Aqui está a **atualização do documento do projeto** incorporando as mudanças feitas na **versão v2**:

**Projeto: Análise e Previsão de Sorteios da Lotofácil**

**1. Objetivo**

Criar um sistema capaz de ler arquivos CSV/XLSX com históricos de sorteios da Lotofácil, realizar análises estatísticas e predições baseadas em inteligência artificial e simulações matemáticas para sugerir combinações otimizadas para futuras apostas.

**2. Estrutura do Projeto**

📂 lotofacil-analise  
┣ 📜 analise\_lotofacil.py → Código principal  
┣ 📜 dados.py → Funções para carregar e limpar dados  
┣ 📜 estatisticas.py → Análises estatísticas dos sorteios  
┣ 📜 predicao.py → Modelagem preditiva e Machine Learning  
┣ 📜 gerador\_jogos.py → Geração dos jogos recomendados  
┣ 📜 banco.py → Gerenciamento de grupos de apostas salvas  
┣ 📜 painel.py → Interface gráfica interativa com Streamlit  
┣ 📂 data/ → Pasta onde o arquivo Lotofácil.xlsx é armazenado

**3. Configuração do Ambiente**

Antes de rodar o projeto, instale as bibliotecas necessárias usando:

pip install pandas numpy matplotlib seaborn scikit-learn streamlit sqlite3

**4. Carregamento e Preparo dos Dados**

📜 **dados.py**  
Agora, os dados dos sorteios são carregados corretamente com tratamento especial para análise preditiva e estatística.

**5. Estatísticas Básicas**

📜 **estatisticas.py**  
A função **obter\_estatisticas()** agora retorna:  
✅ Total de sorteios analisados  
✅ Último sorteio registrado  
✅ Números mais frequentes  
✅ Média de acertos por sorteio

**6. Modelagem e Predição**

📜 **predicao.py**  
✅ O usuário pode escolher entre **RandomForest ou MLP** para prever os números mais prováveis.  
✅ Agora a simulação de Monte Carlo permite definir a quantidade de simulações.

**7. Geração de Jogos**

📜 **gerador\_jogos.py**  
✅ Ajustado para garantir que os jogos gerados sejam **listas corretas de números**, eliminando erros anteriores de formatação.

**8. Interface Gráfica - Modo Painel**

📜 **painel.py**  
✅ Criado um painel interativo em **Streamlit** para visualização e gestão de apostas.  
✅ O painel agora permite:

* **Gerar sugestões de apostas** com modelo de Machine Learning.
* **Salvar grupos de apostas** no banco de dados.
* **Visualizar grupos salvos** e escolher qual exibir.
* **Remover grupos de apostas** armazenados.

**9. Banco de Dados**

📜 **banco.py**  
✅ Implementado o armazenamento de grupos de apostas no banco SQLite.  
✅ Criada a função salvar\_grupo\_apostas() para garantir a correta serialização dos dados.  
✅ Ajustada a conversão de números para evitar erro **"int32 not JSON serializable"**.

**10. Como Executar**

Certifique-se de que **Lotofácil.xlsx** está na pasta data/.  
Execute o painel gráfico com:

streamlit run painel.py

Agora o projeto está evoluindo com uma interface gráfica poderosa e um banco de dados organizado! 🚀

Me avise se quiser mais ajustes no documento ou novos detalhes! 🎯😊  
Agora, podemos seguir com os próximos aprimoramentos! 🔥

AJUSTES NO PROJETO  
  
Aqui está a documentação atualizada do seu projeto **Lotofácil Análise**, incorporando as últimas alterações nos módulos de predição, geração de jogos e interface do painel.

# **Documentação do Projeto Lotofácil Análise**

## ****Descrição Geral****

Este projeto utiliza análise preditiva aplicada à Lotofácil, explorando abordagens como modelos supervisionados (RandomForest e MLP), frequência condicional e clustering. O sistema permite gerar sugestões de apostas com base em padrões históricos, apresentar estatísticas e gerenciar apostas em uma interface interativa.

