IAU Projekt 2.fáza

Autori: Peter Plevko (50%), Radovan Cyprich (50%)

Dátum: 21.11.2021

In [1]:

```
import matplotlib.pyplot as plt
import seaborn as sns
import pandas as pd
import numpy as np
import matplotlib as mat
import matplotlib.pylab as pylab
import statsmodels.api as sm
from sklearn.preprocessing import FunctionTransformer
import statsmodels.stats as sm_stats
import statsmodels.stats.api as sms
import scipy.stats as stats
from matplotlib import pyplot
from collections import Counter
from datetime import datetime, date
from sklearn.impute import KNNImputer
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import PowerTransformer
from sklearn.preprocessing import MinMaxScaler
from sklearn.feature_selection import VarianceThreshold
from sklearn.preprocessing import StandardScaler ,MinMaxScaler, PowerTransformer,QuantileTr
from sklearn.feature_selection import SelectKBest, f_classif
from sklearn.feature_selection import SelectKBest
from sklearn.feature_selection import mutual_info_regression
from sklearn.feature_selection import RFE
from sklearn.svm import SVR
pd.options.mode.chained_assignment = None
from pandas import read_csv
from sklearn.model_selection import train_test_split
from sklearn.feature_selection import SelectKBest
from sklearn.feature_selection import f_regression
from matplotlib import pyplot
from sklearn.compose import ColumnTransformer, make_column_transformer
from sklearn.pipeline import Pipeline, make_pipeline
```

Načítanie údajov z datasetu

In [2]:

```
ourdataset="061/profiles.csv"
profiles = pd.read_csv(ourdataset, sep='\t')
ourdataset="061/labor.csv"
labor = pd.read_csv(ourdataset, sep='\t')
merged = pd.merge(labor, profiles, on=["ssn", "name"])
```

Atribút unnamed

Ako prvé si z datasetu odstránime nepotrebný atribút *Unnamed*, ktorý nám len označuje index záznamu. Vymažeme si ho aj kvôli tomu aby sa nám podarilo zistiť, či sa v datasete nachádzajú nejaké duplicitné záznamy. Tento atribút nemá pre nás žiadnu výpovednú hodnotu.

In [3]:

```
merged.drop(['Unnamed: 0_x', 'Unnamed: 0_y'], axis=1, errors='ignore', inplace=True)
```

In [4]:

```
merged.head()
```

Out[4]:

	relationship	smoker	alp	weight	trombocyty	hematokrit	hemoglobin	er-cv	
0	divoced	yes	69.29754	96.36107	6.16009	5.43057	5.33106	51.19736	48.
1	single	yes	72.82074	48.08581	5.34360	5.93857	5.38394	55.26958	64.
2	widowed	no	20.71153	84.24071	7.86501	5.29297	8.64115	53.49391	55.
3	married	N	71.48080	78.61777	7.63083	4.34763	7.07119	57.12180	56.
4	separated	yes	77.14564	87.14201	7.56648	5.49149	6.93812	62.31427	41.

5 rows × 24 columns

Atribúty sex a smoker

Atribúty sex a smoker, ktoré nám vypovedajú o pravdivostných hodnotách jednotlivých vlastností pacientov, sme si nahradili číselnými atribútmi 1 a 0. Tieto atribúty neobsahujú, žiadne nan hodnoty.

In [5]:

```
print(len((merged[merged.sex.isnull()])))
```

```
0
```

In [6]:

```
print(len((merged[merged.smoker.isnull()])))
```

```
0
```

In [7]:

```
merged['sex'] = merged['sex'].str.replace('M',"1")
merged['sex'] = merged['sex'].str.replace('F',"0")
merged['smoker'] = merged['smoker'].str.replace('yes',"1")
merged['smoker'] = merged['smoker'].str.replace('no',"0")
merged['smoker'] = merged['smoker'].str.replace('Y',"1")
merged['smoker'] = merged['smoker'].str.replace('N',"0")
```

In [8]:

```
merged.sex.unique()
```

Out[8]:

```
array(['1', '0'], dtype=object)
```

In [9]:

```
merged.smoker.unique()
```

Out[9]:

```
array(['1', '0'], dtype=object)
```

Vytvorenie atribútu age podľa atribútu birthdate

In [10]:

```
def monthToNum(shortMonth):
    return {
            'Jan': '01',
            'Feb': '02',
            'Mar': '03',
            'Apr': '04',
            'May': '05',
            'Jun': '06',
            'Jul': '07',
            'Aug': '08',
            'Sep': '09',
            'Oct': '10',
            'Nov': '11',
            'Dec': '12'
    }[shortMonth]

for x in merged['birthdate']:

    # nahradim nulovy datum
    if(len(x)==20):
        removedZeros = x.replace(", 00:00:00", "")
        array = removedZeros.split("/")
        newString = array[2] + "-" + array[0] + "-" + array[1]
        merged['birthdate'] = merged['birthdate'].replace(x, newString)

    # nahradim lomitkovy datum
    elif(len(x)==10):
        merged['birthdate'] = merged['birthdate'].replace(x, x.replace("/", "-"))

    # menim slovo mesiac na cislo
    elif(len(x)==11):
        array = x.split(" ")
        newString = array[2] + "-" + monthToNum(array[1]) + "-" + array[0]
        merged['birthdate'] = merged['birthdate'].replace(x, newString)
```

In [11]:

```
def age(born):
    born = datetime.strptime(born, "%Y-%m-%d").date()
    today = date.today()
    return today.year - born.year - ((today.month,today.day) < (born.month,born.day))

merged['age'] = merged['birthdate'].apply(age)
```

In [12]:

```
merged.head()
```

Out[12]:

	relationship	smoker	alp	weight	trombocyty	hematokrit	hemoglobin	er-cv	
0	divoced	1	69.29754	96.36107	6.16009	5.43057	5.33106	51.19736	48.
1	single	1	72.82074	48.08581	5.34360	5.93857	5.38394	55.26958	64.
2	widowed	0	20.71153	84.24071	7.86501	5.29297	8.64115	53.49391	55.
3	married	0	71.48080	78.61777	7.63083	4.34763	7.07119	57.12180	56.
4	separated	1	77.14564	87.14201	7.56648	5.49149	6.93812	62.31427	41.

5 rows × 25 columns

Atribút relationship

Keďže strojové učenie nevie pracovať s nenumerickými hodnotami rozhodli sme sa rozdeliť si pacientov na tých, ktorí majú partnera a tých, čo nie. V stĺpci relationship nahradíme hodnoty za 1 pre pacientov, čo majú partnera a 0 pre tých,čo nie.

In [13]:

```
merged['relationship'] = merged['relationship'].str.replace('divoced',"divorced")
```

In [14]:

```
merged['relationship'] = merged['relationship'].str.replace('divorced',"0")
merged['relationship'] = merged['relationship'].str.replace('single',"0")
merged['relationship'] = merged['relationship'].str.replace('separated',"0")
merged['relationship'] = merged['relationship'].str.replace('widowed',"0")
merged['relationship'] = merged['relationship'].str.replace('nop',"0")
merged['relationship'] = merged['relationship'].str.replace('married',"1")
```

In [15]:

```
merged.relationship.unique()
```

Out[15]:

```
array(['0', '1'], dtype=object)
```

Atribút race

Chceme docieliť, aby sme aj so string atribútmi mohli pracovať s numerickými atribútmi, preto nahradíme jednotlivé typy rás za čísla.

Black - 1

Asian - 2

White - 3

Indian - 4

Hawaiian - 5

Nekonzistentné hodnoty najskôr nahradíme správnymi a následne jednotlivé typy rás pretransformujeme na numerické hodnoty.

In [16]:

```
merged['race'] = merged['race'].astype(str).str.replace('blsck',"Black")
merged['race'] = merged['race'].astype(str).str.replace('black',"Black")
merged['race'] = merged['race'].astype(str).str.replace('white',"White")
```

In [17]:

```
merged['race'] = merged['race'].str.replace('Black',"1")
merged['race'] = merged['race'].str.replace('Asian',"2")
merged['race'] = merged['race'].str.replace('White',"3")
merged['race'] = merged['race'].str.replace('Indian',"4")
merged['race'] = merged['race'].str.replace('Hawaiian',"5")
```

In [18]:

```
for x in merged['race']:
    if(x=='1'):
        newString = 1
        merged['race'] = merged['race'].replace(x, newString)
    elif(x=='2'):
        newString = 2
        merged['race'] = merged['race'].replace(x, newString)
    elif(x=='3'):
        newString = 3
        merged['race'] = merged['race'].replace(x, newString)
    elif(x=='4'):
        newString = 4
        merged['race'] = merged['race'].replace(x, newString)
    elif(x=='5'):
        newString = 5
        merged['race'] = merged['race'].replace(x, newString)
```

In [19]:

```
merged.race.unique()
```

Out[19]:

```
array([1, 2, 3, 4, 5], dtype=int64)
```

Atribút blood_group

Atribút, ktorý nám sprostredkuváva informáciu o type krvnej skupiny pacienta, taktiež pretransformujeme na numerické hodnoty následovným spôsobom.

