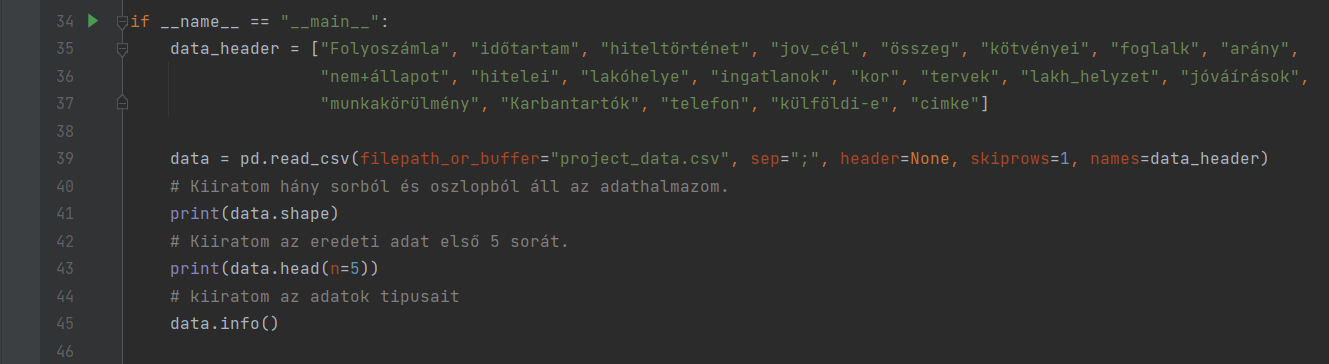
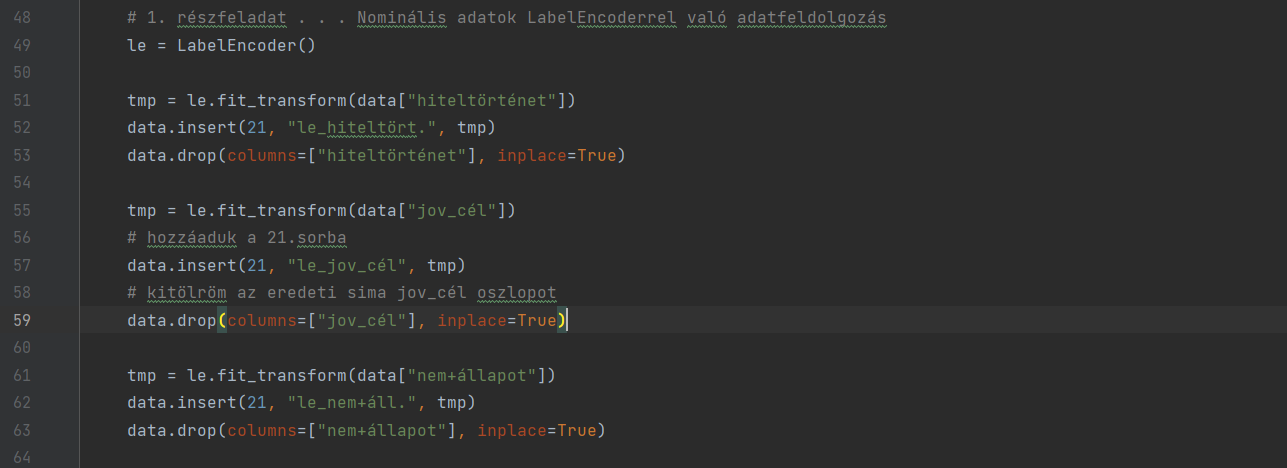
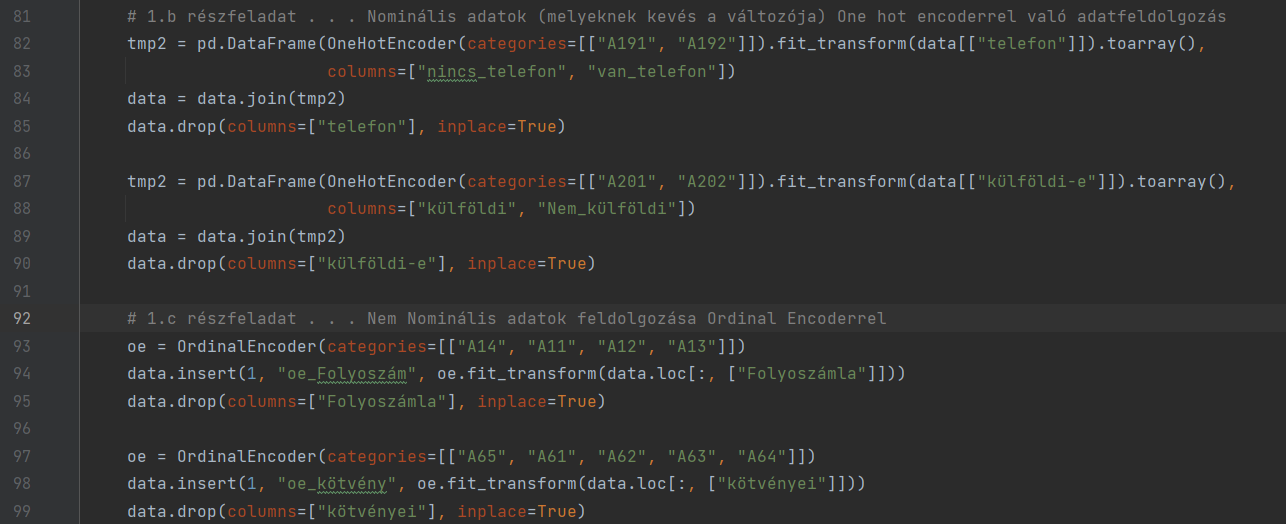
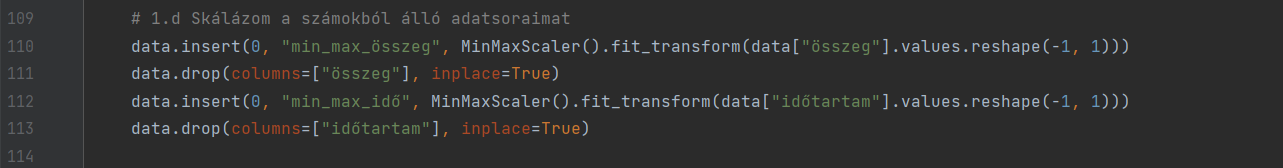
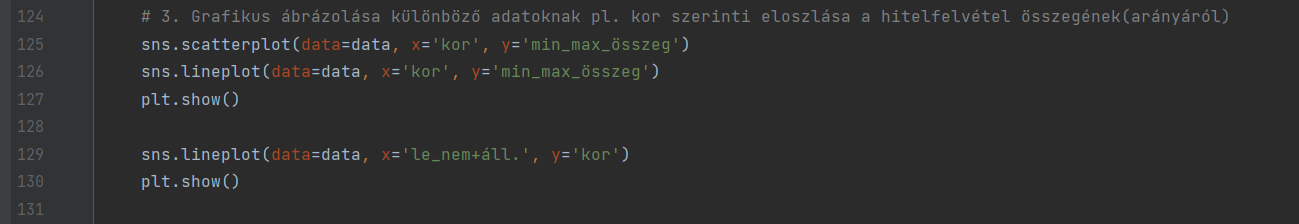
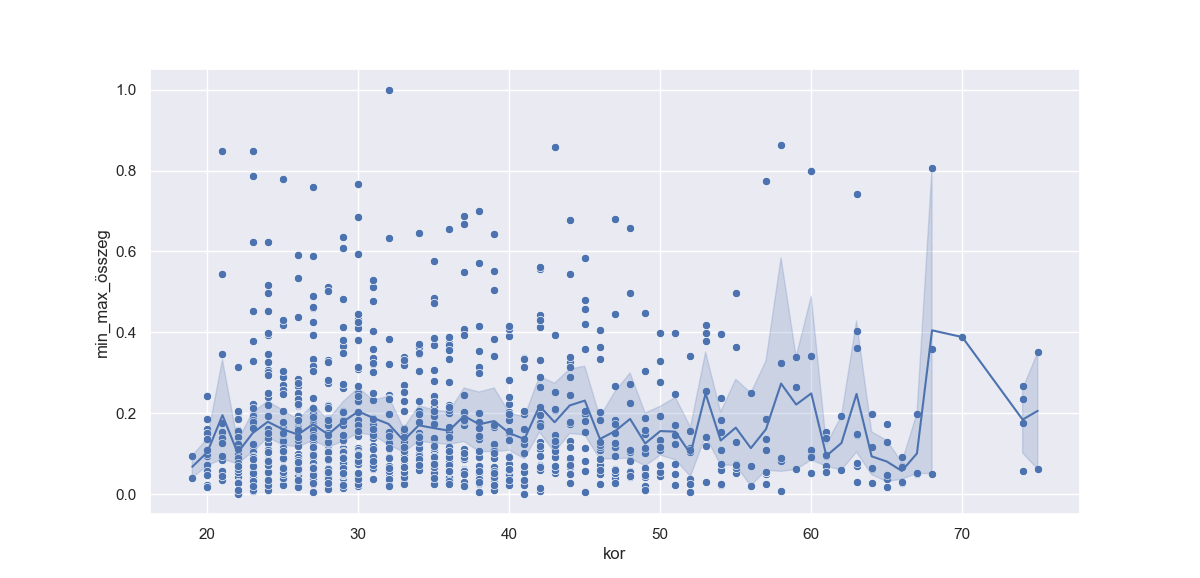
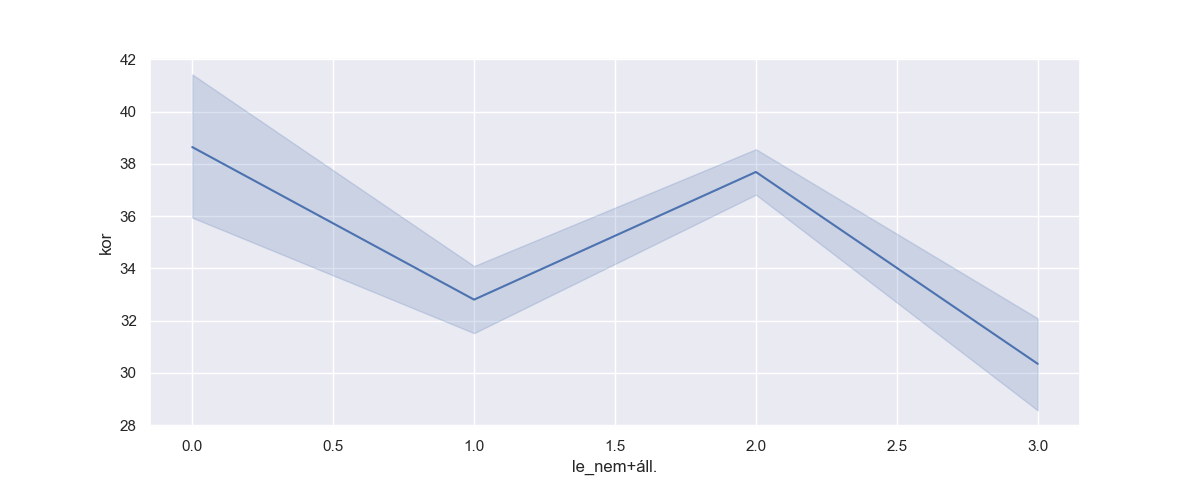
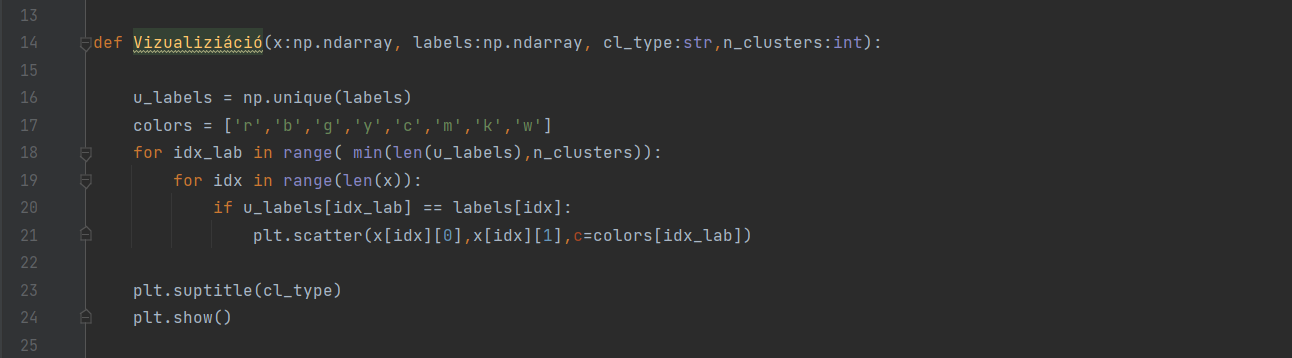
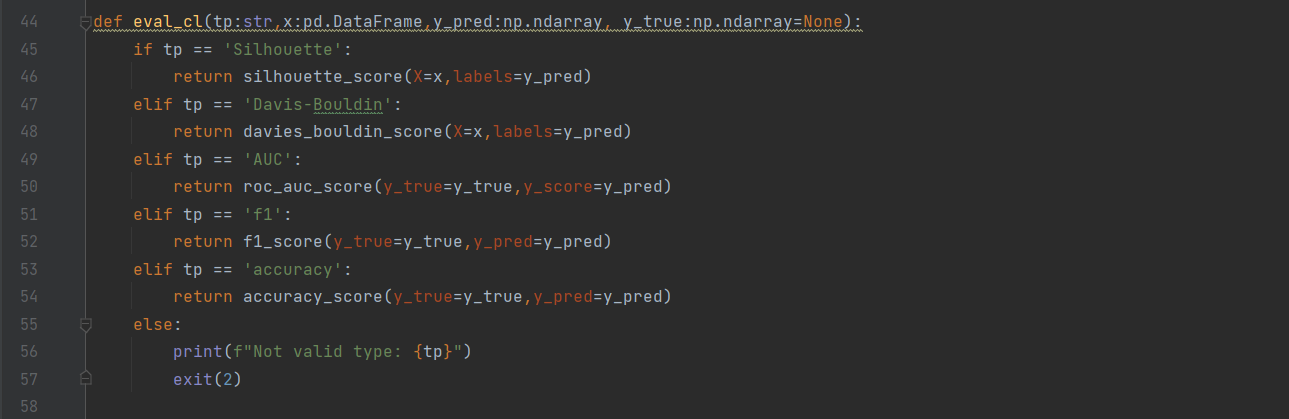
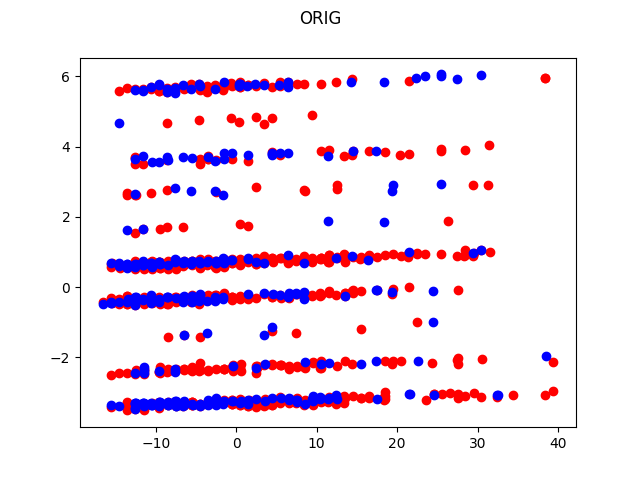
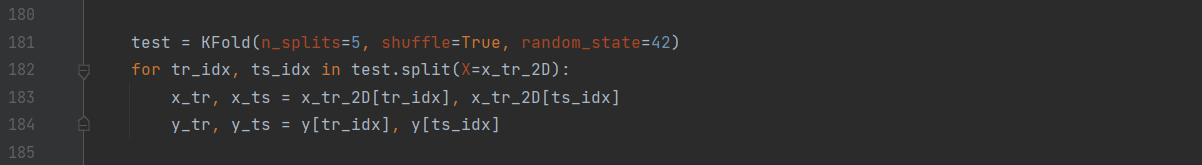
