Machine Learning

Lec # 4
Gender Identification using Scikit-Learn

Introduction

- Aim
  - The main aim of this tutorial is to explain the task of gender identification using Scikit-Learn Machine Learning toolkit.
- Task
  - Learn Input-Output Function
  - Given a human as input predict its gender (output)

Introduction

- Goal
  - The problem of gender prediction is treated as a supervised learning problem.
  - We need
    - Labelled data
    - High quality data
    - Large amount of data

Input and Output

Input:

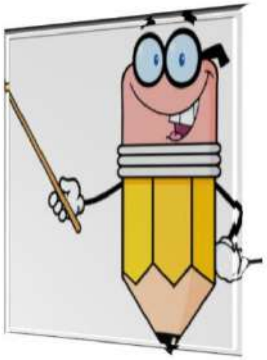
- Human
  - Represented as set of attributes (Height, Weight, Hair Length, Beard, Scarf)

Output:

- Gender of human
- Represented as Gender attribute (Male/Female)

Goal:

- Learn from Input to predict Output



Three Phases of Machine Learning

Training

- Use subset of data (called Train data) to train model (learning)

Testing

- Use subset of data (called Test Data) to evaluate train model

Application

- Use your learned/trained model in real world applications

PHASES 1 & 2: TRAINING AND TESTING

Step 1: Import Libraries

Step 2: Read, Understand and Pre-process Train/Test Data

Step 2.1: Read Data

Step 2.2: Understand Data

Step 2.3: Pre-process Data

PHASES 1 & 2: TRAINING AND TESTING

Step 3: Label Encoding for Train/Test Data

Step 4: Feature Extraction – Values of Attributes

Step 5: Train Machine Learning Algorithms using Train Data

Step 6: Evaluate Machine Learning Algorithms using Test Data

Step 7: Selection of Best Model

PHASE 3: APPLICATION PHASE

Step 8: Application Phase

Step 8.1: Combine Data (Train + Test)

Step 8.2: Train Best Model (see Step 7) on all data (Train + Test)

Step 8.3: Save the Trained Model as Pickle File

PHASE 3: APPLICATION PHASE

Step 9: Make prediction on unseen/new data

Step 9.1: Load the Trained Model (saved in Step 8.3)

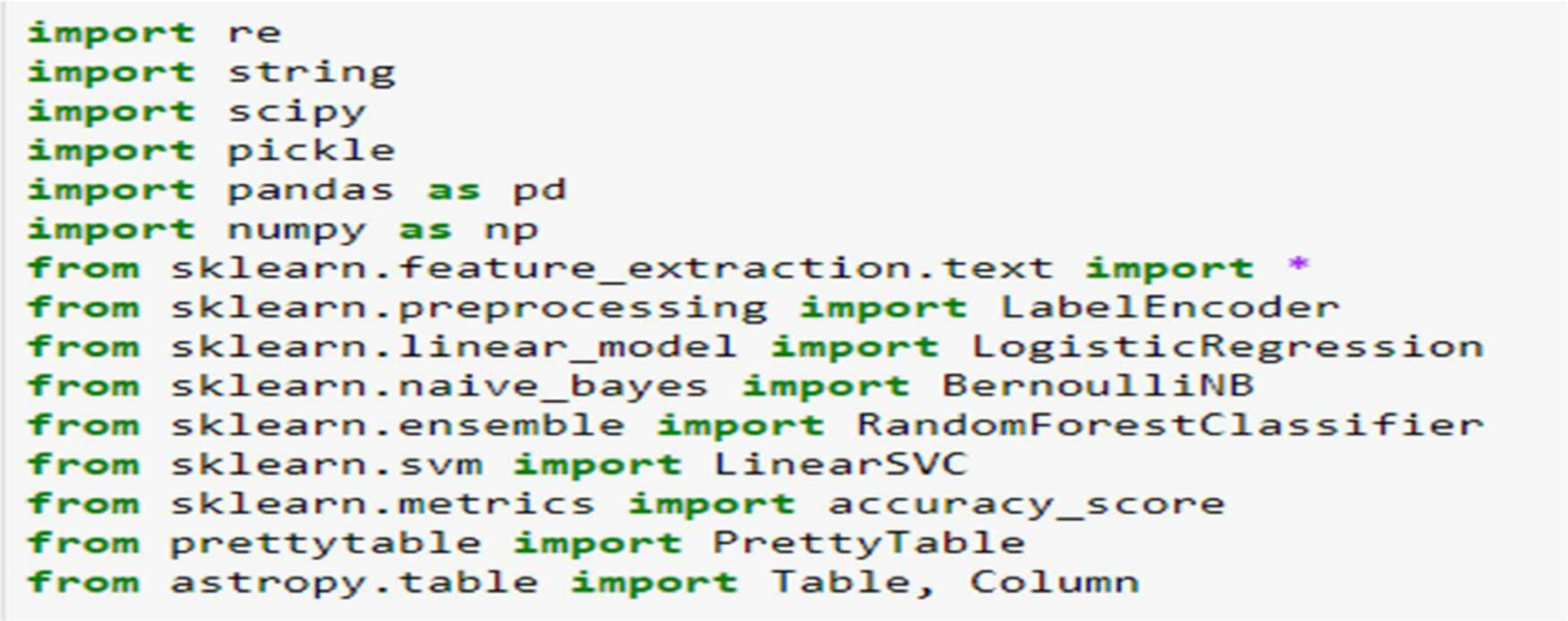
Step 9.2: Take Input from User

PHASE 3: APPLICATION PHASE

Step 9.3: Convert User Input into Feature Vector (Same as Feature Vector of Trained Model)

Step 9.4: Apply Trained Model on Feature Vector of Unseen Data and Output Prediction (Male/Female) to User

Step 1: Import Libraries

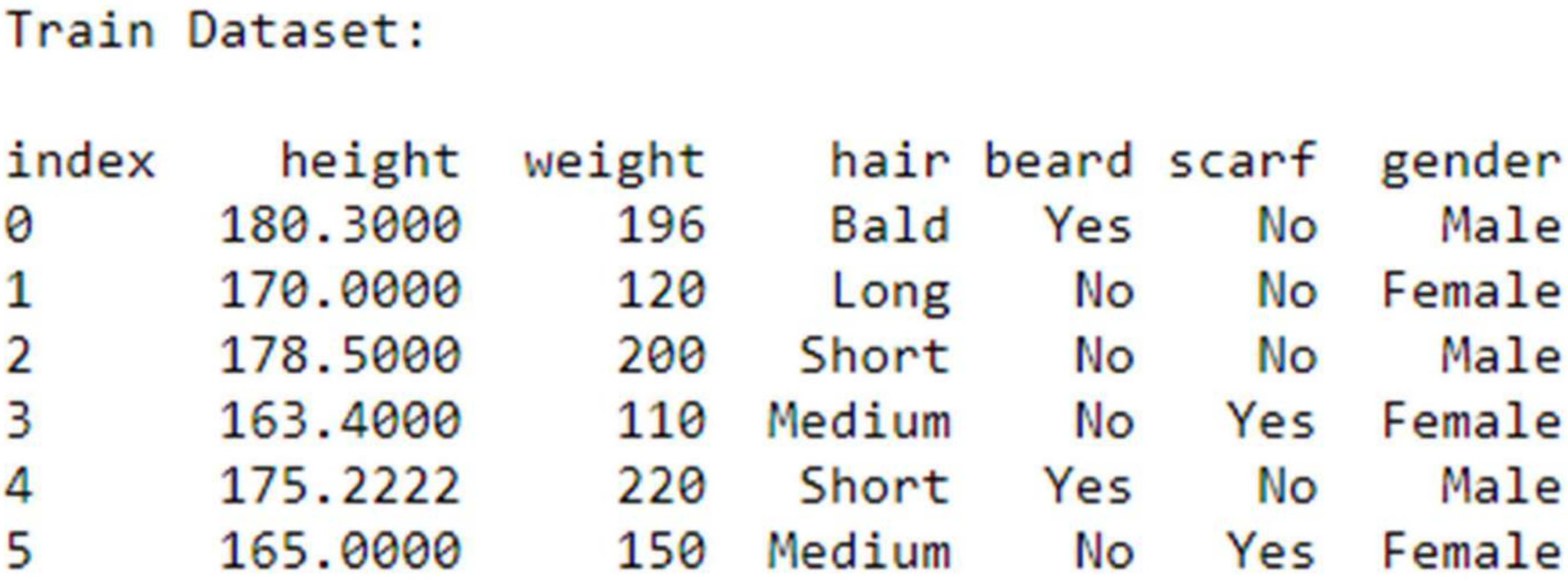


```
import re
import string
import scipy
import pickle
import pandas as pd
import numpy as np
from sklearn.feature_extraction.text import *
from sklearn.preprocessing import LabelEncoder
from sklearn.linear_model import LogisticRegression
from sklearn.naive_bayes import BernoulliNB
from sklearn.ensemble import RandomForestClassifier
from sklearn.svm import LinearSVC
from sklearn.metrics import accuracy_score
from prettytable import PrettyTable
from astropy.table import Table, Column
```

Step 2: Read, Understand and Pre-process Train/Test Data

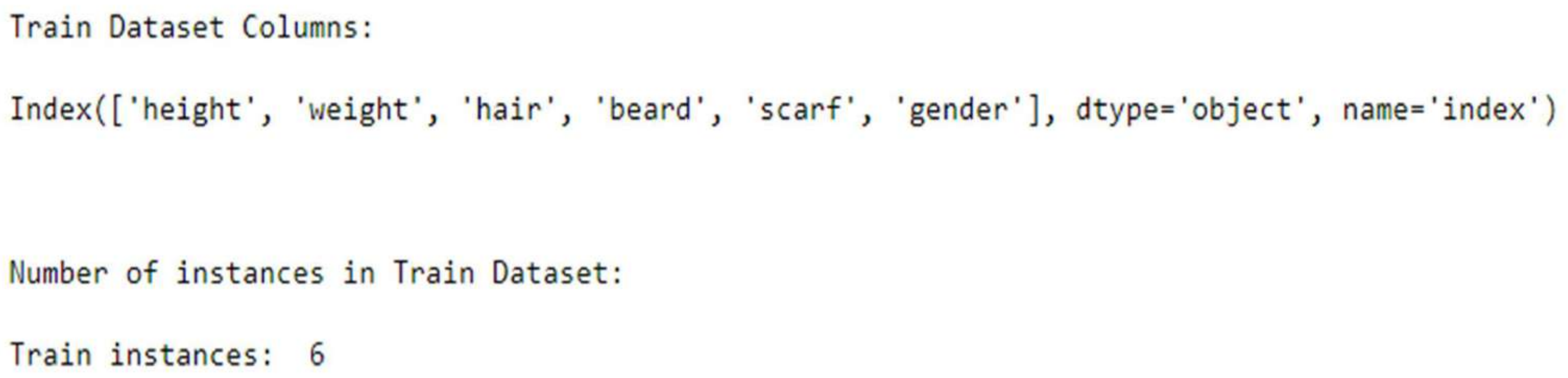
Read, Understand and Pre-process Train/Test Data

