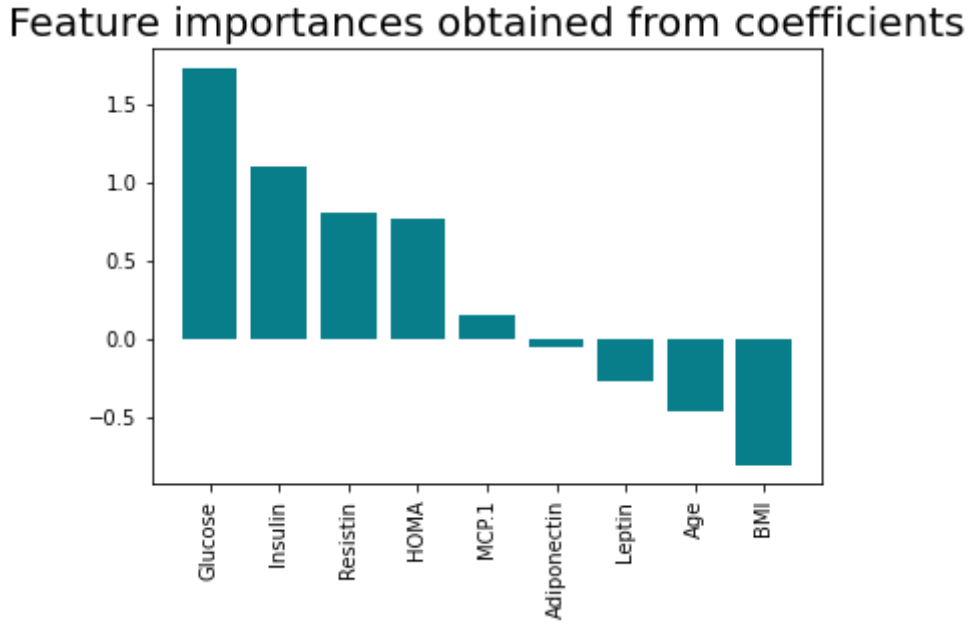
In [28]:

```
#!/usr/bin/env python3
# -*- coding: utf-8 -*-
"""
Created on Fri Oct 15 12:06:07 2021

@author: stephaniewatkins
"""

import pandas as pd


#data exploration
bc=pd.read_csv('~/Desktop/DANN862/breastcancer.csv', sep=',')
bc.columns
bc.shape
print(bc.isnull().sum())
bc.describe()
bc.info()
print(bc.describe())
print(bc.corr())
#30% data
from sklearn.model_selection import train_test_split
from sklearn import metrics
from sklearn import linear_model
from sklearn import tree
from sklearn import naive_bayes
from sklearn.neural_network import MLPClassifier
from sklearn.preprocessing import MinMaxScaler
X = bc.iloc[:, :9]
y = bc.Classification
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size = 0.3, rando

logr = linear_model.LogisticRegression(max_iter=10000)
logr.fit(X_train, y_train)
y_train_logr = logr.predict(X_train)
print('logr train accuracy is ' , metrics.accuracy_score(y_train, y_train_logr))
y_test_logr = logr.predict(X_test)
print(' logr test accuracy is ', metrics.accuracy_score(y_test, y_test_logr))
#naive bayes
nb = naive_bayes.GaussianNB()
nb.fit(X_train, y_train)
y_train_nb = nb.predict(X_train)
print( 'naive bayes train accuracy is ' , metrics.accuracy_score(y_train, y_trai
y_test_nb= nb.predict(X_test)
print( ' naive bayes test accuracy is ', metrics.accuracy_score(y_test, y_test_n
#decision tree
dt= tree.DecisionTreeClassifier()
dt.fit(X_train, y_train)
y_train_dt = dt.predict(X_train)
print( ' decision tree train accuracy is ' , metrics.accuracy_score(y_train, y_t
y_test_dt = dt.predict(X_test)
print (' decision tree test accuracy is ', metrics.accuracy_score(y_test, y_test
#neural nework
scalar = MinMaxScaler()
X_train_scaled = scalar.fit_transform(X_train)
X_test_scaled = scalar.transform(X_test)
nn = MLPClassifier(hidden_layer_sizes = (10,5), max_iter=10000)
nn.fit(X_train_scaled, y_train)
y_train_nn = nn.predict(X_train_scaled)
```

```
print ('   neural network train accuracy is ',  metrics.accuracy_score(y_train,
y_test_nn = nn.predict(X_test_scaled)
print( ' neural network test accuracy is ' , metrics.accuracy_score(y_test, y_te

#Based on the output of the test accuracy results, the neural network performed

from sklearn.linear_model import LogisticRegression

model = LogisticRegression()
model.fit(X_train_scaled, y_train)
importances = pd.DataFrame(data={
    'Attribute': X_train.columns,
    'Importance': model.coef_[0]
})
importances = importances.sort_values(by='Importance', ascending=False)
import matplotlib.pyplot as plt
plt.bar(x=importances['Attribute'], height=importances['Importance'], color='#08
plt.title('Feature importances obtained from coefficients', size=20)
plt.xticks(rotation='vertical')
plt.show()
#from the plot, glucose has the most importance
```

```
Age               0
BMI               0
Glucose           0
Insulin           0
HOMA              0
Leptin            0
Adiponectin       0
Resistin          0
MCP.1             0
Classification    0
dtype: int64
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 116 entries, 0 to 115
Data columns (total 10 columns):
 #   Column          Non-Null Count  Dtype
---  ------          --------------  -----
 0   Age             116 non-null    int64
 1   BMI             116 non-null    float64
 2   Glucose         116 non-null    int64
 3   Insulin         116 non-null    float64
 4   HOMA            116 non-null    float64
