

Olinda 04 de Março de 2024

Edson Henrique Ferreira da Silva

Estudante de Análise e Desenvolvimento de Sistemas

<https://github.com/86HenriqueSilva>

<http://linkedin.com/in/henriquesilvatech>

[Meu Cv](#)

Projeto de Análise e Tratamento de Dados “ [BASE](#) “

"DeepMind Analytica"

“Pode ser traduzido como "Análise Profunda da Mente". Essa sugestão combina "Deep Learning" (Aprendizado Profundo), sugerindo o uso de técnicas avançadas de aprendizado de máquina, com "Analytica" (Análise), indicando o foco na análise de dados. A ideia é transmitir a profundidade e sofisticação da análise de dados usando técnicas avançadas de aprendizado profundo”

O código [BASE](#) compartilhado pode ser dividido em várias etapas. Vamos identificar essas etapas:

1. Importação de Bibliotecas:

Importação das bibliotecas necessárias para manipulação de dados, como pandas para dataframes, numpy para operações numéricas, StringIO para leitura de dados em formato CSV, e outras bibliotecas para manipulação de datas e logging.

2. Configuração do Logging:

Configuração básica para o módulo de registro (logging), permitindo que mensagens informativas, de aviso ou de erro sejam registradas durante a execução do script.

3. Definição da Classe AtualizadorDados:

Declaração da classe que encapsula toda a lógica de atualização dos dados de sorteios.

Criação de uma classe para encapsular toda a lógica relacionada à atualização de dados de sorteios. Isso ajuda a organizar e estruturar o código de maneira modular.

4. Inicialização da Classe:

Configuração inicial da classe com o caminho do arquivo de dados, nomes de colunas e tabela de grupos.

No método `__init__`, são definidos atributos iniciais da classe, como o caminho do arquivo de dados, os nomes das colunas e uma tabela de grupos relacionada aos resultados dos sorteios.

5. Métodos Privados de Leitura e Carregamento de Dados:

`_ler_arquivo_csv` e `_carregar_dados_csv` para ler e carregar os dados do arquivo CSV.

`_ler_arquivo_csv` lê o conteúdo do arquivo CSV.

`_carregar_dados_csv` carrega os dados do arquivo CSV para um dataframe pandas.

6. Métodos Privados de Validação:

`_validar_data_formato` e `_validar_data_concurso` para validar o formato e a validade da data.

`_validar_data_formato` verifica se a data está no formato correto.

`_tratar_erro_valor_invalido` trata erros relacionados à entrada de valores inválidos.

`_validar_data_concurso` valida se a data do concurso é uma quarta-feira ou sábado.

7. Métodos Privados de Manipulação de Dados:

`_tratar_erro_valor_invalido`, `_obter_bicho`, `_obter_nova_linha`, `_atualizar_dados`, `_preencher_MC_e_D`, `_preencher_GP`, `_preencher_BICHO` para diferentes manipulações e preenchimentos dos dados.

Métodos como `_obter_bicho`, `_obter_nova_linha`, `_atualizar_dados`,

`_preencher_MC_e_D`, `_preencher_GP` e `_preencher_BICHO` realizam manipulações específicas nos dados de sorteios.

8. Método `_auto_completar`:

Realiza o preenchimento automático de diversas colunas, como 'MC 1º a D 5º', 'GP 1º a GP 5º' e 'BICHO 1º a BICHO 5º'.

9. Método `_excluir_concurso`:

Permite excluir um concurso específico.

10. Método `menu_opcoes`:

Implementa um menu interativo que permite escolher entre diferentes opções de atualização de dados, exclusão de concursos, auto preenchimento e saída do programa.

11. Execução Principal (`if name == "main":`):

Configuração do caminho do arquivo, inicialização do objeto `AtualizadorDados`, configuração do formato de exibição, e chamada do método `menu_opcoes` para começar a interação com o usuário.

Essas etapas ajudam a entender a estrutura do código e suas funcionalidades. Cada método e seção desempenha um papel específico na execução geral do script. Cada etapa desempenha um papel específico na funcionalidade do script, permitindo a leitura, manipulação e atualização dos dados de sorteios. Se precisar de mais detalhes sobre alguma etapa específica ou tiver perguntas adicionais, estou à disposição!

Cada módulo (ou método) tem uma função específica. No entanto, ao avaliar a complexidade, é importante considerar a extensão e a quantidade de responsabilidades atribuídas a cada parte do código.

Vamos analisar os módulos principais e sua complexidade relativa:

1. menu_opcoes (Método da Classe):

- Este método envolve a lógica principal do programa, gerenciando as opções do menu interativo. Pode-se considerar complexo devido à variedade de funcionalidades que abrange.

