**Sumário**

**1 INTRODUÇÃO**

**2 FUNDAMENTAÇÃO TEÓRICA**

**2.1 Acidente Vascular Cerebral – AVC**

**2.2 Estatística amostral**

2.2.1 Amostragem

2.2.2 Anomalias

2.2.3 Normalização

**2.3 Aprendizado de Máquina**

2.3.1 Máquinas de vetores de Suporte (SVM)

2.3.2 Floresta aleatória

**2.4 Aprendizagem profunda**

2.4.1 Redes neurais convolucionais

**2.5 Sobre ajuste**

2.5.1 Argumentação dos dados

2.5.2 Divisão de treino e teste

2.5.3 Validação cruzada

**2.6 Métricas para avaliação de desempenho e qualidade de um classificador**

2.6.1 Precisão

2.6.2 Taxa de Falso Negativo e Falso Positivo

2.6.3 Curva ROC

**3 METODOLOGIA**

**3.1 Ferramentas utilizadas**

3.1.1 Python

3.1.2 Bibliotecas e frameworks

**3.2 Base de dados**

3.1.3 Dados estruturados

3.1.4 Imagens

**4 DESENVOLVIMENTO**

**4.1 Classificador para dados textuais**

**4.1.1 Preparação dos dados**

**4.1.2 Modelagem: criação do classificador**

**4.1.3 Avaliação dos resultados obtidos**

**4.2 Classificador para imagens**

**4.2.1 Pré-processamento: tratamento dos dados**

**4.1.2 Processamento: análise dos dados e criação do classificador**

**4.1.3 Avaliação do modelo criado**

**5 RESULTADOS**

**5.1 Classificador com base em dados textuais**

**5.2 Classificador de imagens**

**REFERÊNCIAS**

**2 FUNDAMENTAÇÃO TEÓRICA**

**2.1 Estatística**

**2.1.1 Amostragem**

**2.1.2 Testes de normalidade**

**2.1.3 Normalização**

Modelos lineares de Aprendizado de Máquina lidam com distâncias entre os pontos. Se o conjunto de dados utilizado na criação desses modelos apresenta variáveis continuas com intervalos diferentes, isso pode acabar prejudicando o desempenho da solução a qual se deseja treinar. Dessa forma, muitas vezes é importante normalizar os dados para que um melhor resultado seja obtido.

Uma forma de normalização muito comum é a normalização padrão, a qual se calcula a média e o desvio padrão dos dados e atualiza cada valor do conjunto da seguinte maneira:

Esse tipo de normalização tem maior eficácia quando aplicada em dados de distribuição normal. Em casos em que a distribuição dos dados não pode ser assumida como normal, pode-se utilizar outras técnicas de normalização, como a normalização máximos e mínimos, por exemplo. Essa outra técnica normaliza os dados a partir da fórmula:

**2.2 Modelagem**

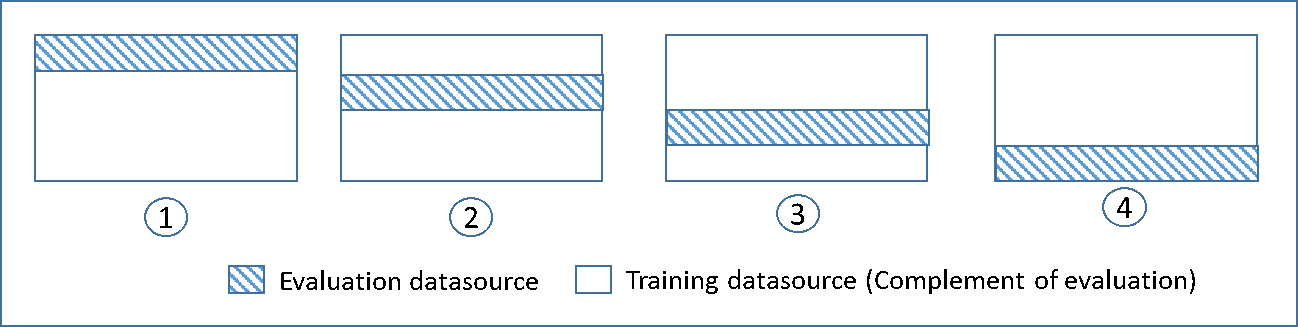
**2.2.1 Sobre ajuste**

**2.2.2 Validação cruzada**

“Validação cruzada é uma técnica para avaliar modelos de ML por meio de treinamento de vários modelos de ML em subconjuntos de dados de entrada disponíveis e avaliação deles no subconjunto complementar dos dados.” ()

Utilizar a validação cruzada na modelagem evita o sobre ajuste do modelo de uma forma muito eficiente, pois torna-se possível dividir os dados de entrada em subconjuntos de dados, conhecidos como dobras. Dessas dobras, uma delas será escolhida para teste do modelo criado e as demais serão utilizadas no treinamento do modelo. Para k dobras, tal processo é repetido k vezes, sendo que cada a cada iteração um subconjunto diferente será reservado para avaliação e os demais para treinamento do modelo.