Step 2.2: Understand Data



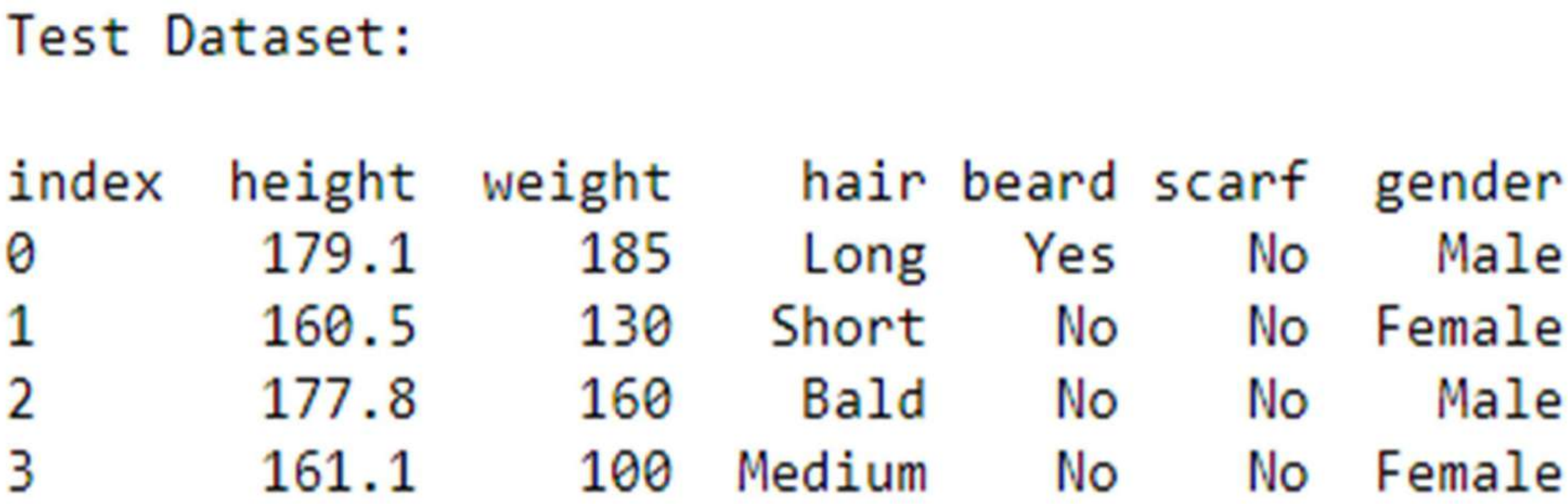
```
Train Dataset:

index    height  weight    hair beard scarf  gender
0      180.3000     196    Bald   Yes    No    Male
1      170.0000     120    Long    No    No  Female
2      178.5000     200   Short    No    No    Male
3      163.4000     110  Medium    No   Yes  Female
4      175.2222     220   Short   Yes    No    Male
5      165.0000     150  Medium    No   Yes  Female
```

Step 2.2: Understand Data



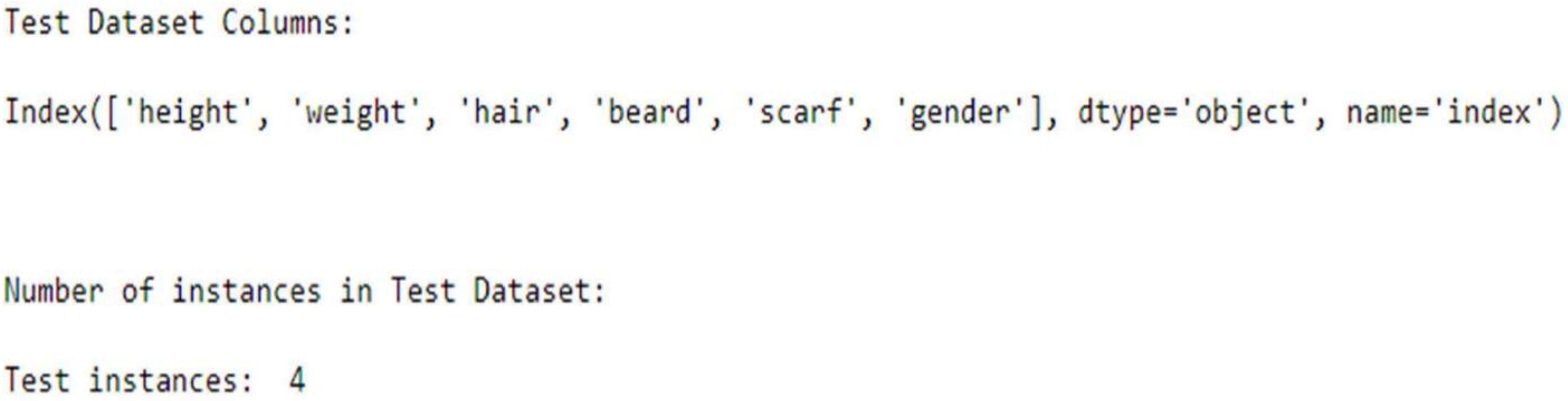
```
Train Dataset Columns:

Index(['height', 'weight', 'hair', 'beard', 'scarf', 'gender'], dtype='object', name='index')


Number of instances in Train Dataset:

Train instances:  6
```

Step 2.2: Understand Data



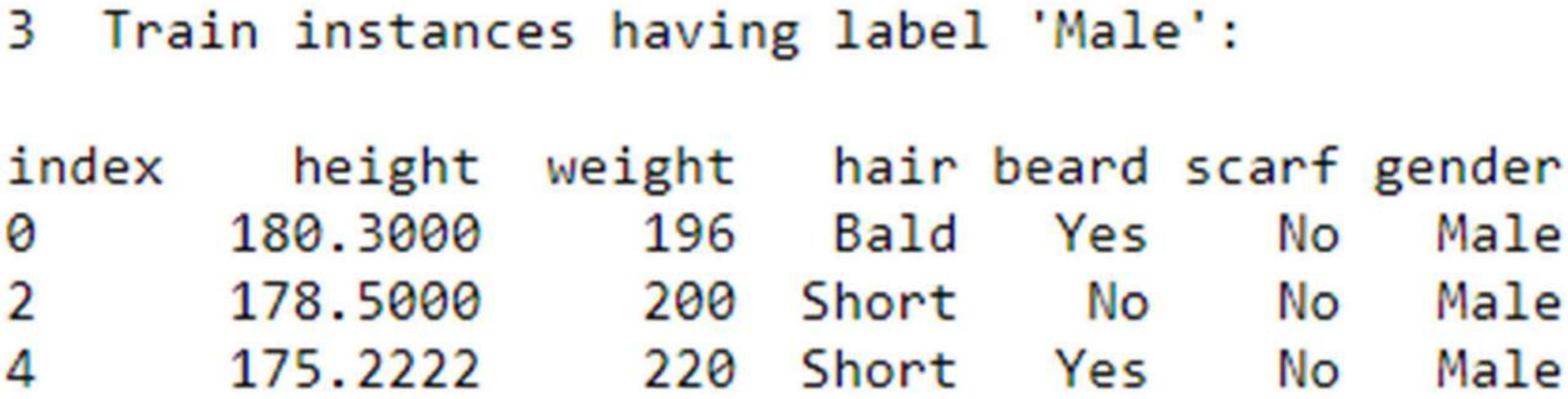
```
Test Dataset:

index  height  weight    hair beard scarf  gender
0       179.1     185    Long   Yes    No    Male
1       160.5     130   Short    No    No  Female
2       177.8     160    Bald    No    No    Male
3       161.1     100  Medium    No    No  Female
```

Step 2.2: Understand Data



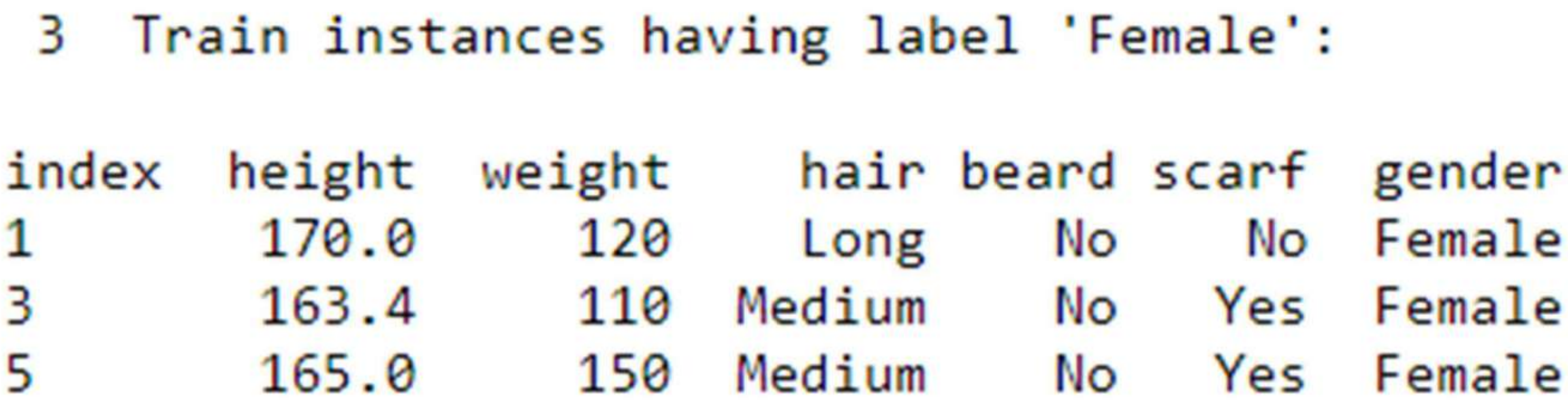
```
Test Dataset Columns:

Index(['height', 'weight', 'hair', 'beard', 'scarf', 'gender'], dtype='object', name='index')


Number of instances in Test Dataset:

Test instances:  4
```

