KAUNO TECHNOLOGIJOS UNIVERSITETAS
INFORMATIKOS FAKULTETAS

Intelektikos pagrindai 2020

Laboratorinio darbo nr. 4 ataskaita

Atliko:

IFF-7/2 gr. studentas

Justas Milišiūnas

2020-05-2

Dėstytojai:

lekt. Audrius Nečiūnas

doc. Agnė Paulauskaitė-Tarasevičienė

KAUNAS 2020

TURINYS

1. Duomenų rinkinio aprašymas

Duomenų rinkinį sudaro 8 stulpeliai, iš kurių 7 tolydus ir 1 kategorinis. Iš viso 257 įrašai.
Stulpeliai:

- mpg – mylios per galoną
- cylinders – variklio cilindrų skaičius
- cubicinches – variklio litražas
- hp – variklio galia
- weightlbs – mašinos svoris
- time-to-60 – įsibegėjimas nuo 0 iki 100 km/h
- year – pagaminimo metai
- brand – gamintojo vieta

2. Eksperimentai

2.1. Duomenys suklasterizuoti į 2 klasterius

Kai klasterizavimui naudojami stulpeliai ,,mpg“ ir ,,hp“:

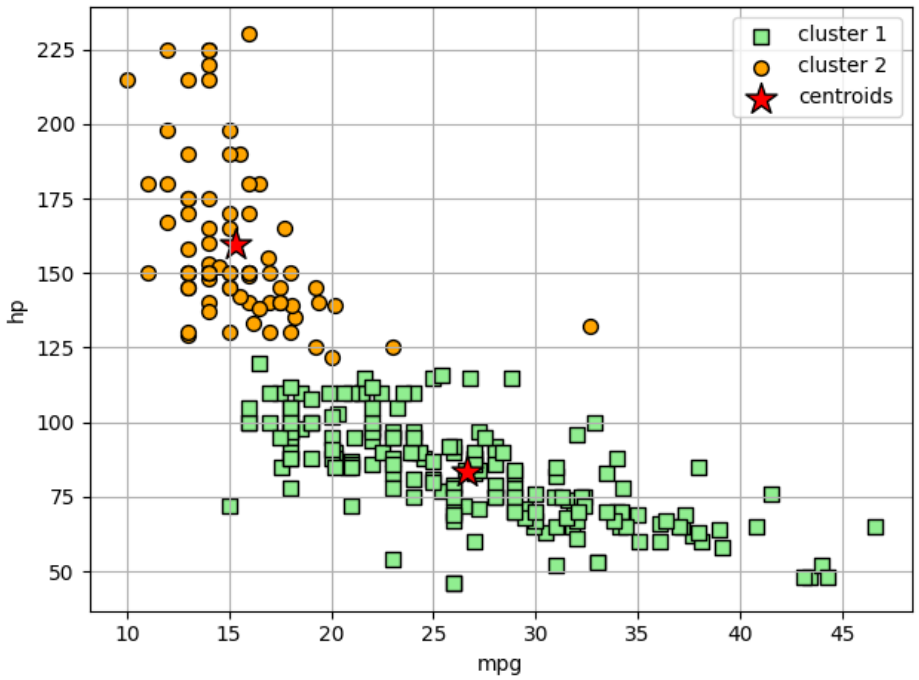

Diagrama 1: Klasterizavimas pagal stulpelius ,,hp" ir ,,mpg"

Šiame grafike matome, kad duomenys nėra lygiai pasiskirstę. Antrajame klasteryje duomenų yra daugiau.

Kai klasterizavimui naudojami stulpeliai „time-to-60“ ir „weightlbs“:

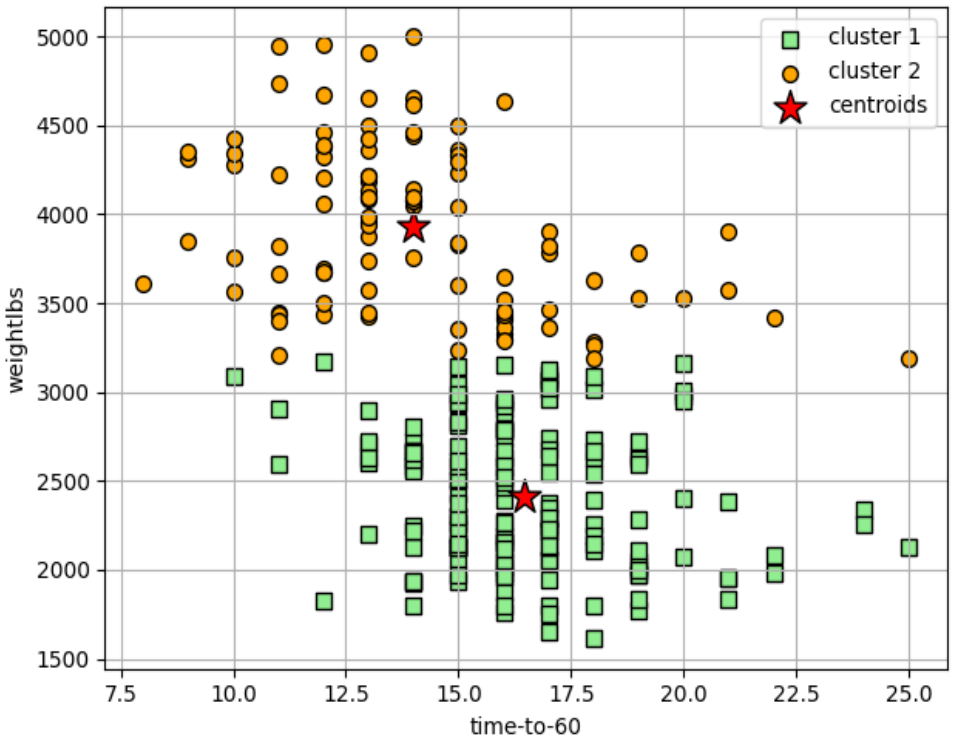

Diagrama 2: Klasterizavimas pagal stulpelius „time-to-60" ir „weightlbs"

Pagal šią diagramą matome, kad 1 klasteryje duomenų vėl yra daugiau. Duomenys nėra tolygiai pasiskirstę.

Kai klasterizavimui naudojami „cubicinches“ ir „weightlbs“:

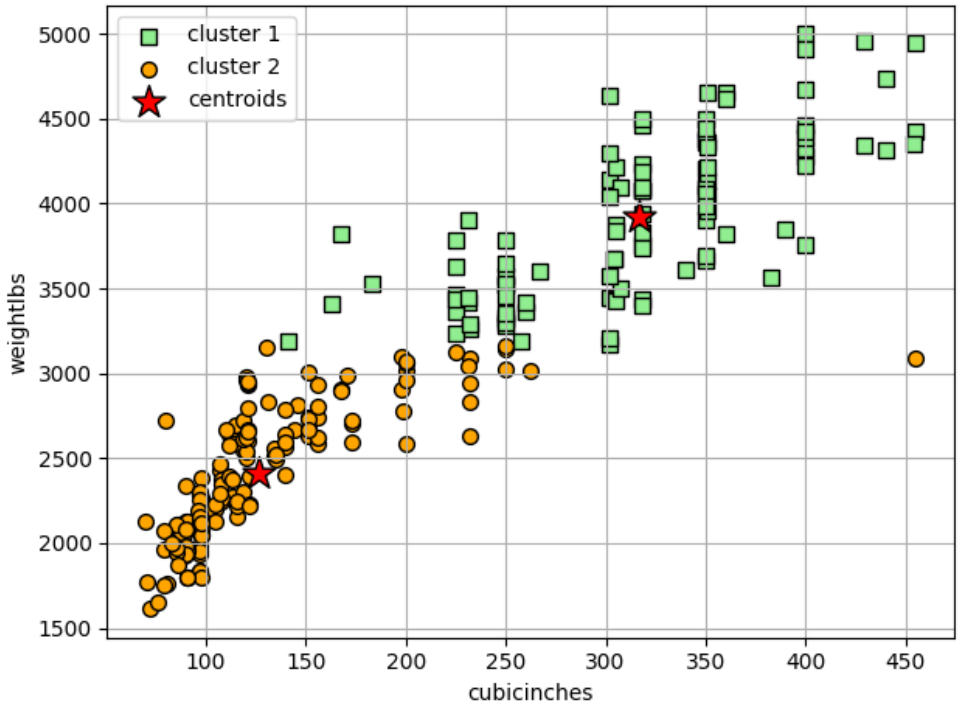

Diagrama 3: Klasterizavimas pagal stulpelius „cubicinches" ir „weightlbs"

Duomenys pasiskirstę daugmaž tolygiai. Pirmajame klasteryje esantys duomenys yra labai išsibarstę.