 5   Leptin          116 non-null    float64
 6   Adiponectin     116 non-null    float64
 7   Resistin        116 non-null    float64
 8   MCP.1           116 non-null    float64
 9   Classification  116 non-null    int64
dtypes: float64(7), int64(3)
memory usage: 9.2 KB
              Age         BMI     Glucose     Insulin        HOMA      Leptin  \
count  116.000000  116.000000  116.000000  116.000000  116.000000  116.000000
mean    57.301724   27.582111   97.793103   10.012086    2.694988   26.615080
std     16.112766    5.020136   22.525162   10.067768    3.642043   19.183294
min     24.000000   18.370000   60.000000    2.432000    0.467409    4.311000
25%     45.000000   22.973205   85.750000    4.359250    0.917966   12.313675
50%     56.000000   27.662416   92.000000    5.924500    1.380939   20.271000
75%     71.000000   31.241442  102.000000   11.189250    2.857787   37.378300
max     89.000000   38.578759  201.000000   58.460000   25.050342   90.280000

       Adiponectin    Resistin        MCP.1  Classification
count   116.000000  116.000000   116.000000      116.000000
mean     10.180874   14.725966   534.647000        1.551724
```

```
std       6.843341   12.390646   345.912663        0.499475
min       1.656020    3.210000    45.843000        1.000000
25%       5.474283    6.881763   269.978250        1.000000
50%       8.352692   10.827740   471.322500        2.000000
75%      11.815970   17.755207   700.085000        2.000000
max      38.040000   82.100000  1698.440000        2.000000
                     Age       BMI   Glucose   Insulin      HOMA    Leptin  \
Age             1.000000  0.008530  0.230106  0.032495  0.127033  0.102626
BMI             0.008530  1.000000  0.138845  0.145295  0.114480  0.569593
Glucose         0.230106  0.138845  1.000000  0.504653  0.696212  0.305080
Insulin         0.032495  0.145295  0.504653  1.000000  0.932198  0.301462
HOMA            0.127033  0.114480  0.696212  0.932198  1.000000  0.327210
Leptin          0.102626  0.569593  0.305080  0.301462  0.327210  1.000000
Adiponectin    -0.219813 -0.302735 -0.122121 -0.031296 -0.056337 -0.095389
Resistin        0.002742  0.195350  0.291327  0.146731  0.231101  0.256234
MCP.1           0.013462  0.224038  0.264879  0.174356  0.259529  0.014009
Classification -0.043555 -0.132586  0.384315  0.276804  0.284012 -0.001078

                Adiponectin  Resistin     MCP.1  Classification
Age               -0.219813  0.002742  0.013462       -0.043555
BMI               -0.302735  0.195350  0.224038       -0.132586
Glucose           -0.122121  0.291327  0.264879        0.384315
Insulin           -0.031296  0.146731  0.174356        0.276804
HOMA              -0.056337  0.231101  0.259529        0.284012
Leptin            -0.095389  0.256234  0.014009       -0.001078
Adiponectin        1.000000 -0.252363 -0.200694       -0.019490
Resistin          -0.252363  1.000000  0.366474        0.227310
MCP.1             -0.200694  0.366474  1.000000        0.091381
Classification    -0.019490  0.227310  0.091381        1.000000
logr train accuracy is  0.8271604938271605
 logr test accuracy is  0.5714285714285714
naive bayes train accuracy is  0.6172839506172839
 naive bayes test accuracy is  0.6857142857142857
 decision tree train accuracy is  1.0
 decision tree test accuracy is  0.5142857142857142
   neural network train accuracy is  1.0
 neural network test accuracy is  0.7428571428571429
```



In [ ]: