

**INSTITUTO FEDERAL DE CIÊNCIA EDUCAÇÃO E TECNOLOGIA DO ESPÍRITO
SANTO - IFES**

**MANUAL DE INSTRUÇÕES
DO SOFTWARE
TOOLBOX_AM**

São Mateus

2022

Sumário

1. Seção 1
 - 1.1. Instalando o Python no Windows
 - 1.2. Como instalar os módulos necessários pelo cmd
2. Seção 2
 - 2.1. Como usar o produto
 - 2.2. Como adicionar e desabilitar opções na Base de Dados
 - 2.3. Escolhendo o tipo de problema
 - 2.4. Como incluir novos Métodos de aprendizado de máquina
 - 2.5. Configurações do Métodos de aprendizado de máquina
 - 2.6. Escolha da Validação Cruzada
 - 2.7. Outras configurações do Experimento
 - 2.8. Métricas
 - 2.9. Como adicionar novas Métricas
 - 2.10. Resultados
 - 2.11. Como adicionar novas opções de Apresentação de Resultados
 - 2.12. Executando e Salvando o experimento desejado

1- Seção 1

1.1- Instalando o Python no Windows

Para este manual considere que a versão do Python instalada é a seguinte:

Python 3.8.10 (64-bit)	Python Software Foundation	02/08/2021	101 MB	3.8.10150.0
Python Launcher	Python Software Foundation	02/08/2021	1,79 MB	3.8.7427.0

O primeiro passo é clicar neste link para baixar o Python (para Windows):

<https://www.python.org/ftp/python/3.8.10/python-3.8.10-amd64.exe>

Ao executar o arquivo baixado, a seguinte janela irá abrir:



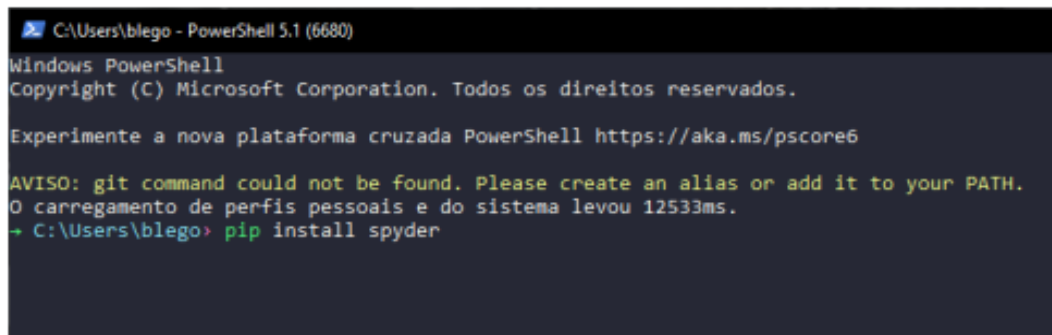
Figura 1: captura de tela do instalador de Python para Windows.

Obs: Existem outras formas de se instalar o Python no Windows. Esta é apenas uma sugestão.

Guarde o local de instalação, pois será necessário posteriormente. Marque a opção “Add Python 3.8 to PATH” e instale clicando em “Install Now”. Se quiser apenas clique nos botões “Next”, mas é sugerido conferir se o instalador não está instalando algo que você não deseja (anúncios, programas indesejados etc).

Para instalar uma IDE (ambiente de programação, similar ao Dev C++, etc) clique com o botão direito no mouse no menu iniciar, e procure pela opção Windows Powershell (pode ser que no seu apareça Prompt de Comando). Nesse exemplo, aparece a opção Windows PowerShell. Abra-o e instale um pacote chamado “spyder”, que é uma das IDEs. Caso não deseje instalar o spyder, instale outra IDE de sua preferência. Note que não é necessário

instalar uma IDE para que o *Toolbox* funcione, já que é possível executar o script diretamente pelo Powershell.



```
C:\Users\blego - PowerShell 5.1 (6680)
Windows PowerShell
Copyright (C) Microsoft Corporation. Todos os direitos reservados.

Experimente a nova plataforma cruzada PowerShell https://aka.ms/pscore6

AVISO: git command could not be found. Please create an alias or add it to your PATH.
O carregamento de perfis pessoais e do sistema levou 12533ms.
C:\Users\blego> pip install spyder
```

Figura 2: captura de tela do instalador do Prompt de Comando do Windows.

