Centro Universitário Carlos Drummond de Andrade

Arthur Murakami dos Santos Silva
Luis Fernando Santos da Silva
Luiz Fhelipe Moreira Vital
Thiago Bueno Gonçalves

Estudo de Dados – Base Adult

São Paulo
2021

Centro Universitário Carlos Drummond de Andrade

Estudo de Dados – Base Adult

Relatório Técnico-Científico apresentado na disciplina de Projeto Integrador para o curso de Análise e Desenvolvimento de Sistemas do Centro Universitário Carlos Drummond de Andrade (UNIDRUMMOND).

São Paulo
2021

SILVA, Arhur Murakami dos Santos; DA SILVA, Luis Fernando Santos; VITAL, Luiz Fhelipe Moreira; GONÇALVES, Thiago Bueno. **Estudo de Dados – Base Adult.** Relatório Técnico-Científico. Análise e Desenvolvimento de Sistemas – **Centro Universitário Carlos Drummond de Andrade**. Tutor: Eduardo Palhares. 2021.

RESUMO

O objetivo do trabalho era de fazer um estudo em cima de uma base de dados. A base de dados escolhida foi a Adult, uma base de dados voltada para censo. O estudo consistia em, com base nas informações presentes sobre as pessoas nessa base de dados, inferir quem possuía um patrimônio maior, menor ou igual a 50 mil dólares. Para isso, foram utilizadas estruturas de dados como DataFrame, dicionário e lista ligada, além da linguagem de programação escolhida para a codificação, que foi Python. Por fim, os resultados finais obtidos eram avaliados por uma função criada no código, onde era gerado um arquivo de texto contendo todo o resultado.

PALAVRAS-CHAVE: Adult; DataFrame; Dicionário; Lista; Python; Dados.

SUMÁRIO

1. INTRODUÇÃO

A proposta do trabalho era acessar uma base de dados, utilizando as informações contidas para a realização de um estudo. A base escolhida foi a base Adult, uma base de dados voltada para o censo. Ela possuía diversas informações sobre as pessoas que participaram da pesquisa, como por exemplo: sexo, ocupação, idade, nível de escolaridade, estado civil, país nativo, etnia, capital ganho e perdido, além de mais algumas informações.

2. DESENVOLVIMENTO

2.1 Objetivos

O objetivo da realização do trabalho era coletar todas essas informações, analisá-las e inferir se os participantes da pesquisa possuíam um patrimônio maior, menor ou igual a 50 mil dólares.

2.2. Justificativa e delimitação do problema

O estudo de dados é algo extremamente importante para que se tire conclusões sobre eles. Visualizar dados é essencial para desenvolver um senso intuitivo de sua natureza, o que guia a decisão de como lidar com eles, possibilitando até mesmo uma tomada de decisão referente ao resultado da análise dos dados.

2. 3. Fundamentação teórica

O que é estrutura de dados?

Estrutura de dados é uma estrutura organizada de dados na memória de um computador ou em qualquer dispositivo de armazenamento, de forma que os dados possam ser utilizados de forma correta.

Essas estruturas encontram muitas aplicações no desenvolvimento e sistemas, sendo que algumas são altamente especializadas e utilizadas em tarefas específicas. Usando as estruturas adequadas através de algoritmos, podemos trabalhar com uma grande quantidade de dados, como aplicações em bancos de dados ou serviços de busca.

Em uma estrutura de dados devemos saber como realizar um determinado conjunto de operações básicas, como por exemplo: Inserir dados, excluir dados, localizar um elemento, percorrer todos os itens constituintes da estrutura para visualização, classificar que se resume em colocar os itens de dados em uma determinada ordem (numérica, alfabética, etc.). (FROES, Anderson, 2021).