Step 2.2: Understand Data



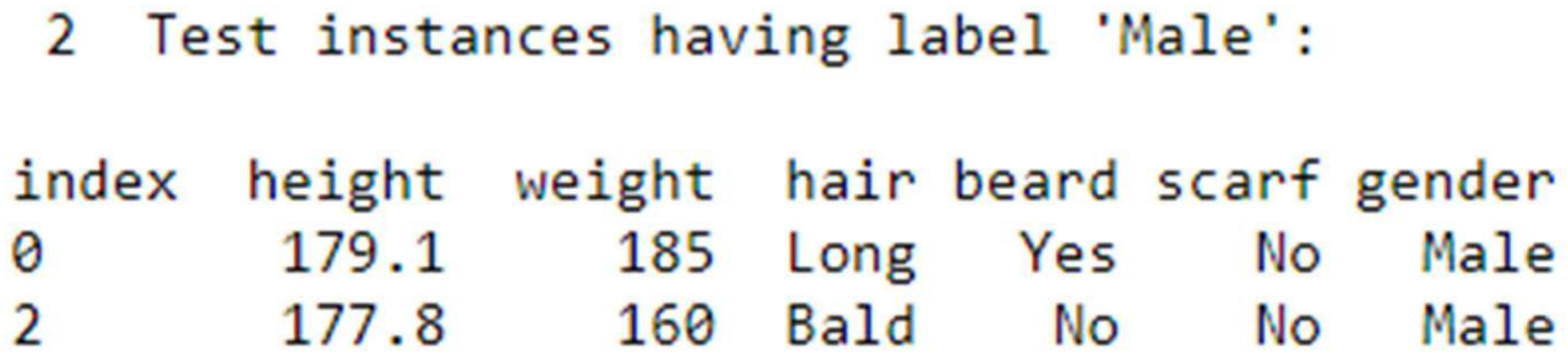
```
3  Train instances having label 'Male':

index      height  weight   hair beard scarf gender
0       180.3000     196   Bald   Yes    No   Male
2       178.5000     200  Short    No    No   Male
4       175.2222     220  Short   Yes    No   Male
```

Step 2.2: Understand Data



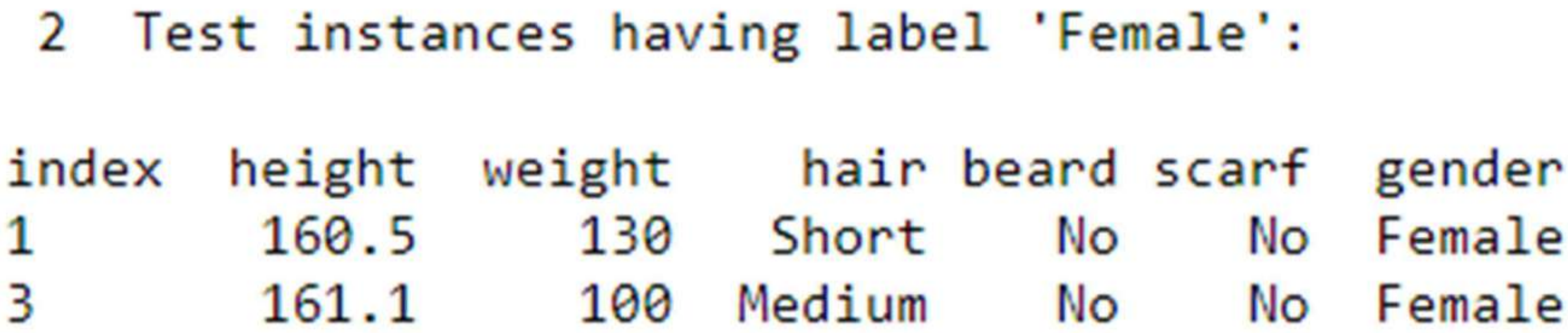
```
 3  Train instances having label 'Female':

index   height   weight     hair beard scarf  gender
1        170.0      120     Long    No    No  Female
3        163.4      110   Medium    No   Yes  Female
5        165.0      150   Medium    No   Yes  Female
```

Step 2.2: Understand Data



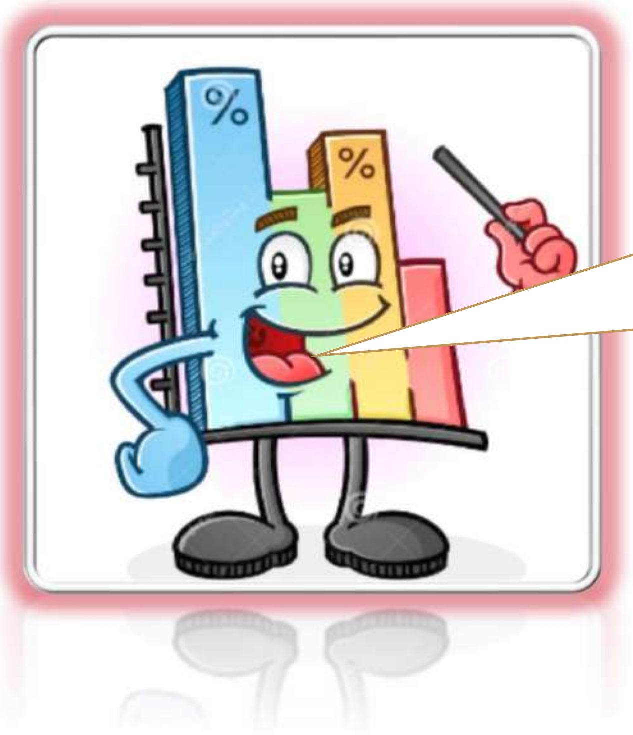
```
 2  Test instances having label 'Male':

index   height   weight   hair beard scarf gender
0        179.1      185   Long   Yes    No   Male
2        177.8      160   Bald    No    No   Male
```

Step 2.2: Understand Data



```
 2  Test instances having label 'Female':

index  height  weight    hair beard scarf  gender
1       160.5     130   Short    No    No  Female
3       161.1     100  Medium    No    No  Female
```

Step 2.2: Understand Data

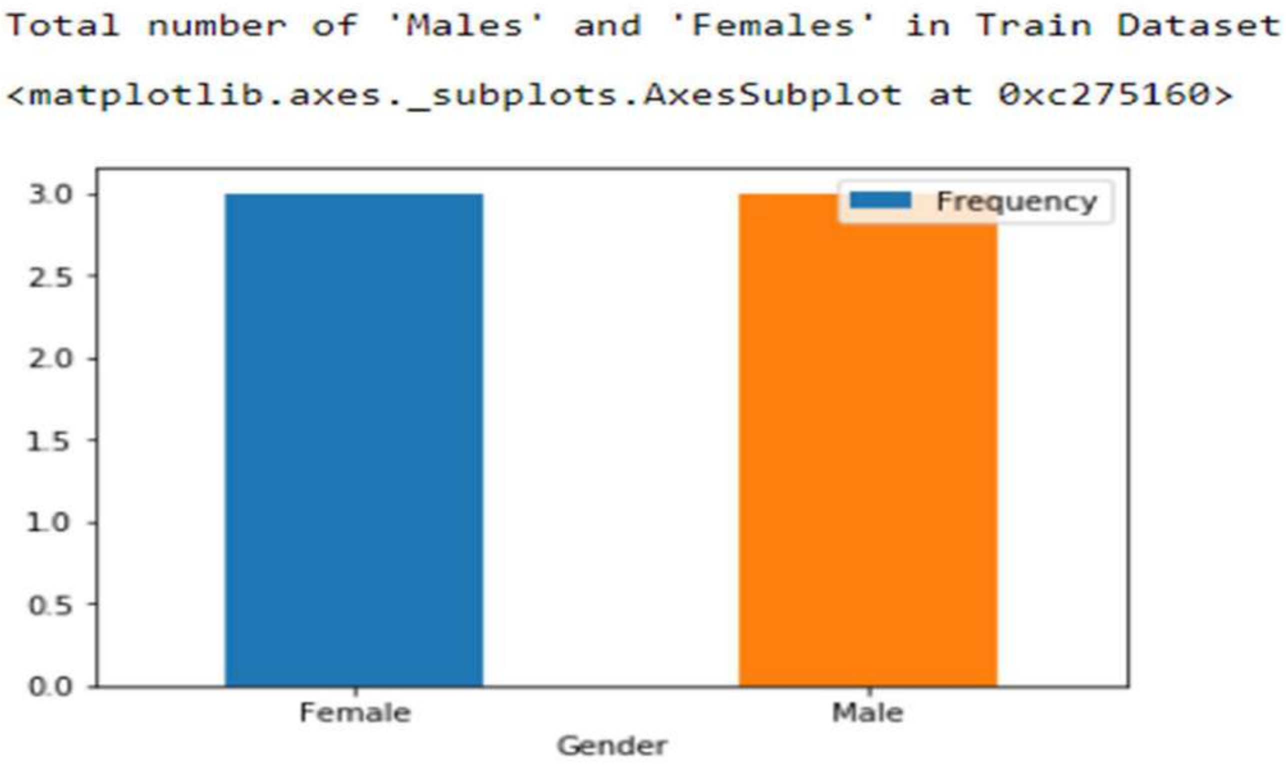


Understanding Data via

GRAPH is easy.