O software “pip” irá instalar vários arquivos e pacotes para instalar o *spyder*, e deve demorar um pouco para terminar de executar o comando da Figura 2. Ele é um gerenciador de pacotes do python (similar a uma playstore, applestore, microsoft store). O Python é uma linguagem de programação bastante utilizada por ser bem fácil, e várias pessoas fazem seus “programas” e disponibilizam eles em módulos neste software “pip”.

No C, nós incluímos algo similar a um pacote com “#include”, por exemplo. Aqui nós instalaremos pacotes com o pip e utilizaremos eles com “import pacote”. No caso do spyder, não utilizaremos ele como pacote, mas sim como um software isolado..

Quando terminar de instalar, vamos na pasta que o python foi instalado. Um exemplo seria “C:\Users\blego\AppData\Local\Programs\Python\Python38” (A janela da Figura 1 mostra esse caminho), então iremos para essa pasta no navegador de arquivos. Nesta pasta, entre na pasta Scripts, e procure por “Spyder.exe”. Adicione este ao menu iniciar ou envie para a área de trabalho como um atalho para facilitar o acesso.

Depois de criar o atalho, abra o spyder. Ele terá a aparência mostrada na Figura 3.

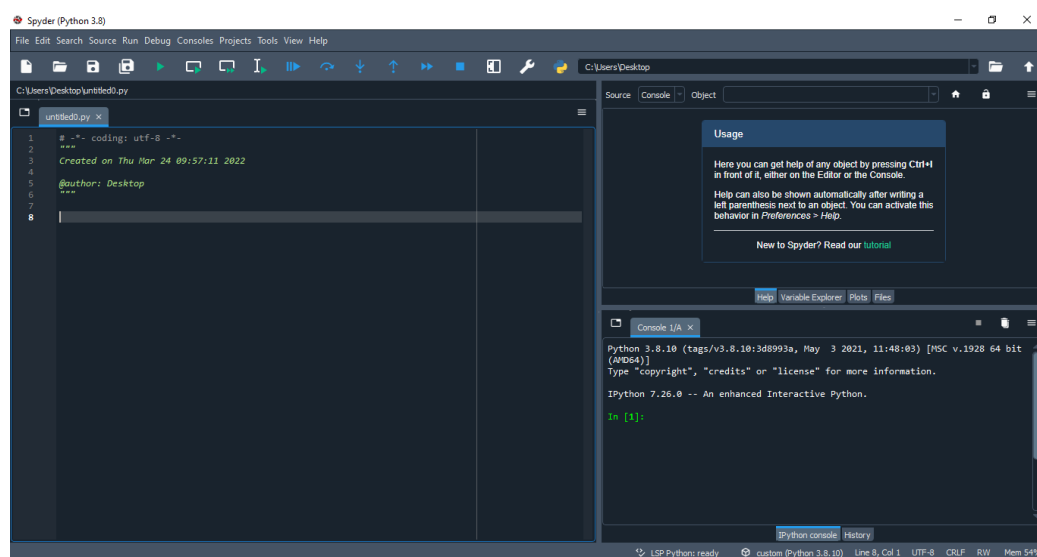


Figura 3: captura de tela da interface do Spyder.

Na parte da esquerda, está o local onde se digita código, ou seja, o editor de texto. Na direita (superior) temos o menu de ajuda, o explorador de variáveis, o visualizador de plots e o gerenciador de arquivos. A aba mais importante aqui é o explorador de variáveis. Já na parte direita inferior está o nosso interpretador de Python.