2. _auto_completar (Método Privado):

- Este método preenche automaticamente várias colunas com base nos dados existentes. Sua complexidade está relacionada à manipulação e transformação dos dados.

3. _atualizar_dados (Método Privado):

- Este método trata da adição de novos dados de sorteios. Envolve interação com o usuário para obter informações e manipulação do dataframe.

4. _obter_nova_linha (Método Privado):

- Este método cria uma nova linha de dados para um concurso específico, envolvendo validação de datas e entrada de resultados.

5. _excluir_concurso (Método Privado):

- Este método lida com a exclusão de um concurso específico do conjunto de dados, incluindo interação com o usuário para obter informações.

Em termos de complexidade, o método `menu_opcoes` pode ser considerado mais complexo devido à sua natureza abrangente, que gerencia várias funcionalidades do programa. No entanto, a complexidade é relativa, e cada método desempenha um papel importante em diferentes aspectos da execução do script.

Vamos falar de forma a explicar detalhadamente o método `_auto_completar`. Este método é responsável por realizar o preenchimento automático de várias colunas específicas do dataframe com base nos dados existentes.

Vamos analisar cada parte do código:

Carregamento de Dados:

```
dados = self._carregar_dados_csv()
```

O método começa carregando os dados do arquivo CSV usando o método privado `_carregar_dados_csv`.

Verificação de Dados:

```
if dados is not None and not dados.empty:
```

Verifica se os dados foram carregados com sucesso e se o dataframe não está vazio.

Conversão da Coluna 'Data':

```
dados['Data'] = pd.to_datetime(dados['Data'])
```

Converte a coluna 'Data' para o tipo `datetime` para facilitar manipulações futuras.

Obtenção do Dia da Semana:

```
dados['D/SEMAN'] = dados['Data'].apply(
    lambda x: format_date(x, 'EEEE', locale=locale) if
pd.notnull(x) else ''
)
```

Obtém o dia da semana em português usando a biblioteca `babel` e preenche a coluna 'D/SEMAN'.

Preenchimento das Colunas 'MC 1° a D 5°':

```
self._preencher_MC_e_D(dados)
```

Chama o método privado `_preencher_MC_e_D` para preencher as colunas 'MC 1° a MC 5°' e 'D 1° a D 5°'.

Preenchimento das Colunas 'S 1° a S 5°':

```
for i in range(1, 6):
coluna_premio = f'{i}° PRÊMIO'
coluna_s = f'S {i}°'
dados[coluna_s] = dados[coluna_premio].apply(
    lambda x: f'[{",".join(str(x).zfill(4))}]' if
pd.notnull(x) else '[0000]'
)
```

Utiliza um loop para preencher as colunas 'S 1° a S 5°' com os resultados formatados.

Preenchimento das Colunas 'GP 1° a GP 5°':

```
self._preencher_GP(dados)
```

Chama o método privado `_preencher_GP` para preencher as colunas 'GP 1° a GP 5°'.

Preenchimento das Colunas 'BICHO 1° a BICHO 5°':

```
self._preencher_BICHO(dados)
```

Chama o método privado `_preencher_BICHO` para preencher as colunas 'BICHO 1º a BICHO 5º'.

Salvamento dos Dados Atualizados:

```
dados.to_csv(self.caminho_arquivo, index=False)
```

Salva os dados atualizados no arquivo CSV.

Mensagens de Log:

```
logging.info("Colunas 'MC 1º a D 5º', 'S 1º a S 5º', 'GP 1º a GP 5º' e 'BICHO 1º a BICHO 5º' preenchidas com sucesso!")
```

Registra uma mensagem de log indicando que as colunas foram preenchidas com sucesso.

Tratamento de Erro:

```
else:
```

```
    logging.error("Não foi possível carregar os dados para auto completar.")
```

Caso não seja possível carregar os dados, registra uma mensagem de erro.

Espero que isso ajude a entender o funcionamento detalhado do método `_auto_completar`.

Bem-vindo ao Centro de Descobertas!

Seja você um explorador experiente de dados ou alguém iniciando sua jornada de descoberta, este projeto é sua bússola para novas possibilidades e insights. Aqui, a magia acontece nas linhas de código, transformando dados em conhecimento.

Esta fonte é mais do que uma ferramenta de extração; é o portal para uma jornada fascinante. Cada linha conta uma história, e cada número oculta um segredo esperando para ser revelado.

Sinta a emoção do desconhecido, mergulhe na vastidão dos dados, e que cada análise seja uma nova aventura. Este é o início de uma jornada repleta de aprendizado, crescimento e conquistas!

Você não está apenas acessando dados; está abrindo as portas para insights extraordinários. Prepare-se para descobrir, inovar e transformar!

Explore. Descubra. Transforme.

Bem-vindo a bordo!