DataFrame

DataFrame é uma estrutura bidimensional de dados, como uma planilha. (FIGUEIREDO, Vinícius, 2018)

```
df = pd.DataFrame({'Aluno' : ["Wilfred", "Abbie", "Harry", "Julia",
"Carrie"],
                   'Faltas' : [3,4,2,1,4],
                   'Prova' : [2,7,5,10,6],
                   'Seminário': [8.5,7.5,9.0,7.5,8.0]})
df
```

	Aluno	Faltas	Prova	Seminário
0	Wilfred	3	2	8.5
1	Abbie	4	7	7.5
2	Harry	2	5	9.0
3	Julia	1	10	7.5
4	Carrie	4	6	8.0

Dicionário

A estrutura de dados do tipo Dict (Dicionário) é uma sequência mutável de objetos mapeados. Esse mapeamento é realizado criando uma relação de chave e valor — para a uma chave temos um valor.

Um Dicionário possui uma quantidade variável de objetos, os quais podem ser adicionados e removidos a qualquer momento. (PESSOA, Willian, 2019).

Lista ligada

Uma lista ligada é uma estrutura que corresponde a uma sequência lógica de entradas ou nós. Tipicamente, em uma lista ligada há um ou dois pontos conhecidos de acesso -- normalmente o topo da lista (seu primeiro elemento) e eventualmente o fim da lista (seu último elemento). Cada nó armazena também a localização do próximo elemento na sequência, ou seja, de seu nó sucessor. Desse modo, o armazenamento de uma lista não requer uma área contígua de memória. (RICARTE, Ivan, 2003).

Importância da estatística para a ciência de dados

Se imagens valem mais do que mil palavras, o mesmo pode ser dito sobre números e tabelas. Visualizar os dados é fundamental para desenvolver um senso intuitivo sobre sua natureza, que guia nossas decisões sobre como abordá-los matematicamente. Isso é especialmente verdadeiro em big data, onde o volume de dados extrapola nossa capacidade analítica meramente numérica. A visualização é importante tanto na fase exploratória dos dados, quanto na interpretação dos resultados. Histogramas e boxplots são formas rápidas de entender a característica das variáveis de estudo e reconhecer dados anômalos. Problemas multidimensionais podem ser analisados visualmente usando técnicas de redução de dimensionalidade, que ainda são capazes de incorporar as relações entre as variáveis para indicar padrões e tendências. São muitas as técnicas voltadas à visualização, mas sem o devido conhecimento estatístico, o cientista de dados não saberá sua aplicabilidade e como extrair informações a partir destas poderosas ferramentas. (CECCON, Denny, 2020).

O que é python?

Python é uma linguagem de programação de alto nível — ou High Level Language —, dinâmica, interpretada, modular, multiplataforma e orientada a objetos — uma forma específica de organizar softwares onde, a grosso modo, os procedimentos estão submetidos às classes, o que possibilita maior controle e estabilidade de códigos para projetos de grandes proporções.

Por ser uma linguagem de sintaxe relativamente simples e de fácil compreensão, ganhou popularidade entre profissionais da indústria tecnológica que não são especificamente programadores, como engenheiros, matemáticos, cientistas de dados, pesquisadores e outros.

Um de seus maiores atrativos é possuir um grande número de bibliotecas, nativas e de terceiros, tornando-a muito difundida e útil em uma grande variedade de setores dentro de desenvolvimento web, e também em áreas como análise de dados, machine learning e IA. (ROVEDA, Ugo, 2013).

2.4. Aplicação das disciplinas estudadas no Projeto Integrador

No projeto, foram utilizados três tipos de estruturas de dados: DataFrame, dicionário e lista ligada. DataFrame é uma estrutura de dados bidimensional com os dados alinhados em forma de tabela (linhas e colunas), tamanho ajustável, semelhante a uma planilha no software Microsoft Excel. A diferença essencial é que, no DataFrame, nomes de colunas e os números de linha são conhecidos como índice de coluna e linha. Já a estrutura de dados conhecida como "dicionário" em Python nos permite acessar os dados por índices, ou seja, de forma mais intuitiva do que as listas. Trata-se de um grupo de dados onde relacionamos uma chave a um valor específico. Dessa forma, recuperamos um valor o acessando através de uma chave. Por fim, foi usada também a lista ligada. Uma lista encadeada ou lista ligada é uma estrutura de dados linear e dinâmica. Ela é composta por várias células que estão interligadas através de ponteiros, ou seja, cada célula possui um ponteiro que aponta para o endereço de memória da próxima célula.