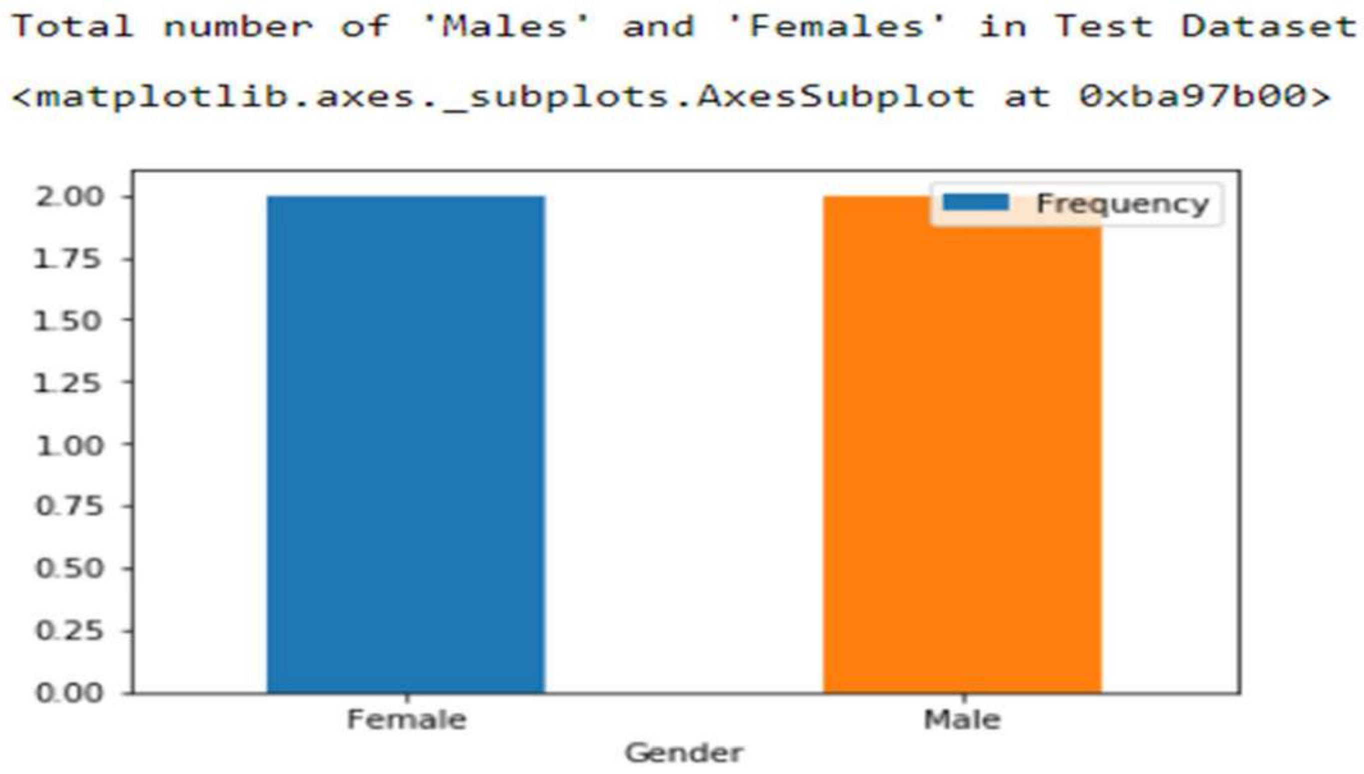
Let's Go!

Step 2.2: Understand Data



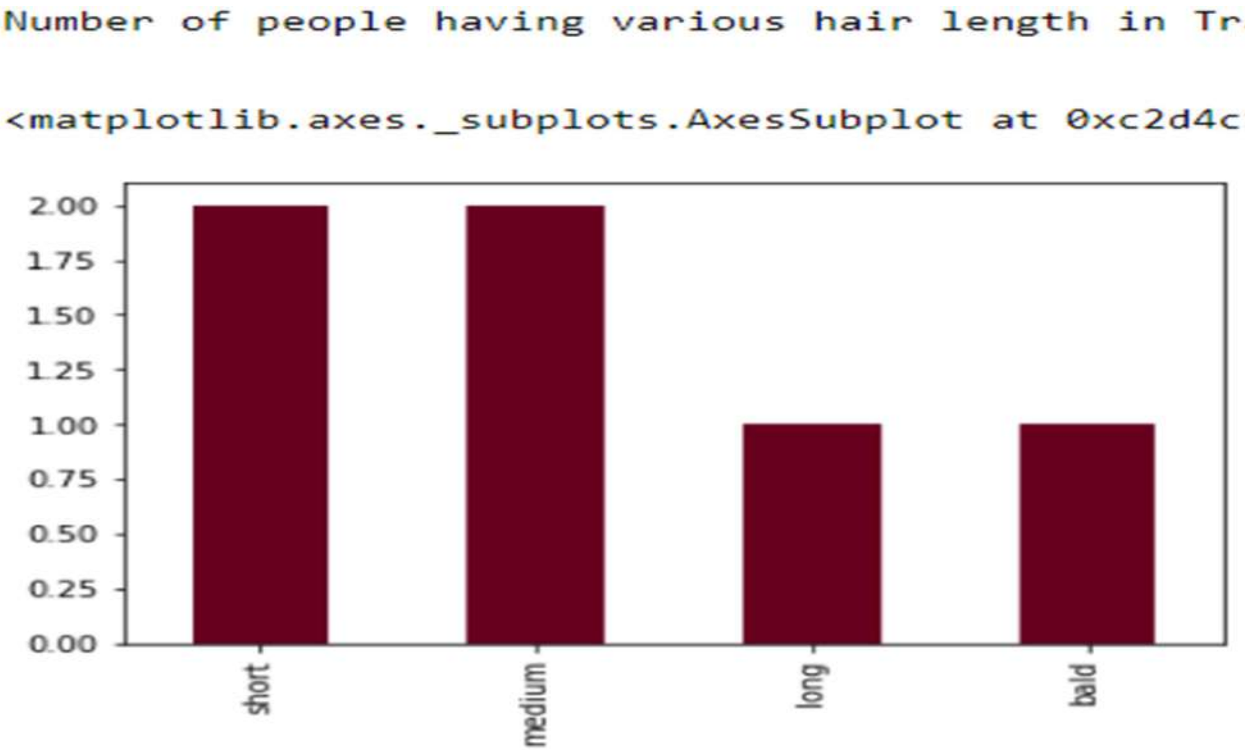
```
Total number of 'Males' and 'Females' in Train Dataset

<matplotlib.axes._subplots.AxesSubplot at 0xc275160>
```

Step 2.2: Understand Data



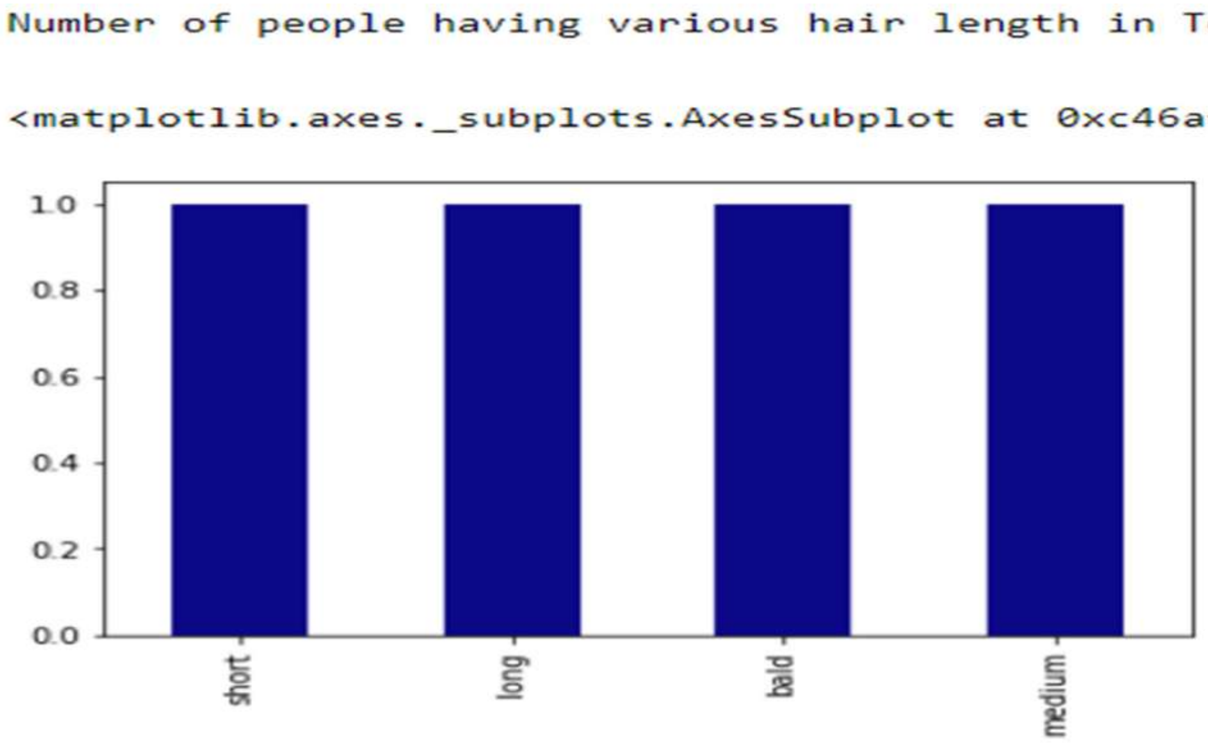
```
Total number of 'Males' and 'Females' in Test Dataset

<matplotlib.axes._subplots.AxesSubplot at 0xba97b00>
```

Step 2.2: Understand Data



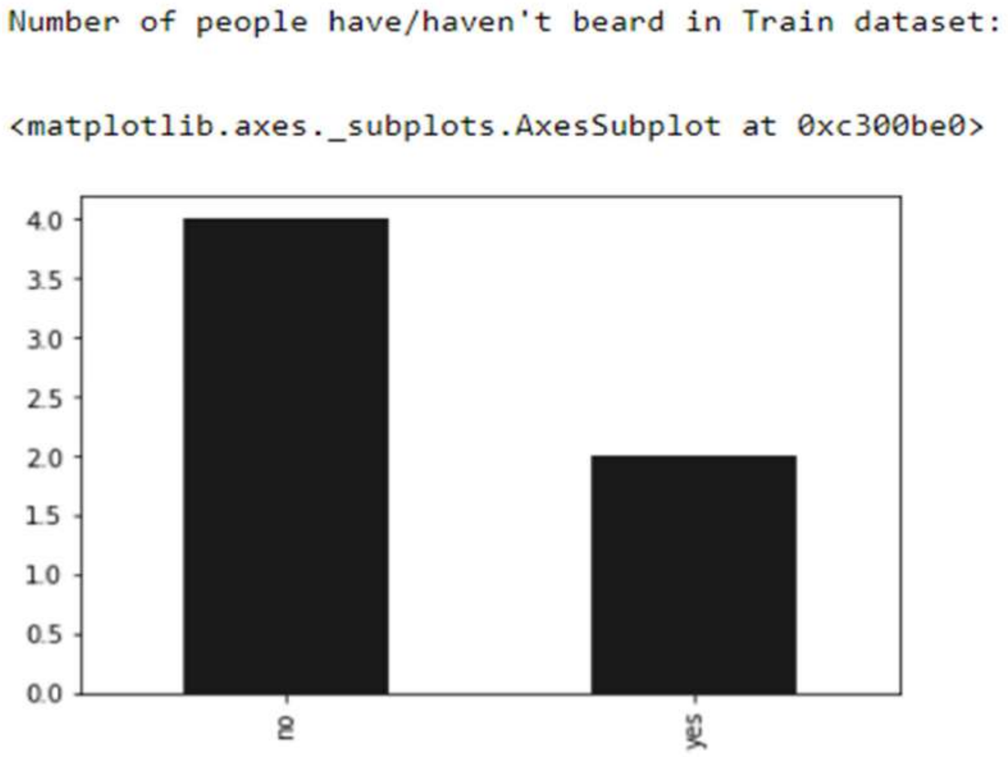
```
Number of people having various hair length in Train dataset:

<matplotlib.axes._subplots.AxesSubplot at 0xc2d4c18>
```