Portanto, realizando a validação cruzada avalia-se o desempenho de um modelo com dados reservados para testagem, os quais são diferentes dos dados usados para ensinamento do respectivo modelo. Além dessa divisão de treino e teste, a validação cruzada permite que diferentes combinações dos dados sejam usadas no treinamento do modelo, tornando o resultado ainda mais fiel para conjunto de dados menores. A imagem abaixo ilustra um processo de validação cruzada para quatro dobras:



Fonte: https://docs.aws.amazon.com/pt\_br/machine-learning/latest/dg/cross-validation.html

**2.2.3 Argumentação de dados**

**4. Desenvolvimento**

**4.1 Classificador para dados textuais**

**4.1.1 Preparação dos dados**

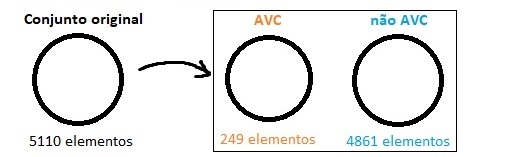
Antes de iniciar o processo de criação de um modelo preditivo, tanto para classificação quanto para regressão, é necessário preparar os dados que serão utilizados nesse processo. Isto é necessário para que o melhor desempenho deste modelo a ser criado seja alcançado.

As tecnologias analíticas que podemos utilizar são poderosas, mas impõem determinados requisitos sobre os dados que usam. Com frequência, elas exigem que os dados estejam em uma forma diferente de como são fornecidos naturalmente, e alguma conversão será necessária. (PROVOST e FAWCETT, 2016, p. 29).

Dessa forma, algumas alterações nos dados foram feitas com intuito de prepara-los para uso correto na criação do classificador. Ao analisar a distribuição da variável alvo do conjunto de dados, foi encontrado a princípio desbalanceamento dos valores. Com 4861 casos de não ocorrência de AVC e apenas 249 casos para ocorrência da doença, seria sintetizado um classificador enviesado para casos de não AVC, classificando dados que receberia de forma tendenciosa.

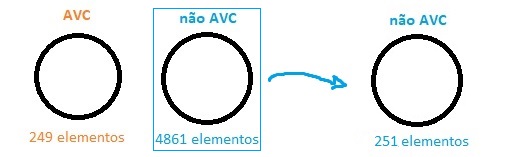
Para realizar o balanceamento dessas duas classes (*AVC* e *não AVC*) foi realizado o seguinte procedimento:

1. Separação do conjunto total a partir das duas classes:



Fonte: autoria própria

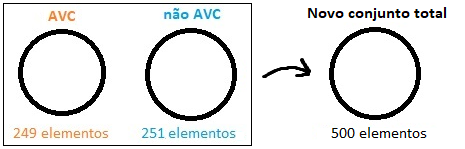
1. Subamostragem do conjunto de dados da classe *não AVC*:

**

Fonte: autoria própria

\*A subamostragem deve ser feita de forma aleatória para que a nova amostra seja bem representativa com relação ao conjunto original em que tal amostragem foi realizada.

1. Junção do conjunto de dados da classe *AVC* com o novo conjunto de dados da classe *não AVC*:



Fonte: autoria própria

Portanto, com o balanceamento das classes presentes na variável alvo, tem-se um novo conjunto de dados de 500 elementos com aproximadamente 50% dos dados sendo casos de AVC e o restante para casos de não AVC. Esse “novo” conjunto de dados criado foi utilizado ao decorrer de todo este trabalho para desenvolver o modelo classificador.

Com o balanceamento feito, chegou-se o momento de analisar o formato de cada variável do conjunto de dados abordado. Como a maioria dos algoritmos de Machine Learning conseguem trabalhar apenas com atributos de formato numérico, variáveis que se encontram em um formato diferente acabam precisando de tratamento para que o classificador consiga interpretá-las da forma correta.

