

ABC Inventory Slotting Analysis App

What is this app?

This is a user-friendly web application that helps warehouse and inventory managers organize their inventory items efficiently using the ABC classification method. The app also assigns optimal warehouse locations to each item based on its importance.

What is ABC Analysis?

ABC analysis is a simple but powerful inventory management technique that categorizes items into three groups:

- **A items:** Your most valuable items (typically 20% of items that represent 80% of your total inventory value)
- **B items:** Medium-value items (typically 30% of items that represent 15% of your total inventory value)
- **C items:** Low-value items (typically 50% of items that represent only 5% of your total inventory value)

This follows the Pareto principle (or 80/20 rule) which helps focus resources on managing the most important items.

What is Warehouse Slotting?

Warehouse slotting is the process of determining the optimal storage location for each item in your warehouse. This app automatically assigns locations in the format **d.aa.bb.ll.pp** where:

- **d** = deposit number
- **aa** = alley number
- **bb** = block number
- **ll** = level number
- **pp** = position number

High-value A items are placed in the most accessible locations for efficient picking.

Getting Started

Prerequisites

- Python 3.7 or higher
- Basic understanding of CSV files

Installation

1. **Clone this repository** to your computer

```
git clone https://github.com/r2talk-bit/r2bit_ABCSlotting.git
cd r2bit_ABCSlotting
```

2. Set up a virtual environment (recommended but optional)

```
python -m venv venv
```

- On Windows:

```
venv\Scripts\activate
```

- On macOS/Linux:

```
source venv/bin/activate
```

3. Install the required packages:

```
pip install -r requirements.txt
```

Quick Start

1. Run the application using one of these methods:

- Using the provided script:

```
run.cmd # On Windows
```

- Using Streamlit directly:

```
streamlit run streamlit_app.py
```

2. Open your web browser - the app should automatically open at <http://localhost:8501>

How to Use the App

Step 1: Prepare Your Data

Create a CSV file with these three columns (names must match exactly, case-insensitive):

- **item_id**: A unique identifier for each inventory item
- **annual_demand**: How many units are used/sold per year
- **unit_cost**: The cost per unit

Example CSV format:

```
item_id,annual_demand,unit_cost
SKU001,1500,10.50
SKU002,300,45.75
SKU003,50,120.00
```

Step 2: Upload and Configure

1. Upload your CSV file using the file uploader in the sidebar
2. Set the ABC classification cutoff percentages (default: A=80%, B=95%)
3. Configure your warehouse dimensions and ABC slotting strategy

Step 3: Run Analysis

Click the "Run ABC Analysis" button to:

- Classify items into A, B, and C categories
- Calculate value percentages and statistics
- Assign optimal warehouse locations
- Generate visualizations

Step 4: Review Results

- View the ABC classification results table
- Examine the Pareto chart showing value distribution
- Check the summary statistics by class
- Review the assigned warehouse locations

Step 5: Download Results

Click the download button to get a CSV file with all results including:

- Original item data
- ABC classifications
- Value calculations
- Assigned warehouse locations

Project Structure

```
r2bit_ABCSlotting/
├── streamlit_app.py      # Main application code
├── utils/
│   └── abc_slotting.py  # ABC analysis algorithm
├── requirements.txt      # Required Python packages
├── Dockerfile            # For containerization
└── cloudbuild.yaml       # For Google Cloud deployment
```

```
|— run.cmd          # Quick start script
|— README.md       # This documentation
```

Features

- **User-friendly interface** built with Streamlit
- **Automatic CSV parsing** with support for both comma and semicolon separators
- **Interactive Pareto chart** visualization
- **Customizable ABC classification** thresholds
- **Warehouse location assignment** based on item value
- **Summary statistics** by class
- **Downloadable results** in CSV format with semicolon separators

Docker Deployment

The app can be containerized and deployed to cloud platforms:

```
# Build Docker image
docker build -t r2bitabcslotting:latest .

# Run container locally
docker run -p 8501:8080 r2bitabcslotting:latest
```

Need Help?

If you encounter any issues or have questions about using this application, please open an issue on the GitHub repository.

Aplicativo de Análise de Endereçamento ABC de Inventário

O que é este aplicativo?

Este é um aplicativo web amigável que ajuda gerentes de armazém e inventário a organizar seus itens de estoque de forma eficiente usando o método de classificação ABC. O aplicativo também atribui localizações ideais de armazém para cada item com base em sua importância.

O que é a Análise ABC?