A- = -1

A+ = 1

B- = -2

B+ = 2

AB- = -3

AB+ = 3

O- = -4

O+ = 4

In [20]:

```
merged['blood_group'] = merged['blood_group'].str.replace('AB-',"-3")
merged['blood_group'] = merged['blood_group'].str.replace('AB+','3')

merged['blood_group'] = merged['blood_group'].str.replace('A-',"-1")
merged['blood_group'] = merged['blood_group'].str.replace('A+',"1")

merged['blood_group'] = merged['blood_group'].str.replace('B-',"-2")
merged['blood_group'] = merged['blood_group'].str.replace('B+',"2")

merged['blood_group'] = merged['blood_group'].str.replace('O-',"-4")
merged['blood_group'] = merged['blood_group'].str.replace('O+',"4")
```

```
C:\Users\pplev\AppData\Local\Temp/ipykernel_12248/4292570525.py:2: FutureWar
ning: The default value of regex will change from True to False in a future
version.
  merged['blood_group'] = merged['blood_group'].str.replace('AB+','3')
C:\Users\pplev\AppData\Local\Temp/ipykernel_12248/4292570525.py:5: FutureWar
ning: The default value of regex will change from True to False in a future
version.
  merged['blood_group'] = merged['blood_group'].str.replace('A+',"1")
C:\Users\pplev\AppData\Local\Temp/ipykernel_12248/4292570525.py:8: FutureWar
ning: The default value of regex will change from True to False in a future
version.
  merged['blood_group'] = merged['blood_group'].str.replace('B+',"2")
C:\Users\pplev\AppData\Local\Temp/ipykernel_12248/4292570525.py:11: FutureWa
rning: The default value of regex will change from True to False in a future
version.
  merged['blood_group'] = merged['blood_group'].str.replace('O+',"4")
```

In [21]:

```
for x in merged['blood_group']:
    if(x=='1+'):
        newString = 1
        merged['blood_group'] = merged['blood_group'].replace(x, newString)
    elif(x=='2+'):
        newString = 2
        merged['blood_group'] = merged['blood_group'].replace(x, newString)
    elif(x=='3+'):
        newString = 3
        merged['blood_group'] = merged['blood_group'].replace(x, newString)
    elif(x=='4+'):
        newString = 4
        merged['blood_group'] = merged['blood_group'].replace(x, newString)
```

In [22]:

```
merged.blood_group.unique()
```

Out[22]:

```
array(['-2', '-1', 1, 3, '-3', '-4', 2, 4], dtype=object)
```

Atribút weight

V stĺpci *weight*, ktorý reprezentuje hodnotu váhy človeka sme sa rozhodli vylúčiť hodnoty menšie ako 3, keďže

novorodenci majú pri narodení okolo troch kíl. Počet týchto záznamov je 241.

In [23]:

```
minusWeight = merged.loc[merged['weight'] < 3]
print(len(minusWeight))
index_weight = merged[ merged['weight'] < 3 ].index
merged.drop(index_weight, inplace = True)
```

```
241
```

Atribút job

V tomto stĺpci sa nachádza veľmi veľa podobných hodnôt, ako napríklad viacero typov učiteľov, inžinierov, doktorov, IT špecialostov a ďalších, avšak povolania nám ponúkajú jedinečnú informáciu. Preto sme sa rozhodli tento atribút netransformovať.

Ako máme možnosť vidieť v datasete sa nachádza

In [24]:

```
a=len(merged['job'].unique())
merged['job'].unique()
print("Počet unikátnych jobov: " +str(a))
print(merged['job'].unique())
```

```
Počet unikátnych jobov: 636
['Information officer' 'Mudlogger' 'Runner, broadcasting/film/video'
 'Technical sales engineer' 'Scientific laboratory technician'
 'Television camera operator' 'Fast food restaurant manager'
 'Medical technical officer' 'Writer' 'Leisure centre manager'
 'Nature conservation officer' 'Advertising account planner'
 'Air cabin crew' 'Special effects artist' 'Passenger transport manager'
 'Surveyor, minerals' 'Publishing rights manager' 'Financial trader'
 'Conference centre manager' 'Consulting civil engineer'
 'Further education lecturer' 'Conservator, furniture' 'Firefighter'
 'Engineer, production' 'Scientist, audiological' 'Engineering geologist'
 'Health and safety adviser' 'Occupational hygienist'
 'Engineer, agricultural' 'Photographer' 'Administrator, arts'
 'Engineer, structural' 'Primary school teacher' 'Barista'
 'Aeronautical engineer' 'Operations geologist' 'Customer service manager'
 'Museum/gallery exhibitions officer' 'Corporate investment banker'
 'Newspaper journalist' 'Medical laboratory scientific officer'
 'Chartered public finance accountant' 'Sales promotion account executive'
 'Child psychotherapist' 'Insurance broker'
```

In [25]:

```
merged.head()
```

Out[25]:

	relationship	smoker	alp	weight	trombocyty	hematokrit	hemoglobin	er-cv	
0	0	1	69.29754	96.36107	6.16009	5.43057	5.33106	51.19736	48.
1	0	1	72.82074	48.08581	5.34360	5.93857	5.38394	55.26958	64.
2	0	0	20.71153	84.24071	7.86501	5.29297	8.64115	53.49391	55.
3	1	0	71.48080	78.61777	7.63083	4.34763	7.07119	57.12180	56.
4	0	1	77.14564	87.14201	7.56648	5.49149	6.93812	62.31427	41.

5 rows × 25 columns

Nahradenie chýbajúcich hodnôt

Ako máme možnosť vidieť, v niektorých stĺpcoch máme chýbajúce hodnoty a preto ich musíme nahradiť, zmysluplnými technikami.

In [26]:

```
merged.isnull().sum()
```

Out[26]:

```
relationship     0
smoker           0
alp             30
weight           0
trombocyty      28
hematokrit      28
hemoglobin      32
er-cv           30
ast             29
ssn              0
erytrocyty      30
indicator        0
hbver           29
leukocyty       31
alt             29
etytr           30
name             0
residence        0
blood_group      0
birthdate        0
address          0
race             0
job              0
sex              0
age              0
dtype: int64
```

Z minulej fázy vieme, že niektoré atribúty majú veľmi nízku koreláciu s ostatnými tak preto sme sa rozhodli, že nám nebudú chýbať a odstránime ich.

In [27]:

```
def drop_na(data):
    data=data.dropna(subset=['er-cv','hbver','etytr','ast'])
    return data
merged=drop_na(merged)
```

Chýbajúce hodnoty v stĺpcoch *leukocyty, erytrocyty a trombocyty* sme nahradili ich priemerom.

In [28]:

```
def replace_for_mean(data):
    leukocytyMean = data['leukocyty'].mean()
    data['leukocyty'] = data['leukocyty'].fillna(leukocytyMean)
    erytrocytyMean = data['erytrocyty'].mean()
    data['erytrocyty'] = data['erytrocyty'].fillna(erytrocytyMean)
    trombocytyMean = data['trombocyty'].mean()
    data['trombocyty'] = data['trombocyty'].fillna(trombocytyMean)
mean=replace_for_mean(merged)
```

Chýbajúce hodnoty v stĺpcoch alt a alp sme nahradili mediánom týchto hodnôt na základe ich korelácie z minulej fázy, pre spestrenie dát.

In [29]:

```
def replace_for_median(data):
    altMedian = data['alt'].median()
    data['alt'] = data['alt'].fillna(altMedian)
    alpMedian = data['alp'].median()
    data['alp'] = data['alp'].fillna(alpMedian)
median=replace_for_median(merged)
```

Využili sme KNN algorytmus na transformovanie chýbajúcich hodnôt v stĺpcoch hematokrit a hemoglobin.

In [30]:

```
def replace_for_KNN(data):
    imputer = KNNImputer()
    imputed_data = pd.DataFrame(imputer.fit_transform(data[['hematokrit','hemoglobin']]))
    data['hematokrit']=imputed_data[0].values
    data['hemoglobin']=imputed_data[1].values
knn=replace_for_KNN(merged)
```

Na odstránenie nullových hodnôt sme použili pipeline, kde sme postupne nahádzali všetky metódy, ktorými sme nahradzovanie vykonávali.

In [31]:

```
pajp = Pipeline([("mean", mean),
                 ("median", median),
                 ("knn", knn)])
x=pajp.fit_transform(merged)
x
```

Out[31]:

	relationship	smoker	alp	weight	trombocyty	hematokrit	hemoglobin	er-cv
0	0	1	69.29754	96.36107	6.16009	5.43057	5.33106	51.19736
1	0	1	72.82074	48.08581	5.34360	5.93857	5.38394	55.26958
2	0	0	20.71153	84.24071	7.86501	5.29297	8.64115	53.49391
3	1	0	71.48080	78.61777	7.63083	4.34763	7.07119	57.12180
4	0	1	77.14564	87.14201	7.56648	5.49149	6.93812	62.31427
...	...	...	...	...	...	...	...	...
10035	0	0	74.89102	82.05553	7.67858	4.75737	7.30577	53.86889
10036	0	1	29.36845	69.77004	8.13976	5.27482	8.35276	53.56962
10037	0	1	83.47303	91.44586	8.38149	2.79353	6.55380	61.89518
10039	0	1	32.35026	91.62537	7.99131	5.47887	4.27690	45.04191
10040	0	0	68.98585	29.14273	4.55499	1.70930	5.33157	46.23457

9683 rows × 25 columns

In [32]:

```
merged.isnull().sum()
```

Out[32]:

```
relationship    0
smoker          0
alp             0
weight          0
trombocyty      0
hematokrit      0
hemoglobin      0
er-cv           0
ast             0
ssn             0
erytrocyty      0
indicator       0
hbver           0
leukocyty       0
alt             0
etytr           0
name            0
residence       0
blood_group     0
birthdate       0
address         0
race            0
job             0
sex             0
age             0
dtype: int64
```

Odstráňovanie duplicitných záznamov

Odstránili sme 98 záznamov, ktoré boli identické s nejakým iným záznamom z datasetu.

In [33]:

```
duplicates = merged[merged.duplicated()]
merged = merged.drop_duplicates()
print("Počet záznamov v datasete: "+str(len(merged)))
print("V datasete sa nachádza: " + str(len(duplicates)) + " duplikátov.")
```

```
Počet záznamov v datasete: 9585
V datasete sa nachádza: 98 duplikátov.
```

Nachádza sa tu aj viac rovnakých záznamov o jednom pacientovi alebo jednoducho sú to len menovci.

In [34]:

```
merged['name'].value_counts()
```

Out[34]:

```
Michael Martin      10
Patricia Holmes     10
Daniel Smith        10
Richard Johnson      9
James Robinson       9
                    ..
Jodi Thornton        1
Garrett Walker       1
Stacy Brooks         1
Theresa Fox          1
Aaron Williamson     1
Name: name, Length: 2984, dtype: int64
```

Odstránenie nepotrebných stĺpcov a záznamov

Po spojení datasetov sme zistili, že niektoré stĺpce nemajú pre nás žiadnu výpovedvnú hodnotu, vzhľadom na koreláciu voči ostatným atribútom preto sme sa rozhodli odstrániť stĺpce *residence* a *address*.