Na figura abaixo é possível ver o formato original de cada variável do conjunto de dados utilizado. Cada variável foi separada pelo seu respectivo tipo, quantitativa ou qualitativa:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Variáveis qualitativas | | | Variáveis quantitativas | |
| Coluna | Formato original | Possíveis valores | Coluna | Formato original |
| Gênero | Texto | *Male* ou *Female* | Idade | Numérico |
| Hipertensão | Numérico (binário) | 0 ou 1 | Nível médio de glicose | Numérico |
| Doença cardíaca | Numérico (binário) | 0 ou 1 | BMI | Numérico |
| Casado | Texto | *Yes* ou *No* |  |  |
| Tipo de emprego | Texto | *Private, self-employed, children, govt\_job, never\_worked* |
| Tipo de residência | Texto | *Urban* ou *Rural* |
| Status com relação ao uso de cigarro | Texto | *Never smoked, unknown, formely smoked, smokes* |
| Ocorrência de AVC | Numérico (binário) | 0 ou 1 |

Como pôde ser observado na tabela acima, algumas colunas de formato original texto apresentam apenas dois possíveis valores. Nesses casos, para corrigir o formato dessas colunas para um formato numérico pode-se substituir um desses valores pelo dígito “1” e o outro pelo ”0”. Abaixo, tem-se um exemplo para a coluna Gênero:

|  |  |  |
| --- | --- | --- |
| Gênero |  | Masculino |
| Masculino | 1 |
| Masculino | 1 |
| Feminino | 0 |
| Feminino | 1 |
| Feminino | 0 |
| Masculino | 0 |
| Feminino | 0 |

Já nos casos em que as colunas apresentam mais do que dois possíveis valores, para fazer a correção destas pode-se utilizar da técnica de *OneHotEncoder*. Tal transformador recebe uma coluna com diferentes possíveis valores textuais e cria novas colunas binárias para cada valor diferente. Para ilustrar o funcionamento dessa técnica, a figura abaixo mostra um exemplo desta transformação utilizando como referência valores da coluna tipo de emprego:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Tipo de emprego |  | Privado | Autônomo | Cargo público | Criança |
| Privado | 1 | 0 | 0 | 0 |
| Privado | 1 | 0 | 0 | 0 |
| Autônomo | 0 | 1 | 0 | 0 |
| Privado | 1 | 0 | 0 | 0 |
| Criança | 0 | 0 | 0 | 1 |
| Cargo público | 0 | 0 | 1 | 0 |
| Autônomo | 0 | 1 | 0 | 0 |

Como visto acima, toda linha obrigatoriamente contém o dígito “1” em pelo menos uma das novas colunas criadas. Isso ocorre, pois, a criação dessas colunas é feita a partir da ocorrência de todos os diferentes possíveis valores, de forma a ter uma nova coluna para cada valor diferente presente. Por essa razão, pode-se deduzir o valor de uma coluna a partir dos valores das demais, já que os valores são exclusivos.

O processo descrito acima foi realizado para as colunas “Tipo de emprego” e “Status com relação ao uso de cigarro”. Em ambos os casos foi omitida uma das colunas criadas pois a mesma, como já mencionado, pode ser deduzida a partir das demais. Para “Tipo de emprego”, a informação omitida foi a nova coluna criada *Criança* e, para “Status com relação ao uso de cigarro”, a coluna omitida foi *Unknown*.

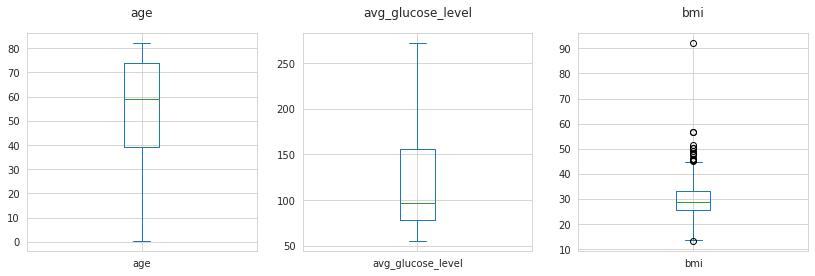
Além do formato numérico das variáveis, algoritmos de Aprendizado de Máquina também não costumam lidar bem com a presença de dados nulos no conjunto de dados. Ela não traz nenhum benefício para o classificador, pois esses dados são apenas casos de falta de informação e podem gerar erros quando passados para um modelo.

