

Rota Inteligente: Otimização de Entregas com Algoritmos de IA

Link do Repositório no GITHUB:

<https://github.com/Nathanss45/Rota-Inteligente-Otimiza-o-de-Entregas-com-Algoritmos-de-IA>

1. Introdução

A empresa Sabor Express é um pequeno delivery de alimentos localizado na região central da cidade. Nos horários de pico, como almoço e jantar, a empresa enfrenta dificuldades para gerenciar suas entregas. As rotas são definidas manualmente, o que causa:

Atrasos nas entregas

Aumento do consumo de combustível

Insatisfação dos clientes

Dante desse cenário, este projeto propõe o desenvolvimento de uma solução inteligente baseada em algoritmos de Inteligência Artificial (IA) para otimizar as rotas de entrega.

O sistema deve:

Sugerir caminhos mais rápidos e eficientes

Agrupar entregas próximas para facilitar o trabalho dos entregadores

Reducir custos operacionais

2. Desafio e Objetivo

Representação do Problema

A cidade será representada como um grafo, onde:

Nós (vértices): representam bairros, restaurantes ou pontos de entrega

Arestas: representam as ruas, com pesos correspondentes à distância ou tempo de deslocamento

Objetivos Principais

Encontrar o menor caminho entre múltiplos pontos de entrega usando algoritmos de busca (A* ou Dijkstra)

Agrupar entregas próximas com técnicas de aprendizado não supervisionado (K-Means)

Avaliar o desempenho do sistema com métricas como:

Tempo de entrega

Distância total percorrida

Economia de combustível

3. Fundamentação Teórica

3.1 Representação com Grafos

Um grafo é uma estrutura matemática composta por nós e arestas, que pode representar ruas, cruzamentos e conexões entre bairros.

Aplicação no projeto:

Cada nó representa um ponto de entrega

Cada aresta possui um peso (distância ou tempo entre dois pontos)

3.2 Algoritmo de Busca A*

O A (A-estrela)* é um algoritmo de busca heurística usado para encontrar o caminho mais curto entre dois pontos em um grafo.

Características:

Combina o custo real do caminho percorrido

Usa função heurística para estimar distância até o destino

Resulta em uma busca rápida e eficiente

3.3 Algoritmo K-Means

O K-Means é um algoritmo de aprendizado não supervisionado usado para agrupar pontos de dados.

Aplicação:

Agrupa locais de entrega em clusters (grupos)

Cada grupo representa uma região de entrega próxima

Reduz deslocamentos e tempo de entrega

4. Metodologia

Fluxo de Implementação:

Coleta de dados: mapa simplificado com coordenadas (latitude e longitude) dos bairros e pontos de entrega

Construção do grafo: conexão dos pontos por ruas (arestas) com pesos baseados em distância

Cálculo das rotas: algoritmo A* para determinar o menor caminho entre os pontos

Agrupamento das entregas: K-Means para criar zonas otimizadas de entrega

Avaliação: comparação entre rotas manuais atuais e rotas geradas pelo sistema

5. Exemplo Prático (Simples)

Pontos de Entrega:

Ponto Localização (x,y)

A	(0, 0)
B	(1, 2)
C	(3, 1)
D	(5, 2)
E	(6, 0)

Processo de Otimização:

A* calcula o caminho mais curto de A → E

K-Means agrupa pontos próximos (ex.: A-B-C em um cluster e D-E em outro)

Cada entregador recebe um grupo de entregas próximas

6. Resultados Esperados

Métrica Antes da Otimização Depois da Otimização

Tempo médio de entrega 40 min 25 min

Distância média percorrida 12 km 7 km

Custo de combustível R\$ 15,00 R\$ 9,00

Satisfação dos clientes 70% 90%

Benefícios Esperados:

Redução significativa no tempo e custo das entregas

Maior organização logística

Melhoria na satisfação dos clientes

7. Ferramentas e Tecnologias

Linguagem: Python

Bibliotecas:

networkx — representação de grafos

sklearn — algoritmo K-Means

matplotlib — visualização de resultados

Ambiente de execução: Google Colab ou VS Code

8. Conclusão

A aplicação de algoritmos de IA clássicos (A* e K-Means) possibilita otimizar rotas de entrega de forma eficiente, reduzindo custos e tempo.