1.2 - Como instalar módulos do Python

Para instalar um módulo do python basta abrir o CMD, digitando essas letras no menu iniciar que logo aparece o ícone, como mostra na imagem logo abaixo:

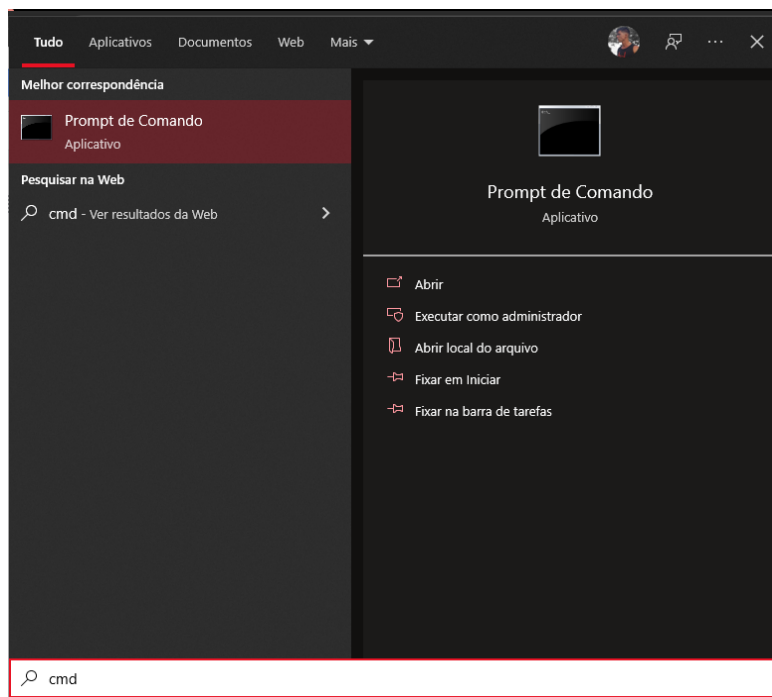


Figura 4: captura de tela para pesquisar o prompt de comando do Windows.

Este processo é similar ao processo de instalação da IDE Spyder. Após clicar no ícone basta digitar na tela “ pip install nome do módulo “ que ele será instalado em alguns segundos. Veja o exemplo para o caso de instalar o módulo “numpy”.

```
C:\Users\Desktop>pip install numpy
Collecting numpy
  Downloading numpy-1.22.3-cp38-cp38-win_amd64.whl (14.7 MB)
    | 14.7 MB 6.4 MB/s
Installing collected packages: numpy
Successfully installed numpy-1.22.3
WARNING: You are using pip version 21.1.1; however, version 22.0.4 is available.
You should consider upgrading via the 'c:\users\desktop\appdata\local\programs\python\python38\python.exe -m pip install --upgrade pip' command.
C:\Users\Desktop>
```

Figura 5: captura de tela após instalar o módulo “numpy”.

Faça o mesmo para as seguintes bibliotecas:

1. pygubu==0.14 (ex.: pip install pygubu==0.14)
2. pandas==1.4.2
3. matplotlib==3.5.2 ou 3.4.3

2- Seção 2

2.1 - Como Usar o produto na Base de Dados

Para todas as colunas o usuário tem acesso a opções que contém informações que são agrupadas seguindo padrão determinado.

A primeira coluna do *software* mostra a lista das bases de dados que o usuário pode escolher para realizar seus testes. Cada base de dados armazena informações sobre algum problema, que pode ser do mundo real ou de simulação, que se deseja analisar. A base de dados pode se referir apenas a problemas de regressão ou problemas de classificação. Na segunda coluna do software, há uma caixa de seleção para escolher um desses dois tipos de problema. Ao escolher, as bases de dados do outro tipo de problema serão automaticamente desabilitadas (isto é, não poderá ser selecionada), como mostra a imagem abaixo (“Pupila” está desabilitada).

