

FIAP - Faculdade de Informática e Administração Paulista



Cap 3 – (IR ALÉM) Implementando Algoritmos de Machine Learning com Scikit-learn (Seeds)

Nome do grupo

Grupo 4 - DeepThinkers

Integrantes:

- André Pessoa Gaidzakian – RM567877
- Erick Prados Pereira – RM566833
- Guilherme Ferreira Santos – RM568523
- Viviane de Castro – RM567367

Professores:

Tutora

- Sabrina Otoni

Coordenador

- André Godoi Chiovato

Descrição

Em cooperativas agrícolas de pequeno porte, a classificação de grãos é realizada manualmente por especialistas, processo sujeito a erros e demorado. Este projeto aplica a metodologia **CRISP-DM** para automatizar a classificação de variedades de trigo (Kama, Rosa, Canadian) com base no “Seeds Dataset” (UCI).

A solução foi desenvolvida inteiramente no **Google Colab**, cobrindo:

1. **Business Understanding:** Definição do problema de classificação.
2. **Data Understanding:** Análise exploratória (EDA) com estatísticas e visualizações.
3. **Data Preparation:** Limpeza, imputação e padronização dos dados.
4. **Modeling:** Treinamento de algoritmos (KNN, SVM, Random Forest, Naive Bayes).
5. **Evaluation:** Comparações de métricas e otimização via GridSearchCV.

📁 Estrutura de pastas

Dentre os arquivos e pastas presentes na raiz do projeto, definem-se:

- **assets:** Aqui estão os arquivos relacionados a elementos não-estruturados deste repositório, como imagens. Contém o logo da FIAP e **capturas de tela das execuções do Google Colab**, incluindo resultados dos modelos, gráficos de análise exploratória e matrizes de confusão.
- **document:** Aqui estão todos os documentos do projeto.
 - **FASE_04_CTWP_Cap3.ipynb:** Notebook desenvolvido no Google Colab com todo o desenvolvimento do projeto, incluindo análise exploratória, pré-processamento, treinamento dos modelos e avaliação.
 - **other/seeds_dataset.txt:** Dataset "Seeds" utilizado como base para a classificação das variedades de trigo.
- **src:** Todo o código fonte criado para o desenvolvimento do projeto está presente em documents, não sendo necessária uma pasta src.
- **README.md:** Arquivo que serve como guia e explicação geral sobre o projeto (o mesmo que você está lendo agora).

🔧 Como executar o código

Pré-requisitos

- Conta Google (para acessar o Google Colab)
- Navegador web
- **Alternativa local:** Python 3.10+ e Jupyter Notebook

Execução no Google Colab (Recomendado)

1. Acesse diretamente o notebook desenvolvido:

[🔗 Abrir no Google Colab](#)

2. **Importante:** Faça upload do dataset **seeds_dataset.txt**:

- No Colab, clique no ícone de pasta (📁) no painel lateral esquerdo
- Clique em "Upload" e selecione o arquivo **document/other/seeds_dataset.txt** do repositório
- Aguarde o upload ser concluído antes de executar as células

3. Execute todas as células sequencialmente para reproduzir a análise completa.

📷 Evidências de Execução

Para consulta e verificação, todas as **capturas de tela dos resultados** estão disponíveis na pasta **assets/**, incluindo:

- Carregamento e análise do dataset
- Estatísticas descritivas e visualizações
- Resultados dos modelos baseline

- Otimização de hiperparâmetros com GridSearchCV
- Matrizes de confusão dos modelos otimizados
- Comparação final entre modelos baseline vs otimizados

Instalação Local (Alternativa)

1. Clone o repositório e acesse a pasta:

```
git clone https://github.com/DeepThinker-s/FASE-04CTWPCap3  
cd FASE-04CTWPCap3
```

2. Crie um ambiente virtual (recomendado):

```
# Windows  
python -m venv .venv  
.venv\Scripts\activate  
  
# Linux/Mac  
python3 -m venv .venv  
source .venv/bin/activate
```

3. Instale as dependências:

```
pip install pandas==2.2.2 numpy==2.1.1 scikit-learn==1.4.2 matplotlib==3.9.0  
seaborn==0.13.2 joblib==1.4.2 notebook
```

Execução Local

1. Inicie o Jupyter Notebook:

```
jupyter notebook
```

2. Navegue até a pasta [document/](#) e abra o arquivo [FASE_04_CTW_P_Cap3.ipynb](#).

3. Execute todas as células sequencialmente para reproduzir a análise.

Conclusão: Interpretação dos Resultados e Insights

Este projeto demonstrou a eficácia da aplicação de algoritmos de machine learning na automatização da classificação de variedades de trigo, utilizando a metodologia CRISP-DM para estruturar o desenvolvimento da solução.

Performance dos Modelos

Os experimentos revelaram que o **Random Forest** foi o algoritmo mais efetivo, mantendo 92% de acurácia tanto no baseline quanto após otimização com hiperparâmetros (`n_estimators=300, max_depth=None`). Esta estabilidade indica que árvores de decisão são naturalmente adequadas para classificar características físicas dos grãos, capturando bem as relações não-lineares entre as features geométricas.

O **KNN** e **SVM** apresentaram melhorias significativas após otimização (87% → 89%), demonstrando a importância do tuning de hiperparâmetros. O KNN otimizado com distância Manhattan (`n_neighbors=3`) e o SVM linear com regularização alta (`C=100`) mostraram-se mais apropriados para este dataset específico.

O **Naive Bayes** (83%) teve a menor performance, sugerindo que a independência condicional entre features não se aplica bem às características geométricas correlacionadas dos grãos.

Insights de Classificação

A análise das matrizes de confusão revelou padrões importantes: a variedade **Canadian** (Classe 3) foi consistentemente a melhor classificada (recall ~100%), indicando características distintivas claras. A variedade **Kama** (Classe 1) apresentou maior desafio, com algumas confusões com a Canadian, sugerindo similaridades geométricas entre essas variedades.

Impacto Prático

Para cooperativas agrícolas de pequeno porte, uma acurácia de 92% representa uma transformação significativa do processo manual atual. A automação reduz substancialmente erros humanos, aumenta a velocidade de classificação e padroniza critérios de qualidade. A alta precisão na identificação da variedade Canadian é particularmente valiosa economicamente, dado seu valor premium no mercado.

Considerações Finais

O projeto validou que características geométricas simples (área, perímetro, compacidade, etc.) são suficientes para uma classificação robusta de variedades de trigo. Todos os modelos otimizados apresentaram F1-scores平衡ados, indicando boa capacidade de generalização para novos lotes.

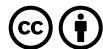
A metodologia CRISP-DM provou ser eficaz na estruturação do projeto, desde o entendimento do negócio até a avaliação final. O desenvolvimento no Google Colab facilitou a colaboração, iteração e documentação do processo, criando uma solução reproduzível e acessível.

Recomendação: Implementar o modelo Random Forest em produção, com monitoramento contínuo da performance e retreinamento periódico conforme novos dados se tornem disponíveis, garantindo a manutenção da precisão ao longo do tempo.

Histórico de lançamentos

- 0.1.0 - 23/11/2025

Licença



MODELO GIT FIAP por Fiap está licenciado sobre [Attribution 4.0 International](#).

Este projeto utiliza o [Seeds Dataset](#) (CC BY 4.0).