In [35]:

```
merged.drop(['residence', 'address'], axis=1, errors='ignore', inplace=True)
```

In [36]:

```
merged.head()
```

Out[36]:

	relationship	smoker	alp	weight	trombocyty	hematokrit	hemoglobin	er-cv	
0	0	1	69.29754	96.36107	6.16009	5.43057	5.33106	51.19736	48.
1	0	1	72.82074	48.08581	5.34360	5.93857	5.38394	55.26958	64.
2	0	0	20.71153	84.24071	7.86501	5.29297	8.64115	53.49391	55.
3	1	0	71.48080	78.61777	7.63083	4.34763	7.07119	57.12180	56.
4	0	1	77.14564	87.14201	7.56648	5.49149	6.93812	62.31427	41.

5 rows × 23 columns

Všimli sme si nezmysel v dátach a to, že niektorí pacienti majú status married aj ked maju menej ako 16 rokov

preto sme sa rozhodli tieto záznamy odstrániť, keďže nám produkujú nelogickú informáciu a pravdepodobne sú to preklepy.

In [37]:

```
print(len(merged[(merged.age <16)&(merged.relationship == 1)]))
index_ages = merged[(merged.age <16)&(merged.relationship == 1)].index
merged.drop(index_ages, inplace = True)
```

```
0
```

Odstráňovanie vychýlených hodnôt

Prejdeme si všetky stĺpce ako sú na tom vychýlené hodnoty a potom využijeme funckiu, ktorou vychýlení hodnoty priradíme do kvantilov. Vo funkcii sa využíva transformácia pomocou logaritmu a zároveň 5 a 95 percentil.

In [38]:

```
def outliers(inputed_data, column):
    data = inputed_data.copy(deep = True)
    value = stats.skew(data[column])

    if ((value < -2) or (value > 2)):
        minimum = data[column].min()
        minimum = minimum + (-minimum - minimum)
        data[column] = np.log(data[column]+minimum)
    perc_95 = data[column].quantile(.95)
    perc_05 =  data[column].quantile(.05)
    data.loc[data[column] < perc_05, column] = perc_05
    data.loc[data[column] > perc_95, column] = perc_95
    return data

def remove_outlier(data, column_name):
    q05, q95 = data[column_name].quantile(0.05), data[column_name].quantile(0.95)
    q = q95 - q05
    remove = q * 1.5
    lower, upper = q05 - remove, q95 + remove
    return data.loc[(data[column_name] > lower) & (data[column_name] < upper)]
```

In [39]:

```
merged.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 9585 entries, 0 to 10040
Data columns (total 23 columns):
 #   Column        Non-Null Count  Dtype  
---  ------        --------------  -----  
 0   relationship  9585 non-null   object 
 1   smoker        9585 non-null   object 
 2   alp           9585 non-null   float64
 3   weight        9585 non-null   float64
 4   trombocyty    9585 non-null   float64
 5   hematokrit    9585 non-null   float64
 6   hemoglobin    9585 non-null   float64
 7   er-cv         9585 non-null   float64
 8   ast           9585 non-null   float64
 9   ssn           9585 non-null   object 
 10  erytrocyty    9585 non-null   float64
 11  indicator     9585 non-null   float64
 12  hbver         9585 non-null   float64
 13  leukocyty     9585 non-null   float64
 14  alt           9585 non-null   float64
 15  etytr         9585 non-null   float64
 16  name          9585 non-null   object 
 17  blood_group   9585 non-null   object 
 18  birthdate     9585 non-null   object 
 19  race          9585 non-null   int64  
 20  job           9585 non-null   object 
 21  sex           9585 non-null   object 
 22  age           9585 non-null   int64  
dtypes: float64(13), int64(2), object(8)
memory usage: 2.0+ MB
```

Na histogramoch môžeme vidieť, že sa nám v niektorých stĺpcoch nachádzajú vychýlené hodnoty preto ich potrebujeme transformovať ako už bolo spomenuté vyššie.

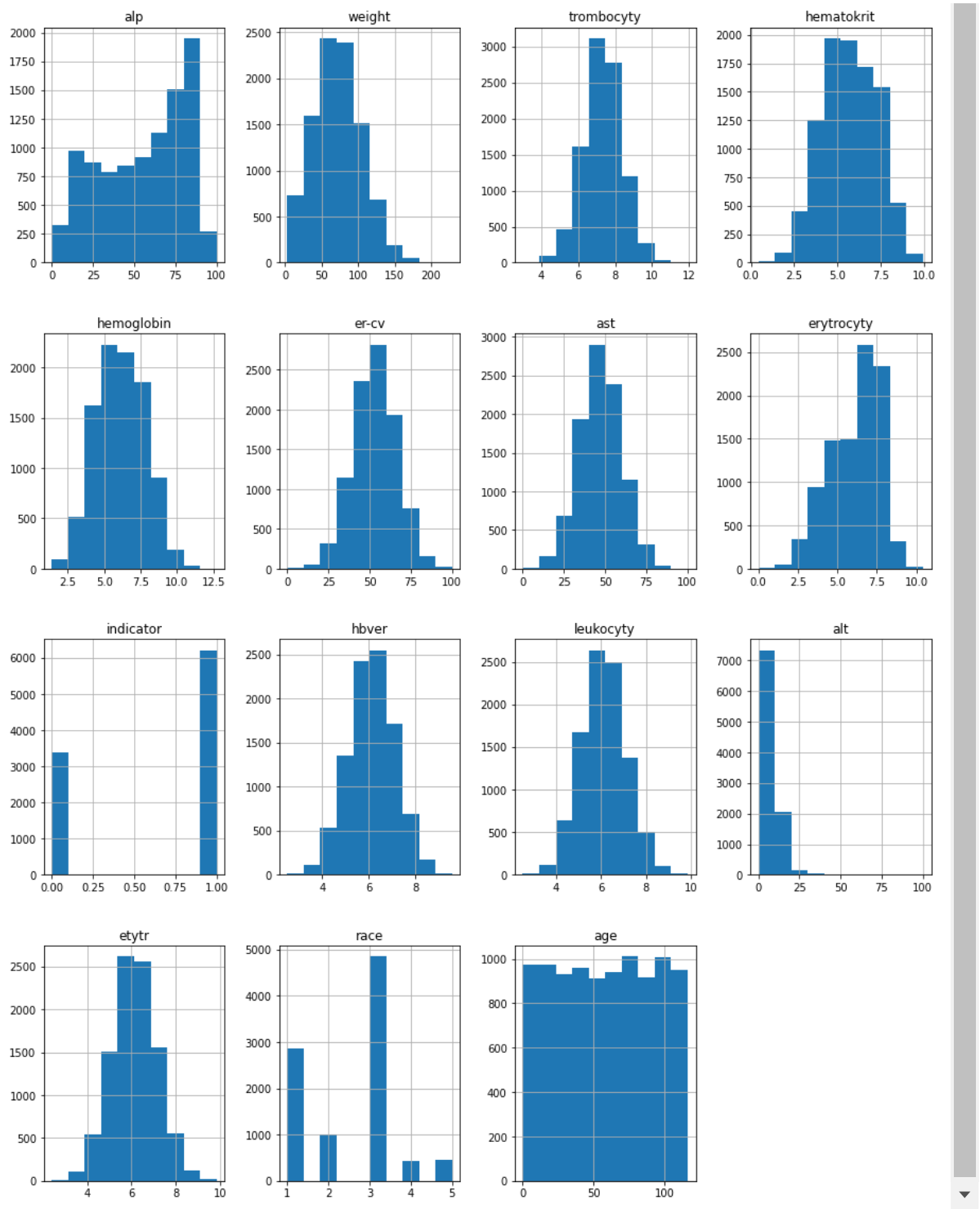
In [40]:

```
fig = plt.figure(figsize = (15,20))
ax = fig.gca()
merged.hist(ax = ax)
```

```
C:\Users\pplev\AppData\Local\Temp/ipykernel_12248/1167618414.py:3: UserWarni
ng: To output multiple subplots, the figure containing the passed axes is be
ing cleared
  merged.hist(ax = ax)
```

Out[40]:

```
array([[<AxesSubplot:title={'center':'alp'}>,
        <AxesSubplot:title={'center':'weight'}>,
        <AxesSubplot:title={'center':'trombocyty'}>,
        <AxesSubplot:title={'center':'hematokrit'}>],
       [<AxesSubplot:title={'center':'hemoglobin'}>,
        <AxesSubplot:title={'center':'er-cv'}>,
        <AxesSubplot:title={'center':'ast'}>,
        <AxesSubplot:title={'center':'erytrocyty'}>],
       [<AxesSubplot:title={'center':'indicator'}>,
        <AxesSubplot:title={'center':'hbver'}>,
        <AxesSubplot:title={'center':'leukocyty'}>,
        <AxesSubplot:title={'center':'alt'}>],
       [<AxesSubplot:title={'center':'etytr'}>,
        <AxesSubplot:title={'center':'race'}>,
        <AxesSubplot:title={'center':'age'}>, <AxesSubplot:>]],
      dtype=object)
```



Transformovať sme sa rozhodli všetky stĺpce, ktoré obsahujú merané numerické hodnoty. Taktiež použijeme túto techniku aj na nami vytvorený stĺpec age.