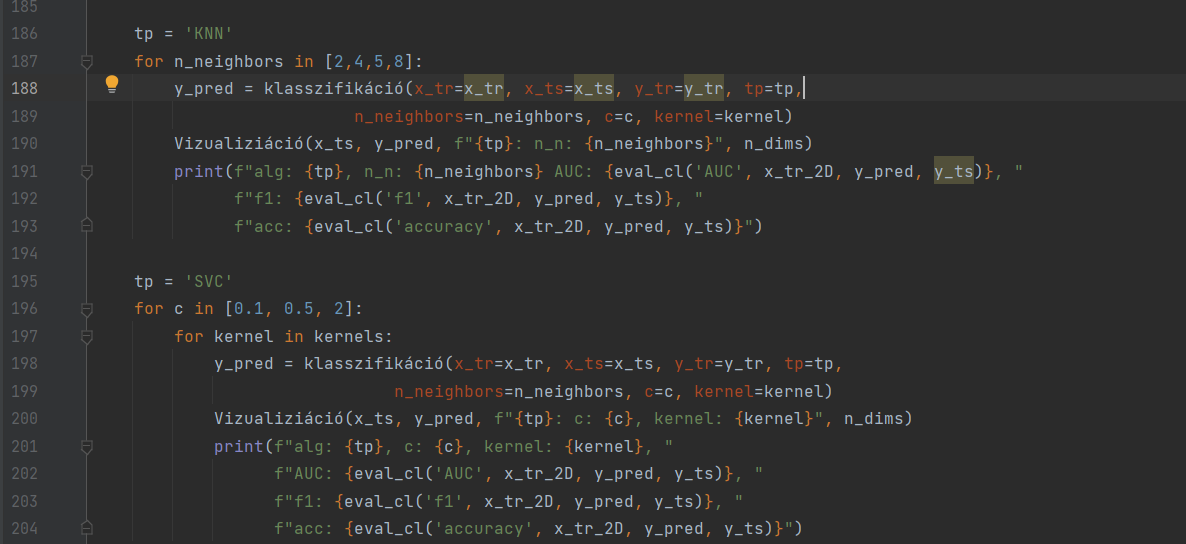
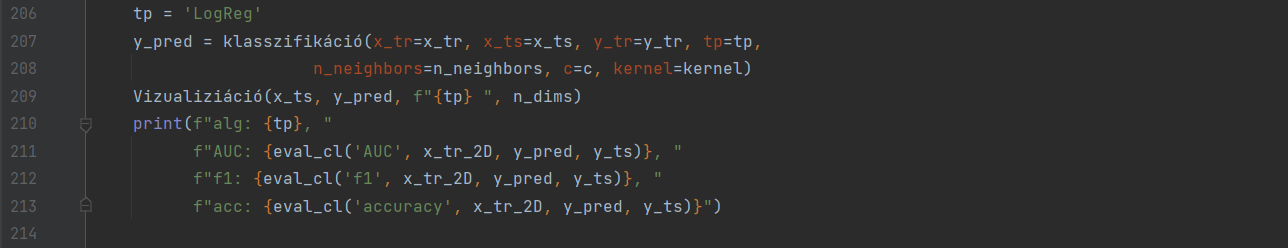
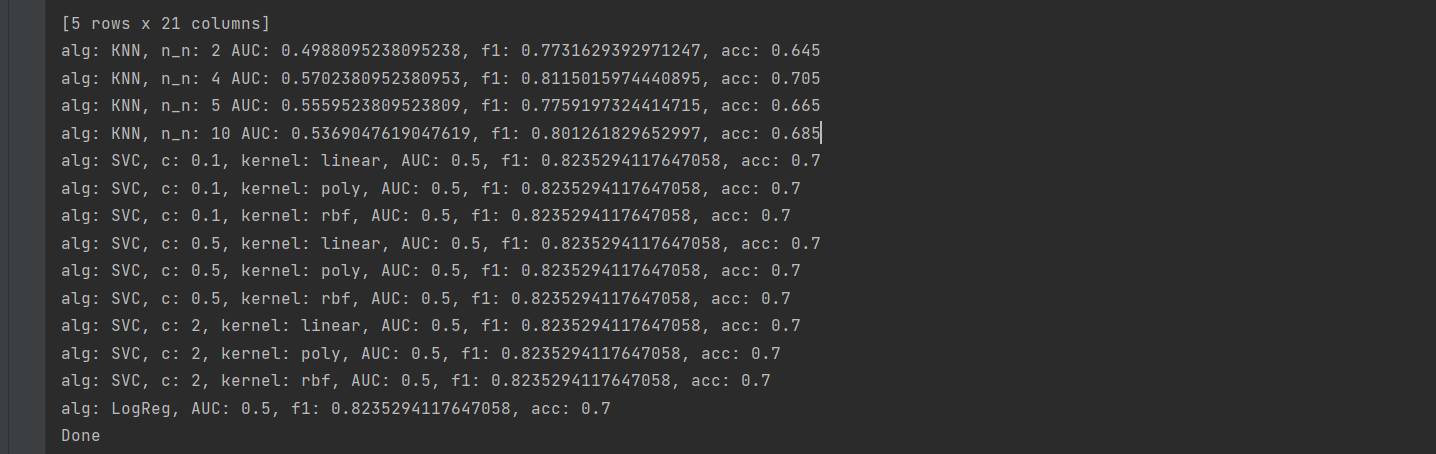
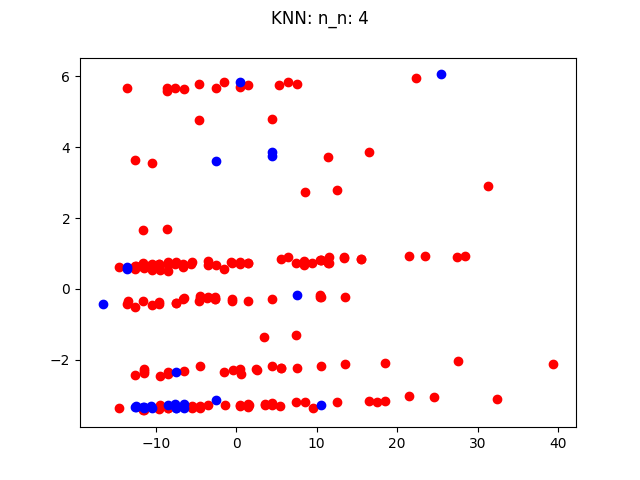
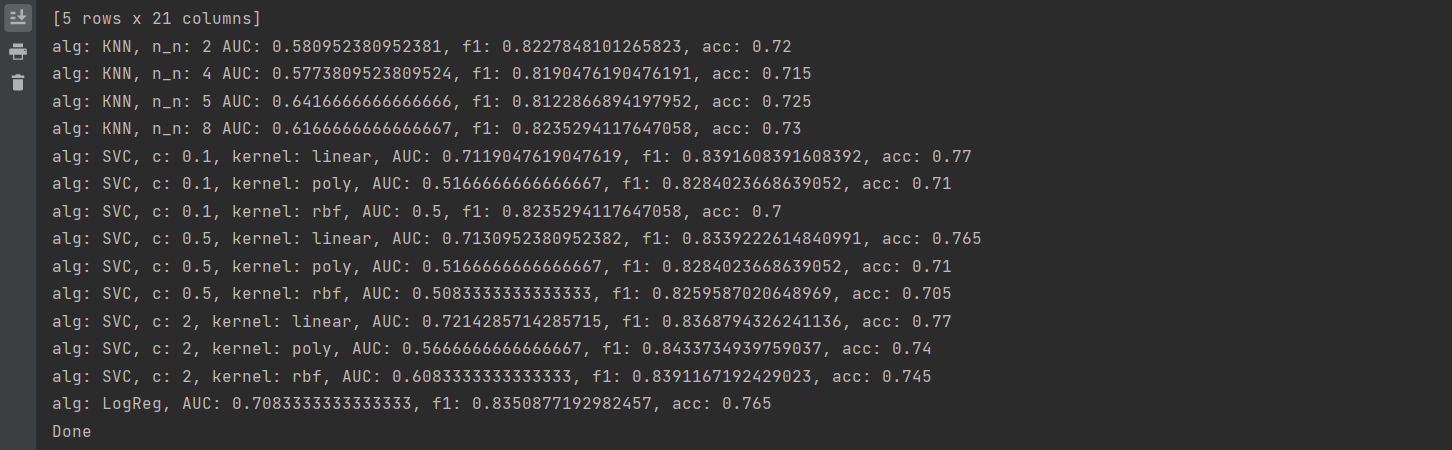
1. Lépésként alaposan elolvastam a „*data\_description.txt*” fájlt, hogy megértsem milyen adatokból áll az adatsorom.
2. Importáltam az összes fontosabb könyvtárat pl. „*import pandas as pd*” vagy „*import numpy as np*” stb.
3. Lépésként beimportáltam az adatokat, majd az eredeti header-ket kicseréltem számomra érthetőbbekre, majd „skiprows=1” paranccsal kihagytam a beolvasásból az eredeti oszlopneveket.  
   Kiírattam az beolvasott adat sorainak, oszlopainak számát valamint az első 5 sorát példaként. Majd a Preprocessing előkészítéséhez kiírattam az adatokról való információkat:
4. Mivel itt még nem tudom, hogy melyik adatokkal fogok foglalkozni ezért a Preprocessing-et (előfeldolgozást) az összeg oszlopon elvégeztem. Valamint az összeg régi oszlopot töröltem.  
   Először vettem a Nominális adatokat (szöveges adatok melyek típusai között nincs alá-fölé rendelő viszony)

A kevés változós adatokat One Hot Encoderrel a nem nominális szöveges adatokat pedig Ordinal Encoderrel dolgoztam fel.  
  
Az alapból számokból álló adataimat pedig skálázom, hogy a gépnek kisebb adatokkal kelljen dolgoznia.

1. Grafikusan ábrázoltam 1-2 adatsor közötti kapcsolatot:

1. Létrehoztam a klasszifikációhoz szükséges „visualize” függvényt melynek hatására vizuálisan is képként láthatom a klasszifikáció eredményét.   
   Valamint utána magát a klasszifikáció függvényt.  
   Megadtam random hiperparamétereket melyeken utána elvégeztem és kirajzoltam az eredeti adathalmazon a klasszifikációt. Melynek eredménye a következő oldalon látható.
2. 
3. Ezután létrehoztam egy teszt adatsort melyen betanítom a programot, azaz azon futtatom majd le a klasszifikációt először.
4. Majd lefuttattam mind a teszt mind az előfeldolgozott adatsoron 3 fajta klasszifikációt.
5.  Ezekkel hiperparaméterekkel a legjobb eredményt a KNN algoritmus hozta, amikor az n\_neighbors értéke 4 volt:  
   **AUC: 0.5702380952380953 f1: 0.8115015974440895 acc: 0.705**
6. Viszont én ezeket az értékeket nem találtam eléggé megfelelőnek emiatt próbálgatások árán megkerestem a legmegfelelőbb hiperparamétereket.   
   n\_dims-et felnöveltem 13-ra figyelve hogy ne legyen túltanulás, azaz se az AUC se az f1 értéke ne menjen 0,85 fölé.
7. Ezek az adatok születtek meg:
8. Akkor lesz a legjobb a klasszifikáció, hogyha az SVC algoritmust használom c=2 kerner=linear hiperparaméterekkel.  
   **AUC: 0.7214285714285715 f1: 0.8368794326241136 acc: 0.77**