Dessa forma, cada variável presente no conjunto de dados foi analisada com relação a presença desse tipo de dado e a única que apresentou os dados nulos foi *BMI*. A distribuição dessa variável com relação a esses dados pode ser observada na imagem abaixo:

|  |  |  |
| --- | --- | --- |
| Distribuição da variável BMI com relação a dados nulos | | |
| Conjunto de dados total | 249 casos de AVC | 209 valores não nulos (84%) |
| 40 valores nulos (16%) |
| 251 casos de não AVC | 245 valores não nulos (98%) |
| 6 valores nulos (2%) |

Para correção dos valores nulos encontrados em *BMI* fora calculada a mediana dos valores não nulos dessa coluna e depois foi feita a substituição dos valores nulos por essa mediana calculada. Após tal procedimento, todos os valores nulos do conjunto de dados foram corrigidos.

Outro aspecto importante que deve ser analisado no conjunto de dados é a presença de anomalias ou *outliers*. Tais valores podem atrapalhar a análise de determinado conjunto de dados, pois são sujeiras e geralmente trazem certo ruído às medidas estatísticas da respectiva distribuição de valores que pertencem. Para identificação desse tipo de valor foi realizada a análise gráfica das distribuições das variáveis quantitativas do conjunto de dados utilizado via *boxplot*:



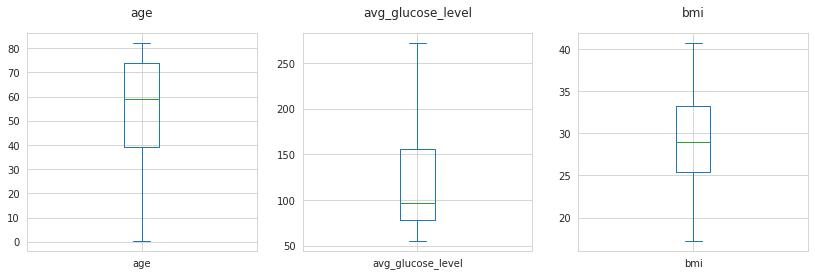
Como é possível observar na imagem acima, das três colunas analisadas apenas a variável *BMI* apresentou dados anômalos, enquanto que as demais estão corretas com relação a esse tipo de dados. Com a identificação das anomalias, agora pode-se iniciar a etapa de tratamento. Para iniciar a correção desses valores foi calculado o IQR do conjunto e posteriormente foram determinados o limite superior e o limite inferior.

**IQR:** 3° quartil – 1° quartil

**Limite superior:** Mediana + 1.5 x IQR

**Limite inferior:** Mediana - 1.5 x IQR

Com tais medidas estabelecidas, agora torna-se possível substituir todo valor maior que o limite superior pelo próprio limite superior e todo valor menor que o limite inferior pelo próprio limite inferior. Após a realização desses procedimentos, as anomalias foram corrigidas e o resultado obtido está expresso logo abaixo:



Como algumas variáveis numéricas apresentaram ainda intervalos diferentes, a normalização foi realizada para que tal fato não prejudicasse a criação do classificador. Modelos de aprendizado de máquina lineares podem acabar sofrendo com a falta de padronização dos intervalos. Uma variável que varia de 0 a 80, enquanto outra varia de 50 a 200, por exemplo, pode acabar prejudicando o modelo linear. Por essa razão, nesses casos é interessante se realizar a normalização.

A normalização é uma técnica geralmente aplicada como parte da preparação de dados para o aprendizado de máquina. O objetivo da normalização é mudar os valores das colunas numéricas no conjunto de dados para usar uma escala comum, sem distorcer as diferenças nos intervalos de valores nem perder informações. A normalização também é necessária para alguns algoritmos para modelar os dados corretamente.

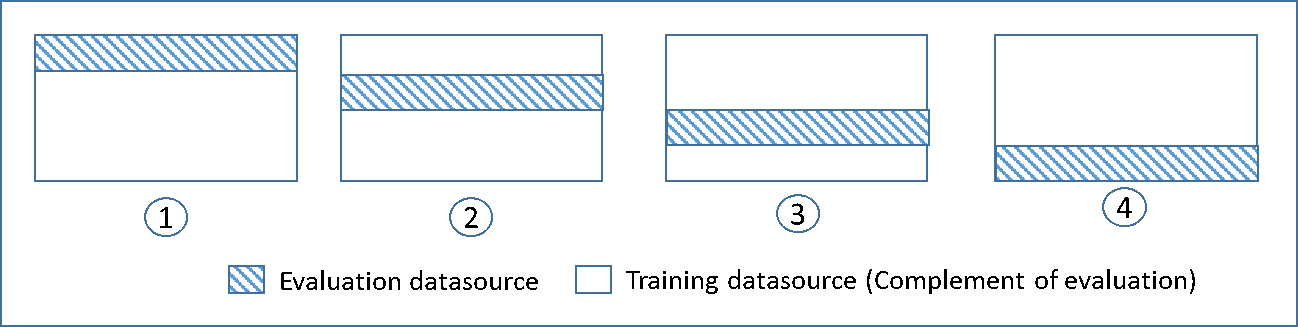
A normalização evita na modelagem problemas com tais diferenças nos intervalos das variáveis. Ela mantém os valores originais, porém em uma escala padrão. Como a normalização utilizada aqui foi a de máximos e mínimos, o padrão de escala ficou entre 0 e 1.

