*Постановка задачи:*

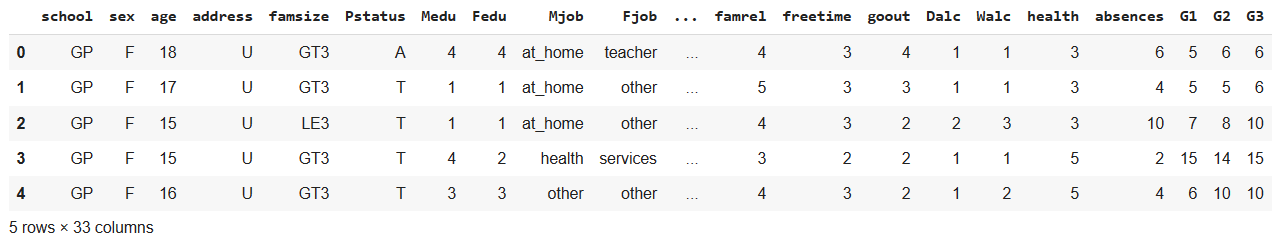
Обучиться структуре байесовской сети, которая должна уметь прогнозировать G1, G2, G3, оценить параметры каждой полученной структуры, оценить ошибку прогноза каждой из получившихся сетей; каждую получившуюся сеть изобразить в виде направленного графа с подписанными вершинами. Сравнить все полученные результаты с какой-либо моделью машинного обучения.

1. *Данные*

Данные представляют собой описание студентов математического курса (student-mat.csv) и курса португальского языка (student-por.csv).

1. *Предварительная обработка*

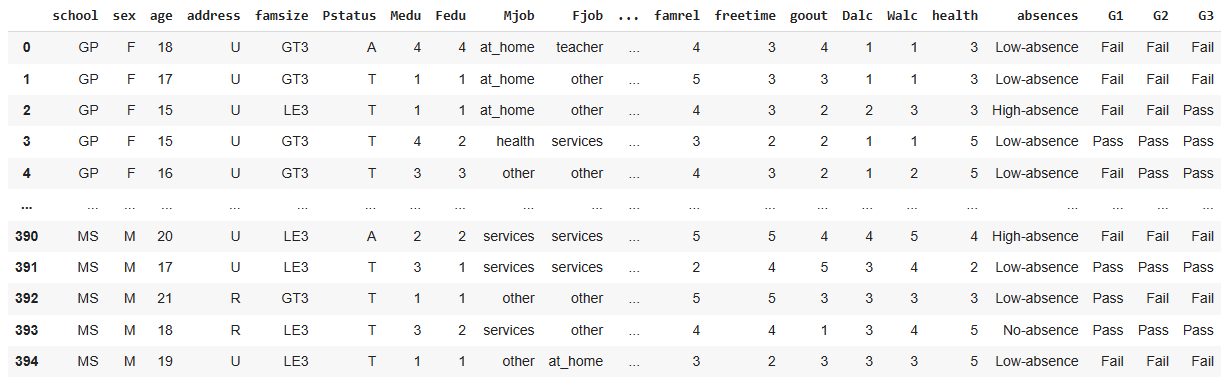
Исходные данные student-mat.csv:



Данные проверены на наличие пропущенных значений, которые не были обнаружены.

Также для уменьшения количества категорий в некоторых категориальных признаках проведено объединение схожих значений. Числовые признаки сделаны категориальными путем дискретизации и заданы метки. Преобразования выполнены для признаков "failures", "studytime", "absences", "G1", "G2", "G3".

Результат:

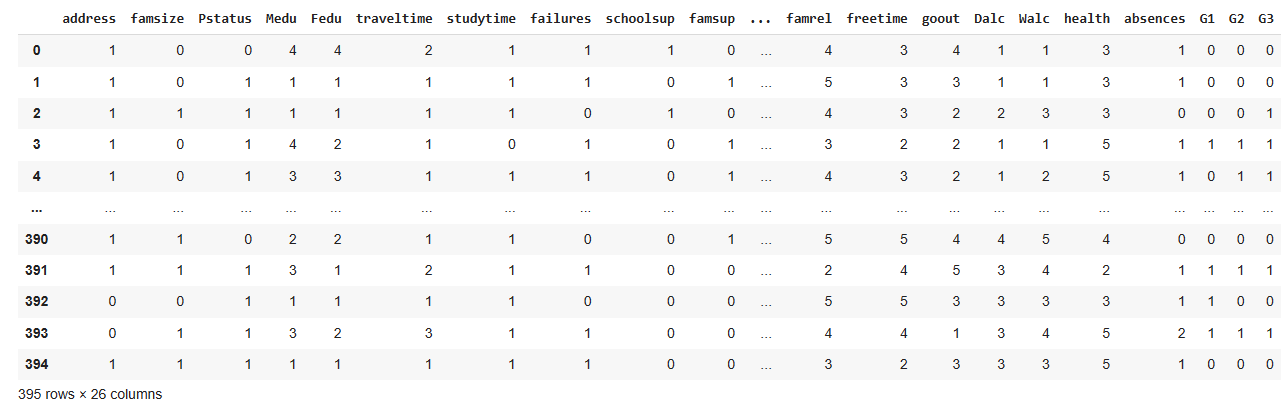


Затем все переменные переведены в числовые.

Для прогнозируемых признаков "G1", "G2", "G3" значение 0 соответствует Fail (при условии, что балл ниже 10), значение 1 – Pass (при условии, что балл > или = 10).

В анализ не включались чувствительные переменные, такие как 'age', 'school', 'sex', 'Mjob', 'Fjob', 'reason', 'guardian'.

Данные, которые использованы для анализа:



1. *Ход работы:*

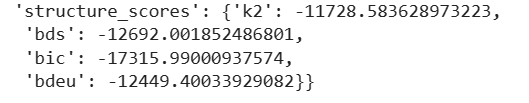
Для структурного обучения байесовской сети использована библиотека bnlearn. Настройкой по умолчанию в [bnlearn](https://erdogant.github.io/bnlearn" \t "_blank) для структурного обучения является метод поиска холма (hillclimbsearch) и оценка BIC. Применение такого метода не показало связей между вершинами графа. Также изучение структуры с более чем 15 узлами вычислительно невозможно при исчерпывающем поиске (exhaustivesearch).

Поэтому в ходе работы рассмотрены Chow-liu и Tree-augmented Naive Bayes (TAN).

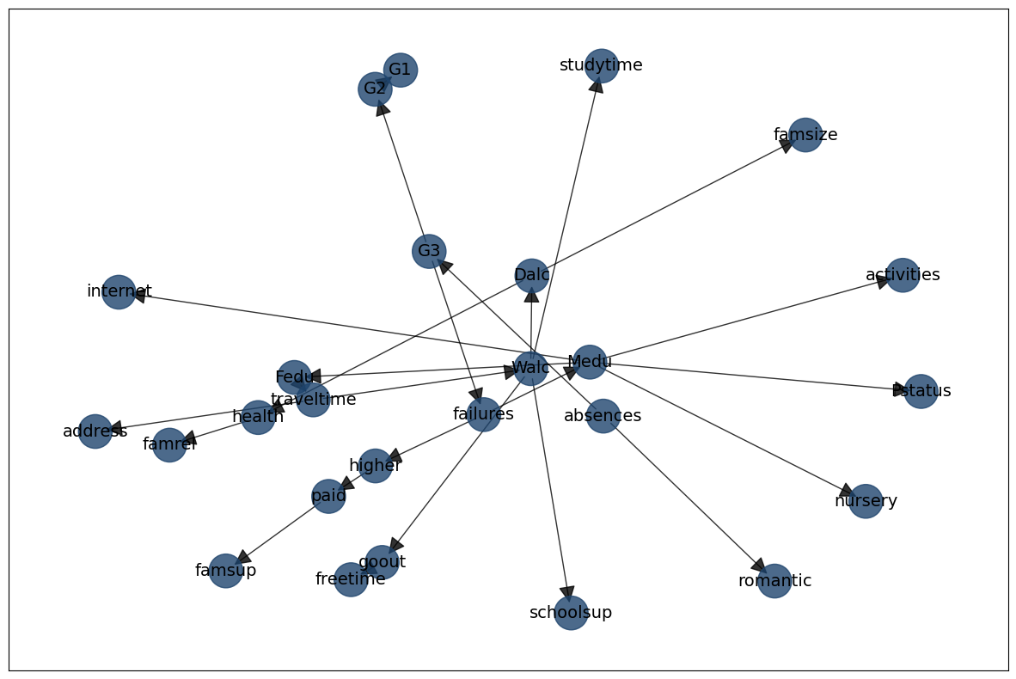
Данные поделены на обучающую выборку (train, 0.9) и тестовую (test, 0.1). Используя обучающую выборку, производится структурное обучение байесовской сети.

