, J. Bralten, J. Oosterlaan, D. Heslenfeld, S. Faraone, B. Franke, J. Buitelaar, and C. Hartman, "Predicting attention-deficit/hyperactivity disorder severity from psychosocial stress and stress-response genes: a random forest regression approach," 2017.

[19] "Osmi."

[20] "Osmi research."

[21] "Equilíbrio entre a saúde física e mental." `http://rumosaudeseguros.com.br/equilibrando-saude-mental-e-fisica/`. Acedido: 2017-16-12.

[22] "Algoritmo K-means." `http://scikit-learn.org/stable/modules/clustering.html`. Acedido: 2017-14-12.

[23] P. Refaeilzadeh, L. Tang, and H. Liu, "Validação cruzada," in *Encyclopedia of database systems*, pp. 532–538, Springer, 2009.

[24] "Precisão e Sensibilidade." `http://scikit-learn.org/stable/auto_examples/model_selection/plot_precision_recall.html`. Acedido: 2017-13-12.

[25] "Área sobre a curva ROC." `http://gim.unmc.edu/dxtests/roc3.htm`. Acedido: 2017-15-12.

[26] "Algoritmo OneR." `http://www.saedsayad.com/oner.htm`. Acedido: 2017-15-12.

[27] "Algoritmo ZeroR." `http://chem-eng.utoronto.ca/~datamining/dmc/zeror.htm`. Acedido: 2017-15-12.

[28] "Python." `https://www.python.org/`. Acedido: 2017-15-12.

[29] "Data mining of mental health."

Anexos

A Anexo A

Código utilizado durante a realização do trabalho

```
df['Age'] =
pd.to_numeric(df['Age'],errors='coerce')
def age_process(age):
    if age>=0 and age<=100:
        return age
    else:
        return np.nan
df['Age'] = df['Age'].apply(age_process)
```

Figura 1- Remover os outliers da Idade, apenas se aceita a idade compreendida entre 0 e 100

```
df['Age_Group'] = pd.cut(df['Age'].dropna(),
                         [0,18,25,35,45,99],
                         labels=['<18','18-
24','25-34','35-44','45+'])
```

Figura 2- Agrupar a idade nos seguintes grupos: "<18"; "18-24"; "25-34";"35-44";">45"

```
fig,ax = plt.subplots(figsize=(8,6))
sns.countplot(data=df,x = 'Age_Group',hue=
'remote_work',ax=ax)
plt.title('Remote Work vs Age Group')
plt.show()
```

Figura 3- Código que permite visualizar a imagem "Remote Work vs Age"

```
country_count =
Counter(df['Country'].dropna().tolist()).most_
common(10)
country_idx = [country[0] for country in
country_count]
country_val = [country[1] for country in
country_count]
fig,ax = plt.subplots(figsize=(8,6))
sns.barplot(x = country_idx,y=country_val ,ax
=ax)
plt.title('Top ten country')
plt.xlabel('Country')
plt.ylabel('Count')
ticks =
plt.setp(ax.get_xticklabels(),rotation=90)
plt.show()
```

Figura 4- Código que permite visualizar o top 10 dos países participantes na pesquisa

```
fig,ax = plt.subplots(figsize=(8,6))
sns.countplot(data=df,x =
'mental_vs_physical',hue=
'mental_vs_physical',ax=ax)
plt.title('Mental Health vs Physical
Health')
plt.show()
```

Figura 5- Código que permite visualizar a Saúde Mental vs Saúde Física

```
fig,ax = plt.subplots(figsize=(8,6))
sns.countplot(data=df,x =
'mental_health_interview',hue=
'mental_health_interview',ax=ax)
plt.title('Mental Health Interview')
plt.show()
```

Figura 6- Código que permite visualizar se as pessoas diriam que tinham um problema de saúde mental numa entrevista

```
fig,ax = plt.subplots(figsize=(8,6))
sns.countplot(data=df,x =
'phys_health_interview',hue=
'phys_health_interview',ax=ax)
plt.title('Physical Health Interview')
plt.show()
```

Figura 7- Código que permite visualizar se as pessoas diriam que tinham um problema de saúde física numa entrevista

```
data.dropna(axis=0, subset =
['work_interfere'], inplace=True)
```

Figura 8- Código que elimina os valores NaN (not a number) da feature Interferência do trabalho

```
listLabelsData = list(data)
for a in listLabelsData[1::]:

    typesOfLabels = data[a].unique()
    #print(typesOfLabels)
    numericalLabels = list(range(0,
len(typesOfLabels)))
    data[a].replace(typesOfLabels,
numericalLabels, inplace = True)
```

Figura 9- Código que transforma as variáveis nominais para variáveis numéricas.

```
count = 0
counters = 0
print(targetVector.unique())
for i in targetVector:
    if (i==1):
        count = count + 1
    else:
        counters = counters + 1
print(count)
print(counters)
```