**4.1.2 Modelagem: criação do classificador**

Com a etapa de preparação dos dados feita, agora o conjunto de dados está pronto para ser processado. Para iniciar o processo de criação, foi dividido o conjunto total de dados disponível em partes distintas de treino e teste do modelo, pois isso evita que o classificador fique sobreajustado. Quando se usa uma parte diferente para treino e uma diferente para teste, a avaliação do modelo é feita a partir de dados desconhecidos pelo modelo, mostrando um resultado mais verídico.

Porém, apenas essa divisão pode não ser suficiente para conjuntos pequenos, pois mesmo criando os conjuntos de treino e teste de forma aleatória, pode ocorrer a seleção de valores mais representativos dependendo da semente definida para aleatoriedade. Por essa razão, uma boa maneira de lidar com conjuntos pequenos é fazer várias divisões de treino e teste no conjunto, pois assim treina-se o modelo e o avalia com diferentes subconjuntos de dados, trazendo resultados mais confiáveis. Uma forma bastante comum de se fazer isso é utilizando a Validação Cruzada e essa foi a forma utilizada aqui.

A divisão dos elementos foi feita com cinco dobras: os 500 elementos presentes no conjunto de dados totais foram divididos em cinco grupos de 100 elementos cada. Com esses grupos criados, foi separado quatro deles para treinamento do modelo e um para teste. Esse processo de separação foi repetido por cinco vezes de forma que o modelo fosse retreinado e novamente testado com diversas combinações de conjuntos. A figura abaixo ajuda a ilustrar o que foi feito e o funcionamento da validação cruzada:



A validação cruzada descrita acima foi utilizada para criação de dois modelos classificadores diferentes, um a partir do algoritmo de Máquina de vetores de suporte, com probabilidade habilitada, e outro a partir do algoritmo de florestas aleatórias.

**4.1.3 Avaliação dos resultados obtidos**

Falar da importância da avaliação e como ela deve ser feita para um problema de classificação (quais métricas utilizar e por quê)

Falar de cada métrica para cada modelo criado

Estabelecer o melhor modelo (melhor eficiência de acordo com o necessário – que melhor resolve o problema abordado)

**Referencias**

PROVOST, Foster; FAWCETT, Tom. Data Science para Negócios: O que Você Precisa Saber Sobre Mineração de Dados e Pensamento Analítico de Dados. Rio de Janeiro, RJ: Alta Books, 2016.

<https://docs.microsoft.com/pt-br/azure/machine-learning/component-reference/normalize-data>

<https://docs.aws.amazon.com/pt_br/machine-learning/latest/dg/cross-validation.html>

<https://scikit-learn.org/stable/modules/generated/sklearn.preprocessing.OneHotEncoder.html>

<https://dco-unesp-bauru.github.io/tcc-bcc-2020-2/BrunaLT/thesis-BrunaLT.pdf#page=39&zoom=100,113,261>

<https://www.aprendadatascience.com/blog/an%C3%A1lise-explorat%C3%B3ria-de-dados/vari%C3%A1veis-categ%C3%B3ricas-enconding>

<https://cloud.google.com/architecture/data-preprocessing-for-ml-with-tf-transform-pt1?hl=pt-br>

<https://www.kaggle.com/datasets/fedesoriano/stroke-prediction-dataset>

<https://revistas.pucsp.br/index.php/redeca/article/download/23377/16824/60265>

<https://datascience.eu/pt/aprendizado-de-maquina/validacao-cruzada-k-fold/>

<https://dco-unesp-bauru.github.io/tcc-bcc-2020-2/BrunaLT/thesis-BrunaLT.pdf#page=39&zoom=100,113,261>