*Chow-liu*

В качестве root\_node взят 'absences'. Вычислены оценки структуры ['k2', 'bds', 'bic', 'bdeu'] для сравнения моделей (чем выше, тем лучше). В результате для scoretype взят 'k2', т.к.:



Полученный граф:



Далее проведен parameter learning и построены CPD. Таблицы условной вероятности (CPD) описывают статистические отношения между каждым узлом и его родителями. В данном задании использована Байесовская оценка.

Пример полученного результата (другие CPD представлены в приложении):

CPD of absences:

+-------------+----------+

| absences(0) | 0.302091 |

+-------------+----------+

| absences(1) | 0.375154 |

+-------------+----------+

| absences(2) | 0.322755 |

+-------------+----------+

CPD of G3:

+----------+---------------------+--------------------+---------------------+

| absences | absences(0) | absences(1) | absences(2) |

+----------+---------------------+--------------------+---------------------+

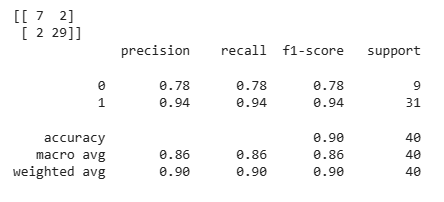
| G3(0) | 0.49022801302931596 | 0.4026229508196722 | 0.49085365853658536 |

+----------+---------------------+--------------------+---------------------+

| G3(1) | 0.509771986970684 | 0.5973770491803279 | 0.5091463414634146 |

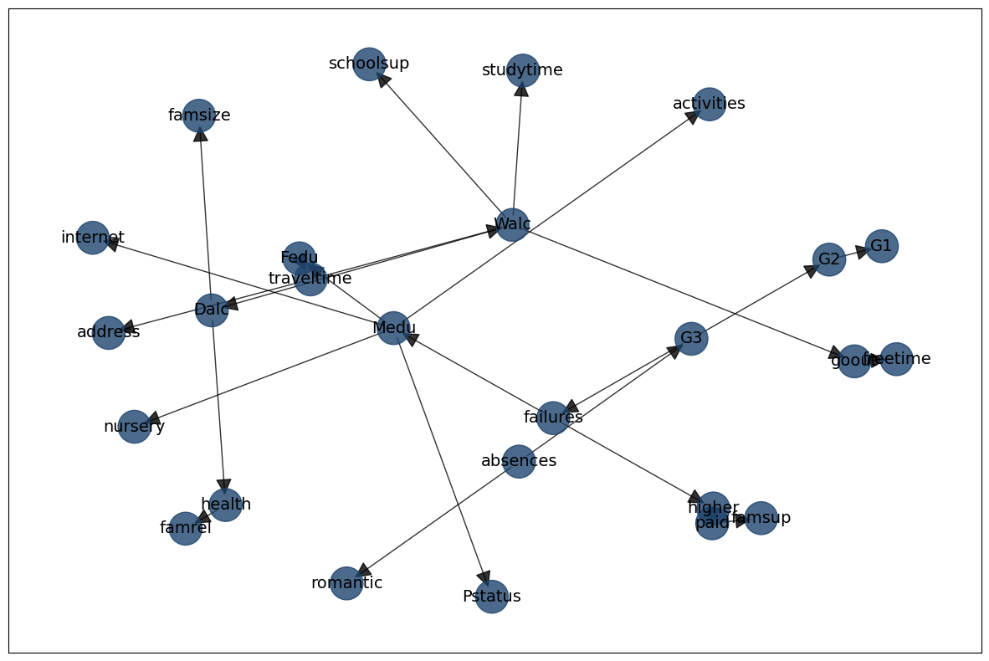
+----------+---------------------+--------------------+---------------------+

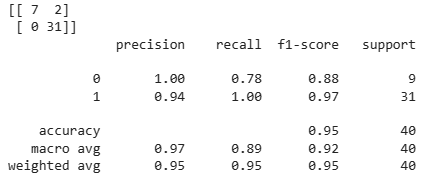
Для оценки качества построенной модели прогноза G1 построена матрица ошибок и отчет с указанием accuracy, precision, recall и F-меры:



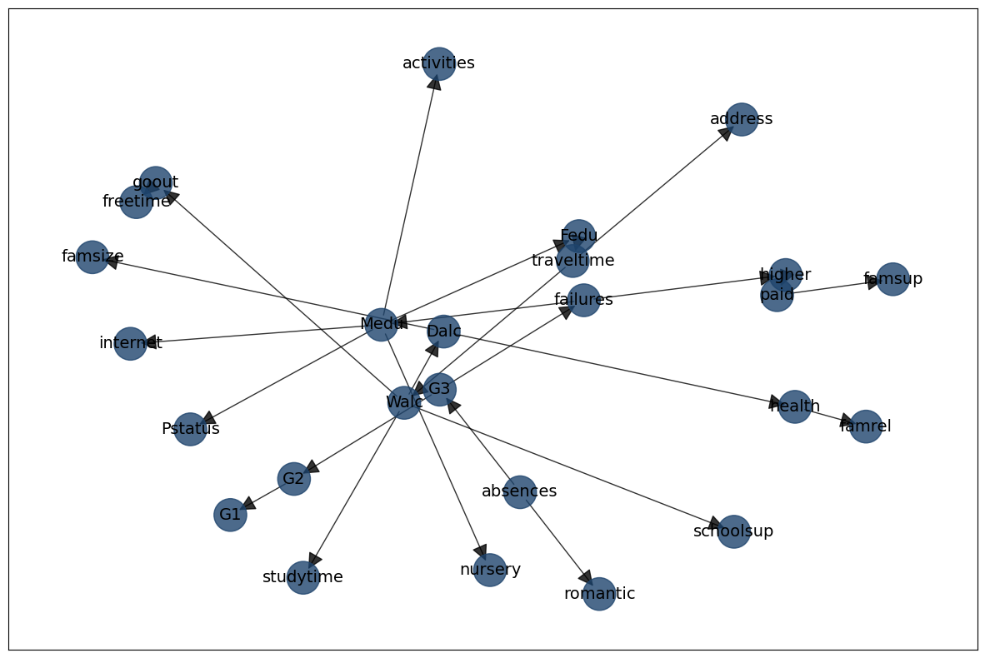
Посмотрев на полученные значения, например, на f1, можно заметить, что модель хорошо предсказывает и тех, кто сдал, и тех, кто не сдал.

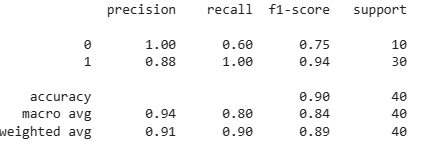
Аналогично построен прогноз и оценено качество для G2:





И G3:



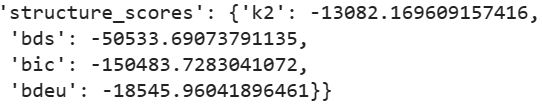


Следовательно, можно сделать вывод, что с помощью построенной модели можно делать прогноз G1, G2, G3.