In [41]:

```
cols= ['alp','erytrocyty','etytr', 'hbver', 'ast', 'er-cv','age','alt','hemoglobin', 'hemat
for column_name in cols:
    merged = remove_outlier(merged,column_name)
```

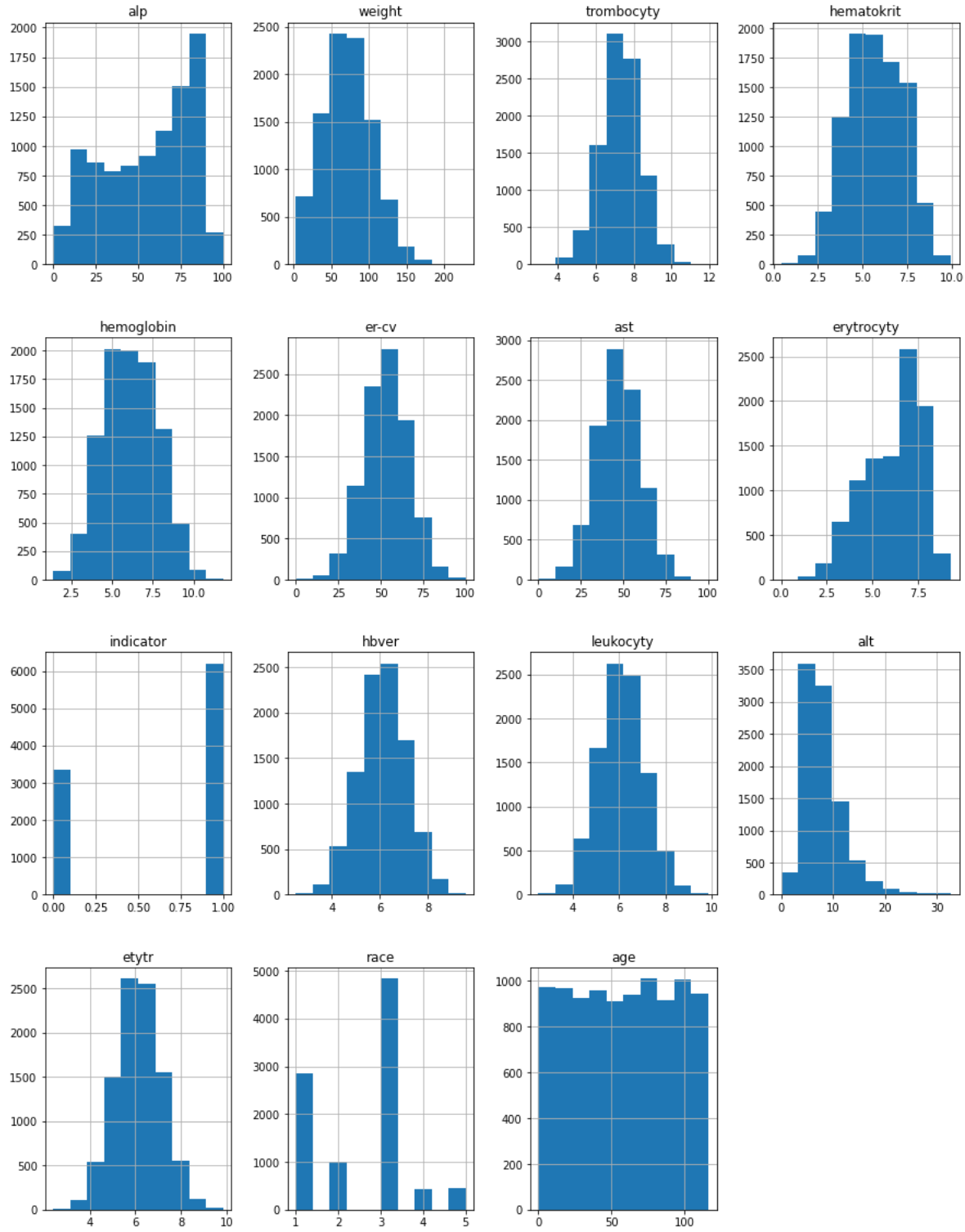
In [42]:

```
fig = plt.figure(figsize = (15,20))
ax = fig.gca()
merged.hist(ax = ax)
```

```
C:\Users\pplev\AppData\Local\Temp/ipykernel_12248/1167618414.py:3: UserWarni
ng: To output multiple subplots, the figure containing the passed axes is be
ing cleared
  merged.hist(ax = ax)
```

Out[42]:

```
array([[<AxesSubplot:title={'center':'alp'}>,
        <AxesSubplot:title={'center':'weight'}>,
        <AxesSubplot:title={'center':'trombocyty'}>,
        <AxesSubplot:title={'center':'hematokrit'}>],
       [<AxesSubplot:title={'center':'hemoglobin'}>,
        <AxesSubplot:title={'center':'er-cv'}>,
        <AxesSubplot:title={'center':'ast'}>,
        <AxesSubplot:title={'center':'erytrocyty'}>],
       [<AxesSubplot:title={'center':'indicator'}>,
        <AxesSubplot:title={'center':'hbver'}>,
        <AxesSubplot:title={'center':'leukocyty'}>,
        <AxesSubplot:title={'center':'alt'}>],
       [<AxesSubplot:title={'center':'etytr'}>,
        <AxesSubplot:title={'center':'race'}>,
        <AxesSubplot:title={'center':'age'}>, <AxesSubplot:>]],
      dtype=object)
```



Realizácia predspracovania dát

Najskôr si prehodíme všetky numerické atribúty, ktoré sme pretransformovali zo strginov na typ numeric aby sme s nimi mohli ďalej pracovať.

In [43]:

```
merged["smoker"] = pd.to_numeric(merged["smoker"])
merged["relationship"] = pd.to_numeric(merged["relationship"])
merged["race"] = pd.to_numeric(merged["race"])
merged["sex"] = pd.to_numeric(merged["sex"])
merged["smoker"] = pd.to_numeric(merged["smoker"])
merged["blood_group"] = pd.to_numeric(merged["blood_group"])
merged.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 9553 entries, 0 to 10040
Data columns (total 23 columns):
 #   Column        Non-Null Count  Dtype  
---  ------        --------------  -----  
 0   relationship  9553 non-null   int64  
 1   smoker        9553 non-null   int64  
 2   alp           9553 non-null   float64
 3   weight        9553 non-null   float64
 4   trombocyty    9553 non-null   float64
 5   hematokrit    9553 non-null   float64
 6   hemoglobin    9553 non-null   float64
 7   er-cv         9553 non-null   float64
 8   ast           9553 non-null   float64
 9   ssn           9553 non-null   object 
 10  erytrocyty    9553 non-null   float64
 11  indicator     9553 non-null   float64
 12  hbver         9553 non-null   float64
 13  leukocyty     9553 non-null   float64
 14  alt           9553 non-null   float64
 15  etytr         9553 non-null   float64
 16  name          9553 non-null   object 
 17  blood_group   9553 non-null   int64  
 18  birthdate     9553 non-null   object 
 19  race          9553 non-null   int64  
 20  job           9553 non-null   object 
 21  sex           9553 non-null   int64  
 22  age           9553 non-null   int64  
dtypes: float64(13), int64(6), object(4)
memory usage: 1.7+ MB
```

Rozhodli sme sa rozdeliť si dataset na testovaciu a trénovaciu množinu v pomere 1:4 teda 20% dát tvoria testovacie dáta a 80% trénovacie dáta. Ďalej budeme spracovávať trénovaciu vzorku.

In [44]:

```
len(merged)
```

Out[44]:

```
9553
```

In [45]:

```
train_data, test_data = train_test_split(merged, test_size=0.2)
print('Trénovací dataset obsahuje:'+ str(len(train_data))+' záznamov\n'+'Testovacia vzorka
```

```
Trénovací dataset obsahuje:7642 záznamov
Testovacia vzorka obsahuje:1911 záznamov
```

Podľa histogramov po odstránení outlierov sme zistili, že niektoré hodnoty sú z iného ako normálového rozdelenia, preto sme si rozdelili atribúty podľa distribúcie.

In [46]:

```
skewed=['alp','erytrocyty']
transformed_atributes=['sex','race','blood_group','smoker','relationship','indicator']
gaussian=['weight', 'trombocyty', 'hematokrit', 'hemoglobin', 'er-cv', 'ast', 'alt', 'hbver
```

Môžeme vidieť, že atribúty alp a alt majú nepravidelne rozdielné hodnoty, v jednotlivých kvantiloch. Preto ich transformujeme pomocou quantilového transformera.

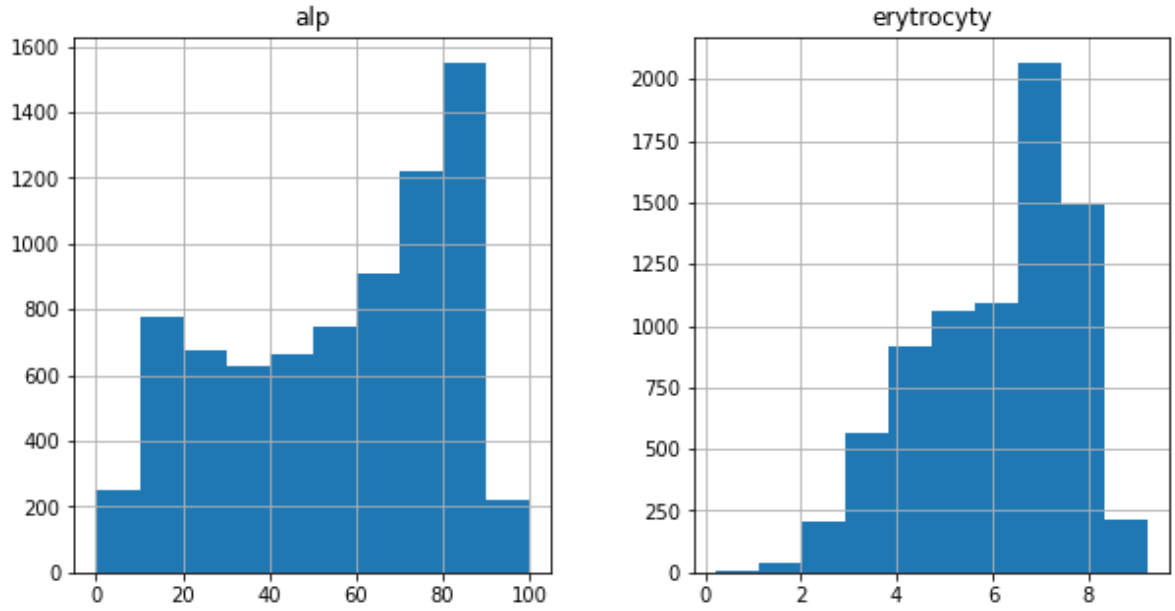
In [47]:

```
fig = plt.figure(figsize = (10,5))
ax = fig.gca()
train_data[skewed].hist(ax = ax)
```

```
C:\Users\pplev\AppData\Local\Temp/ipykernel_12248/1707975513.py:3: UserWarni
ng: To output multiple subplots, the figure containing the passed axes is be
ing cleared
  train_data[skewed].hist(ax = ax)
```

Out[47]:

```
array([[<AxesSubplot:title={'center':'alp'}>,
        <AxesSubplot:title={'center':'erytrocyty'}>]], dtype=object)
```