A análise ABC é uma técnica simples, mas poderosa, de gerenciamento de estoque que categoriza os itens em três grupos:

- **Itens A:** Seus itens mais valiosos (normalmente 20% dos itens que representam 80% do valor total do seu inventário)

- **Itens B:** Itens de valor médio (normalmente 30% dos itens que representam 15% do valor total do seu inventário)
- **Itens C:** Itens de baixo valor (normalmente 50% dos itens que representam apenas 5% do valor total do seu inventário)

Isso segue o princípio de Pareto (ou regra 80/20) que ajuda a concentrar recursos no gerenciamento dos itens mais importantes.

O que é Endereçamento de Armazém?

O endereçamento de armazém é o processo de determinar a localização ideal de armazenamento para cada item em seu armazém. Este aplicativo atribui automaticamente localizações no formato **d.aa.bb.ll.pp** onde:

- **d** = número do depósito
- **aa** = número do corredor
- **bb** = número do bloco
- **ll** = número do nível
- **pp** = número da posição

Itens A de alto valor são colocados nas localizações mais acessíveis para coleta eficiente.

Primeiros Passos

Pré-requisitos

- Python 3.7 ou superior
- Conhecimento básico de arquivos CSV

Instalação

1. **Clone este repositório** para o seu computador

```
git clone https://github.com/r2talk-bit/r2bit_ABCSlotting.git
cd r2bit_ABCSlotting
```

2. **Configure um ambiente virtual** (recomendado, mas opcional)

```
python -m venv venv
```

- No Windows:

```
venv\Scripts\activate
```

- No macOS/Linux:

```
source venv/bin/activate
```

3. Instale os pacotes necessários:

```
pip install -r requirements.txt
```

Início Rápido

1. **Execute o aplicativo** usando um destes métodos:

- Usando o script fornecido:

```
run.cmd # No Windows
```

- Usando o Streamlit diretamente:

```
streamlit run streamlit_app.py
```

2. **Abra seu navegador web** - o aplicativo deve abrir automaticamente em <http://localhost:8501>

Como Usar o Aplicativo

Passo 1: Prepare Seus Dados

Crie um arquivo CSV com estas três colunas (os nomes devem corresponder exatamente, não diferenciando maiúsculas de minúsculas):

- **item_id**: Um identificador único para cada item de inventário
- **annual_demand**: Quantas unidades são usadas/vendidas por ano
- **unit_cost**: O custo por unidade

Exemplo de formato CSV:

```
item_id,annual_demand,unit_cost
SKU001,1500,10.50
SKU002,300,45.75
SKU003,50,120.00
```

Passo 2: Faça Upload e Configure

1. Faça upload do seu arquivo CSV usando o carregador de arquivos na barra lateral
2. Defina as porcentagens de corte da classificação ABC (padrão: A=80%, B=95%)

3. Configure as dimensões do seu armazém e a estratégia de endereçamento ABC

Passo 3: Execute a Análise

Clique no botão "Run ABC Analysis" para:

- Classificar itens nas categorias A, B e C
- Calcular porcentagens de valor e estatísticas
- Atribuir localizações ideais de armazém
- Gerar visualizações

Passo 4: Revise os Resultados

- Veja a tabela de resultados da classificação ABC
- Examine o gráfico de Pareto mostrando a distribuição de valor
- Verifique as estatísticas resumidas por classe
- Revise as localizações de armazém atribuídas

Passo 5: Baixe os Resultados

Clique no botão de download para obter um arquivo CSV com todos os resultados, incluindo:

- Dados originais do item
- Classificações ABC
- Cálculos de valor
- Localizações de armazém atribuídas

Estrutura do Projeto

```
r2bit_ABCTslotting/  
├── streamlit_app.py      # Código principal do aplicativo  
├── utils/  
│   └── abc_slotting.py  # Algoritmo de análise ABC  
├── requirements.txt      # Pacotes Python necessários  
├── Dockerfile            # Para containerização  
├── cloudbuild.yaml       # Para implantação no Google Cloud  
├── run.cmd               # Script de início rápido  
└── README.md             # Esta documentação
```

Recursos

- **Interface amigável** construída com Streamlit
- **Análise automática de CSV** com suporte para separadores de vírgula e ponto-e-vírgula
- **Visualização interativa de gráfico de Pareto**
- **Limites de classificação ABC personalizáveis**
- **Atribuição de localização de armazém** baseada no valor do item
- **Estatísticas resumidas** por classe
- **Resultados para download** em formato CSV com separadores de ponto-e-vírgula

Implantação com Docker

O aplicativo pode ser containerizado e implantado em plataformas na nuvem:

```
# Construir imagem Docker
docker build -t r2bitabcslotting:latest .

# Executar contêiner localmente
docker run -p 8501:8080 r2bitabcslotting:latest
```

Precisa de Ajuda?

Se você encontrar algum problema ou tiver dúvidas sobre o uso deste aplicativo, por favor, abra uma issue no repositório GitHub.