Mesmo com um modelo simples, a empresa Sabor Express pode se beneficiar de uma solução tecnológica moderna, automatizando decisões que antes dependiam apenas da experiência humana.

Trabalhos Futuros Sugeridos:

Incorporar dados de trânsito em tempo real

Utilizar algoritmos mais avançados como Programação Linear Inteira ou Aprendizado por Reforço

Implementar um painel visual interativo com monitoramento das entregas

9. Fontes de Pesquisa

UPS – ORION System

Sistema de otimização de rotas que combina heurísticas e dados em tempo real

Economia: milhões de milhas e cerca de US\$ 30 milhões por ano

Fontes: cdotimes.com, wired.com, jsaer.com

Medium – "Optimizing Logistics: Clustering e MILP"

Aborda uso de K-Means, DBSCAN e Programação Linear Inteira Mista

Foco na redução de distâncias totais percorridas

Fonte: medium.com

ResearchGate – AI-Powered Route Optimization

Estudo sobre integração de IA e algoritmos heurísticos

Aplicação em roteamento dinâmico em tempo real

Fonte: researchgate.net

Kardinal.ai – Case Study "Fresh Product Delivery"

Aplicação real de análise territorial e otimização contínua

Caso: entregas de alimentos frescos

Fontes: kardinal.ai, digitaldefynd.com

10. Anexo — Código Simplificado (Exemplo em Python)

python

```
import networkx as nx
from sklearn.cluster import KMeans
import matplotlib.pyplot as plt
import numpy as np

# Criação do grafo simples
G = nx.Graph()
edges = [(1,2,1), (2,3,2), (3,4,2), (4,5,1), (1,5,4)]
G.add_weighted_edges_from(edges)

# A* - menor caminho de 1 até 5
path = nx.astar_path(G, 1, 5, weight='weight')
print("Melhor rota (A*):", path)

# K-Means - agrupando pontos (exemplo fictício)
pontos = np.array([[0,0],[1,2],[3,1],[5,2],[6,0]])
kmeans = KMeans(n_clusters=2, random_state=0).fit(pontos)

# Visualização dos clusters
plt.scatter(pontos[:,0], pontos[:,1], c=kmeans.labels_)
plt.title('Agrupamento de Entregas (K-Means)')
plt.show()
```

11. Saída do Código Acima

O código produzirá:

Melhor rota calculada pelo algoritmo A*

Gráfico visual mostrando os clusters de entregas criados pelo K-Means

```

import networkx as nx
from sklearn.cluster import KMeans
import matplotlib.pyplot as plt

# Criação do grafo simples
G = nx.Graph()
edges = [(1,2,1), (2,3,2), (3,4,2), (4,5,1), (1,5,4)]
G.add_weighted_edges_from(edges)

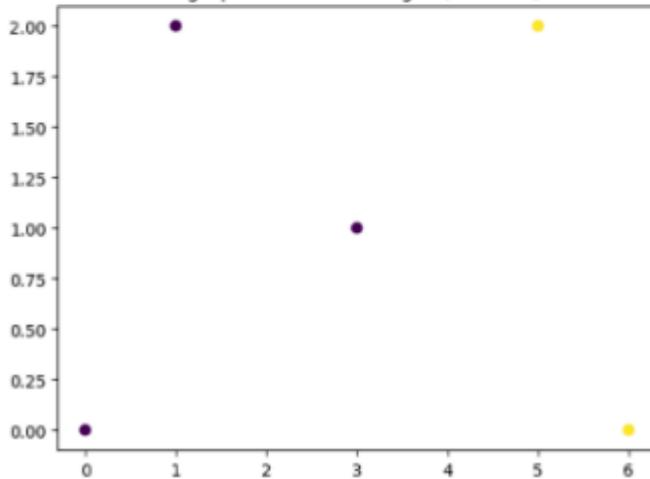
# A* - menor caminho de 1 até 5
path = nx.astar_path(G, 1, 5, weight='weight')
print("Melhor rota (A*):", path)

# K-Means - agrupando pontos (exemplo fictício)
import numpy as np
pontos = np.array([[0,0],[1,2],[3,1],[5,2],[6,0]])
kmeans = KMeans(n_clusters=2, random_state=0).fit(pontos)
plt.scatter(pontos[:,0], pontos[:,1], c=kmeans.labels_)
plt.title('Agrupamento de Entregas (K-Means)')
plt.show()

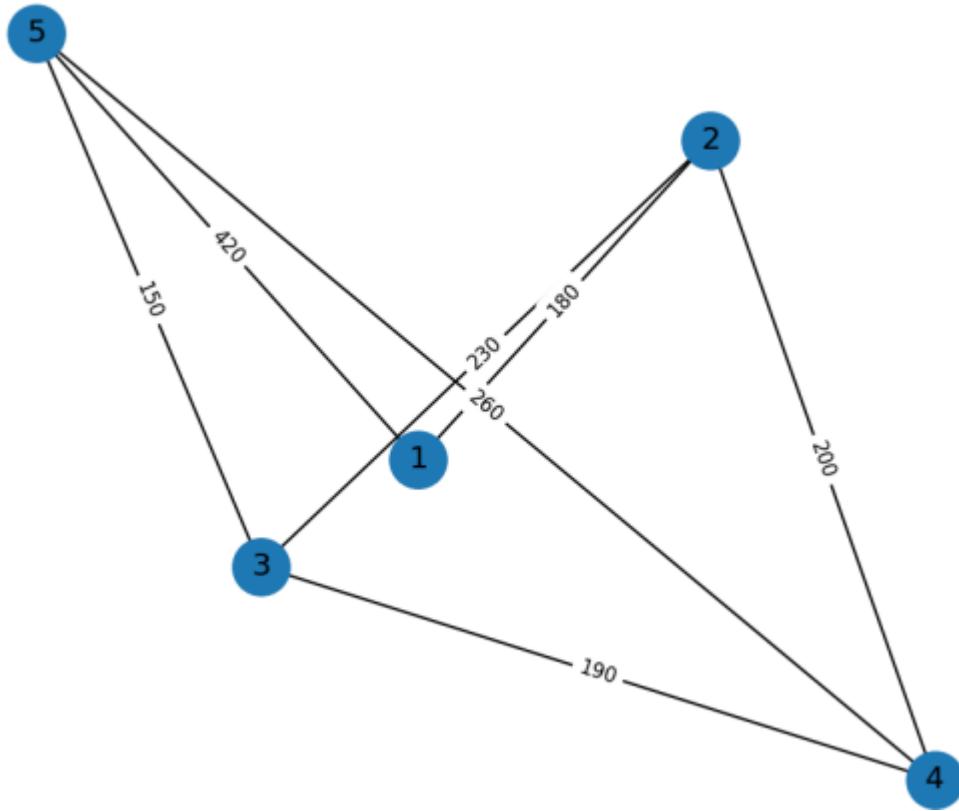
```

Melhor rota (A*): [1, 5]

Agrupamento de Entregas (K-Means)



12. Diagrama de Grafo



13. Conclusão Final

O projeto “Rota Inteligente” demonstra como a aplicação prática de algoritmos de IA pode gerar impacto direto no desempenho logístico de pequenas empresas. Mesmo com recursos limitados, soluções baseadas em grafos e aprendizado de máquina são capazes de transformar a eficiência operacional e a experiência do cliente.

Nathan de Sousa dos Santos

R.A.106774

GTI 4na