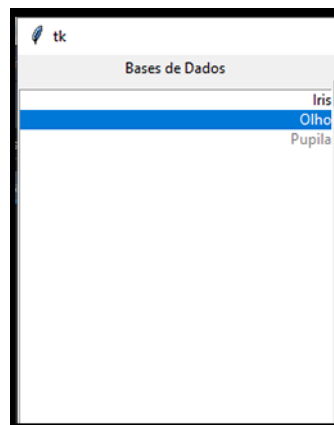


Figura 6: captura de tela da interface no segmento Base de Dados.

2.2 - Como adicionar e desabilitar opções na Base de Dados

Caso deseje, o usuário pode incluir novas bases de dados a seu critério, desde que a inclua seguindo o padrão de bases de dados definido pelo programa. Para adicionar uma base de dados, o usuário deve fazer o seguinte (para facilitar o entendimento, chamaremos a base de dados a ser adicionada de BD).

1. Criar uma pasta, dentro da pasta BaseDeDados, com o nome da base de dados a ser adicionada. Sugerimos utilizar a primeira letra maiúscula. Por exemplo: “BD”
2. A pasta criada no passo (1), isto é, “BD”, deve ter **no mínimo** um arquivo Python com o mesmo nome da base de dados. Isto é, “BD.py”.
3. Este arquivo .py criado deve definir, PELO MENOS, duas funções:
 - a. `read()` -> retorna uma tupla de 2 ou 4 elementos. Caso sejam dois, o primeiro elemento deve representar as características da base de dados armazenadas uma *array* da biblioteca *numpy*, enquanto o segundo elemento deve representar os alvos do problema de classificação ou regressão, também armazenadas uma *array* da biblioteca *numpy*. Caso seja de Classificação, sugere utilizar o chamado *one-hot-encoding*.
 - b. `is_regression()` -> retorna “True” ou “False”. Caso seja uma base de dados de regressão, deve retornar o valor lógico True, e caso seja uma base de dados de classificação, deve retornar o valor lógico False .

Caso o usuário queira outras opções além das oferecidas, ele pode consultar os exemplos de códigos que foram incluídos no produto e modificar de acordo com seu objetivo.

2.3 - Escolhendo o tipo de problema

Na segunda coluna do *software*, dois elementos estão disponíveis. O primeiro é uma caixa de seleção onde o usuário decide entre classificação e regressão, como mostra a imagem abaixo, dependendo de qual problema de aprendizado de máquina ele quer tratar. Dependendo da opção que ele escolher, as opções disponíveis em algumas colunas são alteradas drasticamente.

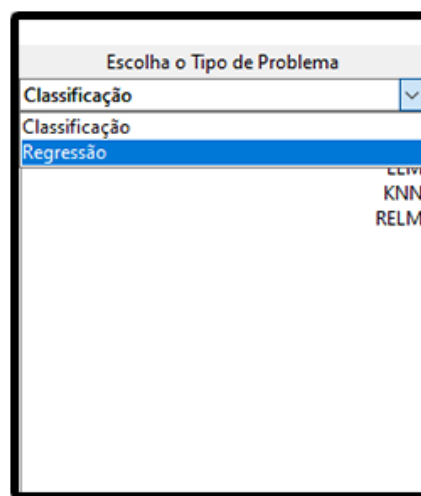


Figura 7: captura de tela da interface no segmento Escolha o Tipo de Problema.

O segundo elemento da coluna é uma lista de métodos de aprendizado de máquina disponíveis para utilização no *software*. Na imagem abaixo, por exemplo, todos os métodos disponíveis podem ser utilizados para problemas de classificação. Caso algum método não for adequado para a tarefa escolhida, ele seria impossibilitado de ser selecionado, assim como a base de dados “Pupila” na Figura 6.

