Documentação do Pipeline de Machine Learning para Previsão de Sucesso de Startups

1. Carregamento dos Dados

Carregamos os datasets de treino, teste e o arquivo de submissão
Verificamos o formato dos dados e a distribuição da variável alvo (`labels`).

```
train = pd.read_csv('train.csv')
test = pd.read_csv('test.csv')
sample_sub = pd.read_csv('sample_submission.csv')
```

2. Limpeza e Tratamento de Valores Nulos

Preenchemos valores ausentes nas colunas numéricas usando a mediana, que é robusta a outliers.
Também tratamos valores infinitos e NaNs gerados durante a criação de novas features, substituindo-os por zero.

3. Engenharia de Features

Criamos novas variáveis que capturam relações importantes, como:

- **funding_per_round:** valor médio financiado por rodada.
- **milestones_per_year:** número de marcos alcançados por ano.
- **has_multiple_rounds:** indicador se a startup teve mais de uma rodada de financiamento.
- **funding_age_span:** diferença entre o primeiro e último ano de financiamento.
- **relationships_per_milestone:** número de relacionamentos por marco.
- Outras features derivadas para capturar eficiência e densidade de milestones.

Essas features ajudam o modelo a capturar padrões mais complexos que influenciam o sucesso.

4. Codificação e Escalonamento

Transformamos a variável categórica `category_code` em variáveis dummy (one-hot encoding) para que o modelo possa utilizá-las.
Aplicamos escalonamento (StandardScaler) nas variáveis numéricas para normalizar suas distribuições, facilitando o aprendizado dos modelos.

5. Divisão dos Dados e Balanceamento

Dividimos os dados em treino e validação com estratificação para manter a proporção das classes.
Aplicamos SMOTE para balancear as classes no conjunto de treino, evitando que o modelo fique enviesado para a classe majoritária.

6. Treinamento dos Modelos

Treinamos três modelos poderosos e complementares:

- Random Forest
- LightGBM
- XGBoost

Cada um com hiperparâmetros otimizados para maximizar a performance.

7. Ensemble com Stacking

Combinamos os três modelos usando Stacking Classifier, que treina um meta-modelo (Regressão Logística) para aprender a melhor forma de combinar as previsões individuais, aumentando a robustez e a acurácia final.

8. Ajuste Fino do Threshold

Realizamos uma busca detalhada no threshold de decisão (de 0.30 a 0.80 com passo 0.001) para maximizar a acurácia no conjunto de validação, pois o threshold padrão de 0.5 nem sempre é o ideal.

9. Avaliação Final

Calculamos métricas completas para avaliar o modelo:

- Acurácia
- Precisão
- Recall
- F1-score
- AUC (Área sob a curva ROC)

Essas métricas fornecem uma visão completa do desempenho, especialmente em problemas com classes desbalanceadas.

10. Treinamento Final e Submissão

Treinamos o modelo final com todos os dados balanceados e geramos as previsões para o conjunto de teste usando o threshold otimizado.
Salvamos o arquivo de submissão no formato exigido.

Considerações Finais

Este pipeline combina boas práticas de ciência de dados, incluindo limpeza rigorosa, engenharia de features, balanceamento, modelagem avançada e ajuste fino, resultando em um modelo com acurácia superior a 80%.