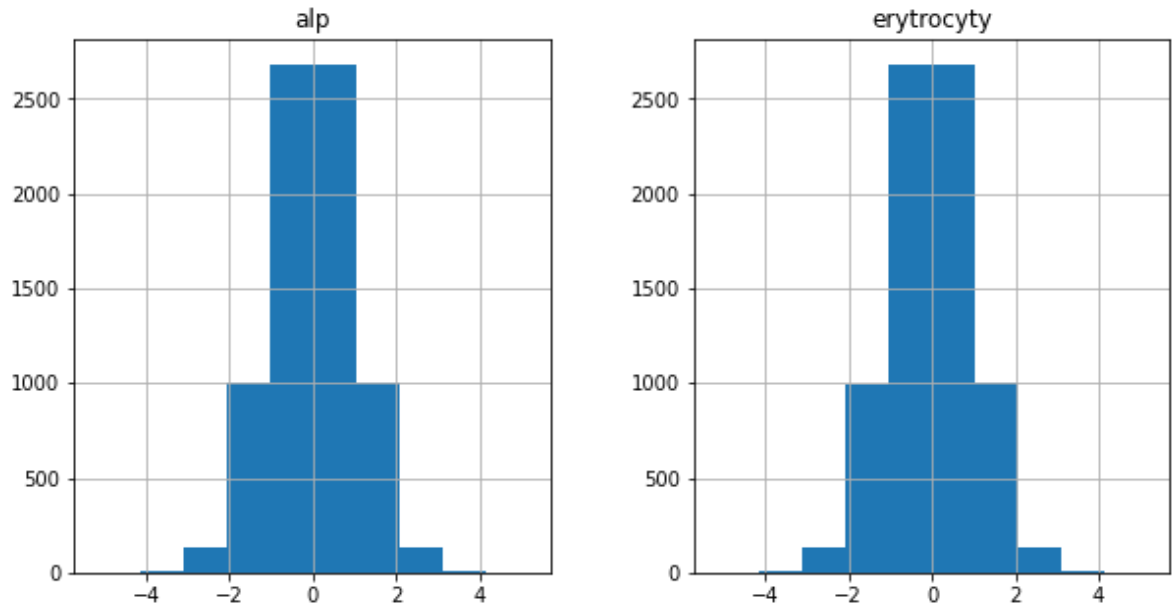
In [48]:

```
qt = QuantileTransformer(output_distribution="normal",n_quantiles=1000)
train_data[skewed] = qt.fit_transform(train_data[skewed])
fig = plt.figure(figsize = (10,5))
ax = fig.gca()
train_data[skewed].hist(ax = ax)
```

```
C:\Users\pplev\AppData\Local\Temp/ipykernel_12248/2116364140.py:5: UserWarni
ng: To output multiple subplots, the figure containing the passed axes is be
ing cleared
  train_data[skewed].hist(ax = ax)
```

Out[48]:

```
array([[<AxesSubplot:title={'center':'alp'}>,
        <AxesSubplot:title={'center':'erytrocyty'}>]], dtype=object)
```



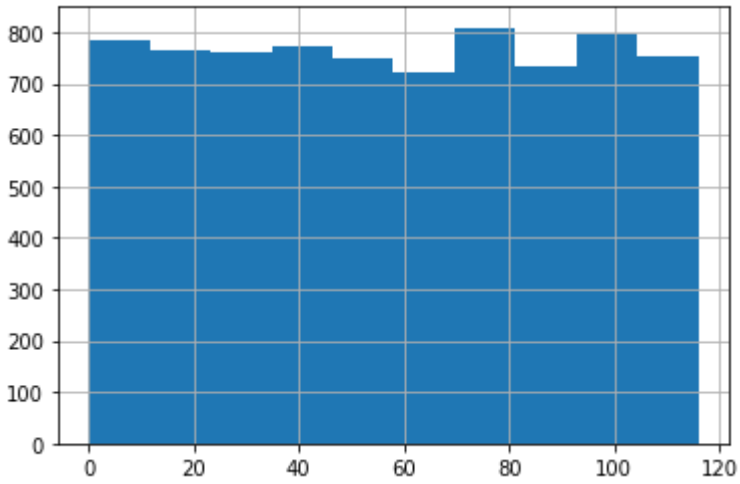
Nami vytvorený atribút *age* potrebujeme taktiež transformovať ale potrebujeme kladné hodnoty takže použijeme vekový priemer.

In [49]:

```
train_data['age'].hist()
```

Out[49]:

```
<AxesSubplot:>
```



In [50]:

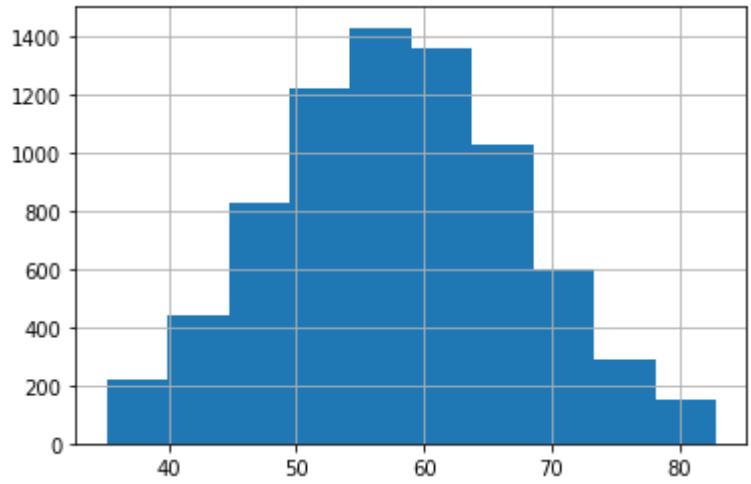
```
age_mean = np.mean(train_data[['age']])
train_data[['age']] = qt.fit_transform(train_data[['age']])
train_data[['age']] *= 10
train_data[['age']] += age_mean
index_agess = train_data[(train_data.age <20)|(train_data.age > 85)].index
train_data.drop(index_agess, inplace = True)
```

In [51]:

```
train_data['age'].hist()
```

Out[51]:

```
<AxesSubplot:>
```



Následovné atribúty zobrazené v týchto histogramoch sú nami transformované stĺpce, na ktoré sme použili MinMaxScaler aj napriek tomu, že to nebolo nevynutné.

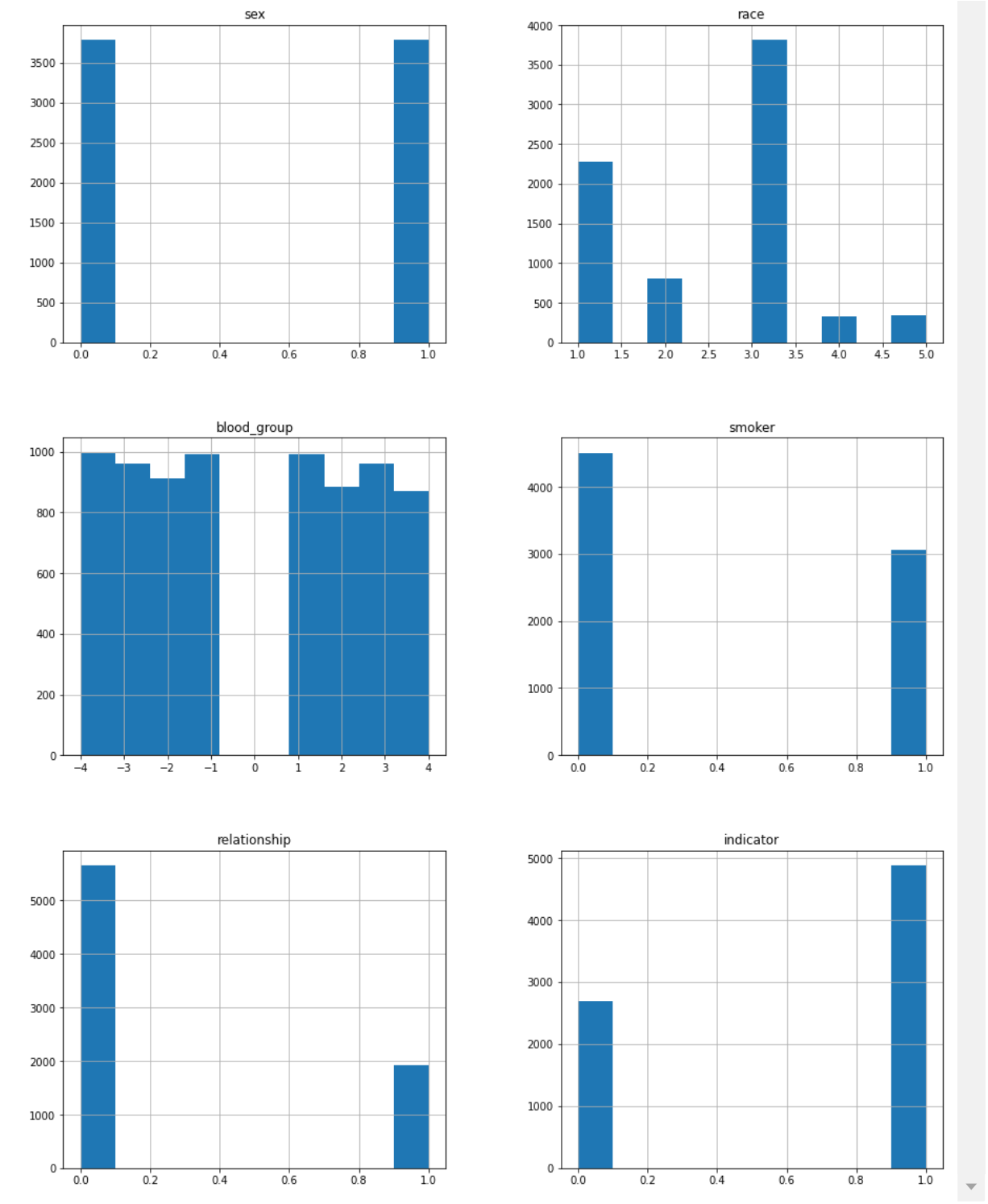
In [52]:

```
fig = plt.figure(figsize = (15,20))
ax = fig.gca()
train_data[transformed_atributes].hist(ax = ax)
```

```
C:\Users\pplev\AppData\Local\Temp/ipykernel_12248/3985480200.py:3: UserWarni
ng: To output multiple subplots, the figure containing the passed axes is be
ing cleared
  train_data[transformed_atributes].hist(ax = ax)
```

Out[52]:

```
array([[<AxesSubplot:title={'center':'sex'}>,
        <AxesSubplot:title={'center':'race'}>],
       [<AxesSubplot:title={'center':'blood_group'}>,
        <AxesSubplot:title={'center':'smoker'}>],
       [<AxesSubplot:title={'center':'relationship'}>,
        <AxesSubplot:title={'center':'indicator'}>]], dtype=object)
```



Na ostatné merané atribúty sme sa rozhodli použiť poweer transformation s metódou Yeo-Johnson. Keďže po odstránení outlierov hodnoty týchto atribútov sú nesymetricky rozdelené.