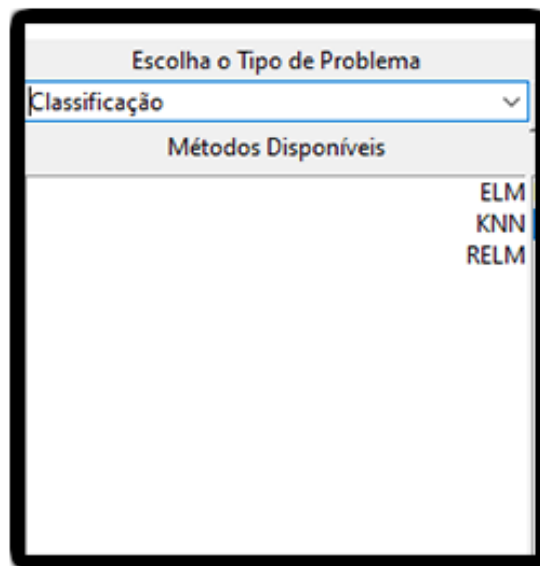


Figura 8: captura de tela da interface no segmento Escolha o Tipo de Problema, quanto à classificação.

2.4 - Como incluir novos métodos de aprendizado de máquina

Caso deseje, o usuário pode incluir novos métodos a seu critério, desde que o inclua seguindo o padrão definido pelo programa. Para adicionar um método, o usuário deve fazer o seguinte (para facilitar o entendimento, chamaremos o método de “MT”)

1. Criar um arquivo “mt.py” - sugerimos utilizar o nome do método todo em letra minúscula - dentro da pasta “MetodosDisponiveis”.
2. Este arquivo “mt.py” **deve atender, no mínimo**, as seguintes restrições
 - a. deve definir uma classe, com nome do método utilizando letras maiúsculas. Ex.: “MT”
 - b. Essa classe deve estender a classe “_util.Util”. O script “util” está incluído na pasta de métodos disponíveis, e ele define a classe Util.
 - i. Ex.: “class ELM(util.Util)”
 - c. A classe do método **deve implementar, no mínimo**, os seguintes métodos:
 - i. `train(self, X,Y)` - recebe X (matriz de características) e Y (matriz de alvos), e treina o modelo de aprendizado de máquina. O modo de treinar depende da técnica, obviamente
 - ii. `predict(self, X)` - recebe X (matriz de características) e tenta prever, através do conhecimento aprendido pela técnica, os alvos associados às amostras de entrada.
 - d. A classe também deve definir dois atributos booleanos, herdados com valor True da classe _util.Util:
 - i. `self.isRegressionMethod = True/False` (True se o método pode ser utilizado para problemas de regressão e False caso contrário)
 - ii. `self.isClassificationMethod = True/False` (True se o método pode ser utilizado para problemas de classificação e False caso contrário)
 - e. A classe também deve definir os hiperparâmetros que poderão ser modificados pelo software. Para isso, existe uma variável

- “self._accepted_params” que contém uma lista os nomes de tais variáveis. Ao criar um novo método, estenda essa lista, adicionando as novas variáveis.
- f. (OPCIONAL, mas importante) Adicione uma breve explicação dos hiperparâmetros adicionados na letra (e) ao dicionário de hiperparâmetros. Ou seja, para cada parâmetro adicionado, faça o seguinte:
 - i. self._accepted_params_docs[NOME_PARAMETRO] = DESCRIÇÃO

E também, caso o usuário queira outras opções além das oferecidas, ele pode consultar os exemplos de códigos que foram inclusos no produto e modificar de acordo com seu objetivo.

2.5 - Configurações dos Métodos de Aprendizado de Máquina

A terceira coluna do *software* é composta de três partes. A primeira parte se refere aos hiperparâmetros dos Métodos, que consiste em um botão chamado “Configurações dos hiperparâmetros”. Ao clicar neste botão, vai para a interface mostrada na imagem abaixo, que é uma janela dinâmica composta de vários campos referentes aos hiperparâmetros dos métodos selecionados na coluna anterior (apenas dos que foram selecionados - na imagem abaixo, por exemplo, os métodos selecionados foram o ELM e o RELM). Ao passar o mouse em cima dos campos, o usuário pode receber informações sobre o preenchimento do respectivo parâmetro, como mostrado na figura 9.

