



FIAP

alura

POSTECH

MACHINE LEARNING ENGINEERING

Tech Challenge 01

Grupo F-61

Cenário: Painel Inteligente para Análise e Projeção de Tendências na Produção e Venda de Vinhos e Sucos

1. Contextualização

1.1 Embrapa Uva e Vinho

Embrapa Uva e Vinho é um centro de pesquisa da Embrapa (Empresa Brasileira de Pesquisa Agropecuária). Este centro mantém um site que disponibiliza dados sobre a Viticultura no Brasil, como produção de uvas, vinhos e sucos, além de estatísticas do setor desde 1970 - URL: http://vitibrasil.cnpuv.embrapa.br/index.php?opcao=opt_01.

1.2 API Restfull

Como escopo do projeto do Tech Challenge 1, foi construída uma API Restfull que, por meio de técnicas de webscraping, permite o acesso a todos os dados históricos de uvas e vinhos de diversas fontes: Produção, Processamento, Comercialização, Importação e Exportação. Foi preocupação deste projeto a disponibilidade de dados (por meio de redundância) e segurança da informação (por meio de autorização). A documentação da API também seguiu padrões técnicos de mercado.

Algumas funcionalidades da API são:

- Coleta automática e atualizada de dados do site Vitibrasil da Embrapa Uva e Vinho;
- Armazenamento dos dados extraídos do site Vitibrasil para garantia de disponibilidade;
- Fornecimento de endpoints para consulta de dados de diversas fontes tais como Produção, Processamento, Comercialização, Importação e Exportação, podendo filtrar por ano;
- Cenários futuros: são propostos modelos de Machine Learning (ML) para implementação futura, para as diversas fontes de dados.

Potenciais usuários desta API são:

- Produtores rurais e viticultores – buscam informações sobre cultivo e produtividade;
- Pesquisadores e acadêmicos – utilizam os dados da Embrapa para estudos, publicações científicas e desenvolvimento de novas tecnologias para a viticultura;
- Empresas e cooperativas do setor – produtoras de vinhos, sucos e uvas de mesa que acompanham estatísticas de produção e consumo;
- Órgãos governamentais e formuladores de políticas – utilizam os dados para subsidiar políticas públicas voltadas para a agricultura e o setor vitivinícola;
- Consumidores e entusiastas do vinho – Alguns podem acessar o site para entender melhor a produção nacional e suas tendências.

A partir dos dados colhidos pela API, tornou-se possível sugerir cenários para o desenvolvimento de Modelos de Machine Learning - para este intento, é proposto um **Painel Inteligente para Análise e Projeção de Tendências na Produção e Venda de Vinhos e Sucos.**

