columns = all data.columns.tolist()[1:] columns.remove("smoking") Feature engineering 1. Remove outliers We will remove outliers from the data to reduce the effect of large values on the model. In [31]: columns to remove outliers = { "weight(kg)": [30, 120], "HDL": [20, 110], "triglyceride": [0, 410], "fasting blood sugar": [50, 200], "Cholesterol": [80, 320], "LDL": [0, 260], "AST": [0, 100], "Gtp": [0, 230],

In [30]: import pandas as pd

import numpy as np

from helpers import plot histograms, plot

data path = "tahkeer_data.csv" all_data = pd.read_csv(data_path)

"eyesight(left)": [0, 2], "waist(cm)": [50, 120], "serum creatinine": [0, 2], "systolic": [80, 170], for column_name, values in columns_to_remove_outliers.items(): min value = values[0] max value = values[1] all_data = all_data[(all_data[column_name] >= min_value) & (all_data[column_name] <= max_value)]</pre> plot histograms (all data, columns, 3) In [32]: weight(kg) HDL hearing(left) 3000 100000 14000 2500 12000 80000 2000 10000 60000 8000 1500 6000 40000 1000 4000 20000 500

2000 0 0 120 1.0 1.2 1.8 2.0 weight(kg) hearing(left) triglyceride fasting blood sugar Cholesterol 2500 smoking smoking smokina 3500 3500 2000 3000 2500 1500 2000 2000 1000 500 500 500 200 250 120 200 fasting blood sugar triglyceride LDL AST Gtp 2500 6000 4000 5000 2000 3000 4000 1500 S 3000 Count 2000 1000 2000 500 1000 0 0 LDL AST Gtp eyesight(left) waist(cm) hearing(right) smoking smoking 4000 20000 0 100000 3500 17500 3000 80000 15000 2500 12500 60000 00000 S 2000 1500 7500 40000 5000 1000 20000 2500 500 0 0 0 0.50 0.75 1.00 1.25 1.50 1.75 2.00 50 110 eyesight(left) systolic serum creatinine dental caries smoking smoking smoking 6000 70000 17500 5000 60000 15000 50000 4000 12500 10000 40000 3000 30000 7500 2000 5000 20000 1000 2500 10000 0 0 0 0.25 0.50 0.75 1.00 1.25 1.50 1.75 2.00 0.0 0.2 0.4 0.6 0.8 1.0 140 160 120 serum creatinine dental caries systolic 2. Apply log We will apply log to the data to reduce the effect of skewed data. For example, the data for Gtp and triglyceride is skewed to the right. In [33]: logged_columns = ['Gtp', 'triglyceride', 'AST', 'LDL', 'Cholesterol', 'serum creatinine'] for column in logged_columns: all_data[column] = np.log(all_data[column])

plot_histograms(all_data, columns, 3)

In [34]:

features comparable.

for column in columns:

3. Min-Max normalization

min value = all data[column].min() max_value = all_data[column].max()

weight(kg)

14000 100000 2500 12000 80000

HDL

hearing(left)

60000

all_data[column] = (all_data[column] - min_value) / (max_value - min_value)

3000

2000

We will use min-max normalization to normalize the data to the range [0, 1] To reduce the effect of features with large values and make the

8000 1500 6000 40000 1000 4000 20000 500 2000 0 0 0 0.0 0.0 0.2 0.4 0.6 1.0 0.2 0.4 1.0 0.0 0.2 0.4 0.6 0.8 1.0 HDL weight(kg) hearing(left) triglyceride fasting blood sugar Cholesterol 4000 2500 smoking smoking smoking 2500 3500 3000 2000 2500 1500 1500 Count 2000 1000 1500 1000 1000 500 500 500 0.0 0.2 1.0 0.4 0.6 0.8 1.0 0.2 0.4 0.6 0.8 0.6 0.8 1.0 triglyceride fasting blood sugar LDL AST Gtp smoking 2500 6000 4000 2000 5000 3000 4000 1500 Count 3000 2000 1000 500 1000 0 0 0.0 1.0 0.0 1.0 0.0 0.4 0.6 0.8 0.4 1.0 LDL AST eyesight(left) waist(cm) hearing(right) smokina 4000 smokina smokina 20000 0 100000 3500 17500 80000 15000 2500 12500 60000 Count 10000 S 2000 1500 7500 40000 5000 1000 20000 2500 500 0 0 0 0.0 0.2 eyesight(left) serum creatinine dental caries systolic smoking 6000 70000 17500 60000 15000 4000 12500 3000 10000 40000 30000 7500 2000 20000 5000 1000 2500 10000 0 0 0 0.0 0.0 0.0 0.2 1.0 0.2 1.0 0.4 0.6 0.8 0.2 0.6 1.0 0.4 0.6 0.8 0.4 0.8 systolic serum creatinine dental caries 4. Combine features We create new information that is more helpful for our model. weight = "weight(kg)" In [35]: waist = "waist(cm)" all_data[weight + "_" + waist + "_interaction"] = all_data[weight] * all_data[waist] columns.append(weight + "_" + waist + "_interaction") columns.remove(weight) columns.remove(waist) cholesterol = "Cholesterol" ldl = "LDL"