2.2. Duomenys suklasterizuoti į 3 klasterius

Kai klasterizavimui naudojami stulpeliai „mpg“ ir „hp“:

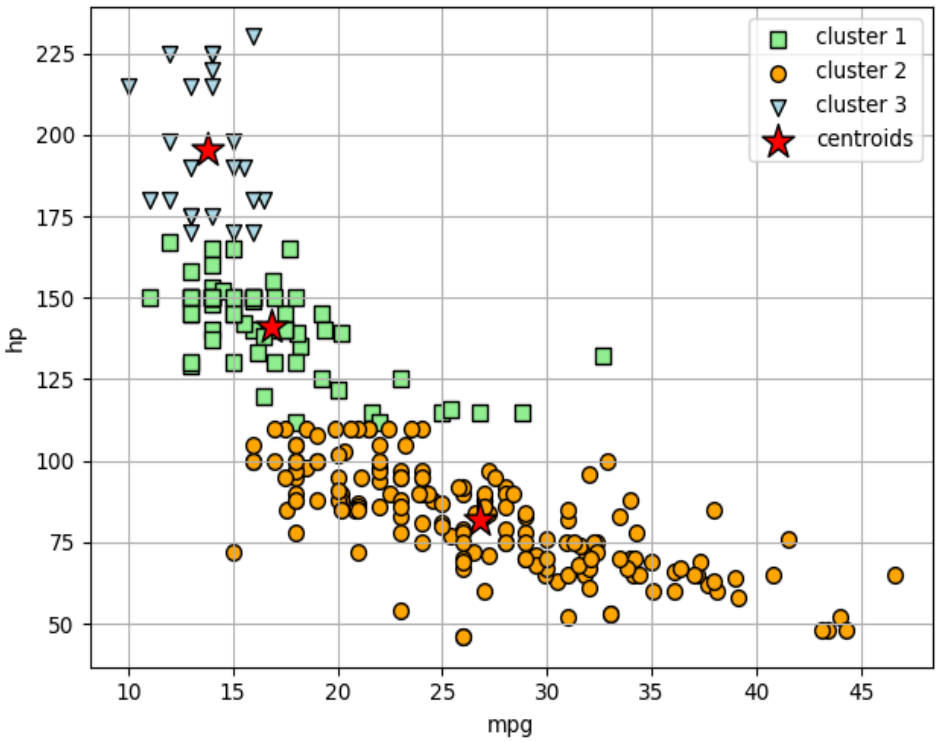

Diagrama 4: Klasterizavimas pagal stulpelius „hp" ir „mpg"

Matome, kad duomenys pasiskirstę labai netolygiai. Didžioji dalis duomenų yra priskirti 2 klasteriui.

Kai klasterizavimui naudojami stulpeliai „time-to-60“ ir „weightlbs“:

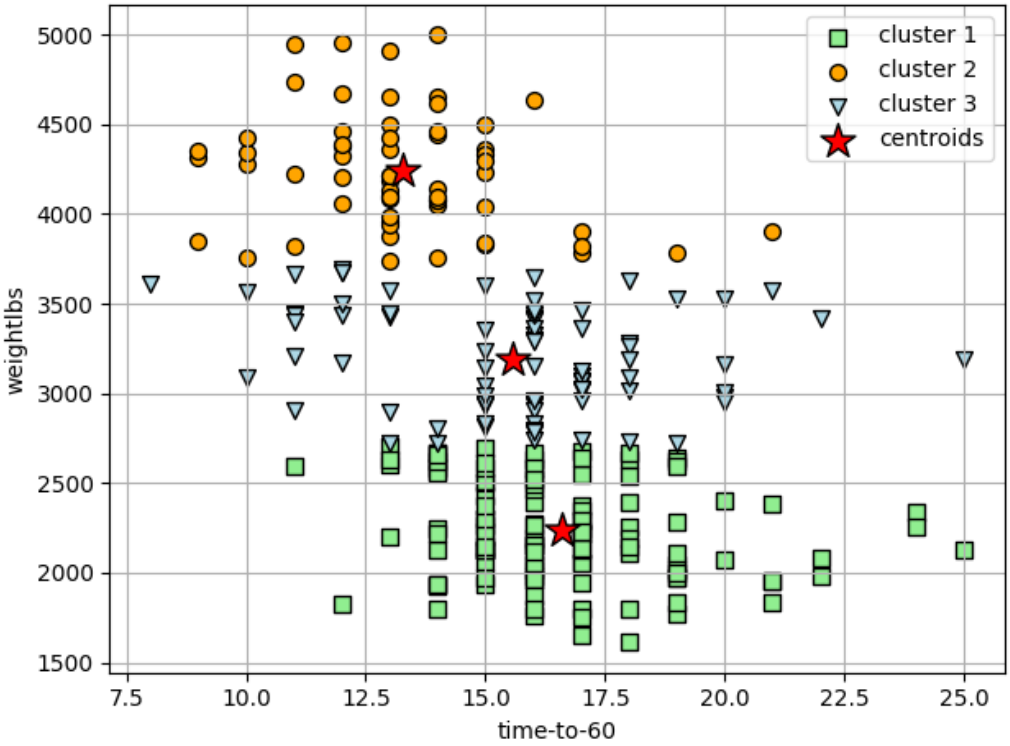

Diagrama 5: Klasterizavimas pagal stulpelius „time-to-60" ir „weightlbs"

Duomenys pasiskirstę tolygiai. Geriau nei su 2 klasteriais.

Kai klasterizavimui naudojami „cubicinches“ ir „weightlbs“:

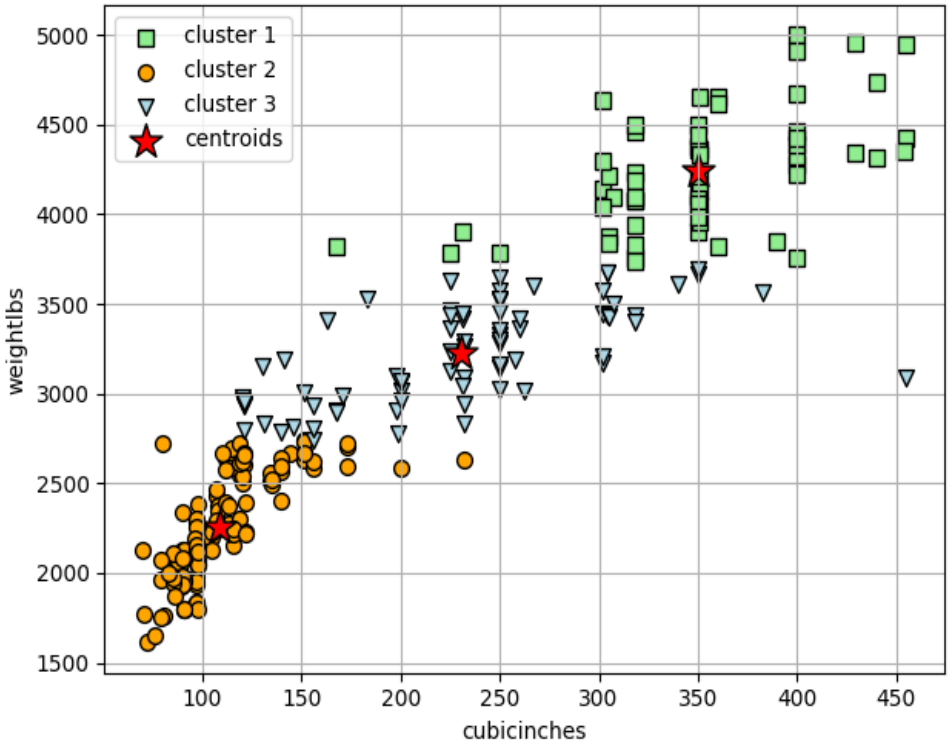

Diagrama 6: Klasterizavimas pagal stulpelius „cubicinches" ir „weightlbs"
Duomenys pasiskirstę daugmaž tolygiai. 1 ir 3 klasterio duomenys labai nutolę nuo centroidžių.

2.3. Duomenys suklasterizuoti į 4 klasterius

Kai klasterizavimui naudojami stulpeliai „mpg“ ir „hp“:

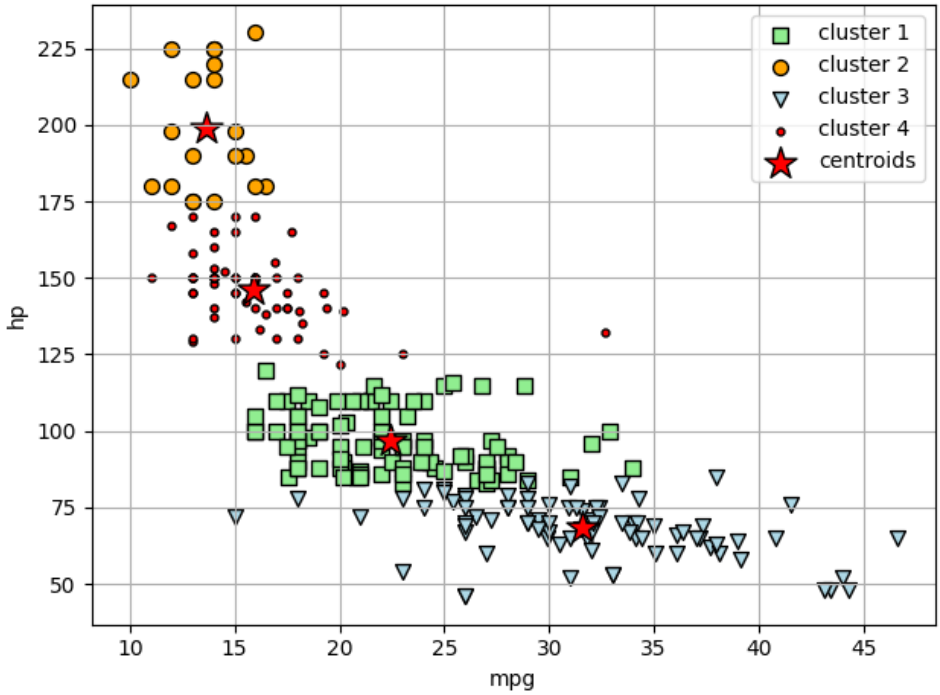

Diagrama 7: Klasterizavimas pagal stulpelius „hp" ir „mpg"
1 ir 3 klasteriuose duomenų yra žymiai daugiau nei 2 ir 4.

Kai klasterizavimui naudojami stulpeliai „time-to-60“ ir „weightlbs“:

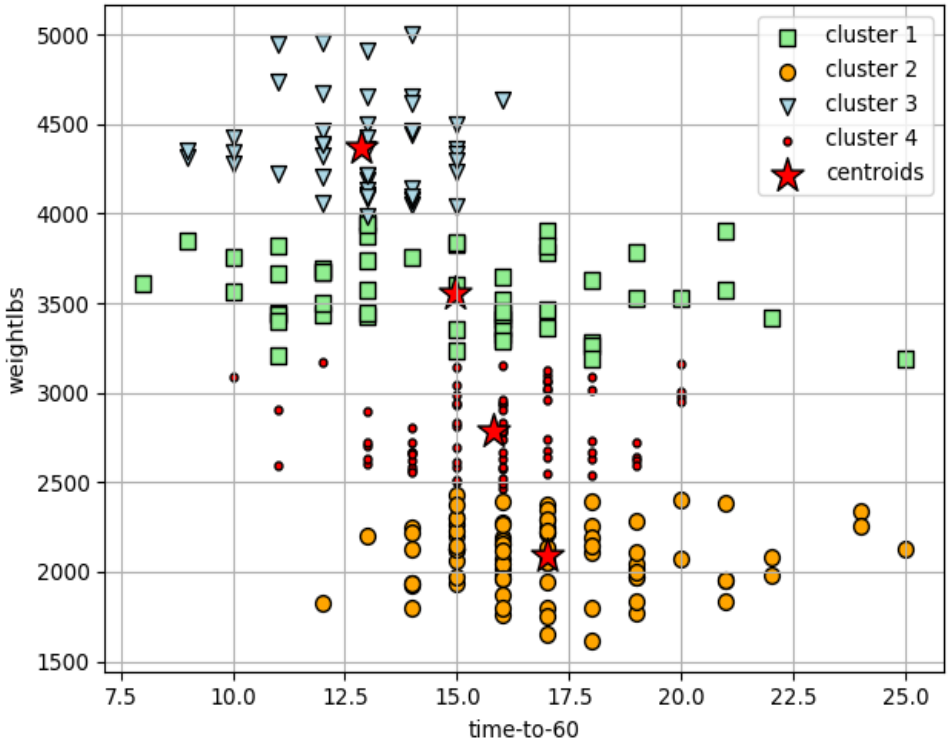

Diagrama 8: Klasterizavimas pagal stulpelius „time-to-60" ir „weightlbs"
1, 4, 2 klasteriai labai persidengia tarpusavyje.

Kai klasterizavimui naudojami „cubicinches“ ir „weightlbs“:

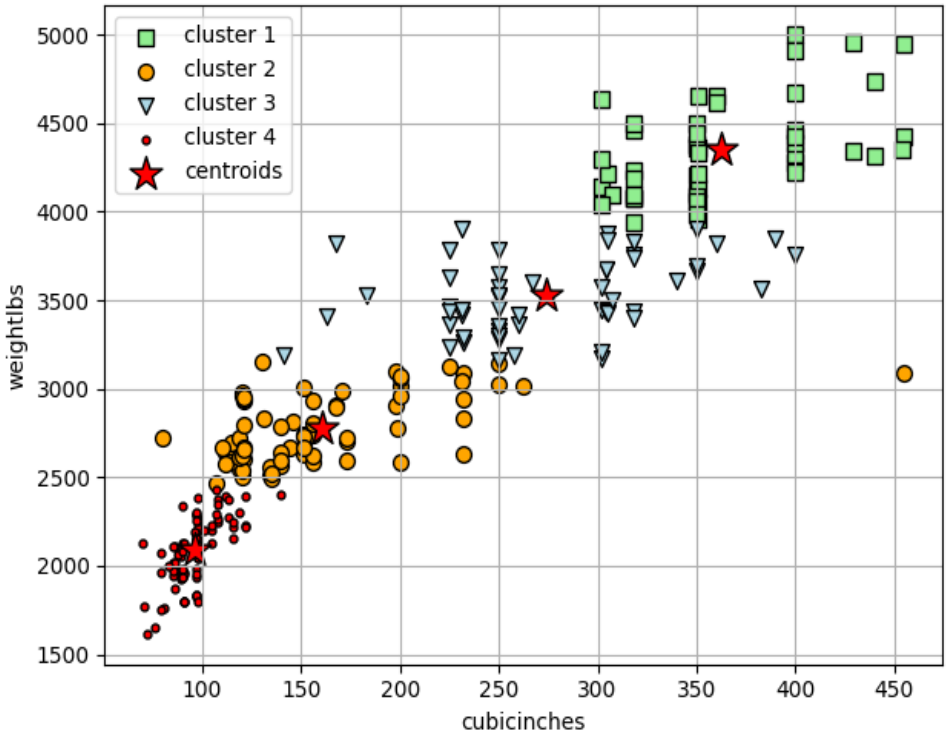

Diagrama 9: Klasterizavimas pagal stulpelius „cubicinches" ir „weightlbs"
Duomenys pasiskirstę tolygiai tarp visų klasterių.

3. Programos kodas

Naudota sklearn bibliotekos Kmeans algoritmo implementacija.

```
import matplotlib.pyplot as plt
import pandas as pd
from sklearn.cluster import KMeans


def show_clusters(data, columns, n_centroids=2):
    extracted_data = data[columns].to_numpy()

    km = KMeans(n_clusters=n_centroids)
    y_km = km.fit_predict(extracted_data)

    colors = ['lightgreen', 'orange', 'lightblue', 'red']
    markers = ['s', 'o', 'v', '.']

    # Plot all clusters
    for i in range(n_centroids):
        plt.scatter(
            extracted_data[y_km == i, 0], extracted_data[y_km == i, 1],
            s=50, c=colors[i],
            marker=markers[i], edgecolor='black',
            label=f"cluster {i + 1}"
        )

    # Plot centroids
    plt.scatter(
        km.cluster_centers_[:, 0], km.cluster_centers_[:, 1],
        s=250, marker='*',
        c='red', edgecolor='black',
        label='centroids'
    )

    plt.legend(scatterpoints=1)
    plt.grid()
    plt.xlabel(columns[0])
    plt.ylabel(columns[1])
    plt.show()


df = pd.read_csv('cars.csv')
show_clusters(df, ['cubicinches', 'weightlbs'], n_centroids=4)
```

4. Išvados

1. Šio duomenų rinkinio, geriausi atributai klasterizavimui yra ,,hp“ ir ,,mpg“. Lyginant su kitų atributų grafikais, matome, kad atributų ,,hp“ ir ,,mpg“ reikšmės yra mažiausiai persidengenčios.
2. Blogiausi atributai klasterizavimui yra ,,time-to-60“ ir ,,weightlbs“ dėl per mažo reikšmių kardinalumo ir smarkaus klasterių persidengimo.
3. Klasterių dydžiai labiausiai supanašėjo naudojant 4 centroides.
4. Dydžių skirtumas didžiausias tarp klasterių kur atributai ,,hp“ ir ,,mpg“ su 2 centroidėmis.
5. Šiam duomenų rinkiniui optimaliausias klasterių skaičius yra 4(sunku tiksliai nustatyti, nes visi klasteriai šalia vienas kito). Dėl mažiausio reikšmių nuokrypio nuo centroidžių.