Figura 9: captura de tela da interface na parte de Configurações de hiperparâmetros.

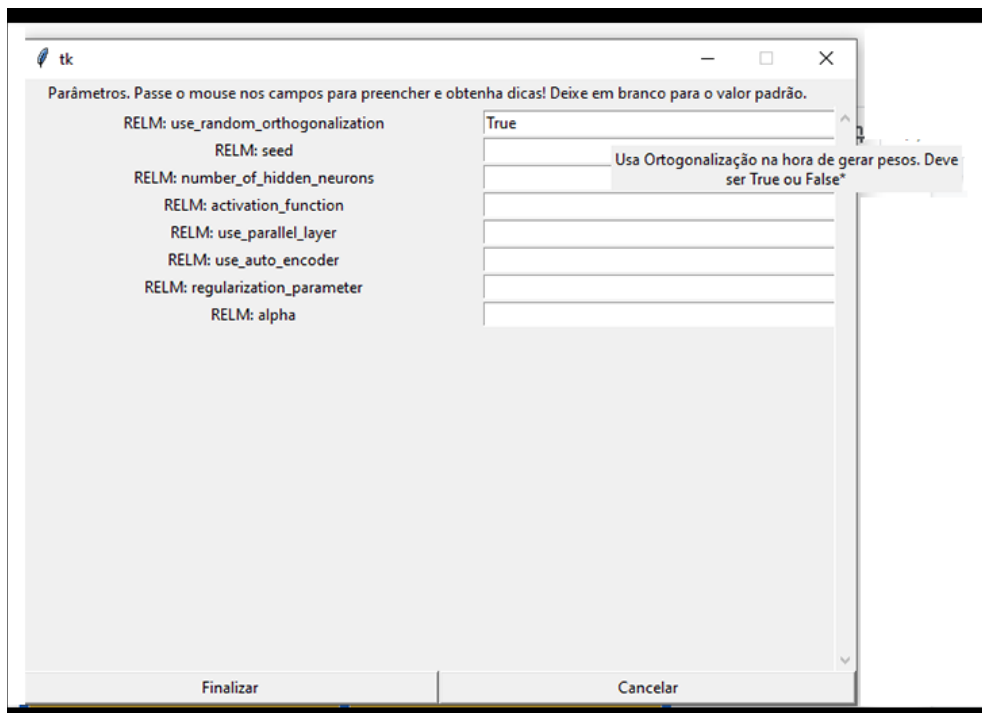


Figura 10: captura de tela da interface na parte das Dicas de Configurações de hiperparâmetros.

Na tela das Figuras 8 e 9, são apresentados os locais apropriados para que uma pessoa possa configurar os métodos escolhidos na coluna anterior através dos hiperparâmetros listados. No caso mais simples, o usuário pode digitar apenas um valor (que segue o padrão mostrado na Dica). Mas caso o usuário deseje testar mais de um valor (por exemplo, testar dois valores e escolher aquele que se mostra mais adequado), basta apenas preencher os valores desejados separados por vírgula simples. Exemplo, se o usuário quiser testar dois valores de “alpha”, como 0 e 1, ele pode digitar apenas “0, 1”.

2.6 - Escolha da Validação Cruzada

A segunda parte desta coluna é uma lista de métodos de validação que o usuário pode selecionar, como mostra a imagem abaixo. Dois exemplos de validação são o Holdout e o Kfold. Nesta lista, apenas um método pode ser selecionado.

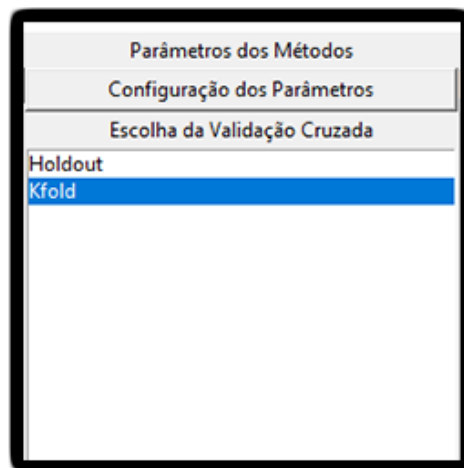


Figura 11: captura de tela da interface no segmento quanto aos hiperparâmetros dos Métodos.