2. Arquitetura da Solução proposta

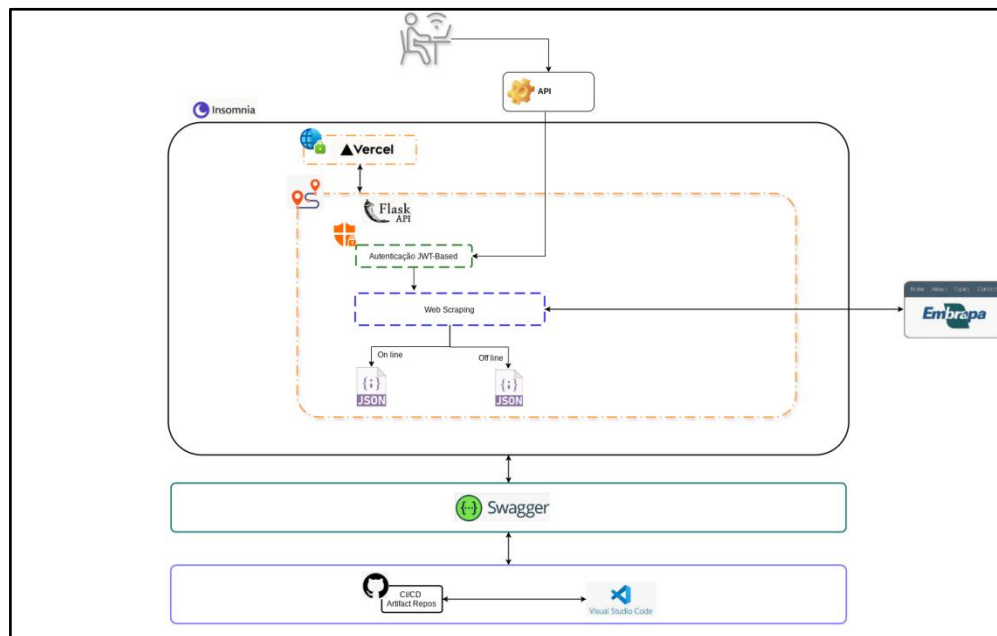


Figura 1 - Arquitetura da API

A proposta é que os dados obtidos online pelo *webscraping* sejam utilizados para os modelos preditivos - em caso de indisponibilidade do site da Embrapa, a fonte alternativa serão os dados off-line, conforme demonstra a Figura 1 anterior.

Como exemplo de integração da arquitetura API para consumo de uma possível rota que permita a aplicação de modelo de Machine Learning, como exemplo será adtimida a “Fonte de dados: Produção”, descrita no Cenário 3.3.2 adiante. Para a “Produção”, a seguir no Quadro 1 apresenta-se um recorte de alguns dados (formato JSON) retornados para um determinado ano pela rota correspondente:

```
{
  "producao": "[
    {
      Produto:VINHO DE MESA,
      Quantidade
(L.):217208604.0
    },
    {
      Produto:Tinto,
      Quantidade
(L.):174224052.0
    },
    {
      Produto:Branco,
      Quantidade
(L.):748400.0
    },
    {
      Produto:Rosado,
      Quantidade
(L.):42236152.0
    },
    {
      Produto:VINHO FINO DE MESA
(VINIFERA),
      Quantidade (L.):23899346.0
    }
  ]"
```

Quadro 1 - Retorno de dados da Rota “Produção” (formato JSON)

Como para este cenário propõe-se trabalhar com Previsão de produção, em Litros, por Produto categorizado, para um setor de Planejamento de Vendas, deve ser construída uma nova rota “Produção_Período” que permita informar um intervalo de anos para que se possa recuperar dados históricos de produção com “Ano”, “Produto” e “Quantidade” produzida. Com estes dados históricos, é possível a aplicação de EDA e treinamento de um modelo preditivo, usando técnicas como Regressão Linear ou Séries Temporais.

A seguir no Quadro 2 um exemplo de possíveis dados retornados (formato JSON) pela nova rota “Produção_Período”:

```
{
  "producao": "[
    {
      Ano: 2001, Produto:VINHO DE MESA,
      Quantidade
      (L.):217208604.0
    },
    {
      Ano: 2001, Produto:Tinto,
      Quantidade
      (L.):174224052.0
    },
    {
      Ano: 2001, Produto:Branco,
      Quantidade
      (L.):748400.0
    },
    {
      Ano: 2002, Produto:Rosado,
      Quantidade
      (L.):42236152.0
    },
    {
      Ano: 2002, Produto:VINHO FINO DE MESA
      (VINIFERA),
      Quantidade (L.):23899346.0
    }
  ]"
```

Quadro 2 - Retorno de dados da Rota “Produção_Período” (formato JSON) Proposta

Assim, pretende-se obter uma Previsão de Produção, conforme Gráfico 1 abaixo. Para este intento foi elaborado código Python apresentado no Anexo A deste documento.

