```python
#Solution to group assignment
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import LabelEncoder
from sklearn.ensemble import RandomForestClassifier
from sklearn.metrics import accuracy_score

# Load the dataset
df = pd.read_csv('https://archive.ics.uci.edu/ml/machine-learning-databases/credit-screening/crx.data', header=None)

# Replace missing values with NaN
df = df.replace('?', pd.NaT)

# Encode categorical columns as numerical values
le = LabelEncoder()
for col in df.columns:
    if df[col].dtype == 'object':
        df[col] = le.fit_transform(df[col].astype(str))

# Split the data into features and labels
X = df.iloc[:, :-1]
y = df.iloc[:, -1]

# Split the data into training and testing sets
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

# Instantiate a Random Forest classifier with 100 trees
clf = RandomForestClassifier(n_estimators=100, random_state=42)

# Train the classifier on the training data
clf.fit(X_train, y_train)

# Use the trained classifier to make predictions on the testing data
y_pred = clf.predict(X_test)

# Evaluate the performance of the classifier using accuracy score
acc = accuracy_score(y_test, y_pred)
print("Accuracy:", acc)
```