2.7 - Outras configurações do experimento

A terceira parte da terceira coluna consiste em campos de configuração do experimento. Três das configurações disponíveis nesta versão se referem a divisão do conjunto de dados para escolha dos hiperparâmetros. Num primeiro momento, o conjunto é dividido entre treino e teste (e validação, se necessário) sem repetição das amostras. O conjunto de teste é separado, e o conjunto de treino (e o de validação) são utilizados no experimento para escolha dos hiperparâmetros dos métodos. Como esses hiperparâmetros se referem a divisão do conjunto de dados, a soma deles deve ser 100%, dependendo de qual validação cruzada for escolhida na parte anterior. Se o "holdout" estiver selecionado, os três hiperparâmetros abaixo (Dados para Treino, Dados para Validação e Dados para Teste) devem somar 100. Se "Kfold" for escolhido, apenas Dados para Treino e Dados para Teste devem somar 100 (O parâmetro "% Dados para validação" será ignorado)

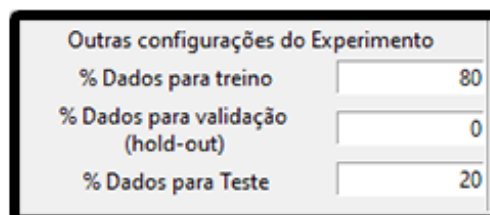


Figura 12: captura de tela da interface no segmento de Outras configurações de Experimento.

Ainda nesta parte da terceira coluna, existem dois campos que podem ser preenchidos: Número de Repetições e Semente Aleatória inicial. O primeiro campo se refere ao número de vezes que o teste deve ser repetido - e o resultado final a ser mostrado é a média dos resultados de cada repetição. Já o segundo campo se refere a semente aleatória que deve ser utilizada nos testes. Cada teste envolve números aleatórios (usados para dividir as bases de dados em dois conjuntos distintos, ou ainda em outros trechos do *software*) - e este parâmetro controla esses números, tornando possível a reprodução dos testes.

Número de Repetições	5
Semente Pseudoaleatória Inicial	12345

Figura 13: captura de tela da interface quanto a configurações opcionais do experimento.

2.8 - Métricas

Na quarta coluna do *software*, três elementos estão disponíveis. O primeiro é uma lista onde o usuário deve selecionar quais métricas serão consideradas em seus experimentos. Na segunda coluna do software, há uma caixa de seleção para escolher entre problemas de Classificação ou de Regressão. Ao escolher, as métricas comumente utilizadas para o outro tipo de problema serão automaticamente desabilitadas (isso é, não poderá ser selecionada), como mostra a imagem abaixo (“RMSE” e outras estão desabilitadas).

Métricas	
<input checked="" type="checkbox"/>	ACCURACY
<input type="checkbox"/>	ARRMSE
<input type="checkbox"/>	MAPE
<input type="checkbox"/>	RMSE
<input type="checkbox"/>	RRMSE
<input type="checkbox"/>	SMAPE

Figura 14: Captura de tela da interface no segmento Métricas.

2.9 - Como adicionar novas opções de métricas

Caso o usuário deseje adicionar novas opções de métricas ele precisa criar o arquivo “.py” com letras minúsculas e a classe deve receber esse mesmo nome, mas em letras maiúsculas. Ex.: se o arquivo tem o nome de “teste_aleatorio.py” a classe deve ter o nome de “TESTE_ALEATORIO”. Cada Métrica deve ter 4 funções que são:

- **worst_case:** É o pior caso para alguma métrica. Geralmente é 0;
- **is_regression_metric:** pode ser True ou False. Por padrão, o estado é “True”;
- **calculate:** Calcula a métrica de fato;
- **is_better:** Compara dois valores de métricas e retorna qual for melhor para o caso.