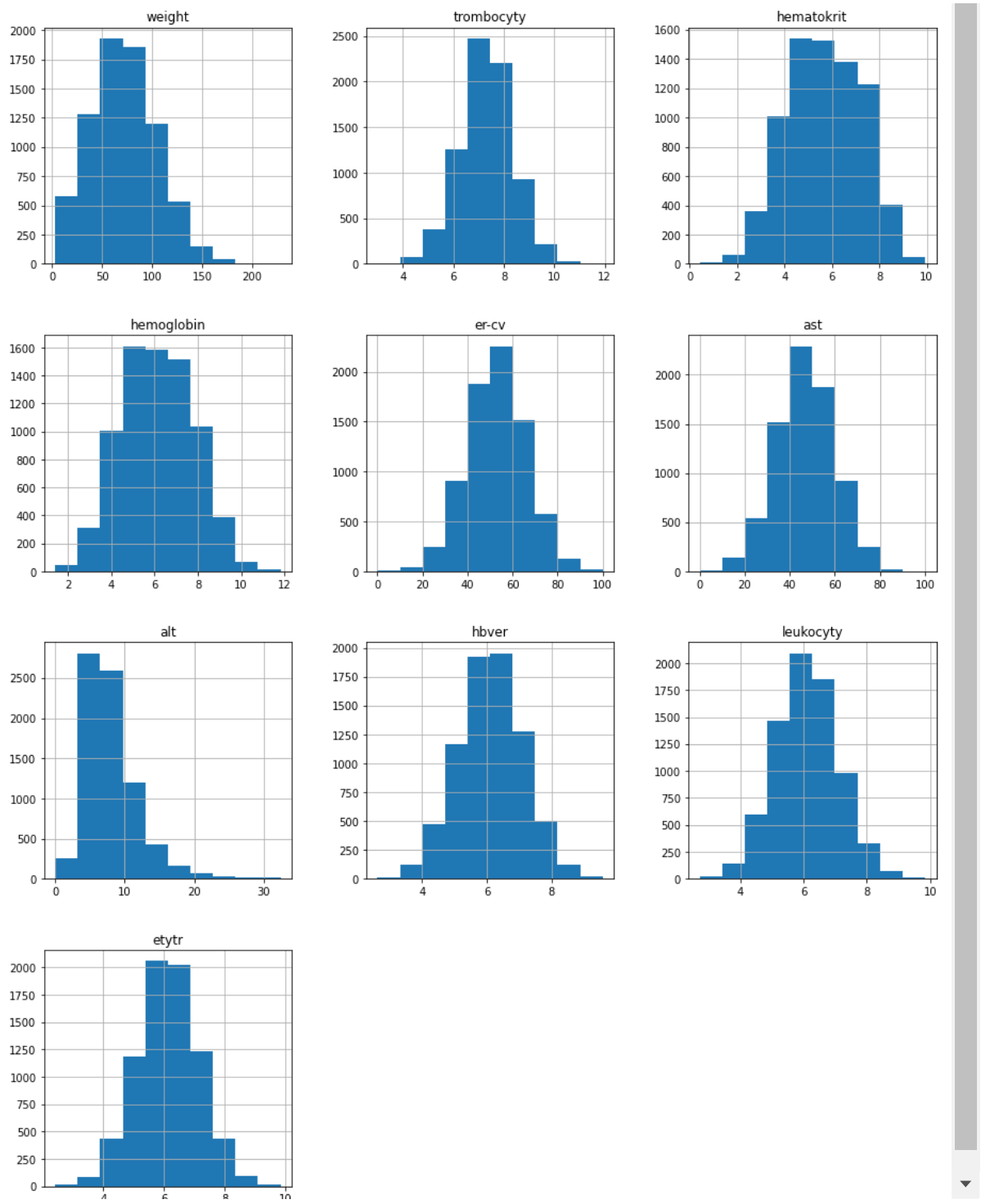
In [53]:

```
fig = plt.figure(figsize = (15,20))
ax = fig.gca()
train_data[gaussian].hist(ax = ax)
```

```
C:\Users\pplev\AppData\Local\Temp/ipykernel_12248/4084925039.py:3: UserWarni
ng: To output multiple subplots, the figure containing the passed axes is be
ing cleared
  train_data[gaussian].hist(ax = ax)
```

Out[53]:

```
array([[<AxesSubplot:title={'center':'weight'}>,
        <AxesSubplot:title={'center':'trombocyty'}>,
        <AxesSubplot:title={'center':'hematokrit'}>],
       [<AxesSubplot:title={'center':'hemoglobin'}>,
        <AxesSubplot:title={'center':'er-cv'}>,
        <AxesSubplot:title={'center':'ast'}>],
       [<AxesSubplot:title={'center':'alt'}>,
        <AxesSubplot:title={'center':'hbver'}>,
        <AxesSubplot:title={'center':'leukocyty'}>],
       [<AxesSubplot:title={'center':'etytr'}>, <AxesSubplot:>,
        <AxesSubplot:>]], dtype=object)
```



In [54]:

```
power = PowerTransformer(method='yeo-johnson', standardize=True)
train_data[gaussian] = power.fit_transform(train_data[gaussian])
```

Posledný krok pre normalizovanie atribútov je použiť standard scaler na všetky nekategorické atribúty, okrem atribútu vek, ten sme si upravili špeciálne.

In [55]:

```
standard_scaler = StandardScaler()
train_data[gaussian+skewed] = standard_scaler.fit_transform(train_data[gaussian+ skewed])
```

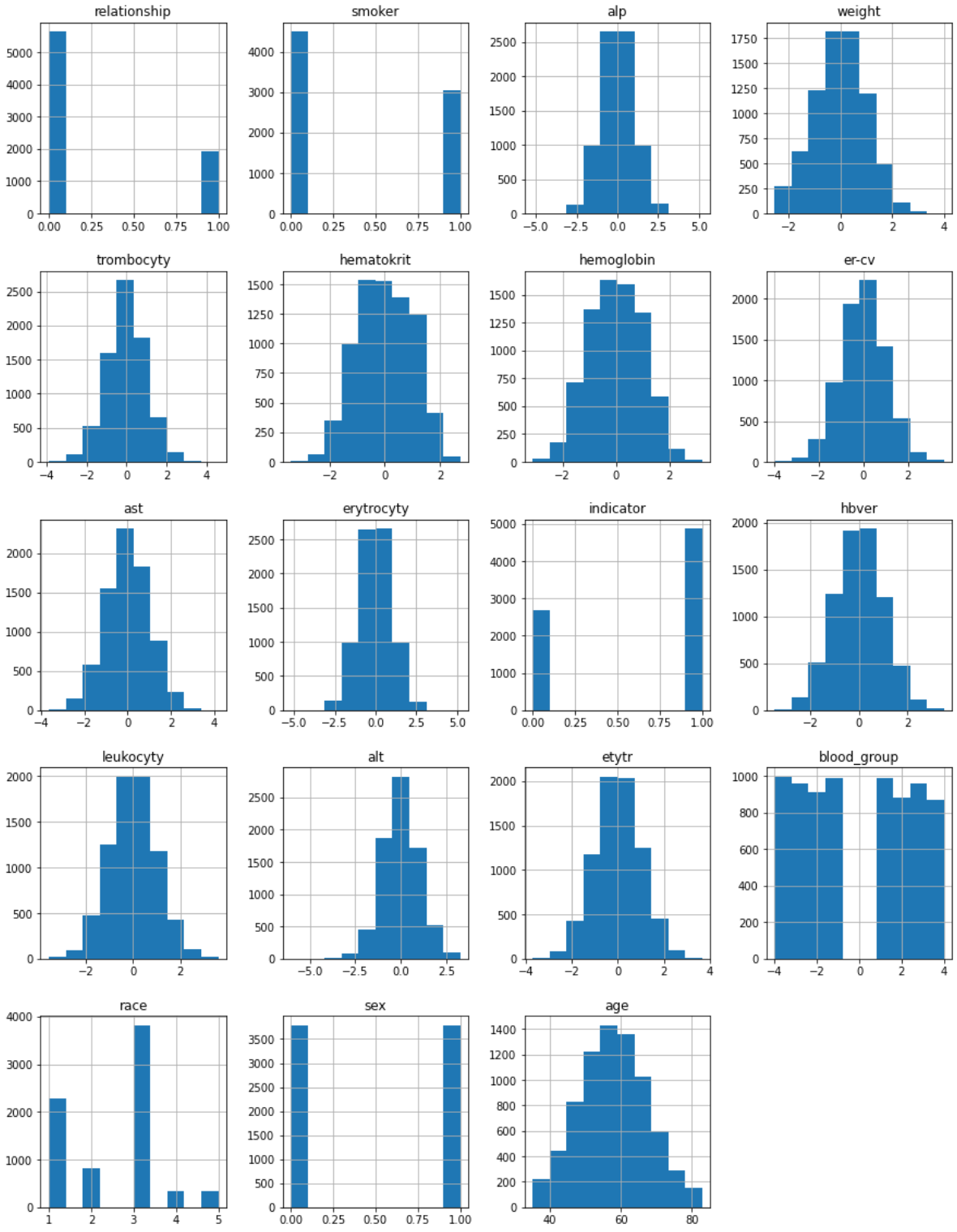
In [56]:

```
fig = plt.figure(figsize = (15,20))
ax = fig.gca()
train_data.hist(ax = ax)
```

```
C:\Users\pplev\AppData\Local\Temp/ipykernel_12248/2748110766.py:3: UserWarni
ng: To output multiple subplots, the figure containing the passed axes is be
ing cleared
  train_data.hist(ax = ax)
```

Out[56]:

```
array([[<AxesSubplot:title={'center':'relationship'}>,
        <AxesSubplot:title={'center':'smoker'}>,
        <AxesSubplot:title={'center':'alp'}>,
        <AxesSubplot:title={'center':'weight'}>],
       [<AxesSubplot:title={'center':'trombocyty'}>,
        <AxesSubplot:title={'center':'hematokrit'}>,
        <AxesSubplot:title={'center':'hemoglobin'}>,
        <AxesSubplot:title={'center':'er-cv'}>],
       [<AxesSubplot:title={'center':'ast'}>,
        <AxesSubplot:title={'center':'erytrocyty'}>,
        <AxesSubplot:title={'center':'indicator'}>,
        <AxesSubplot:title={'center':'hbver'}>],
       [<AxesSubplot:title={'center':'leukocyty'}>,
        <AxesSubplot:title={'center':'alt'}>,
        <AxesSubplot:title={'center':'etytr'}>,
        <AxesSubplot:title={'center':'blood_group'}>],
       [<AxesSubplot:title={'center':'race'}>,
        <AxesSubplot:title={'center':'sex'}>,
        <AxesSubplot:title={'center':'age'}>, <AxesSubplot:>]],
      dtype=object)
```



Výber atribútov pre strojové učenie

V datasete sa nachádza viacero informatívnych atribútov k atribútu indicator, ktorý reprezentuje stav pacienta. Podľa heat mapy môžeme vidieť, že niektoré atribúty s ním korelujú viac a niektoré menej, ale v podstate sú k nemu aj tak informatívne.