Step 2.2: Understand Data



```
Number of people having various hair length in Test dataset:

<matplotlib.axes._subplots.AxesSubplot at 0xc46af28>
```

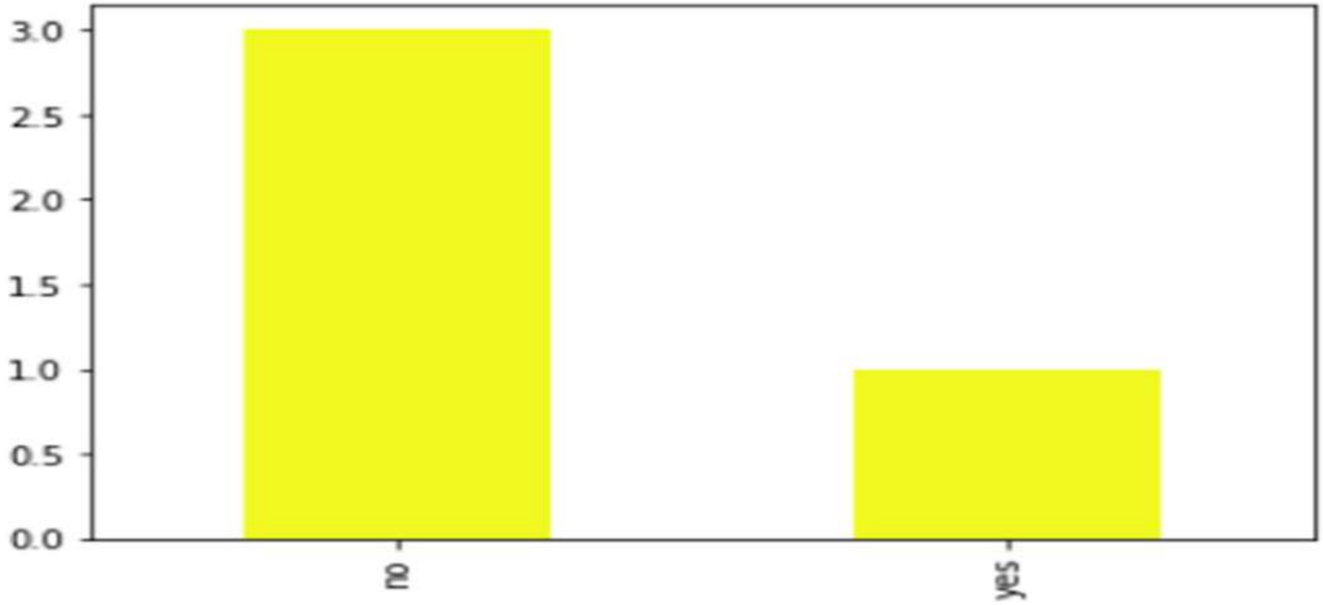
Step 2.2: Understand Data



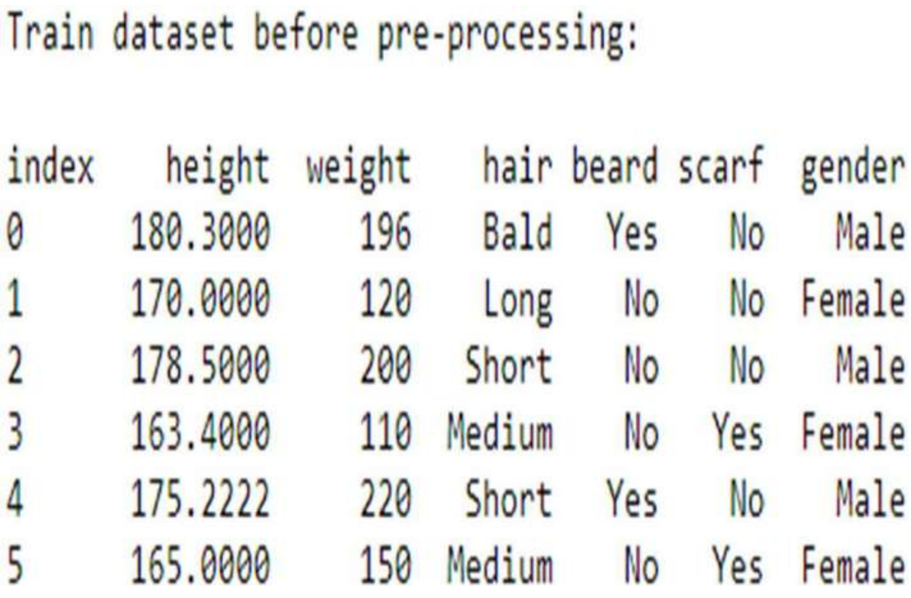
```
Number of people have/haven't beard in Train dataset:

<matplotlib.axes._subplots.AxesSubplot at 0xc300be0>
```

Step 2.2: Understand Data

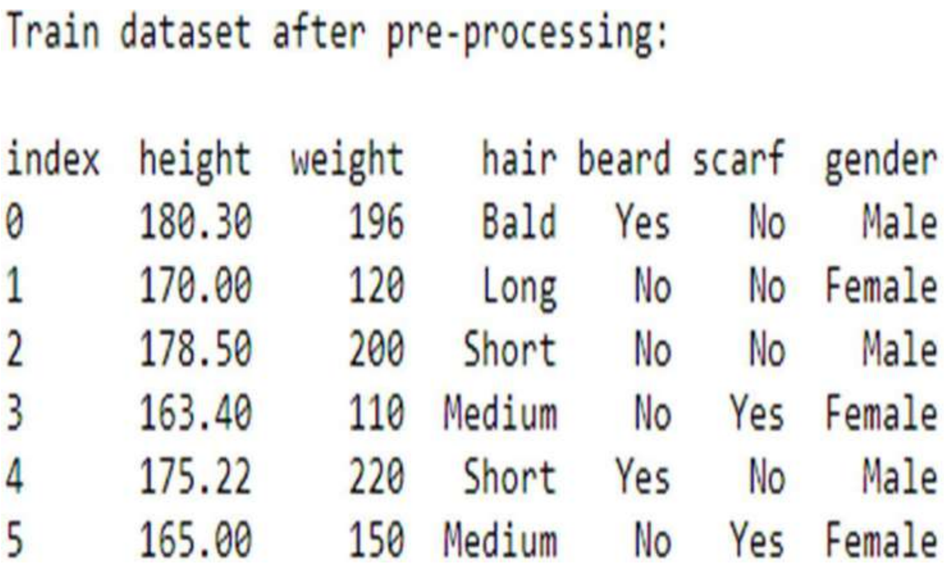
```
Number of people have/haven't beard in Test dataset:

<matplotlib.axes._subplots.AxesSubplot at 0xc4d2cc0>
```



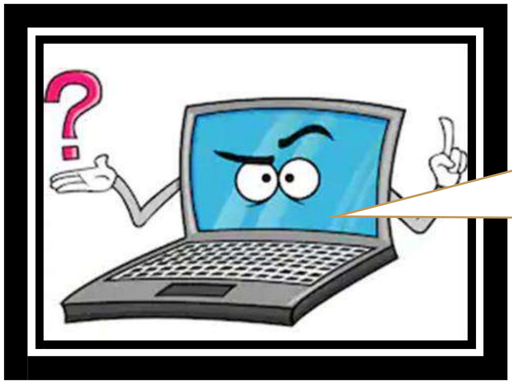
Step 2.3: Pre-Process Data



```
Train dataset before pre-processing:

index     height  weight    hair beard scarf  gender
0       180.3000     196    Bald   Yes    No    Male
1       170.0000     120    Long    No    No  Female
2       178.5000     200   Short    No    No    Male
3       163.4000     110  Medium    No   Yes  Female
4       175.2222     220   Short   Yes    No    Male
5       165.0000     150  Medium    No   Yes  Female
```