## ****1. Estrutura do Projeto****

A estrutura principal do projeto inclui os seguintes módulos:

| **Módulo** | **Descrição** |
| --- | --- |
| **dados.py** | Responsável por carregar e validar os dados históricos da Lotofácil. |
| **frequencia.py** | Implementa a análise baseada em frequência condicional, gerando uma matriz de co-ocorrência entre os números sorteados. |
| **clustering.py** | Aplica técnicas de clustering (K-Means) para identificar agrupamentos nos sorteios, permitindo detectar padrões ocultos. |
| **predicao.py** | Contém os métodos de predição (supervisionada, frequência condicional e clustering) que geram combinações sugeridas. |
| **gerador\_jogos.py** | Substitui a simulação de Monte Carlo e agora gera combinações aleatórias de números para apostas. |
| **painel.py** | Interface interativa em **Streamlit**, permitindo que o usuário visualize estatísticas, selecione métodos de predição e gere apostas. |
| **banco.py** | Gerencia o armazenamento de apostas em um banco de dados e permite recuperação e remoção de grupos de apostas. |

## ****2. Métodos de Predição****

O sistema oferece três abordagens para gerar previsões de apostas:

### ****2.1 Predição Supervisionada****

* Usa modelos **RandomForest** e **MLP**.
* Treina um modelo supervisionado com dados históricos para prever quais números serão sorteados.
* Ajusta a saída para garantir que sejam selecionados exatamente **15 números**.

### ****2.2 Frequência Condicional****

* Calcula uma matriz de co-ocorrência de números baseada nos sorteios históricos.
* Gera recomendações escolhendo os **15 números** com maior probabilidade condicional com base no último sorteio.

### ****2.3 Clustering****

* Representa cada sorteio como um vetor binário e aplica **K-Means** para detectar padrões.
* Identifica o cluster mais próximo do último sorteio e sugere os **15 números** mais relevantes dentro desse grupo.

## ****3. Geração de Jogos****

### ****3.1 Atualização de**** gerador\_jogos.py

Este módulo foi atualizado para remover a dependência da função simulacao\_monte\_carlo. Agora, ele gera jogos aleatórios sem necessidade de aprendizado de máquina.

import random

import pandas as pd

def gerar\_jogos(df, modelo, mlb, quantidade=5):

"""

Gera apostas aleatórias para Lotofácil com 15 números cada.

"""

jogos = [sorted(random.sample(range(1, 26), 15)) for \_ in range(quantidade)]

return pd.DataFrame(index=[tuple(jogo) for jogo in jogos])

## ****4. Interface do Painel****

### ****4.1 Atualização de**** painel.py

A interface foi ajustada para permitir a escolha entre três métodos de predição (**Supervisionada, Frequência Condicional e Clustering**) e incluir uma opção adicional para selecionar o modelo (RandomForest ou MLP) ao escolher a predição supervisionada.

metodo\_predicao = st.selectbox("Selecione o método de predição:", ["Supervisionada", "Frequência Condicional", "Clustering"])

if metodo\_predicao == "Supervisionada":

modelo\_escolhido = st.radio("Selecione o modelo:", ["RandomForest", "MLP"])

Após a escolha do método, a função correspondente é chamada para gerar a sugestão de aposta.

## ****5. Fluxo de Uso****

1. **Carregar o painel**:
2. streamlit run painel.py
3. **Navegar pelo menu**:
   * **Dashboard**: Exibe estatísticas dos sorteios históricos.
   * **Gerar Apostas**: Permite escolher um método de predição e gerar sugestões.
   * **Gerenciar Apostas**: Exibe apostas já registradas e permite a remoção de grupos.
4. **Gerar uma aposta**:
   * Escolher um método de predição.
   * Clicar no botão "Gerar sugestão de aposta".
   * Visualizar a combinação sugerida e os jogos aleatórios gerados.
5. **Salvar as apostas**:
   * Registrar as apostas no banco de dados clicando em "Salvar Grupo de Apostas".

## ****6. Controle de Versão****

Para enviar as mudanças para o **GitHub**, use os seguintes comandos:

git add .

git commit -m "Atualização do painel e ajustes no gerador de jogos"

git push origin main

### ****Conclusão****

Este projeto fornece uma abordagem analítica e exploratória para prever padrões na Lotofácil, combinando aprendizado de máquina, estatísticas e geração de jogos. As últimas atualizações melhoraram a interface, corrigiram problemas na predição supervisionada e simplificaram a geração de apostas.

Se precisar de mais ajustes ou melhorias, estou à disposição para ajudar! 🚀🎲