all_data[cholesterol + "_" + ldl + "_aggregation"] = all_data[[cholesterol, ldl]].mean(axis=1)

feature_importance = pd.DataFrame({'feature': columns, 'importance': feature_importance}) feature importance = feature importance.sort values(by='importance', ascending=False)

scatter plot = sns.scatterplot(feature importance, x='feature', y='importance')

feature importance

Gtp

HDL

AST

fasting blood sugar 0.078605

systolic

scatter_plot.set_xticklabels(scatter_plot.get_xticklabels(), rotation=45, ha='right')

0.169605

0.137969

0.118695

0.111283

0.088075 0.085270

0.076413

0.072150

0.045704

import matplotlib.pyplot as plt print(feature_importance) plt.figure(figsize=(20, 6))

plt.show()

5

2

11

12

0

8

3

10

4

6

import seaborn as sns

columns.remove(cholesterol)

5. Feature selection

columns.remove(ldl)

In [36]:

In [37]:

9 dental caries 0.010481 1 hearing(left) 0.002881 7 hearing(right) 0.002870

eyesight(left)

triglyceride

serum creatinine

weight(kg)_waist(cm)_interaction

Cholesterol LDL aggregation

columns.append(cholesterol + "_" + ldl + "_aggregation")

from sklearn.ensemble import RandomForestClassifier

model = RandomForestClassifier(random_state=42)

feature_importance = model.feature_importances_

We will use random forest to select the most important features and discard the rest.

model.fit(all data[columns].values, all data['smoking'].values)

0.175 0.150 0.125 0.100 <u>0</u> .≝ 0.075 0.050 0.025 0.000

C:\Users\fadis\AppData\Local\Temp\ipykernel 15724\887045158.py:8: UserWarning: set ticklabels() shoul

d only be used with a fixed number of ticks, i.e. after set_ticks() or using a FixedLocator. scatter plot.set xticklabels(scatter plot.get xticklabels(), rotation=45, ha='right')

In [38]: threshold = 0.05useful_features = feature_importance[feature_importance['importance'] > threshold]['feature'].tolist() print("Useful features: " + str(useful_features)) Useful features: ['Gtp', 'triglyceride', 'weight(kg) waist(cm) interaction', 'Cholesterol LDL aggrega tion', 'HDL', 'serum creatinine', 'fasting blood sugar', 'systolic', 'AST']

In [39]: plot_histograms(all_data, useful_features, 3) triglyceride weight(kg)_waist(cm)_interaction smoking 0 1 smoking 0 1 3000 2500 4000 2500 2000 2000 3000 1500 O 1500 Count O 1500 2000 1000 1000 1000 500 500 0 0 0 0.0 0.0 0.4 0.6 triglyceride 0.0 0.2 1.0 0.2 0.8 1.0 0.2 0.4 0.6 0.8 Gtp weight(kg)_waist(cm)_interaction HDL Cholesterol_LDL_aggregation serum creatinine 3000 smoking 0 1 smoking 0 1 smoking 0 2500 17500 2500 2000 15000 2000 12500 1500 Count 10000 1500 1000 7500 1000 5000 500 500 2500 0 0 0.6 HDL 0.4 0.6
Cholesterol_LDL_aggregation 0.0 0.4 0.6 serum creatinine 1.0 0.2 0.4 0.8 0.0 0.2 1.0 fasting blood sugar systolic AST 4000 smoking 0 1 smoking 0 1 smoking 0 1 6000 6000 3500 5000 5000 3000 4000 2500 4000 O 3000 2000 3000 1500 2000 2000 1000 1000 1000 500 0 0 0.4 0.6 fasting blood sugar 0.0 0.0 1.0 0.0 0.4 AST In [11]: all_data = all_data[useful_features + ['smoking']] all_data.to_csv("tahkeer_data_cleaned.csv", index=False)