2.5. Metodologia

Foram importadas as bibliotecas e feito os carregamentos dos arquivos através do numpy com objetivo maior de realizar algumas operações matemáticas dentro dos Arrays Multidimensionais. Uma base DataFrame com aproximadamente 20 colunas é mantida para o armazenamento dos dados em uma planilha, sendo que cada linha está associada há um índice, que serve para ordenar e selecionar dados endereçados dentro de sua estrutura, e com a informação do DataFrame é possível analisar a quantidade de memória utilizada, as variáveis, quantidade de linhas e registros encontrados na base.

Após localização do ambiente no local correto é retrabalhado e descrito suas variáveis em quais são úteis e quais são inúteis, e dado função correlação entre as variáveis numéricas através da análise de atributos, por fim, excluindo as linhas faltantes e discretizando as colunas com mesmo sentido.

Em análise técnica referindo-se ao mapeamento foi adicionado um dicionário para elevar o código texto em numérico, decidindo quem ganhará mais ou menos reescrevendo as variáveis e trocando a última coluna para valores binários.

A Variável país nativo começa com uma tabela de países e a primeira coisa que é preciso ser feito é remover as linhas com "?", que são dados faltantes para nossa tabela.

Já que não sabemos da origem desse dado, consequentemente ele não é um dado informativo e precisa ser excluído da tabela. Para isso usamos o código df.head para trocarmos o "?" por NaN e usaremos o código df.dropna para retirarmos os NaN da tabela.

Agora nós iremos transformar todas as pessoas que não forem dos Estados Unidos em 'Non-US' e para isso iremos usar o mesmo conceito do dicionário, vamos transformar em 0 ou em 1. ou seja, { 'US': 1, 'Non-US' :0}.

Na variável estado civil, temos 7 classes e iremos separarmos da seguinte forma, se a pessoa for divorciada, nunca foi casada, estiver separada ou for viúva iremos considerar que ela é solteira e será considerada (Single), se ela estiver casada no papel ou morando junto será considerado (Couple) e será separado na tabela a quantidade de incidência dentro de cada sub classe e será separado com o mesmo conceito do dicionário em 0 ou em 1.

Usaremos um código para separarmos a proporção de Couple e Single e separamos dentro da cada um, Pegamos a sub divisões e vemos se ela é Single ou Couple.

Começamos a variável relacionamento com 6 classes de relacionamento, e o primeiro passo é criar um dicionário com 5 classes que são igualmente relevantes, ou seja, Unmarried 0, Wife 1, Husband 2, Not-in-family 3, Own-child 4 e other-relative 5.

Em seguida usamos o código print para estabilizarmos as classes conforme mencionado acima e criar a coluna que são separados horas trabalhadas, e país nativo que estão separados, os que são de 'US' 1 e os divergentes 0, na tabela temos uma inconsistência nos nomes para relacionamentos, mas se olharmos no Spyder essa coluna se reajusta. E a partir disso aparecem outras classes de relacionamento que vão de 0 à 5 que se juntam com as classes igualmente relevantes.

Para a variável raça iremos começar usando o código print para separarmos em 5 classes, dentre elas temos separadas por classes as etnias amer-indian-eskimo 0, asian-pac-islander 1, black 2, white 3 e other 4 e separamos ela em uma coluna para obtermos os dados de quais raças temos na nossa tabela e separarmos as etnias para podermos visualizar as idades, horas trabalhadas, país nativo e classe de trabalho. E juntando esses dados temos uma separação de cada pessoa para sabermos as etnias de cada pessoa juntamente com outros dados coletados.