Para mais informações, o usuário pode consultar as funções já incluídas no *software* para ver o que elas fazem.



Figura 15: Exemplo de como deve ser inserida a classe e o nome do arquivo “.py” respectivamente.

2.10 - Seção de Apresentação de Resultados

A segunda parte da quarta coluna contém uma lista onde o usuário pode escolher como deseja que os seus resultados sejam apresentados. Na apresentação dos resultados pode ser gerado gráficos ou imagens dependendo de como foi feita a análise.

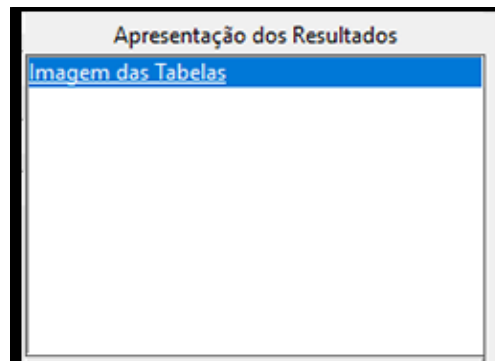


Figura 16: Captura de tela da interface no segmento de Apresentação dos Resultados.

2.11 - Como adicionar novas opções de Apresentação de Resultados

Para incluir novas opções o usuário precisa criar um arquivo do tipo “nome.py” e aloca-lo dentro da pasta “ApresentacaoResultados”. Além disso, a classe deve ter o nome em letras maiúsculas e as funções “getName” e “generate” (responsável por criar a imagem ou gráfico). Na imagem abaixo mostra um exemplo em que apenas a imagem das tabelas é como opção.

A função *generate* deve receber uma única variável, que é um DataFrame (estrutura de dados do pacote pandas, do Python) contendo os resultados obtidos no experimento realizado. A função não necessita retornar nada, podendo apresentar os resultados de diversas formas, como abrindo uma figura, abrindo um popup com os resultados, criando um arquivo de uma planilha no excel com os resultados salvos, entre outras opções. Para mais informações, o usuário pode consultar as funções já incluídas no *software* para ver o que elas fazem.

2.12 - Executando e Salvando o Experimento Desejado

Após ter selecionado todas as opções necessárias, o usuário pode ter a opção de iniciar o experimento de uma vez ou ainda ele pode carregar configurações já feitas anteriormente. E por fim, pode salvar as configurações para acessos futuros, em que será criado um arquivo de backup.

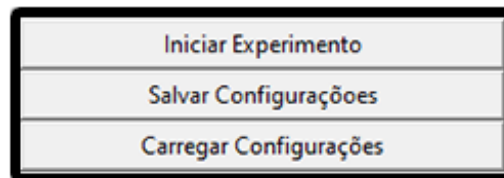


Figura 17: Captura de tela da interface na parte dos principais botões.

A quarta coluna ainda possui uma barra de progresso, como mostra a Figura 18, que é preenchida à medida com que as etapas do experimento iniciado são executadas.

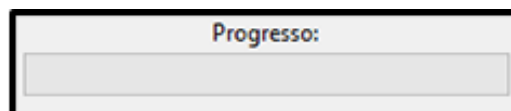


Figura 18: Captura de tela da interface na parte da barra de progresso antes de iniciar o experimento.

Tendo concluído o progresso a barra será preenchida, como mostra a Figura 19. Além disso, as opções de resultados escolhidas na seção de "Apresentação de Resultados" serão criadas ou mostradas na tela.

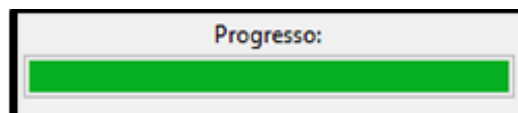


Figura 19: Captura de tela da interface na parte da barra de progresso depois de iniciar o experimento.