```
import os
import pandas as pd
from sklearn.preprocessing import MinMaxScaler
from sklearn.neural_network import MLPClassifier

def write_performance(dict):
    df = pd.DataFrame(dict)
    head = os.path.isfile('Perform.csv')
    df.to_csv('Perform.csv', sep=';', mode='a', index = False, header = not head)

# Read Data
data = pd.read_csv('./DatenSS2018/measures.csv',sep=';')
data_filled=data.fillna(data.mean())

prediction = pd.read_csv('./DatenSS2018/to_predict.csv',sep=';')
prediction_filled=prediction.fillna(prediction.mean())

# Umcodieren der Altersklasse
i = 0
coded_class = pd.DataFrame(columns=['original','coded'])
for x in data_filled['P-Altersklasse'].unique():
    data_filled.loc[data_filled['P-Altersklasse']==x,'P-Altersklasse'] = i
    coded_class=coded_class.append(pd.DataFrame(data={'original':[x],'coded':[i]}))
    i += 1

# Umcodieren der Geschlechtsklasse
data_filled.loc[data_filled['P-Geschlecht']=='m','P-Geschlecht'] = 0
data_filled.loc[data_filled['P-Geschlecht']=='w','P-Geschlecht'] = 1
prediction_filled.loc[prediction_filled['P-Geschlecht']=='m','P-Geschlecht'] = 0
prediction_filled.loc[prediction_filled['P-Geschlecht']=='w','P-Geschlecht'] = 1

# Korrelationskoeffizienten
corrcoef_class=data_filled.corr()
corr_alter=corrcoef_class['P-Altersklasse']

# Auswahl der n Features mit größter positive/negative Korrelation
n = 10
data_class_corr = pd.DataFrame(data_filled[(corr_alter.drop('P-Altersklasse').nlargest(n=n).index) |
(corr_alter.drop('P-Altersklasse').nsmallest(n=n).index)])
predict_corr = pd.DataFrame(prediction_filled[(corr_alter.drop('P-Altersklasse').nlargest(n=n).index)
| (corr_alter.drop('P-Altersklasse').nsmallest(n=n).index)])

corrcoef_class2 = data_class_corr.corr()

# MLP Einstellungen
activation = 'logistic'
solver = 'adam'
learning_rate_init = 0.3
max_iter = 100

# Erstellen der Trainingsdaten + Label
train_X = data_class_corr.astype('float64').values
train_y = data_filled['P-Altersklasse'].astype('float64').values
```

```
# Skalierung
scaler = MinMaxScaler(feature_range=(0,1))
scaler.fit(train_X)
train_X = scaler.transform(train_X)

# Training
clf = MLPClassifier(activation=activation,
                    solver=solver,
                    learning_rate_init=learning_rate_init,
                    max_iter=max_iter)

clf.fit(train_X,train_y)

# Prediction für to_predict
predict_X = predict_corr.astype('float64').values
predict_X = scaler.transform(predict_X)

prediction=pd.DataFrame(clf.predict(predict_X),columns=['P-Altersklasse'])
prediction.to_csv('my_prediction.csv', index=False)
```