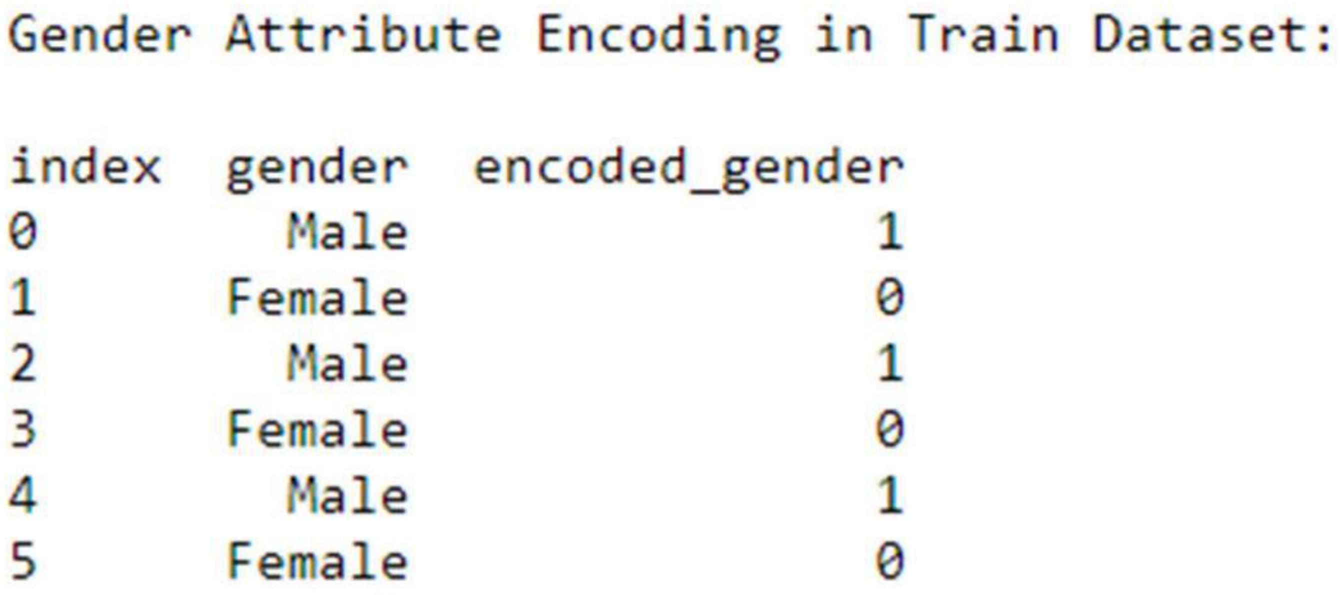


```
Train dataset after pre-processing:

index  height  weight    hair beard scarf  gender
0      180.30     196    Bald   Yes    No    Male
1      170.00     120    Long    No    No  Female
2      178.50     200   Short    No    No    Male
3      163.40     110  Medium    No   Yes  Female
4      175.22     220   Short   Yes    No    Male
5      165.00     150  Medium    No   Yes  Female
```



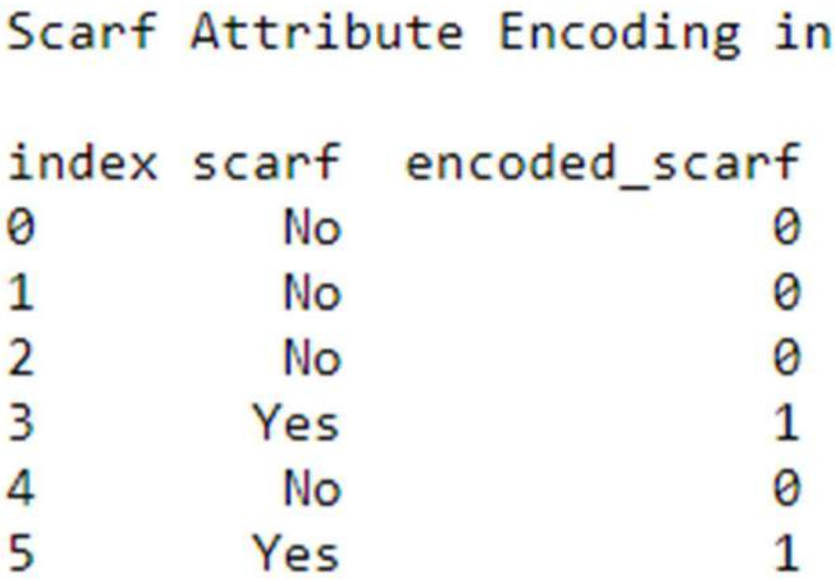
Please convert data to a form that I can understand

Step 3: Label Encoding for Train/Test Data



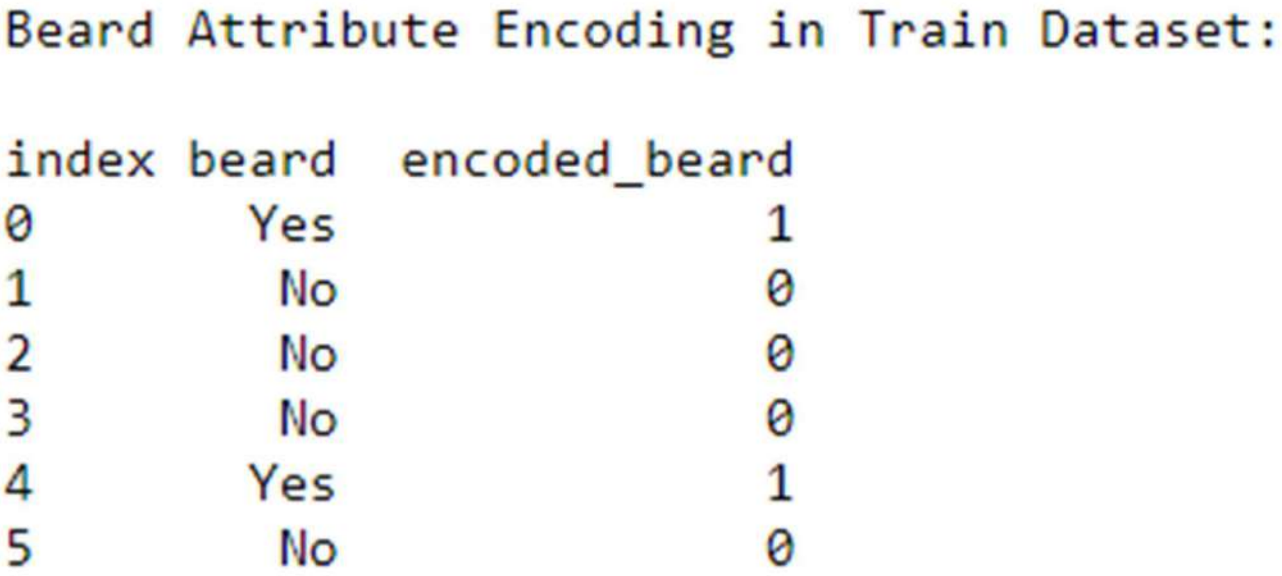
```
Gender Attribute Encoding in Train Dataset:

index  gender  encoded_gender
0        Male               1
1      Female               0
2        Male               1
3      Female               0
4        Male               1
5      Female               0
```

Step 3: Label Encoding for Train/Test Data



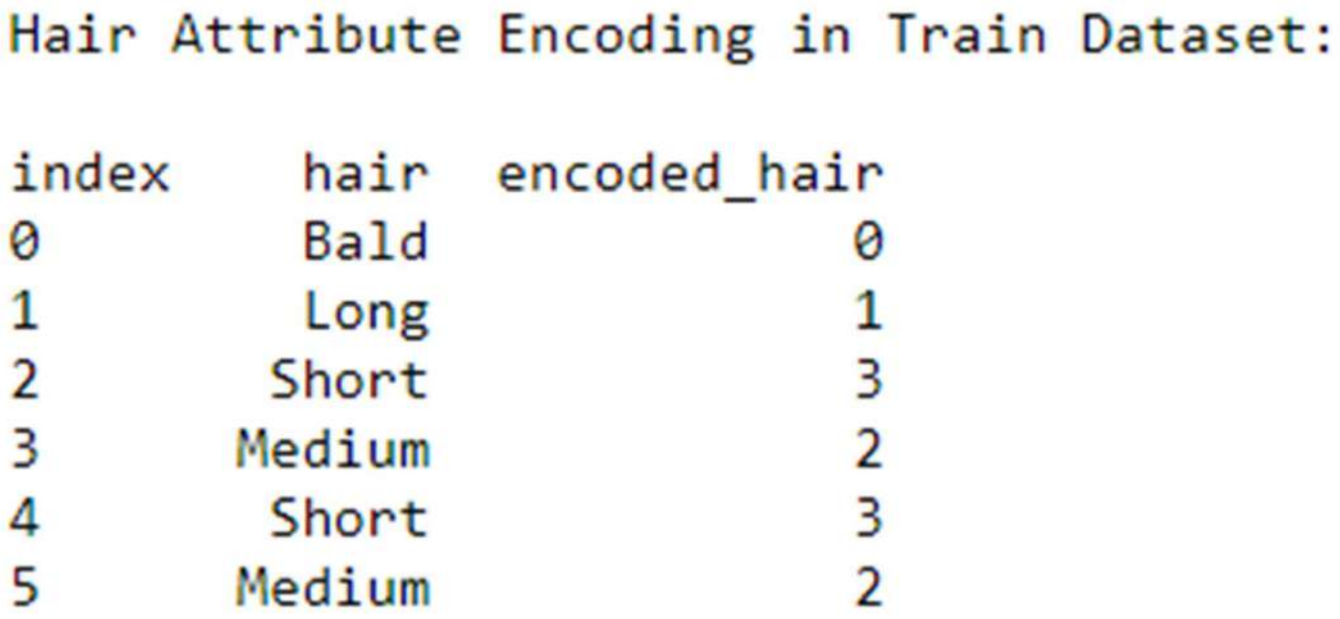
```
Scarf Attribute Encoding in Train Dataset:

index scarf  encoded_scarf
0        No              0
1        No              0
2        No              0
3       Yes              1
4        No              0
5       Yes              1
```

Step 3: Label Encoding for Train/Test Data



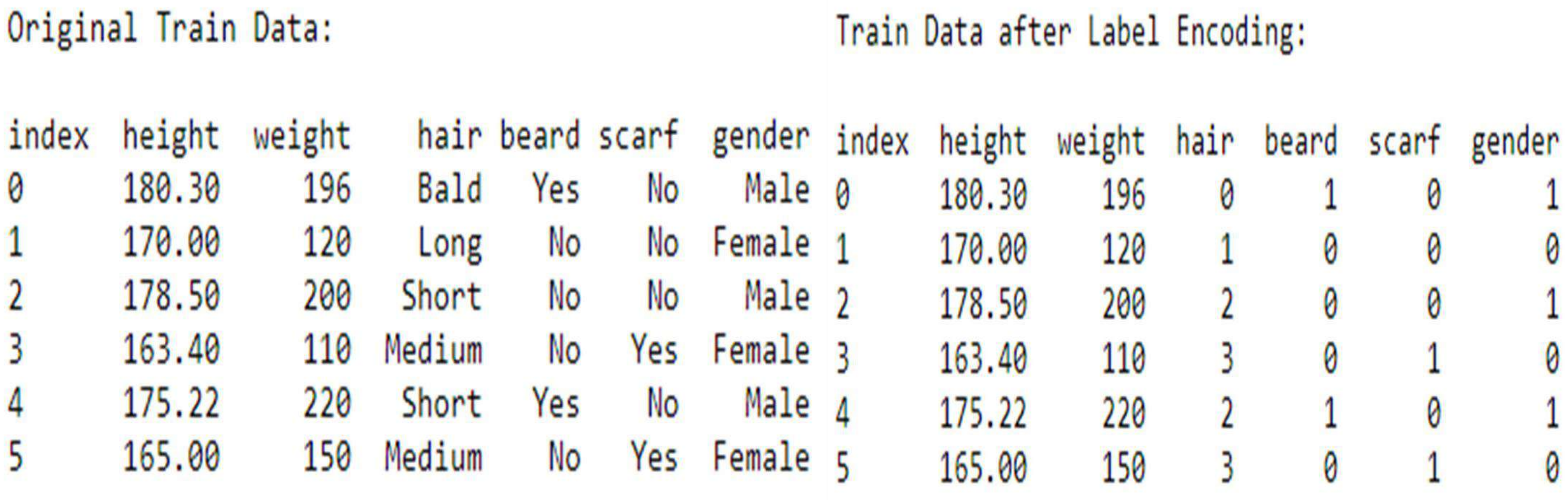
```
Beard Attribute Encoding in Train Dataset:

index beard  encoded_beard
0        Yes              1
1         No              0
2         No              0
3         No              0
4        Yes              1
5         No              0
```