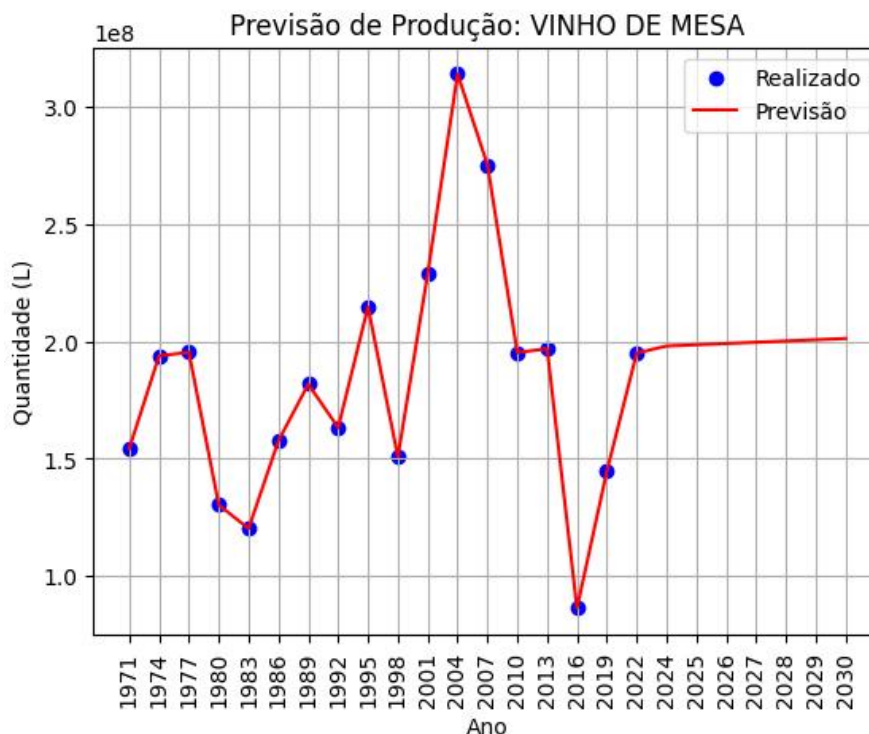


Gráfico 1 - Previsão de Produção: VINHO DE MESA

3. Pipeline

3.1 Coleta de Dados

Webscraping é feito a partir das fontes de dados descritas anteriormente, com rotas disponíveis para cada fonte em particular.

Tecnologias empregadas: Python, Flask, Pandas e BeautifulSoup. A autenticação foi feita com JWT. A documentação foi feita com Swagger (biblioteca Flasgger). A aplicação foi organizada e modularizada usando-se blueprints.

3.2 Pré-processamento

Os dados coletados são tratados de forma a garantir um nível de qualidade mínima - por exemplo, quando um valor vem com “nd” é substituído por “0.00”.

Tecnologias empregadas: Python e Pandas.

3.3 Cenários de Modelos para Machine Learning

Para a proposta de cenários, serão abordadas as fontes de dados com potenciais tarefas de Machine Learning (ML) correlatas para possíveis setores de interesse. Também serão propostos potenciais algoritmos para endereçar solução para cada tarefa de ML proposta.

3.3.1 Fonte de dados: Processamento

Tarefa de ML: Previsão de processamento, em Kg, por tipo de Cultivar (Uva)

Setor Interessado: Planejamento da Produção

Algoritmos possíveis: Regressão Linear ou Séries Temporais

3.3.2 Fonte de dados: Produção

Tarefa de ML: Previsão de produção, em Litros, por Produto categorizado

Setor Interessado: Planejamento de Vendas

Algoritmos possíveis: Regressão Linear ou Séries Temporais

Tarefa de ML: Agrupamento de Variedades de Produtos com melhor desempenho

Setor Interessado: Planejamento de Produção, em toda sua cadeia, desde o plantio

Algoritmos possíveis: K-Means ou DBSCAN

3.3.3 Fonte de dados: Exportação

Tarefa de ML: Previsão de preço, em US\$, por Pais categorizado

Setor Interessado: Planejamento de Vendas, identificando sazonalidades de mercado

3.3.4 Fonte de Dados: Importação

Tarefa de ML: Previsão de preço, em US\$, por Pais categorizado

Setor Interessado: Planejamento de Compras e Vendas

3.3.5 Fonte de Dados: Importação

Tarefa de ML: Previsão de processamento, em Litros, por tipo de Cultivar (Uva)

Setor Interessado: Planejamento de Vendas, identificando sazonalidades de mercado

4. Modelo de Deploy

Para o atendimento dos cenários de Machine Learning sugeridos, a seguir na Tabela 1 são apresentadas as etapas e tecnologias passíveis de serem aplicadas para o Deploy.

Etapas	Descrição
Containerização	Uso de Docker para isolar API e Webscraper
Hospedagem da API	Render, Heroku ou AWS EC2
Banco de Dados	PostgreSQL (recomendado para dados relacionais) ou MongoDB (para flexibilidade)
Agendamento do Scraping	Cron, Celery + Redis ou GitHub Actions programado
Visualização dos Dados	Gráficos com Plotly/Dash, Streamlit ou integração com BI (ex: Metabase)
Monitoramento	Logs com Flask-Logging + observabilidade via Grafana ou serviços da nuvem

Tabela 1 - Etapas e Tecnologias para o Deploy de API de Machine Learning

Anexo A - Código Python no formato Jupyter Notebook - Previsão de Produção

```
# Importação de Bibliotecas e Leitura Arquivo CSV
import pandas as pd
df = pd.read_csv('Producao.csv', sep=';')

# Tratamento dos dados lidos: Primeira linha do CSV tem histórico de um Produto;
# Transposição dos dados
df = df2 = df.iloc[[0]] # Pega a primeira linha como DataFrame
df3 = df2.T # Transpõe (anos viram índice, produto vira coluna)
df3 = df3.reset_index()
df3.columns = ['Ano', 'Quantidade (L)']
df4 = df3.iloc[2:]
produto = df.iloc[0, 1] # Vinho de Mesa
print(produto)
df5 = df4.iloc[1:]
df5 = df5[df5['Ano'].astype(int) % 3 == 0] # Intervalo de 3 anos (melhorar
exibição)
df5
```

VINHO DE MESA

	Ano	Quantidade (L)
4	1971	154264651
7	1974	193875345
10	1977	195359778
13	1980	130308185
16	1983	120261544
19	1986	157711522
22	1989	181801961
25	1992	163248419
28	1995	214788304
31	1998	150814943
34	2001	228932458
37	2004	313962284
40	2007	275287908
43	2010	195267980

	Ano	Quantidade (L)
46	2013	196904222
49	2016	86319015
52	2019	144629737
55	2022	195031611

```
# Importação de Bibliotecas de ML e Previsão com Linear Regression
from sklearn.linear_model import LinearRegression
import numpy as np
import matplotlib.pyplot as plt

# Preparação dos dados para o modelo ML
X = df5['Ano'].values.reshape(-1, 1) # variável independente (Ano)
y = df5['Quantidade (L)'].values

# Criar e ajustar o modelo
model = LinearRegression()
model.fit(X, y)

# Prever para os próximos anos, por exemplo 2024 a 2031
anos_futuros = np.arange(2024, 2031).reshape(-1, 1)
previsoes = model.predict(anos_futuros)

# Mostrar resultados
for ano, pred in zip(anos_futuros.flatten(), previsoes):
    print(f'Ano {ano}: previsão {int(pred)} litros')

Ano 2024: previsão 198035280 litros
Ano 2025: previsão 198572381 litros
Ano 2026: previsão 199109483 litros
Ano 2027: previsão 199646584 litros
Ano 2028: previsão 200183685 litros
Ano 2029: previsão 200720787 litros
Ano 2030: previsão 201257888 litros

import matplotlib.pyplot as plt

# Plot dos dados históricos
plt.scatter(df5['Ano'], df5['Quantidade (L)'], color='blue', label='Realizado')

# Plot da linha de regressão (tanto dados históricos quanto previsões)
import numpy as np
X_all = np.concatenate((df5['Ano'].values.astype(str), anos_futuros.flatten()))
y_all = np.concatenate((df5['Quantidade (L)'].values, previsoes))
```

```

# Ordena os valores para o gráfico ficar certinho
# sorted_indices = np.argsort(X_all)
#X_all_sorted = X_all[sorted_indices]
#y_all_sorted = y_all[sorted_indices]

plt.plot(X_all, y_all, color='red', label='Previsão')

plt.xlabel('Ano')
plt.ylabel('Quantidade (L)')
plt.xticks(rotation=90)
plt.title('Previsão de Produção: '+produto)
plt.legend()
plt.grid(True)
plt.show()

```