Para a variável ocupação da classe de trabalho nós criamos uma função para sabermos as funções de trabalho. Se é federal, municipal ou estadual o código irá retornar para a nossa tabela que ele é do governo, se ele for privado vai retornar que ele é privado e se for autônomo formal

ou informal vai constar como autônomo. E a partir disso vamos usar o código df.head para criarmos uma coluna ajustada sobre quais ocupações de cargos temos na nossa coluna, o próximo passo é implementarmos o código income group.by para obtermos os dados concretos da pesquisa e logo após podemos rodar o código df.head para vermos nossa tabela com a váriavel ocupação de classe de trabalho atualizada.

A variável capital gain tem a função de nós mostrarmos o quanto de capital cada cidadão recebe em média, e para vermos essa média iremos usar uma tabela que foi introduzida a partir de alguns códigos, como o x=df, plt.hist, pltxlabel. Assim que jogamos esse código iremos o programa nós mostra o gráfico que contém o ganho do histograma de distribuição sem considerar o salário e podemos ver que a variável é praticamente binária e com isso, iremos ao próximo código que é o df.loc[(df['capital.gain'] para fazermos a discretização em 0 ou 1, que funciona da seguinte forma, se o código for maior que 0 é 1 e se for igual a 0 é 0.

E após isso vamos rodar o código plt de novo e podemos observar no gráfico atualizado que iremos atualizar a tabela com os códigos que fizemos acima para obtermos o loss de cada cidadão e é mostrado a quantidade de loss (perda de capital).

Para a variável capital loss, iremos realizar a mesma abordagem do código acima para vermos a quantidade de capital loss (perda de capital) de cada cidadão e logo após vamos usar o código df.loc[(df['capital.loss'] para colocarmos a nossa variável capital loss para discretização das variáveis. Nessa variável usamos apenas os códigos introduzidos na variável capital gain pois conseguimos obter todos os dados necessarios para deixarmos nossa tabela com a quantidade correta de perda de capital de cada cidadão em média no mundo.

Vamos explicar a variável education ou variável de nível educacional, que é considerada uma variável não explicativa. Para começarmos, usaremos o código print(df[['education' , ' income']].groupby(['education']).mean()) e com isso é mostrado uma tabela com vários níveis educacionais e com a tabela já podemos ver quem se educou pouco com certeza a maioria deles acaba não tendo uma renda alta, ou seja, quem foi somente até os primeiros anos e não possui um bom nível educacional e na tabela vemos casos de pessoas que fizeram doutorado e são apenas 25%, e temos outros casos como o professor de escola possui um nível apenas de 25% também de pessoas que se inscrevem enquanto muitos nem terminam os primeiros anos e tem muita dificuldade em receber um salário de mais de 40 mil dólares e por isso que criamos a variável educação, para podermos introduzir a realidade de várias pessoas.

E vemos que essa variável é uma das poucas que são tão explicativas, ela mostra a realidade de vários cidadãos e faz uma média de salário para cada qual com base no nível

educacional e com a possibilidade bem maior de receber um salário fora do comum para muitos que não tem um nível educacional, mas devemos considerar que ela também é uma variável redundante, pois não consta a variável anos de educação.

A variável occupation é uma variável muito pulverizada, o que a torna pouco explicativa e ela tem a finalidade de nós mostrarmos os níveis de cargos com base na variável acima, ela vem para ser uma auto explicação dos dados que obtivemos na variável anterior e agora iremos fazer a parte de remoção de variáveis não explicativas conforme foi mencionado na variável education. Começamos usando o código df.drop(labels=['education' , 'occupation'],axis=1, inplace=True), esse código tem a função de mostrar a tabela da discretização das variáveis e a partir disso conseguimos observar que nosso código fez com que as variáveis education e occupation foram removidas por serem não explicativas e temos apenas 13 colunas agora, as outras 2 foram removidas com êxito.

Dentre as variáveis não-discretizáveis, temos as variáveis: hours per week (horas trabalhadas por semana), education num (número de anos na escola) e a variável age (idade). Essas variáveis são tratadas dessa forma por serem muito dispersas, sendo melhor deixá-las apenas numéricas.

Logo após, foi feita a construção dos classificadores. Para isso, primeiramente, foi realizada a importação do dataSplit, para que ocorresse a divisão dos dados. O conjunto de dados foi dividido da seguinte forma: 50% para treinamento, 20% para validação e 30% para teste. Depois disso, foi feita a importação dos classificadores. Após essa importação, foi criada uma lista ligada com os seguintes métodos: Nearest Neighbors, Naive Bayes, Decision Tree, Random Forest, Logistic Regression e SVC. Esses métodos são os modelos de treinamento. Posteriormente, foram importados os métodos de treinamento e as bibliotecas de avaliação.

Por fim, foi importada a função que realiza a avaliação dos resultados obtidos, que cria um arquivo de texto contendo essas informações.

2.6. Relatório de erros

Na criação do algoritmo que são utilizadas técnicas de machine learning para prever possíveis resultados, passamos por algumas barreiras e tivemos algumas dificuldades no meio do caminho.

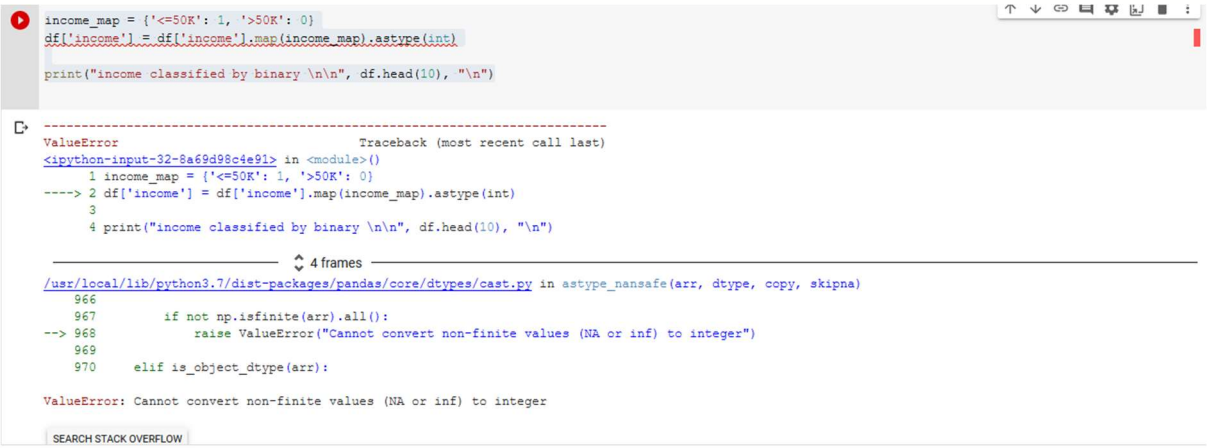
No bloco onde criamos o caminho para a importação dos planilha que está alocada no google drive, estávamos encontrando um erro onde a linha que montava os dados estava com a barra invertida

```
drive.mount("\content\drive")
```

Onde após alguns minutos analisando o código, percebemos que o erro estava nas barras invertidas dando prosseguimento com o código.

```
drive.mount("/content/drive")
```

No bloco de discretização da coluna income, onde utilizamos o dicionário para atribuir valores binários a maior dificuldade foi converter as colunas do "<=50K" e ">50k", o script não conseguia fazer a conversão para os inteiros 0 e 1, o collab apresentava o erro "Cannot convert non-finite values (NA or inf) to integer "(Não é possível converter valores não finitos (NA ou inf) em inteiros).

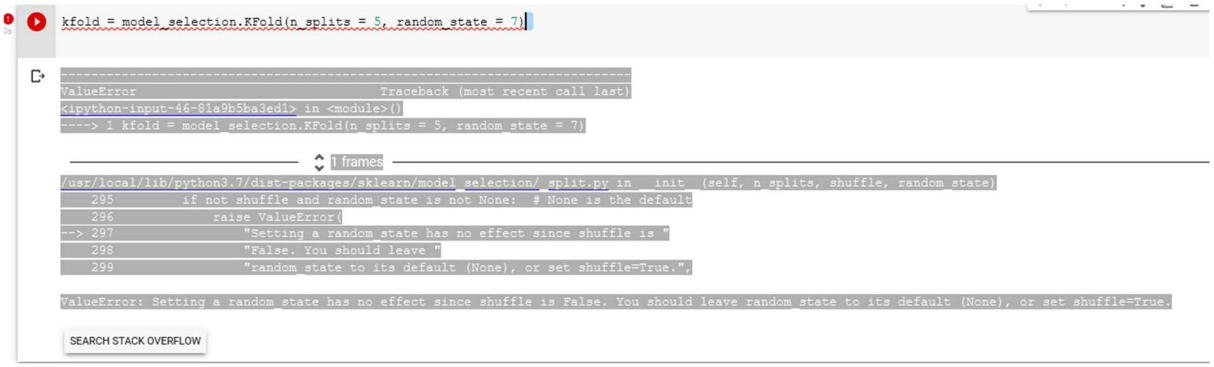


Com algumas pesquisas no google, conseguimos resolver esses problemas. Ao Invés de usar o nome do índice da planilha para fazer a conversão, como o dicionário dentro do .map(), ficando assim:

```
df['income'] = df['income'].map({'<=50K': 1, '>50K': 0}).astype(int)
```

para fazer a conversão dos dados finitos para dados inteiros.

No bloco do KFold, estamos com um erro onde não estamos achando uma solução para resolver o problema, já fizemos algumas buscas na web para o entendimento e solução do bloco, mas acabamos não tendo êxito a tempo de finalizar o código para apresentação.



Apesar de que tenhamos parado nesse problema, sabemos que o script está sendo utilizado para armazenar as variáveis de treino e teste para ser utilizados no treinamento do modelo.

3. RESULTADOS

Após todo o processo realizado no projeto, ao fim, os resultados finais obtidos foram avaliados por uma função criada no código, gerando assim, um arquivo de texto contendo todas as informações referentes ao resultado do estudo dos dados. Foram medidas a precisão e a acurácia de cada método na tentativa de estimar quem ganha mais ou menos de 50 mil dólares.

4. CONSIDERAÇÕES FINAIS

O objetivo inicial do trabalho, que era realizar um estudo em cima de uma base de dados, foi concluído com sucesso. Como foi mencionado, o estudo de dados é algo importante para tirar conclusões sobre dados coletados, a fim de entender ou esclarecer algo, possibilitando até uma tomada de decisão contra alguma problemática constatada após a análise de dados.

REFERÊNCIAS

FROES, Anderson. **Aprenda o que são Estrutura de Dados e Algoritmos - (Material Curso Dio)**. [S. l.], Fevereiro 2021. Disponível em: https://digitalinnovation.one/artigos/aprenda-o-que-sao-estrutura-de-dados-e-algoritmos-material-curso-dio. Acesso em: 2 dez. 2021.

FIGUEIREDO, Vinícius. **Seus primeiros passos como Data Scientist: Introdução ao Pandas!**. [S. l.], Maio 2018. Disponível em: https://medium.com/data-hackers/uma-introdução-simples-ao-pandas-1e15eea37fa1. Acesso em: 2 dez. 2021.

PESSOA, Willian. **Estruturas de Dados IV: Mapeamento (Dicionários).** [S. l.], Maio 2019. Disponível em: https://medium.com/reflexão-computacional/estruturas-de-dados-iv-mapeamento-dicionários-53f4d75bd7d7. Acesso em: 2 dez. 2021.

RICARTE, Ivan. **Listas ligadas**. [S. l.], Fevereiro 2003. Disponível em: https://www.dca.fee.unicamp.br/cursos/EA876/apostila/HTML/node27.html. Acesso em: 2 dez. 2021.

CECCON, Denny. **A importância da Estatística para a Ciência de Dados**. [S. l.], Setembro 2021. Disponível em: https://iaexpert.academy/2020/09/14/importancia-da-estatistica-para-ciencia-de-dados/. Acesso em: 2 dez. 2021.

ROVEDA, Ugo. **O que é Python, para que serve e por que aprender?**. [S. l.], Outubro 2013. Disponível em: https://kenzie.com.br/blog/o-que-e-python/. Acesso em: 2 dez. 2021.