Step 3: Label Encoding for Train/Test Data



```
Hair Attribute Encoding in Train Dataset:

index     hair  encoded_hair
0         Bald             0
1         Long             1
2        Short             3
3       Medium             2
4        Short             3
5       Medium             2
```

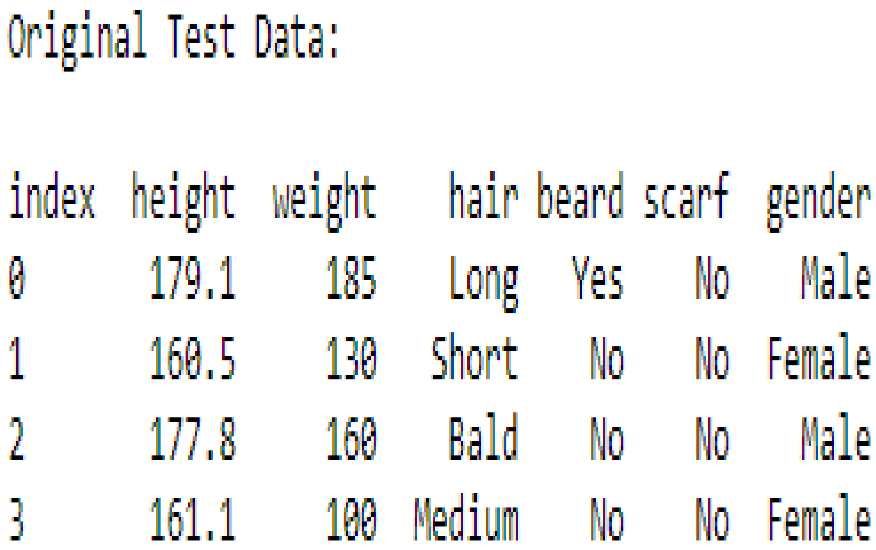
Step 3: Label Encoding for Train/Test Data



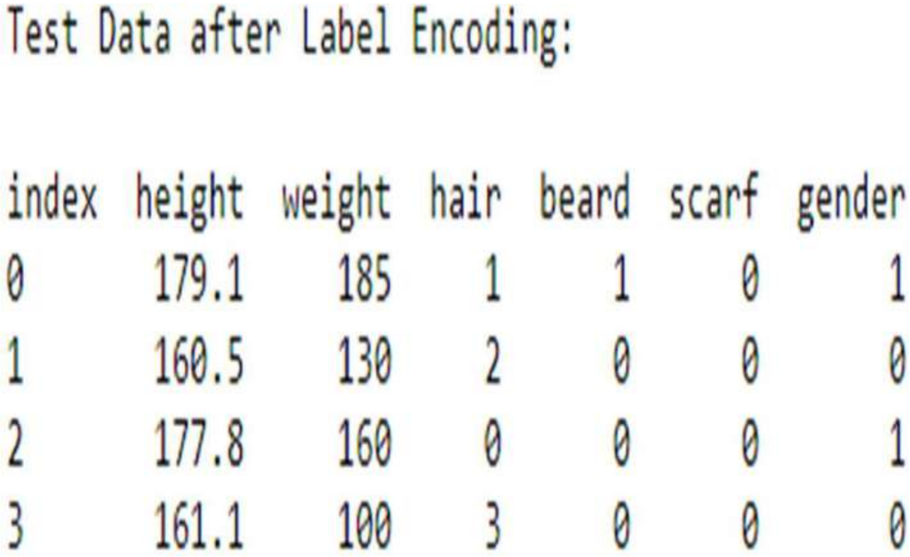
```
Original Train Data:

index  height  weight    hair beard scarf  gender
0      180.30     196    Bald   Yes    No    Male
1      170.00     120    Long    No    No  Female
2      178.50     200   Short    No    No    Male
3      163.40     110  Medium    No   Yes  Female
4      175.22     220   Short   Yes    No    Male
5      165.00     150  Medium    No   Yes  Female
```

```
Train Data after Label Encoding:

index  height  weight  hair  beard  scarf  gender
0      180.30     196     0      1      0       1
1      170.00     120     1      0      0       0
2      178.50     200     2      0      0       1
3      163.40     110     3      0      1       0
4      175.22     220     2      1      0       1
5      165.00     150     3      0      1       0
```

Step 3: Label Encoding for Train/Test Data

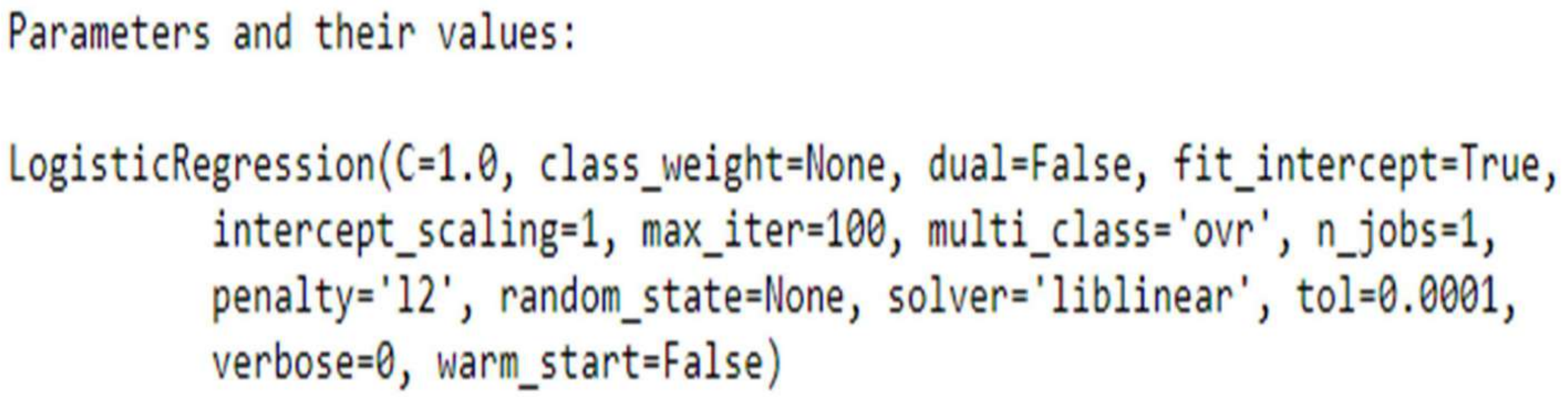


```
Original Test Data:

index  height  weight    hair beard scarf  gender
0       179.1     185    Long   Yes    No    Male
1       160.5     130   Short    No    No  Female
2       177.8     160    Bald    No    No    Male
3       161.1     100  Medium    No    No  Female
```



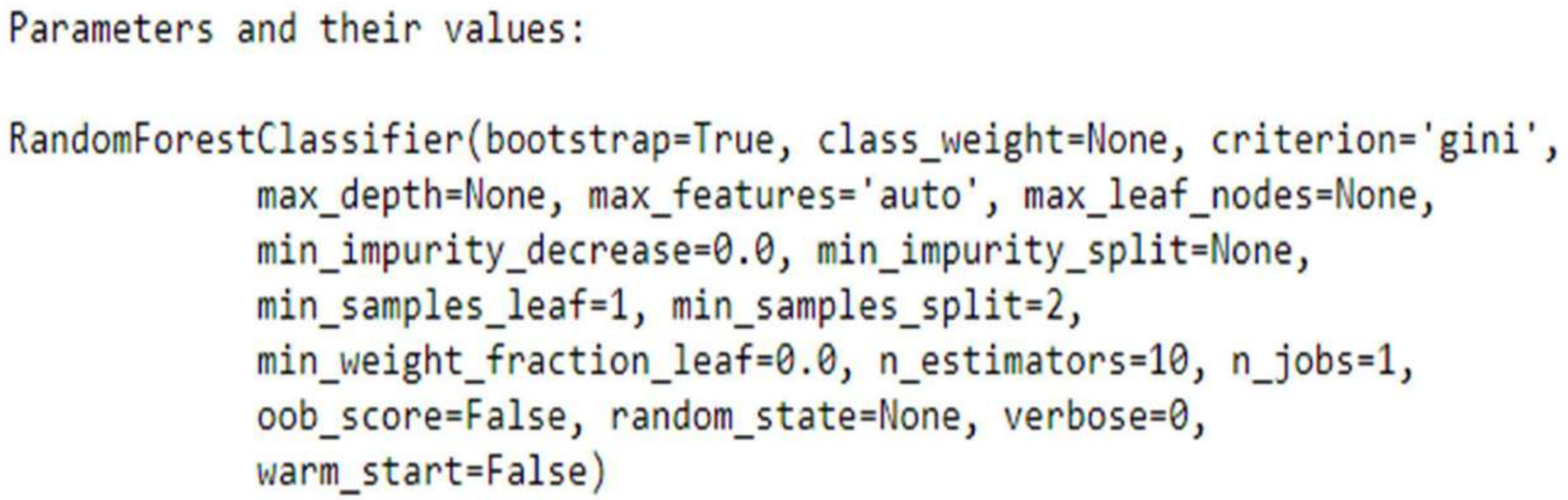
```
Test Data after Label Encoding:

index  height  weight  hair  beard  scarf  gender
0       179.1     185     1      1      0       1
1       160.5     130     2      0      0       0
2       177.8     160     0      0      0       1
3       161.1     100     3      0      0       0
```