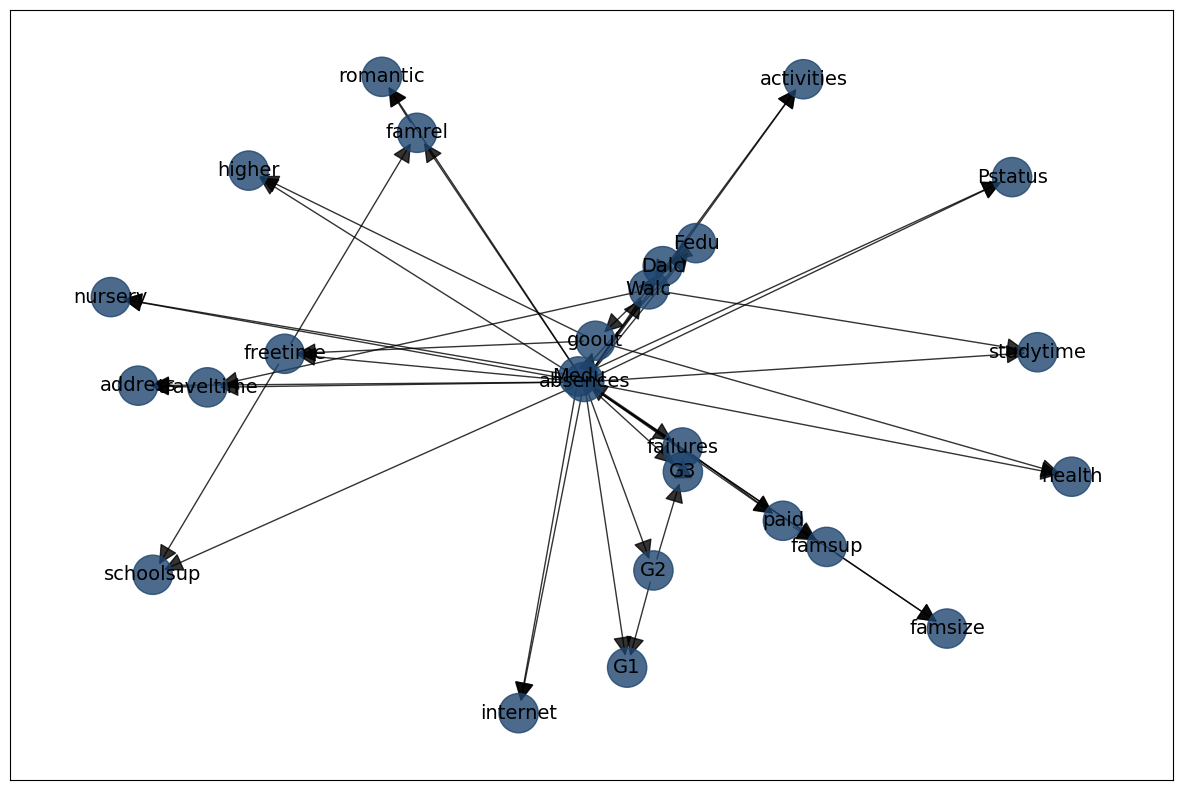
*Tree-augmented Naive Bayes (TAN):*

Аналогично для TAN.

В качестве class\_node взят 'absences'. Вычислены оценки структуры ['k2', 'bds', 'bic', 'bdeu'] для сравнения моделей (чем выше, тем лучше). В результате для scoretype взят 'k2', т.к.:



Полученный граф:



Пример CPD:

+----------+-------------------+--------------------+---------------------+

| absences | absences(0) | absences(1) | absences(2) |

+----------+-------------------+--------------------+---------------------+

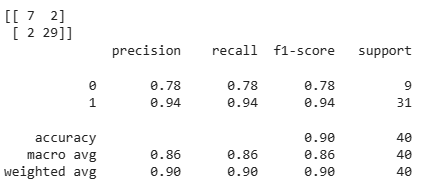
| G2(0) | 0.497557003257329 | 0.4301639344262296 | 0.49085365853658536 |

+----------+-------------------+--------------------+---------------------+

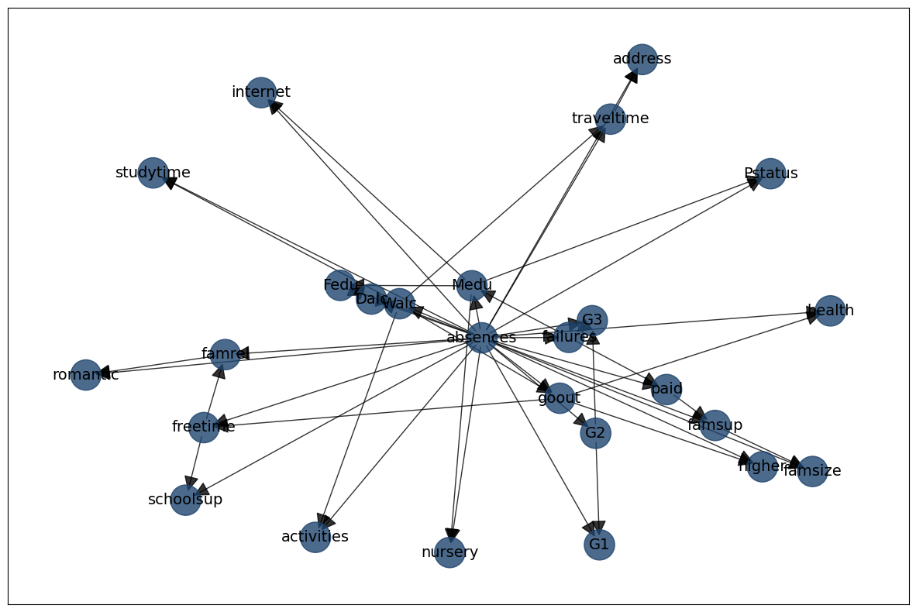
| G2(1) | 0.502442996742671 | 0.5698360655737705 | 0.5091463414634146 |

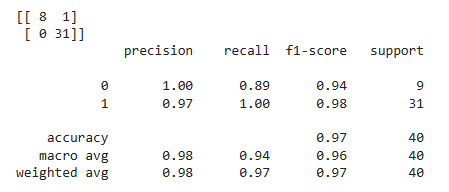
+----------+-------------------+--------------------+---------------------+

Оценка качества модели:

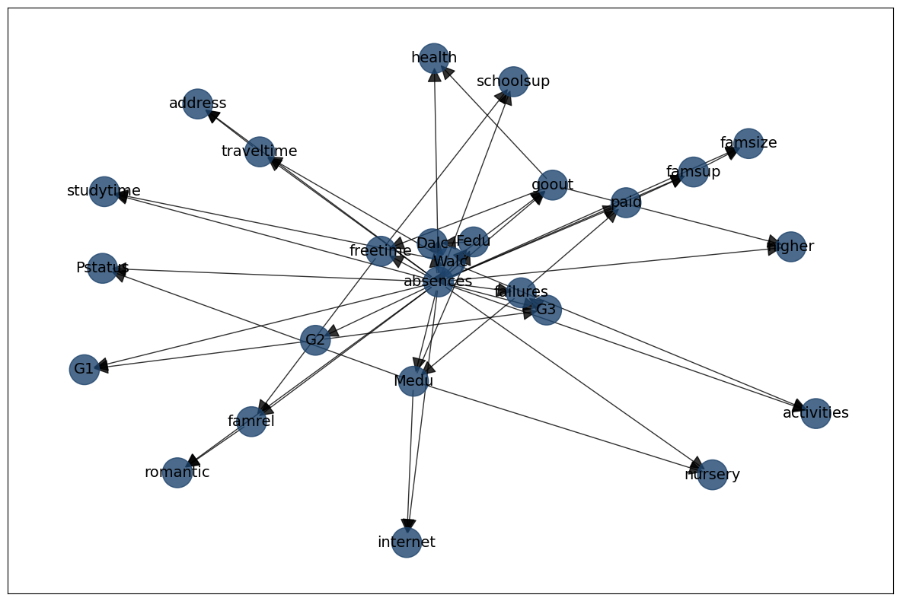


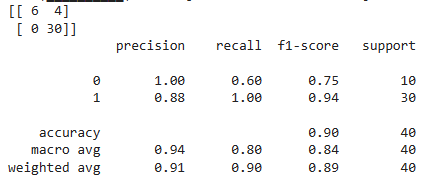
 Для G2:





Для G3:



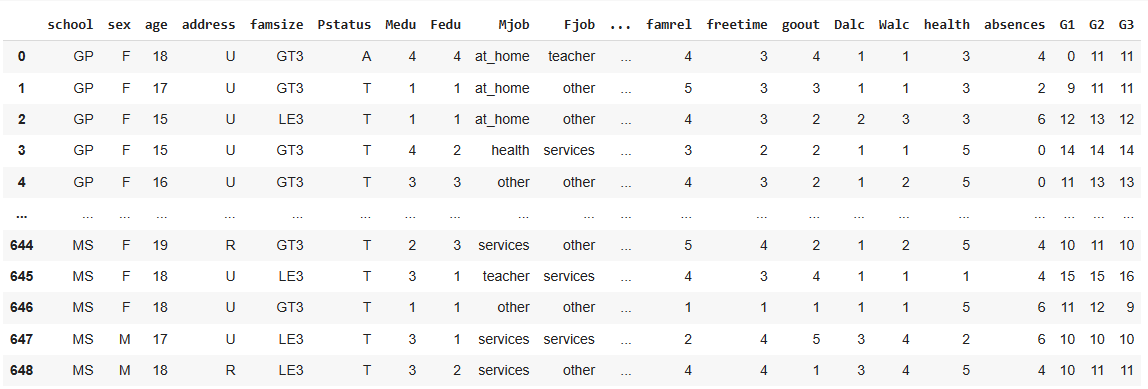


Аналогично, можно сделать вывод, что с помощью построенной модели можно делать прогноз G1, G2, G3.

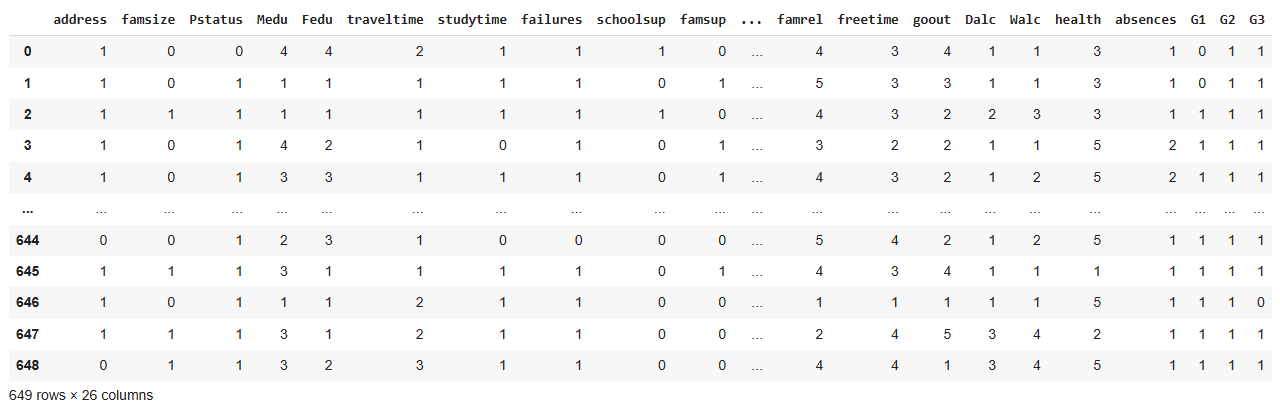
1. *Исследование на данных student-por.csv.*

Аналогично.

Исходные данные:

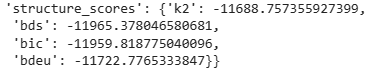


Предобработанные данные:

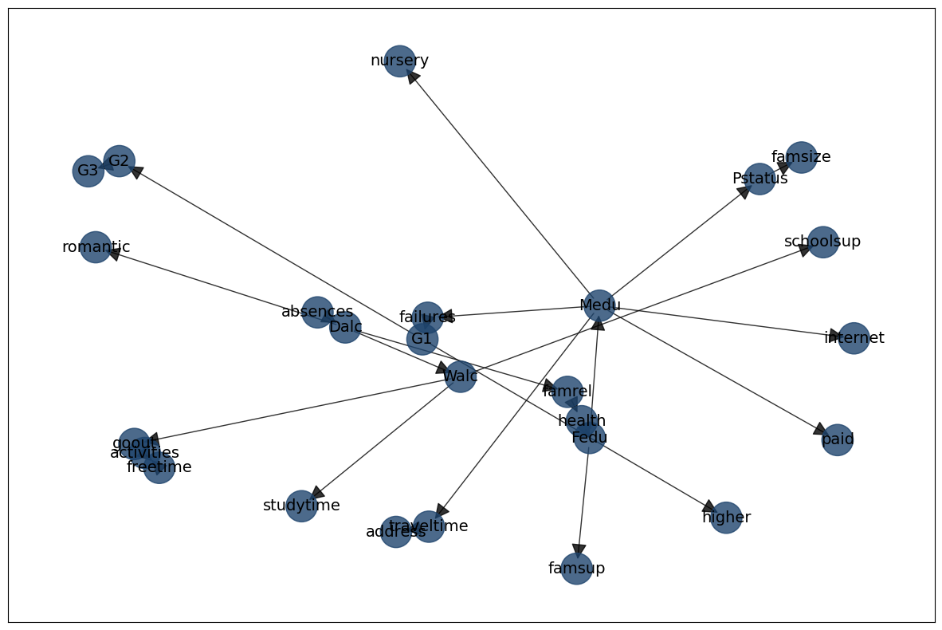


*Chow-liu*

В качестве root\_node взят 'absences'. Для scoretype взят 'k2', т.к.:



Граф:



CPD of absences:

+-------------+----------+

| absences(0) | 0.251473 |

+-------------+----------+

| absences(1) | 0.400463 |

+-------------+----------+

| absences(2) | 0.348064 |

+-------------+----------+

CPD of G1:

+----------+-------------------+--------------------+

| failures | failures(0) | failures(1) |

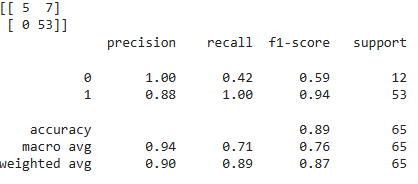
+----------+-------------------+--------------------+

| G1(0) | 0.531986531986532 | 0.3323232323232323 |

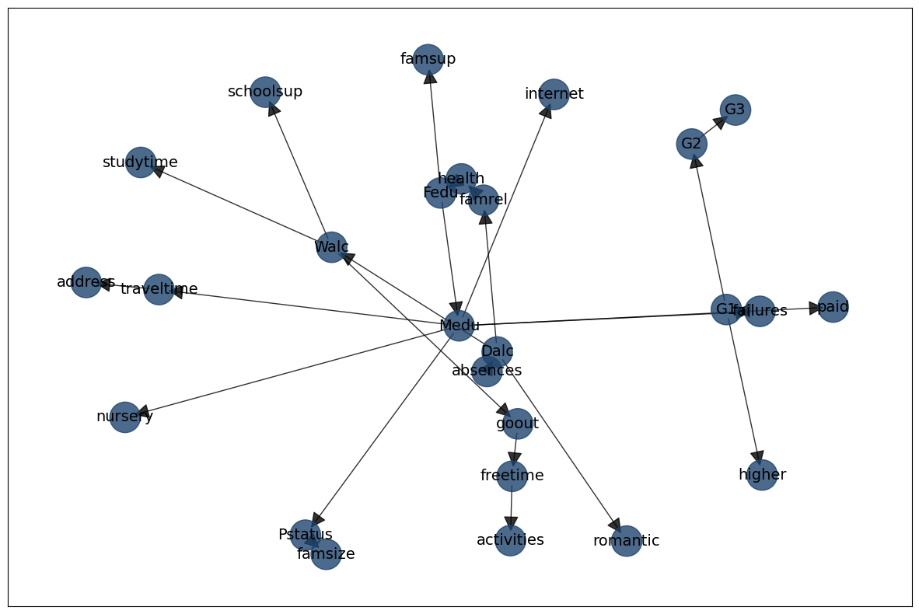
+----------+-------------------+--------------------+

