```
import pandas as pd
from sklearn.preprocessing import StandardScaler
from sklearn.decomposition import PCA
from sklearn.ensemble import RandomForestClassifier
from sklearn.model_selection import cross_val_score


data = pd.read_csv('/content/drive/MyDrive/Complab/heart.csv')
print(data.head())
print(data.info())


data['age_group'] = pd.cut(data['age'], bins=[0, 40, 50, 60, 100],
labels=['<40', '40-50', '50-60', '60+'])
data['chol_category'] = pd.cut(data['chol'], bins=[0, 200, 239, 1000],
labels=['normal', 'borderline', 'high'])
data['age_chol'] = data['age'] * data['chol']
data['thalach_age'] = data['thalachh'] * data['age']


features = data.drop(columns=['output', 'age_group', 'chol_category'])
features = pd.get_dummies(features, drop_first=True)

# Scale features
scaler = StandardScaler()
features_scaled = scaler.fit_transform(features)

# Perform PCA
pca = PCA(n_components=0.95)  # Retain 95% of the variance
features_pca = pca.fit_transform(features_scaled)

# Output PCA results
explained_variance = pca.explained_variance_ratio_
n_components = pca.n_components_

print(features_pca.shape)
print(explained_variance)
print(n_components)


# Train RandomForestClassifier with all features
X = pd.get_dummies(data.drop(columns=['output', 'age_group', 'chol_category']),
drop_first=True)
y = data['output']

rf_model = RandomForestClassifier(n_estimators=100, random_state=42)
rf_model.fit(X, y)

# Extract and display feature importances
feature_importances = rf_model.feature_importances_
```

```
important_features = pd.Series(feature_importances,
index=X.columns).sort_values(ascending=False)

print(important_features.head(15))

# Train RandomForestClassifier using only the top 15 important features
top_features = important_features.head(15).index
X_top = X[top_features]

rf_model_top = RandomForestClassifier(n_estimators=100, random_state=42)
rf_model_top.fit(X_top, y)

# Perform cross-validation
scores = cross_val_score(rf_model_top, X_top, y, cv=5)
print(f'Cross-validated scores: {scores}')
print(f'Mean score: {scores.mean()}')
```

```
   age  sex  cp  trtbps  chol  fbs  restecg  thalachh  exng  oldpeak  slp  \
0   63    1   3     145   233    1        0       150     0      2.3    0
1   37    1   2     130   250    0        1       187     0      3.5    0
2   41    0   1     130   204    0        0       172     0      1.4    2
3   56    1   1     120   236    0        1       178     0      0.8    2
4   57    0   0     120   354    0        1       163     1      0.6    2

   caa  thall  output
0    0      1       1
1    0      2       1
2    0      2       1
3    0      2       1
4    0      2       1
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 303 entries, 0 to 302
Data columns (total 14 columns):
 #   Column    Non-Null Count  Dtype
---  ------    --------------  -----
 0   age       303 non-null    int64
 1   sex       303 non-null    int64
 2   cp        303 non-null    int64
 3   trtbps    303 non-null    int64
 4   chol      303 non-null    int64
 5   fbs       303 non-null    int64
 6   restecg   303 non-null    int64
 7   thalachh  303 non-null    int64
 8   exng      303 non-null    int64
 9   oldpeak   303 non-null    float64
 10  slp       303 non-null    int64
 11  caa       303 non-null    int64
 12  thall     303 non-null    int64
 13  output    303 non-null    int64
dtypes: float64(1), int64(13)
memory usage: 33.3 KB
None
(303, 12)
[0.20769271 0.1671179  0.09044119 0.08171712 0.06844876 0.06815014
 0.05846558 0.05555468 0.05028512 0.04747908 0.04314118 0.03525778]
```

```
12
cp             0.121679
oldpeak        0.110464
caa            0.104801
thalachh       0.102865
thall          0.098752
thalach_age    0.078321
age_chol       0.067467
age            0.064543
chol           0.057642
trtbps         0.057511
exng           0.045652
slp            0.041447
sex            0.028861
restecg        0.013131
fbs            0.006862
dtype: float64
Cross-validated scores: [0.83606557 0.86885246 0.78688525 0.78333333 0.8       ]
Mean score: 0.8150273224043716
```