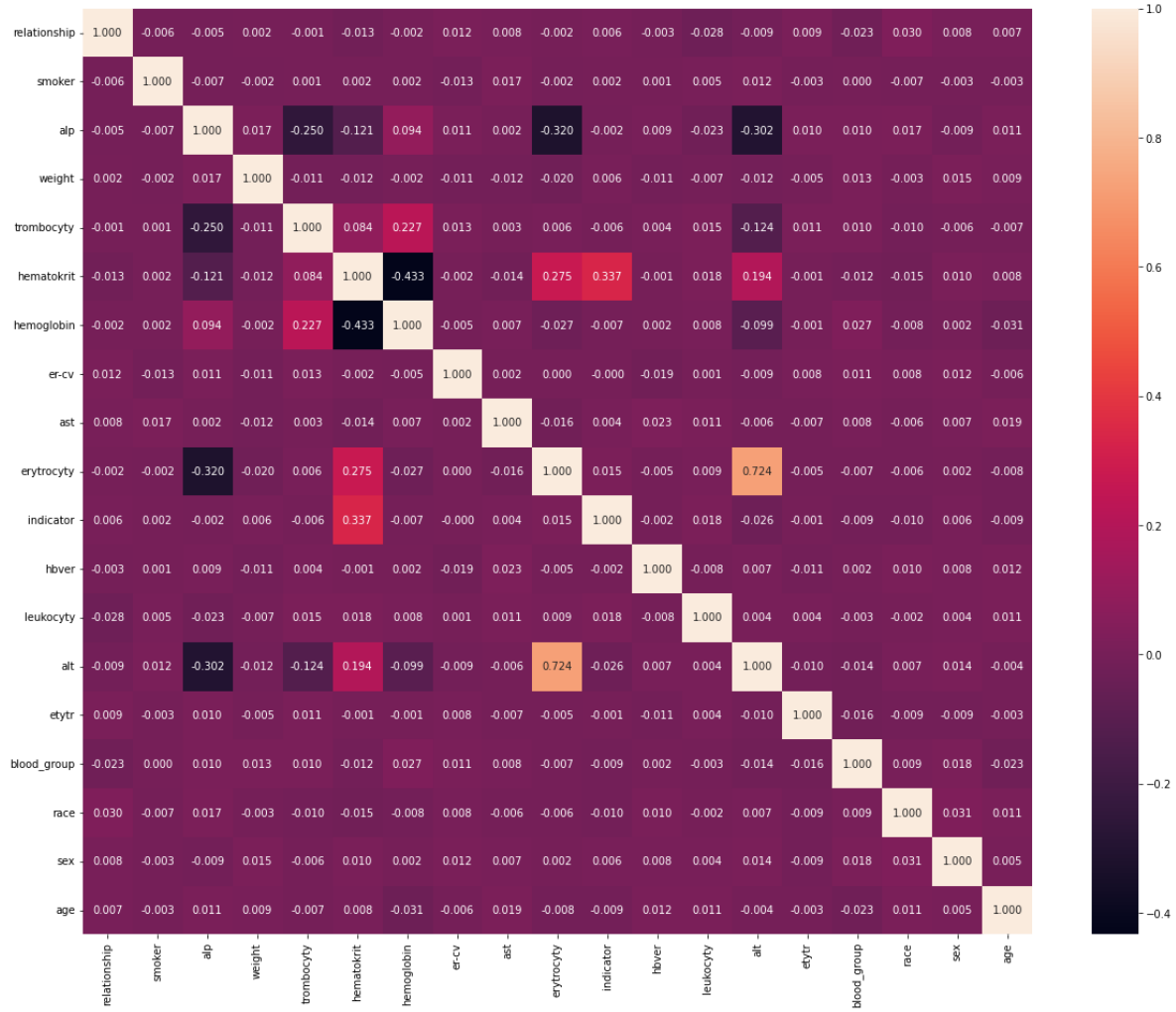
Heatmapa pred transformáciou dát.

In [57]:

```
figure, ax = plt.subplots(figsize=(20,16))
sns.heatmap(merged.corr(),ax=ax, annot =True, fmt = ".3f")
```

Out[57]:

<AxesSubplot:>



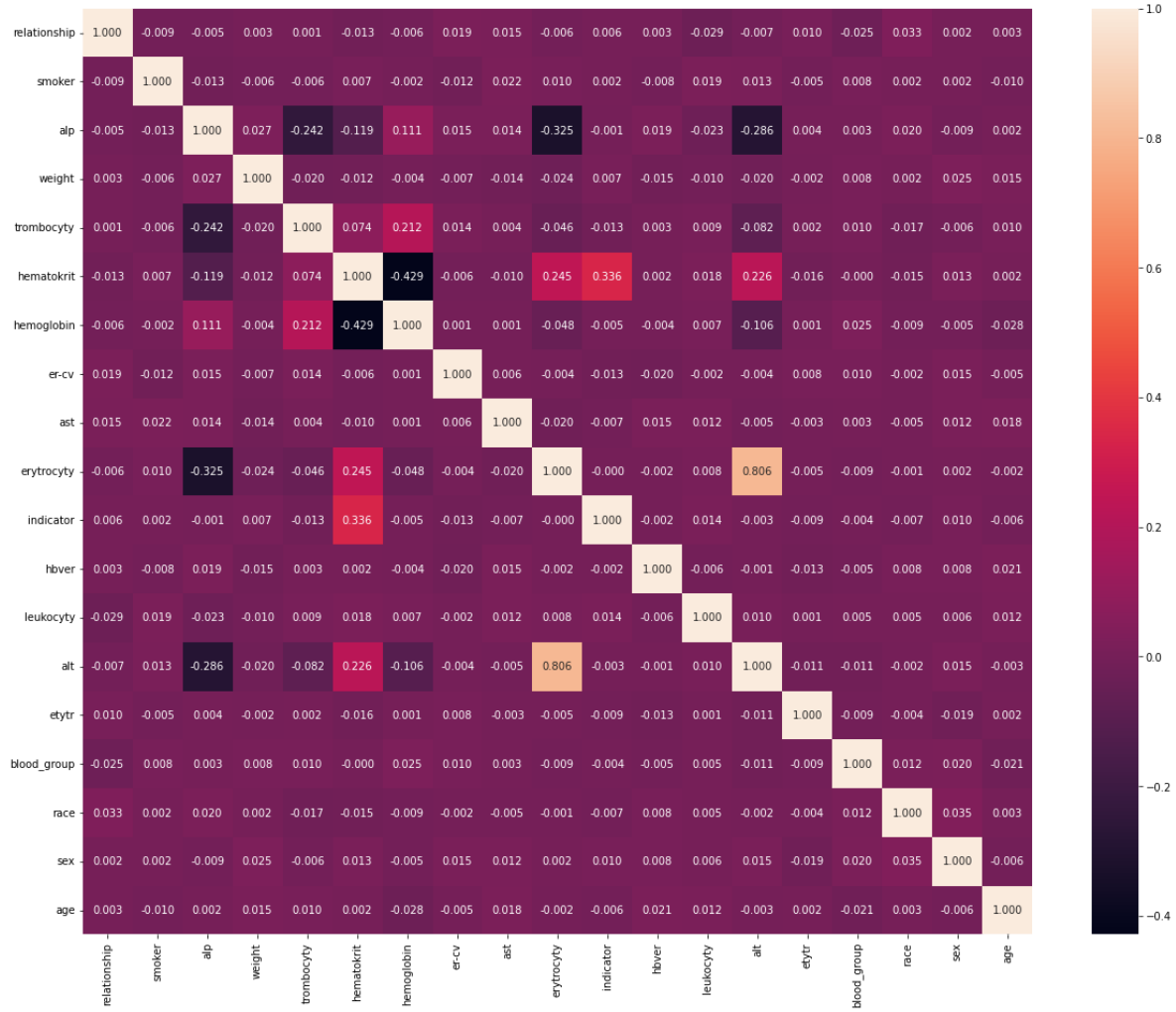
Heatmapa po transformácii dát.

In [58]:

```
figure, ax = plt.subplots(figsize=(20,16))
sns.heatmap(train_data.corr(),ax=ax, annot =True, fmt = ".3f")
```

Out[58]:

<AxesSubplot:>



Ako porvé dáme preč úplne všetky nenumerické atribúty keďže tie sú v tejto časti zbytočné.

In [59]:

```
numeric_train_data = train_data.select_dtypes([np.number])
```

In [60]:

```
print (numeric_train_data.dtypes)
```

```
relationship      int64
smoker            int64
alp             float64
weight          float64
trombocyty      float64
hematokrit      float64
hemoglobin      float64
er-cv           float64
ast             float64
erytrocyty      float64
indicator       float64
hbver           float64
leukocyty       float64
alt             float64
etytr           float64
blood_group       int64
race              int64
sex               int64
age             float64
dtype: object
```

Ako máme možnosť vidieť, korelácie k atribútu indicator su veľmi nízke, kvôli tomu, že sme transformovali všetky atribúty a korelačná relácia k jednotlivým atribútom sa signifikantne nezmenila.