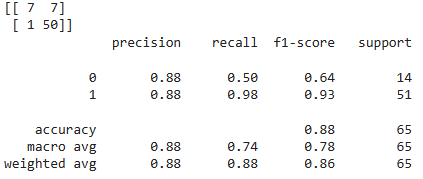
| G1(1) | 0.468013468013468 | 0.6676767676767676 |

+----------+-------------------+--------------------+

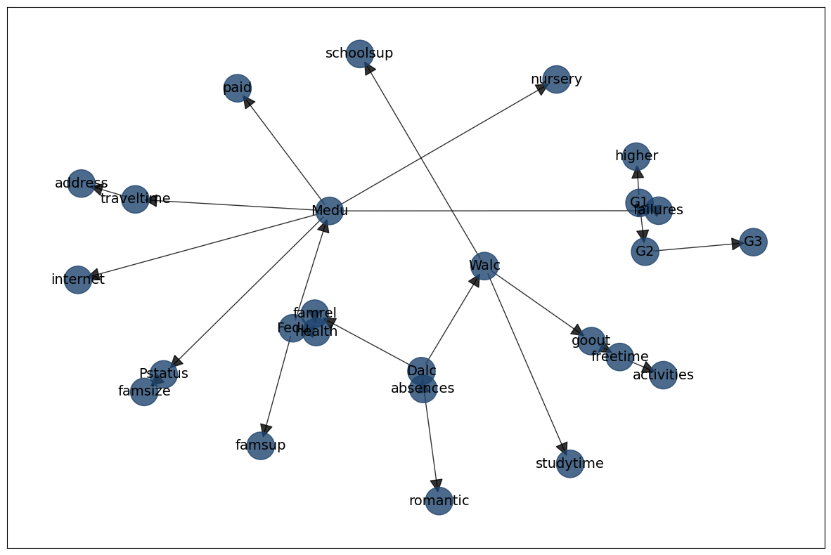


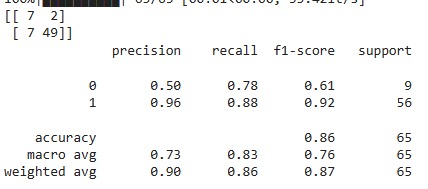
G2:





G3:

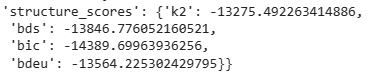




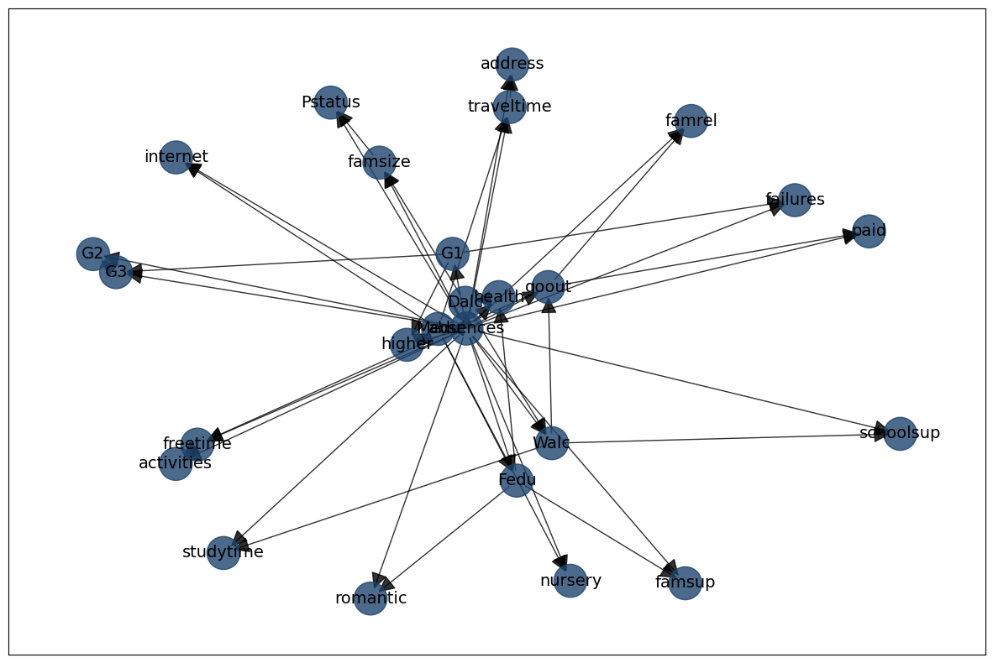
В данном случае метрики показывают, что прогноз для тех, кто не сдал хуже, но все еще можно использовать.

*Tree-augmented Naive Bayes (TAN):*

В качестве class\_node взят 'absences'. Для scoretype взят 'k2', т.к.:



Граф:



CPD:

CPD of G1:

+----------+--------------------+-------------------+--------------------+

| absences | absences(0) | absences(1) | absences(2) |

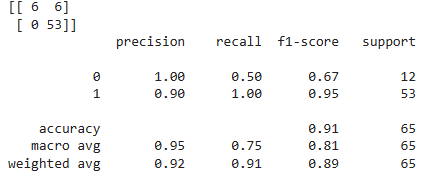
+----------+--------------------+-------------------+--------------------+

| G1(0) | 0.4711297071129707 | 0.395165528113505 | 0.3748488512696494 |

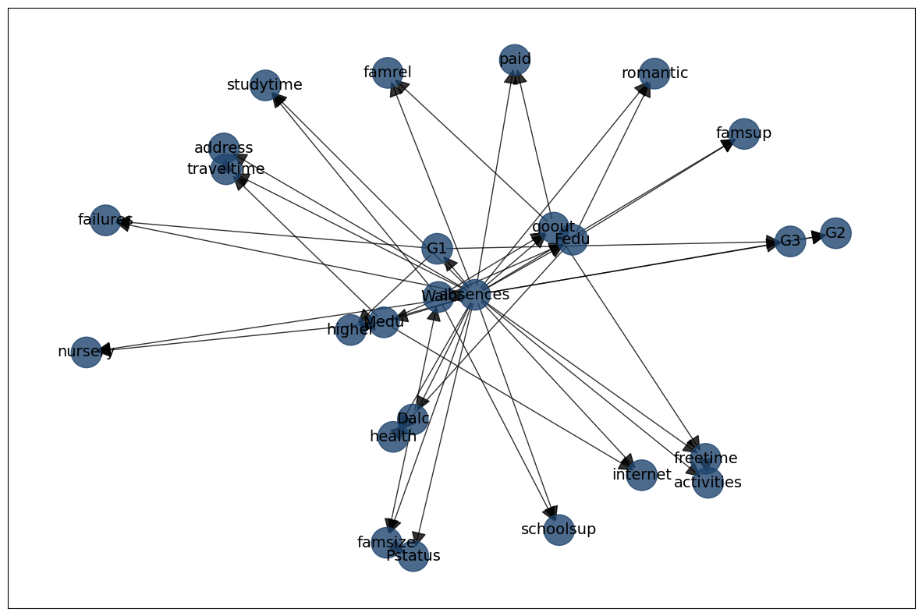
+----------+--------------------+-------------------+--------------------+

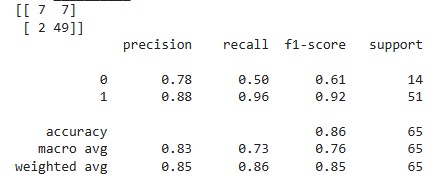
| G1(1) | 0.5288702928870292 | 0.604834471886495 | 0.6251511487303507 |

+----------+--------------------+-------------------+--------------------+

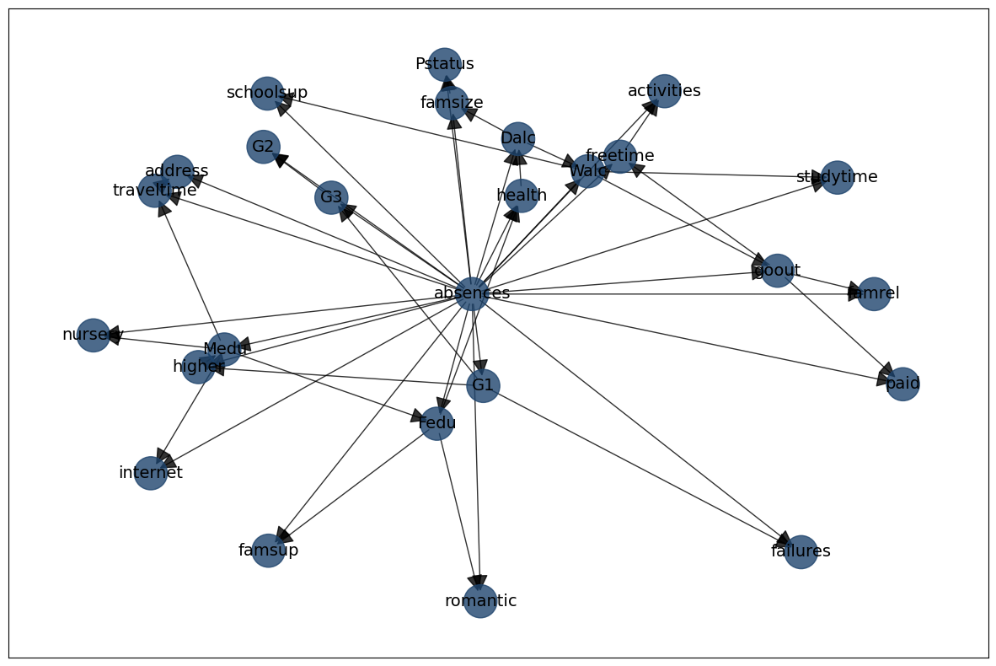


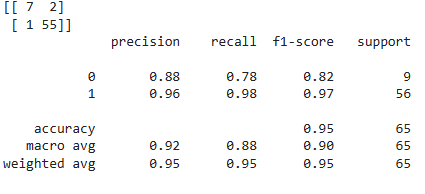
G2:





G3:



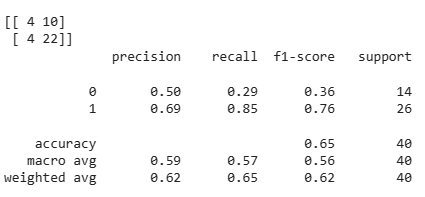


Как можно видеть из полученных результатов, построенные модели можно использовать для прогноза G1, G2, G3.

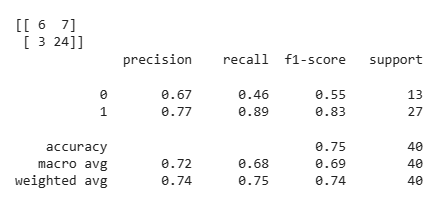
1. *Линейная регрессия*

Исходные данные student-mat.csv:

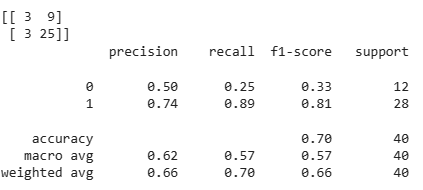
G1:



G2:



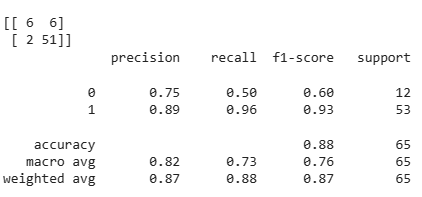
G3:



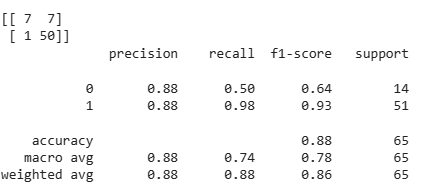
Как можно видеть, в случае линейной регрессии модель чаще ошибается при определении людей, которые не сдали. Для тех, кто сдал, модель работает хорошо.

Исходные данные student-por.csv:

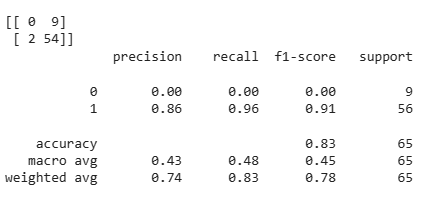
G1:



G2:



G3:



Здесь аналогично, модель часто ошибается при определении людей, которые не сдали.

В целом, можно сделать вывод, что Байесовская сеть для исходных данных работает лучше, чем модель линейной регрессии.

Приложение 1. Код

# import libraries

! pip install causalnex

!pip install pgmpy

import numpy as np

import pandas as pd

from time import time

from sklearn.metrics import f1\_score

import pgmpy.estimators as ests

import os

import random

import warnings

import numpy as np

import pandas as pd

import networkx as nx

import statsmodels

from scipy.io import arff

from scipy.signal.\_signaltools import \_centered

import matplotlib.pyplot as plt

import pgmpy.estimators as ests

from pgmpy.models import BayesianNetwork

from pgmpy.metrics import structure\_score

from sklearn.preprocessing import MinMaxScaler

from sklearn.metrics import mean\_squared\_error

# warnings.simplefilter("ignore")

!pip install bnlearn

import bnlearn as bn

from sklearn.preprocessing import LabelEncoder

# Learn the DAG using Chow-liu

model\_cl      = bn.structure\_learning.fit(struct\_data, methodtype='cl', root\_node='absences', scoretype= 'k2')

# Plot detected DAG

G = bn.plot(model\_cl)

# Learn the DAG using Tree-augmented Naive Bayes

model\_tan = bn.structure\_learning.fit(struct\_data, methodtype='tan', class\_node='absences', scoretype= 'k2')

# Plot detected DAG

G = bn.plot(model\_tan)

# Read data

# student\_mat\_data = pd.read\_csv("student-mat.csv")

# data = student\_mat\_data

student\_por\_data = pd.read\_csv("student-por.csv")

data = student\_por\_data

# data preprocessing

data.notnull()

struct\_data = data

discretised\_data = struct\_data.copy()

data\_vals = {col: struct\_data[col].unique() for col in struct\_data.columns}

failures\_map = {v: 'no-failure' if v == [0]

                else 'have-failure' for v in data\_vals['failures']}

studytime\_map = {v: 'short-studytime' if v in [1,2]

                 else 'long-studytime' for v in data\_vals['studytime']}

discretised\_data["failures"] = discretised\_data["failures"].map(failures\_map)

discretised\_data["studytime"] = discretised\_data["studytime"].map(studytime\_map)

from causalnex.discretiser import Discretiser

discretised\_data["absences"] = Discretiser(method="fixed",

                          numeric\_split\_points=[1, 10]).transform(discretised\_data["absences"].values)

discretised\_data["G1"] = Discretiser(method="fixed",

                          numeric\_split\_points=[10]).transform(discretised\_data["G1"].values)

discretised\_data["G2"] = Discretiser(method="fixed",

                          numeric\_split\_points=[10]).transform(discretised\_data["G2"].values)

discretised\_data["G3"] = Discretiser(method="fixed",

                          numeric\_split\_points=[10]).transform(discretised\_data["G3"].values)

absences\_map = {0: "No-absence", 1: "Low-absence", 2: "High-absence"}

G1\_map = {0: "Fail", 1: "Pass"}

G2\_map = {0: "Fail", 1: "Pass"}

G3\_map = {0: "Fail", 1: "Pass"}

discretised\_data["absences"] = discretised\_data["absences"].map(absences\_map)

discretised\_data["G1"] = discretised\_data["G1"].map(G1\_map)

discretised\_data["G2"] = discretised\_data["G2"].map(G2\_map)

discretised\_data["G3"] = discretised\_data["G3"].map(G3\_map)