Figura 10- Código que conta o número de amostras em que possuem tratamento mental e das que não possuem.

```
treatment = data.loc[(data.treatment ==
1)]
non_treatment = data.loc[(data.treatment
== 0)]

treatment = treatment.sample(frac=1)
treatment = treatment[0:352]

newData = pd.concat([treatment,
non_treatment])
newData = newData.sample(frac=1)

newData.to_csv("newData.csv")
```

Figura 11- Código que permite eliminar amostras em excesso da classe Sem Tratamento, de modo a ter um conjunto de dados balanceado, baralhando no fim as amostras.

```
indexes = np.random.rand(len(newData)) <
0.7
train = newData[indexes]
test = newData[~indexes]

targetVector = newData.treatment #No - 1,
Yes - 2
```

Figura 12- Código que divide o conjunto de dados em treino e teste.

```
#SVM


classifier1 = svm.SVC()
classifier1.fit(train, train.treatment)
predictions1 = classifier1.predict(test)

tn, fp, fn, tp =
sk.metrics.confusion_matrix(test.treatmen
t, predictions1).ravel()
accuracy = (tp + tn) / (tp + tn + fn +
fp)
sensitivity = tp / (tp + fn)
specificity = tn / (tn + fp)

print('SVM')
print('Accuracy: ', accuracy,
'\nSensitivity: ', sensitivity,
'\nSpecificity: ', specificity)
print('\nConfusion
Matrix:\n',sk.metrics.confusion_matrix(te
st.treatment, predictions1))
```

Figura 13- Código que permite treinar o classificador SVM, testá-lo e avaliar a performance deste.

```
#KNN

print('\nKNN')
listAccuracy = []
listNeighbors = []

for x in range(1, len(train)):
    clf =
neighbors.KNeighborsClassifier(x)
    knn_model = clf.fit(train,
train.treatment)
    preds_KNN = clf.predict(test)
    tn2, fp2, fn2, tp2 =
sk.metrics.confusion_matrix(test.treatmen
t, preds_KNN).ravel()
    accuracy2 = (tp2 + tn2) / (tp2 + tn2
+ fn2 + fp2)
    sensitivity2 = tp2 / (tp2 + fn2)
    specificity2 = tn2 / (tn2 + fp2)

    listAccuracy.append(accuracy2)
    listNeighbors.append(x)

plt.figure(1)
plt.title('Accuracy vs Number of Nearest
Neighbours')
plt.plot(listNeighbors, listAccuracy)
plt.xlabel('Number of Nearest Neigbours')
plt.ylabel('Accuracy')
plt.show()

print('Max Accuracy: ',
max(listAccuracy), '\nNumber of
Neighbours: ',
listNeighbors[listAccuracy.index(max(list
Accuracy))])
```

Figura 14- Código que procurar o numero ideal de vizinhos de modo a maximizar a performance do classificador K-Nearest Neighbour

```
clf =
neighbors.KNeighborsClassifier(listNeighb
ors[listAccuracy.index(max(listAccuracy))
])
knn_model = clf.fit(train,
train.treatment)
preds_KNN = clf.predict(test)
tn2, fp2, fn2, tp2 =
sk.metrics.confusion_matrix(test.treatmen
t, preds_KNN).ravel()
accuracy2 = (tp2 + tn2) / (tp2 + tn2 +
fn2 + fp2)
sensitivity2 = tp2 / (tp2 + fn2)
specificity2 = tn2 / (tn2 + fp2)

print('\nFinal Values: \nAccuracy: ',
accuracy2, '\nSensitivity: ',
sensitivity2, '\nSpecificity: ',
specificity2)
print('\nConfusion
Matrix:\n',sk.metrics.confusion_matrix(te
st.treatment, preds_KNN))
```

Figura 15- Código que permite treinar o classificador KNN, testá-lo e avaliar a performance deste.

```
clf = RandomForestClassifier(n_estimators
= 200, oob_score = True, n_jobs = -1,
random_state =50,
                             max_features
= "auto", min_samples_leaf = 100)
clf.fit(train, train.treatment)
preds = clf.predict(test)
tn1, fp1, fn1, tp1 =
sk.metrics.confusion_matrix(test.treatmen
t, preds).ravel()
accuracy1 = (tp1 + tn1) / (tp1 + tn1 +
fn1 + fp1)
sensitivity1 = tp1 / (tp1 + fn1)
specificity1 = tn1 / (tn1 + fp1)

print('\nRANDOM Forest')
print('Accuracy: ', accuracy1,
'\nSensitivity: ', sensitivity1,
'\nSpecificity: ', specificity1)
print('\nConfusion
Matrix:\n',sk.metrics.confusion_matrix(te
st.treatment, preds))
```

Figura 16- Código que permite treinar o classificador Random Forest, testá-lo e avaliar a performance deste.