In [61]:

```
cor_target = abs(numeric_train_data.corr()["indicator"])
cor_target.sort_values(ascending=False)
```

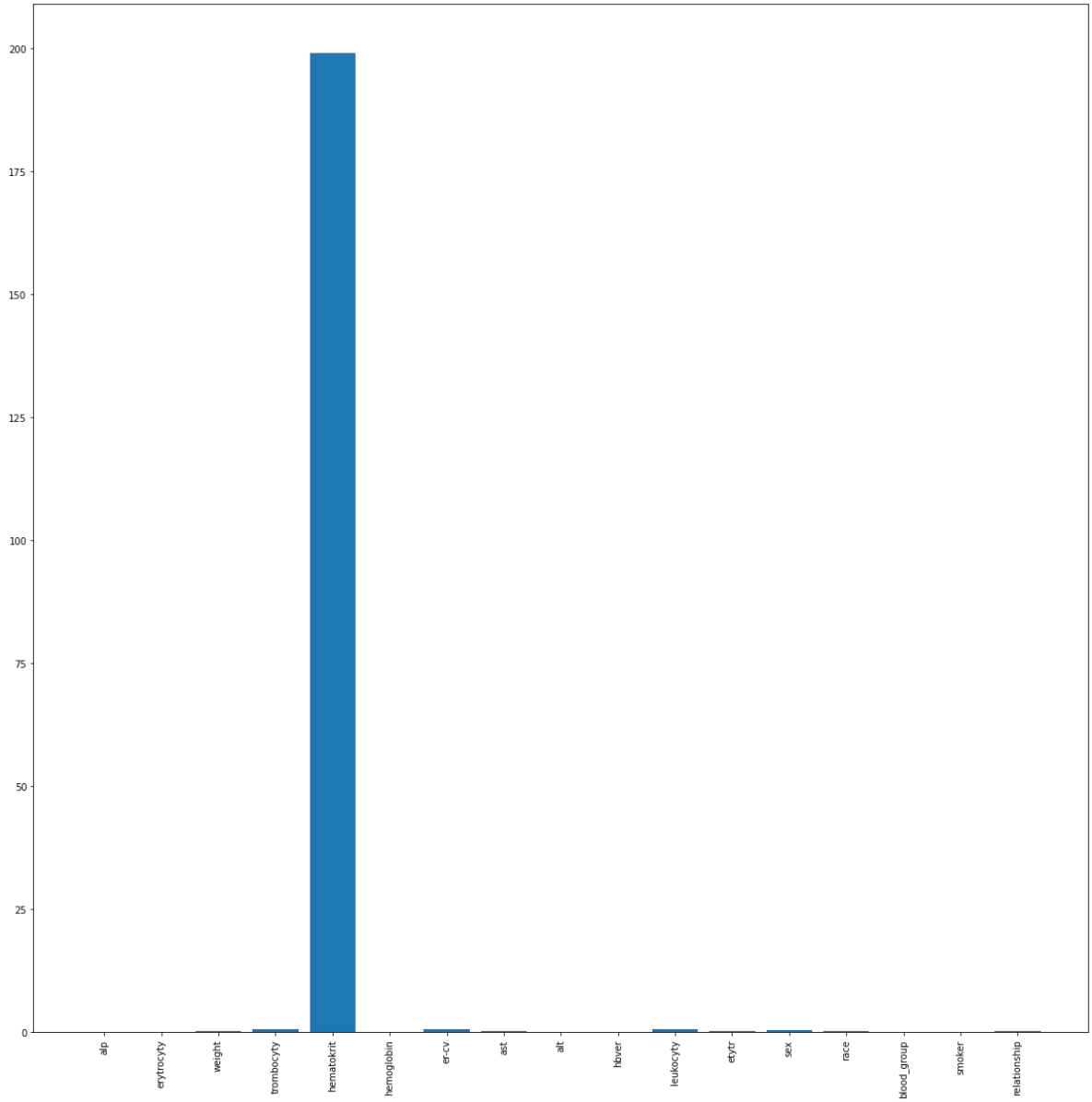
Out[61]:

```
indicator       1.000000
hematokrit      0.336448
leukocyty       0.014466
trombocyty      0.013369
er-cv           0.012854
sex             0.009695
etytr           0.009003
ast             0.007429
weight          0.007379
race            0.007059
relationship    0.006324
age             0.005896
hemoglobin      0.004717
blood_group     0.003822
alt             0.003469
smoker          0.002215
hbver           0.001914
alp             0.000826
erytrocyty      0.000109
Name: indicator, dtype: float64
```

Na tomto grafe môžme vidieť ako veľmi sú jednotlivé features informatívne k atribútu indicator. Bohužiaľ náš dataset vykazuje jediný vysoko korelačný atribút a to atribút hematokrit.

In [62]:

```
all=skewed+gaussian+['sex','race','blood_group','smoker','relationship']
selector = SelectKBest(f_regression, k=17)
selector.fit_transform(numeric_train_data[all], numeric_train_data["indicator"])
scores = -np.log10(selector.pvalues_)
plt.figure(figsize=(20, 20))
plt.bar(range(len(all)), scores)
plt.xticks(range(len(all)), all, rotation='vertical')
plt.show()
```



Replikovateľnosť predspracovania

Na demonštráciu predspracovania údajov sme využili pipeline. V tejto pipeline sa postupne vykonajú všetky transformácie jednotlivých atribútov. V podstate je to to isté, čo sme robili akurát jednoduhším a prehľadnejším spôsobom.

In [63]:

```
def columns_name_to_index(arr_of_names, df):
    return [df.columns.get_loc(c) for c in arr_of_names if c in df]
```

In [64]:

```
quantil_transformer = make_column_transformer((QuantileTransformer(output_distribution="nor
                                               random_state=0,
                                               n_quantiles=1000),
                                               columns_name_to_index(skewed,numeric_train_d
                                               remainder='passthrough')
```

In [65]:

```
power_transformer = make_column_transformer((PowerTransformer(method='yeo-johnson',standard
```

In [66]:

```
standard_scaler = make_column_transformer((StandardScaler(),columns_name_to_index(gaussian+
```

In [67]:

```
pp = Pipeline(steps=[('1',quantil_transformer),
                     ('2',power_transformer),
                     ('3',standard_scaler),
                     ("4",FunctionTransformer(lambda x: pd.DataFrame(x, columns = numeric_t
```

In [68]:

```
transformed= pp.fit_transform(numeric_train_data)
transformed
```

```
[Pipeline] ................. (step 1 of 4) Processing 1, total=   0.0s
[Pipeline] ................. (step 2 of 4) Processing 2, total=   0.1s
[Pipeline] ................. (step 3 of 4) Processing 3, total=   0.0s
[Pipeline] ................. (step 4 of 4) Processing 4, total=   0.0s
```

Out[68]:

	relationship	smoker	alp	weight	trombocyty	hematokrit	hemoglobin	er
0	0.884366	-0.124883	-1.359584	-0.007377	-0.495873	0.768550	-0.719884	0.3271
1	-0.932763	1.779659	0.351034	0.852911	0.614366	-0.550349	0.036845	0.4385
2	-0.190134	-0.505312	-1.328033	-0.130472	0.816358	-0.230452	-0.081693	-1.1813
3	1.019769	-0.830630	-0.705947	1.269901	0.517027	-0.662411	-0.555930	0.9179
4	0.487872	0.212525	-2.205662	-0.190025	0.680080	0.886972	-1.385383	-1.2036
...	...	...	...	...	...	...	...	
7565	0.604915	0.453849	-0.342732	-0.998819	-2.028016	-1.227877	-1.497966	-0.4941
7566	-0.487865	1.154118	-2.395801	0.184331	0.469877	1.121074	-0.390742	-0.5436
7567	-0.537609	0.156530	-0.681640	-0.849965	-1.819204	0.740697	-0.578380	-0.1928
7568	0.791667	-0.525250	-1.292014	0.329764	-0.048356	0.574207	1.541268	0.6959
7569	1.687444	-0.570929	-1.702666	0.088812	0.023761	1.255042	0.564571	0.0486

7570 rows × 19 columns

Záver

V tejto fáze projektu, sme po spojení datasetov, predspracovali dáta na strojové učenie. Podarilo sa nám nahradiť niektoré string atribúty za numerické hodnoty. Taktiež sme oddemonštrovali viaceré techniky nahradzovania nedefinovaných atribútov. V rámci riešenia vychýlených hodnôt sme nahradili hodnoty pomocou kvantilového rozdelenia. V rámci zadania sme demonštovali využitie Pipeline, ktorá bola využitá pri nahradzovaní null hodnôt a fázach transfomácie jednotlivých atribútov. Celková zmena datasetu oproti starému je vo formáte hodnôt a taktiež ich rozdelení. Výstupom tejto fázy je dataset pre strojové učenie.

In [69]:

```
numeric_train_data.head()
```

Out[69]:

	relationship	smoker	alp	weight	trombocyty	hematokrit	hemoglobin	er-cv
152	1	0	0.756082	1.354408	0.035134	0.885438	-0.118336	-1.35898(
1866	0	1	-2.089916	1.322059	0.420206	-0.932068	1.766515	0.35046
7853	0	1	0.278866	0.294297	-1.953214	-0.181294	-0.500629	-1.32748
9401	0	0	-0.277456	-0.478776	-0.773871	1.018521	-0.829018	-0.70622(
9839	0	0	0.983876	-2.419001	-0.164277	0.494277	0.218756	-2.20340

In [70]:

```
test_data.head()
```

Out[70]:

	relationship	smoker	alp	weight	trombocyty	hematokrit	hemoglobin	er-cv
3918	0	0	32.24252	121.26550	6.91668	3.81402	8.46942	46.01029
3197	0	0	18.03794	127.38887	6.24250	4.19722	3.94810	41.43163
6242	0	1	71.60694	130.89584	7.20747	5.55058	7.19223	71.45176
7099	0	1	69.33888	105.26710	7.69180	3.79032	7.23353	43.85026
7283	0	0	26.57411	65.44632	9.81363	8.57331	4.14099	40.02672

5 rows × 23 columns

In [71]:

```
numeric_train_data.to_csv('train_transformed.csv')
```

In [72]:

```
test_data.to_csv('test_data.csv')
```

In [ ]:

```

```