G\_ = "G1"

struct\_data = discretised\_data.copy()

non\_numeric\_columns = list(struct\_data.select\_dtypes(exclude=[np.number]).columns)

from sklearn.preprocessing import LabelEncoder

le = LabelEncoder()

for col in non\_numeric\_columns:

    struct\_data[col] = le.fit\_transform(struct\_data[col])

drop\_col = ['school','sex','age','Mjob', 'Fjob','reason','guardian']

struct\_data = struct\_data.drop(columns=drop\_col)

from sklearn.model\_selection import train\_test\_split

train, test = train\_test\_split(struct\_data, train\_size=0.9, test\_size=0.1, random\_state=7)

# Learn the DAG using Chow-liu

model\_cl = bn.structure\_learning.fit(train, methodtype='cl', root\_node='absences', scoretype= 'k2')

# Plot detected DAG

G = bn.plot(model\_cl)

# Parameter learning

model = bn.parameter\_learning.fit(model\_cl, train, verbose=3);

# Learn the DAG using Tree-augmented Naive Bayes

# model\_tan = bn.structure\_learning.fit(struct\_data, methodtype='tan', class\_node='absences', scoretype= 'k2')

# # Plot detected DAG

# G = bn.plot(model\_tan)

# # Parameter learning

# model = bn.parameter\_learning.fit(model\_tan, train, verbose=3);

Pout = bn.predict(model, test, variables=[G\_])

# print(Pout)

from sklearn.metrics import classification\_report, confusion\_matrix, accuracy\_score

predictions = bn.predict(model, test, variables=[G\_])[G\_]

print("")

print(confusion\_matrix(test[G\_], predictions))

print(classification\_report(test[G\_], predictions))

# Linear Regression

from sklearn import linear\_model

from sklearn.linear\_model import LinearRegression

# Extract feature columns

feature\_cols = list(struct\_data.columns[:-3])

#feature\_cols

# Extract target column 'passed'

#target\_col = student\_data.columns[-3:]

target\_col\_G1 = struct\_data.columns[-3]

target\_col\_G2 = struct\_data.columns[-2]

target\_col\_G3 = struct\_data.columns[-1]

target\_col = target\_col\_G3

#target\_col

# Separate the data into feature data and target data (X\_all and y\_all, respectively)

X\_all = struct\_data[feature\_cols]

y\_all = struct\_data[target\_col]

# split the dataset

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X\_all, y\_all, test\_size=0.1, random\_state=0)

model = LinearRegression().fit(X\_train, y\_train)

y\_pred = model.predict(X\_test)

print("")

print(confusion\_matrix(np.array(y\_test), np.rint(y\_pred)))

print(classification\_report(np.array(y\_test), np.rint(y\_pred)))

Приложение 2. CPD.

[bnlearn] >CPD of G3:

+----------+---------------------+--------------------+---------------------+

| absences | absences(0) | absences(1) | absences(2) |

+----------+---------------------+--------------------+---------------------+

| G3(0) | 0.49022801302931596 | 0.4026229508196722 | 0.49085365853658536 |

+----------+---------------------+--------------------+---------------------+

| G3(1) | 0.509771986970684 | 0.5973770491803279 | 0.5091463414634146 |

+----------+---------------------+--------------------+---------------------+

[bnlearn] >CPD of romantic:

+-------------+---------------------+---------------------+--------------------+

| absences | absences(0) | absences(1) | absences(2) |

+-------------+---------------------+---------------------+--------------------+

| romantic(0) | 0.512214983713355 | 0.5836065573770491 | 0.5274390243902439 |

+-------------+---------------------+---------------------+--------------------+

| romantic(1) | 0.48778501628664495 | 0.41639344262295086 | 0.4725609756097561 |

+-------------+---------------------+---------------------+--------------------+

[bnlearn] >CPD of G2:

+-------+---------------------+--------------------+

| G3 | G3(0) | G3(1) |

+-------+---------------------+--------------------+

| G2(0) | 0.5854838709677419 | 0.3727891156462585 |

+-------+---------------------+--------------------+

| G2(1) | 0.41451612903225804 | 0.6272108843537415 |

+-------+---------------------+--------------------+

[bnlearn] >CPD of failures:

+-------------+--------------------+--------------------+

| G3 | G3(0) | G3(1) |

+-------------+--------------------+--------------------+

| failures(0) | 0.482258064516129 | 0.3795918367346939 |

+-------------+--------------------+--------------------+

| failures(1) | 0.5177419354838709 | 0.6204081632653061 |

+-------------+--------------------+--------------------+

[bnlearn] >CPD of G1:

+-------+---------------------+--------------------+

| G2 | G2(0) | G2(1) |

+-------+---------------------+--------------------+

| G1(0) | 0.5667189952904239 | 0.3788300835654596 |

+-------+---------------------+--------------------+

| G1(1) | 0.43328100470957615 | 0.6211699164345403 |

+-------+---------------------+--------------------+

[bnlearn] >CPD of Medu:

+----------+---------------------+---------------------+

| failures | failures(0) | failures(1) |

+----------+---------------------+---------------------+

| Medu(0) | 0.17474048442906576 | 0.13127413127413126 |

+----------+---------------------+---------------------+

| Medu(1) | 0.21107266435986158 | 0.16859716859716858 |

+----------+---------------------+---------------------+

| Medu(2) | 0.21107266435986158 | 0.22651222651222652 |

+----------+---------------------+---------------------+

| Medu(3) | 0.21453287197231835 | 0.21492921492921493 |

+----------+---------------------+---------------------+

| Medu(4) | 0.18858131487889274 | 0.25868725868725867 |

+----------+---------------------+---------------------+

[bnlearn] >CPD of higher:

+-----------+---------------------+---------------------+

| failures | failures(0) | failures(1) |

+-----------+---------------------+---------------------+

| higher(0) | 0.45501730103806226 | 0.32947232947232946 |

+-----------+---------------------+---------------------+

| higher(1) | 0.5449826989619377 | 0.6705276705276705 |

+-----------+---------------------+---------------------+

[bnlearn] >CPD of Fedu:

+---------+---------------------+-----+---------------------+

| Medu | Medu(0) | ... | Medu(4) |

+---------+---------------------+-----+---------------------+

| Fedu(0) | 0.19704433497536947 | ... | 0.13225806451612904 |

+---------+---------------------+-----+---------------------+

| Fedu(1) | 0.2019704433497537 | ... | 0.13225806451612904 |

+---------+---------------------+-----+---------------------+

| Fedu(2) | 0.20689655172413793 | ... | 0.1774193548387097 |

+---------+---------------------+-----+---------------------+

| Fedu(3) | 0.19704433497536947 | ... | 0.23870967741935484 |

+---------+---------------------+-----+---------------------+

| Fedu(4) | 0.19704433497536947 | ... | 0.3193548387096774 |

+---------+---------------------+-----+---------------------+

[bnlearn] >CPD of internet:

+-------------+--------------------+-----+---------------------+

| Medu | Medu(0) | ... | Medu(4) |

+-------------+--------------------+-----+---------------------+

| internet(0) | 0.4975369458128079 | ... | 0.34838709677419355 |

+-------------+--------------------+-----+---------------------+

| internet(1) | 0.5024630541871922 | ... | 0.6516129032258065 |

+-------------+--------------------+-----+---------------------+

[bnlearn] >CPD of nursery:

+------------+--------------------+-----+---------------------+

| Medu | Medu(0) | ... | Medu(4) |

+------------+--------------------+-----+---------------------+

| nursery(0) | 0.5024630541871922 | ... | 0.36129032258064514 |

+------------+--------------------+-----+---------------------+

| nursery(1) | 0.4975369458128079 | ... | 0.6387096774193548 |

+------------+--------------------+-----+---------------------+

[bnlearn] >CPD of Pstatus:

+------------+---------------------+-----+--------------------+

| Medu | Medu(0) | ... | Medu(4) |

+------------+---------------------+-----+--------------------+

| Pstatus(0) | 0.49261083743842365 | ... | 0.3548387096774194 |

+------------+---------------------+-----+--------------------+

| Pstatus(1) | 0.5073891625615764 | ... | 0.6451612903225806 |

+------------+---------------------+-----+--------------------+

[bnlearn] >CPD of activities:

+---------------+-----+---------------------+

| Medu | ... | Medu(4) |

+---------------+-----+---------------------+

| activities(0) | ... | 0.47096774193548385 |

+---------------+-----+---------------------+

| activities(1) | ... | 0.5290322580645161 |

+---------------+-----+---------------------+

[bnlearn] >CPD of paid:

+---------+--------------------+---------------------+

| higher | higher(0) | higher(1) |

+---------+--------------------+---------------------+

| paid(0) | 0.5163776493256262 | 0.5047846889952153 |

+---------+--------------------+---------------------+

| paid(1) | 0.4836223506743738 | 0.49521531100478466 |

+---------+--------------------+---------------------+

[bnlearn] >CPD of traveltime:

+---------------+-----+---------------------+

| Fedu | ... | Fedu(4) |

+---------------+-----+---------------------+

| traveltime(1) | ... | 0.3807829181494662 |

+---------------+-----+---------------------+

| traveltime(2) | ... | 0.24555160142348753 |

+---------------+-----+---------------------+

| traveltime(3) | ... | 0.18861209964412812 |

+---------------+-----+---------------------+

| traveltime(4) | ... | 0.18505338078291814 |

+---------------+-----+---------------------+

[bnlearn] >CPD of famsup:

+-----------+---------------------+--------------------+

| paid | paid(0) | paid(1) |

+-----------+---------------------+--------------------+

| famsup(0) | 0.5057971014492754 | 0.4345864661654135 |

+-----------+---------------------+--------------------+

| famsup(1) | 0.49420289855072463 | 0.5654135338345865 |

+-----------+---------------------+--------------------+

[bnlearn] >CPD of address:

+------------+---------------------+-----+---------------------+

| traveltime | traveltime(1) | ... | traveltime(4) |

+------------+---------------------+-----+---------------------+

| address(0) | 0.32426778242677823 | ... | 0.5038759689922481 |

+------------+---------------------+-----+---------------------+

| address(1) | 0.6757322175732218 | ... | 0.49612403100775193 |

+------------+---------------------+-----+---------------------+

[bnlearn] >CPD of Walc:

+------------+---------------------+-----+---------------------+

| traveltime | traveltime(1) | ... | traveltime(4) |

+------------+---------------------+-----+---------------------+

| Walc(1) | 0.28870292887029286 | ... | 0.19767441860465115 |

+------------+---------------------+-----+---------------------+

| Walc(2) | 0.20711297071129708 | ... | 0.19767441860465115 |

+------------+---------------------+-----+---------------------+

| Walc(3) | 0.20711297071129708 | ... | 0.1937984496124031 |

+------------+---------------------+-----+---------------------+

| Walc(4) | 0.1589958158995816 | ... | 0.1937984496124031 |

+------------+---------------------+-----+---------------------+

| Walc(5) | 0.13807531380753138 | ... | 0.21705426356589147 |

+------------+---------------------+-----+---------------------+

[bnlearn] >CPD of Dalc:

+---------+---------------------+-----+---------------------+

| Walc | Walc(1) | ... | Walc(5) |

+---------+---------------------+-----+---------------------+

| Dalc(1) | 0.5194029850746269 | ... | 0.1894273127753304 |

+---------+---------------------+-----+---------------------+

| Dalc(2) | 0.12238805970149254 | ... | 0.19823788546255505 |

+---------+---------------------+-----+---------------------+

| Dalc(3) | 0.11940298507462686 | ... | 0.2026431718061674 |

+---------+---------------------+-----+---------------------+

| Dalc(4) | 0.11940298507462686 | ... | 0.19383259911894274 |

+---------+---------------------+-----+---------------------+

| Dalc(5) | 0.11940298507462686 | ... | 0.21585903083700442 |

+---------+---------------------+-----+---------------------+

[bnlearn] >CPD of goout:

+----------+---------------------+-----+---------------------+

| Walc | Walc(1) | ... | Walc(5) |

+----------+---------------------+-----+---------------------+

| goout(1) | 0.16417910447761194 | ... | 0.1762114537444934 |

+----------+---------------------+-----+---------------------+

| goout(2) | 0.25970149253731345 | ... | 0.18502202643171806 |

+----------+---------------------+-----+---------------------+

| goout(3) | 0.24776119402985075 | ... | 0.19383259911894274 |

+----------+---------------------+-----+---------------------+

| goout(4) | 0.191044776119403 | ... | 0.2026431718061674 |

+----------+---------------------+-----+---------------------+

| goout(5) | 0.1373134328358209 | ... | 0.2422907488986784 |

+----------+---------------------+-----+---------------------+

[bnlearn] >CPD of studytime:

+--------------+-----+---------------------+

| Walc | ... | Walc(5) |

+--------------+-----+---------------------+

| studytime(0) | ... | 0.45374449339207046 |

+--------------+-----+---------------------+

| studytime(1) | ... | 0.5462555066079295 |

+--------------+-----+---------------------+

[bnlearn] >CPD of schoolsup:

+--------------+-----+---------------------+

| Walc | ... | Walc(5) |

+--------------+-----+---------------------+

| schoolsup(0) | ... | 0.5462555066079295 |

+--------------+-----+---------------------+

| schoolsup(1) | ... | 0.45374449339207046 |

+--------------+-----+---------------------+

[bnlearn] >CPD of health:

+-----------+---------------------+-----+---------------------+

| Dalc | Dalc(1) | ... | Dalc(5) |

+-----------+---------------------+-----+---------------------+

| health(1) | 0.1547085201793722 | ... | 0.19617224880382775 |

+-----------+---------------------+-----+---------------------+

| health(2) | 0.15022421524663676 | ... | 0.19138755980861244 |

+-----------+---------------------+-----+---------------------+

| health(3) | 0.24439461883408073 | ... | 0.19617224880382775 |

+-----------+---------------------+-----+---------------------+

| health(4) | 0.18385650224215247 | ... | 0.21052631578947367 |

+-----------+---------------------+-----+---------------------+

| health(5) | 0.26681614349775784 | ... | 0.20574162679425836 |

+-----------+---------------------+-----+---------------------+

[bnlearn] >CPD of famsize:

+------------+--------------------+-----+---------------------+

| Dalc | Dalc(1) | ... | Dalc(5) |

+------------+--------------------+-----+---------------------+

| famsize(0) | 0.6524663677130045 | ... | 0.5167464114832536 |

+------------+--------------------+-----+---------------------+

| famsize(1) | 0.3475336322869955 | ... | 0.48325358851674644 |

+------------+--------------------+-----+---------------------+

[bnlearn] >CPD of freetime:

+-------------+---------------------+-----+---------------------+

| goout | goout(1) | ... | goout(5) |

+-------------+---------------------+-----+---------------------+

| freetime(1) | 0.1981981981981982 | ... | 0.1659919028340081 |

+-------------+---------------------+-----+---------------------+

| freetime(2) | 0.20270270270270271 | ... | 0.17408906882591094 |

+-------------+---------------------+-----+---------------------+

| freetime(3) | 0.20270270270270271 | ... | 0.21862348178137653 |

+-------------+---------------------+-----+---------------------+

| freetime(4) | 0.20270270270270271 | ... | 0.21862348178137653 |

+-------------+---------------------+-----+---------------------+

| freetime(5) | 0.19369369369369369 | ... | 0.22267206477732793 |

+-------------+---------------------+-----+---------------------+

[bnlearn] >CPD of famrel:

+-----------+---------------------+-----+---------------------+

| health | health(1) | ... | health(5) |

+-----------+---------------------+-----+---------------------+

| famrel(1) | 0.17842323651452283 | ... | 0.13069908814589665 |

+-----------+---------------------+-----+---------------------+

| famrel(2) | 0.17427385892116182 | ... | 0.1337386018237082 |

+-----------+---------------------+-----+---------------------+

| famrel(3) | 0.1908713692946058 | ... | 0.1762917933130699 |

+-----------+---------------------+-----+---------------------+

| famrel(4) | 0.23651452282157676 | ... | 0.3252279635258359 |

+-----------+---------------------+-----+---------------------+

| famrel(5) | 0.21991701244813278 | ... | 0.23404255319148937 |

+-----------+---------------------+-----+---------------------+