Step 5: Train ML Algorithms using Train Data



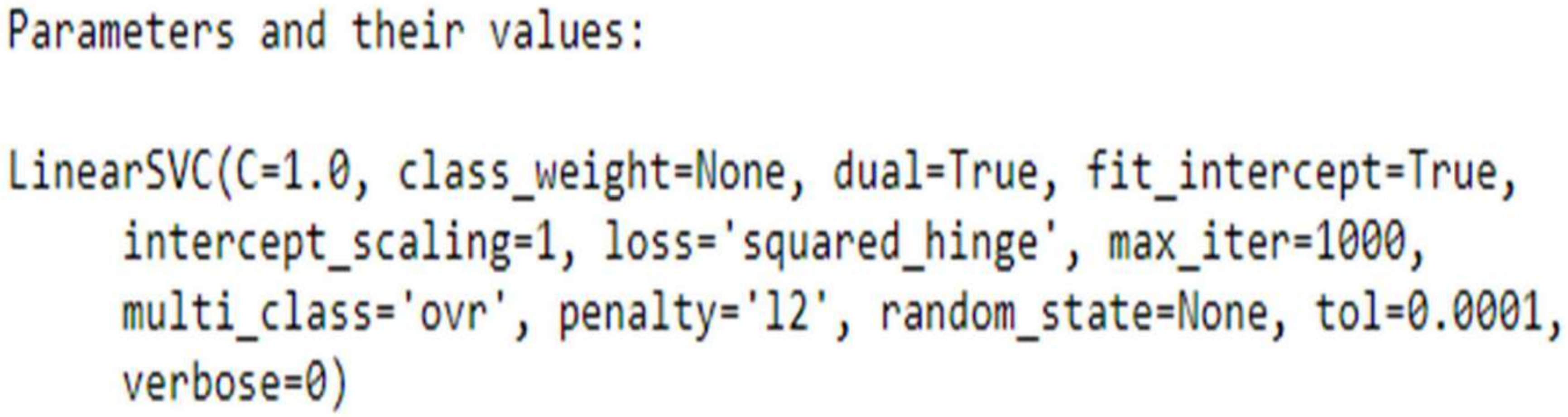
```
Parameters and their values:

LogisticRegression(C=1.0, class_weight=None, dual=False, fit_intercept=True,
          intercept_scaling=1, max_iter=100, multi_class='ovr', n_jobs=1,
          penalty='l2', random_state=None, solver='liblinear', tol=0.0001,
          verbose=0, warm_start=False)
```

Step 5: Train ML Algorithms using Train Data



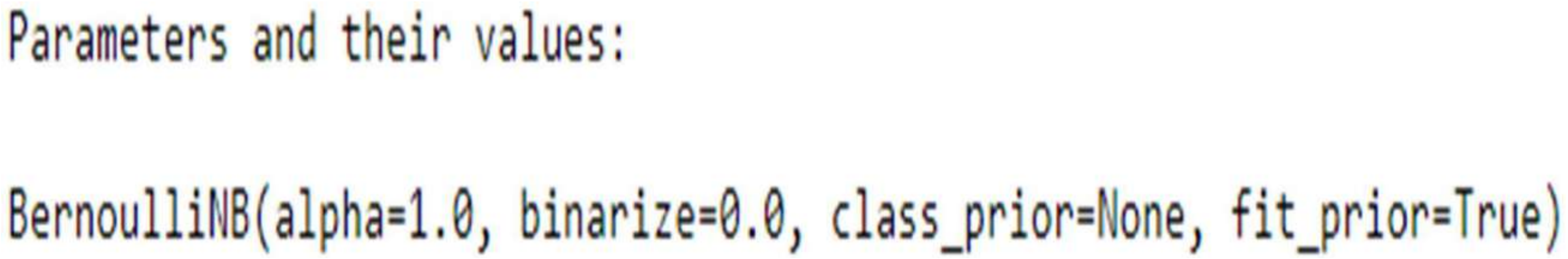
```
Parameters and their values:

RandomForestClassifier(bootstrap=True, class_weight=None, criterion='gini',
            max_depth=None, max_features='auto', max_leaf_nodes=None,
            min_impurity_decrease=0.0, min_impurity_split=None,
            min_samples_leaf=1, min_samples_split=2,
            min_weight_fraction_leaf=0.0, n_estimators=10, n_jobs=1,
            oob_score=False, random_state=None, verbose=0,
            warm_start=False)
```

Step 5: Train ML Algorithms using Train Data



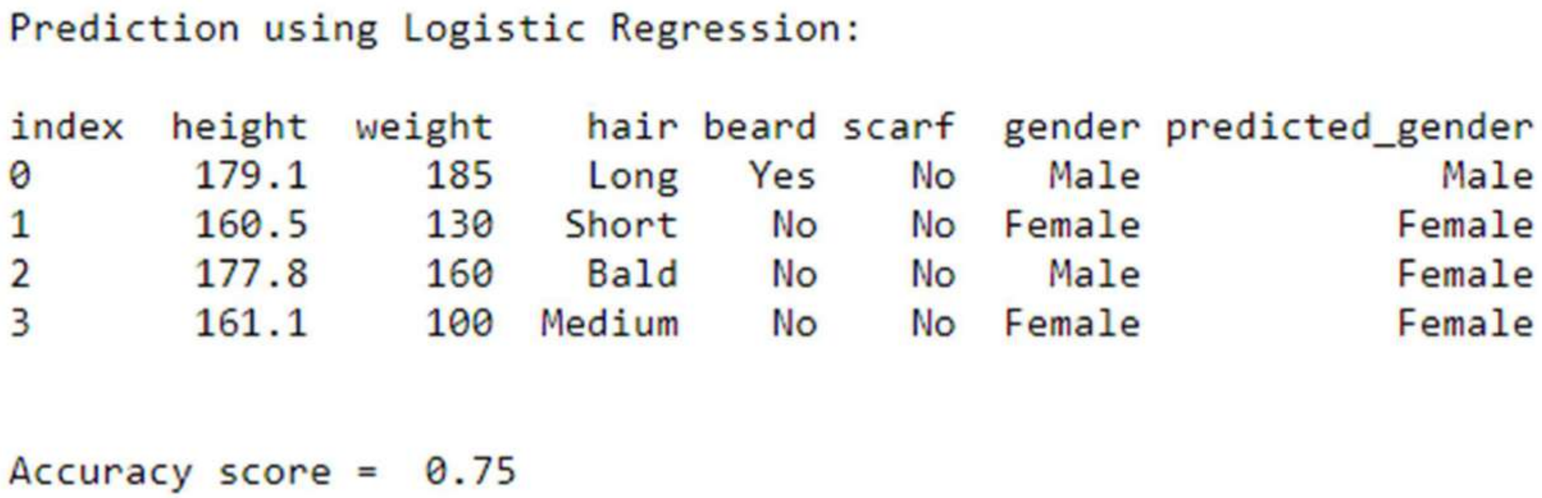
```
Parameters and their values:

LinearSVC(C=1.0, class_weight=None, dual=True, fit_intercept=True,
     intercept_scaling=1, loss='squared_hinge', max_iter=1000,
     multi_class='ovr', penalty='l2', random_state=None, tol=0.0001,
     verbose=0)
```

Step 5: Train ML Algorithms using Train Data



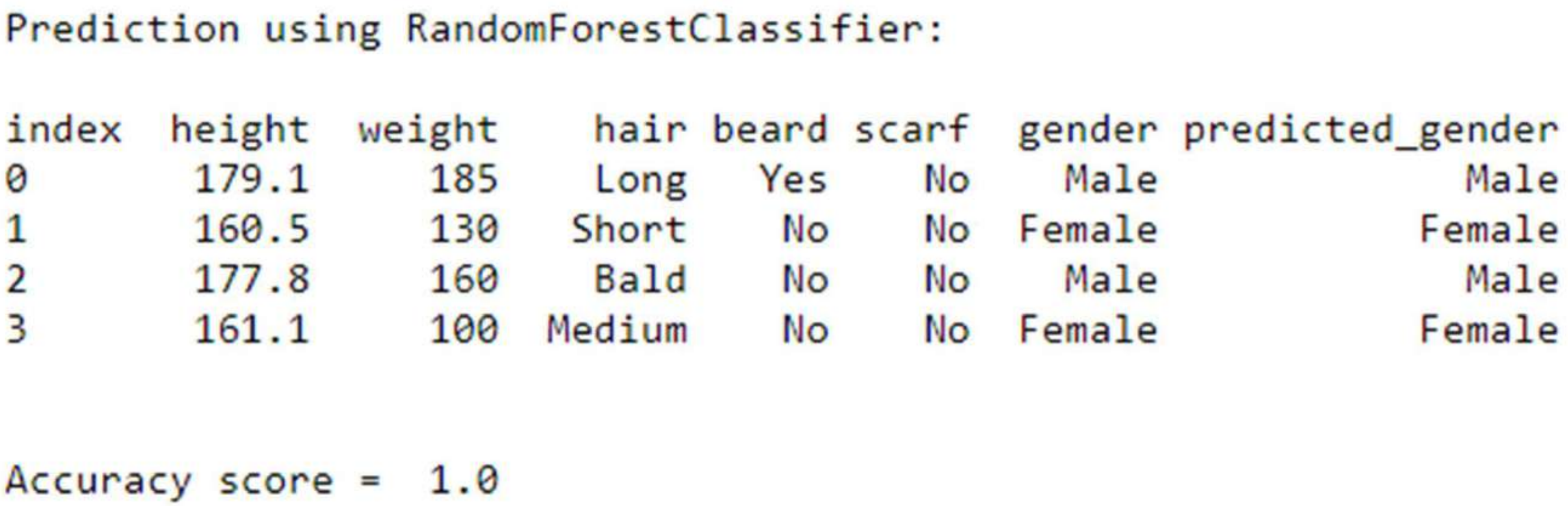
```
Parameters and their values:

BernoulliNB(alpha=1.0, binarize=0.0, class_prior=None, fit_prior=True)
```

Step 6: Evaluate ML Algorithms using Test Data



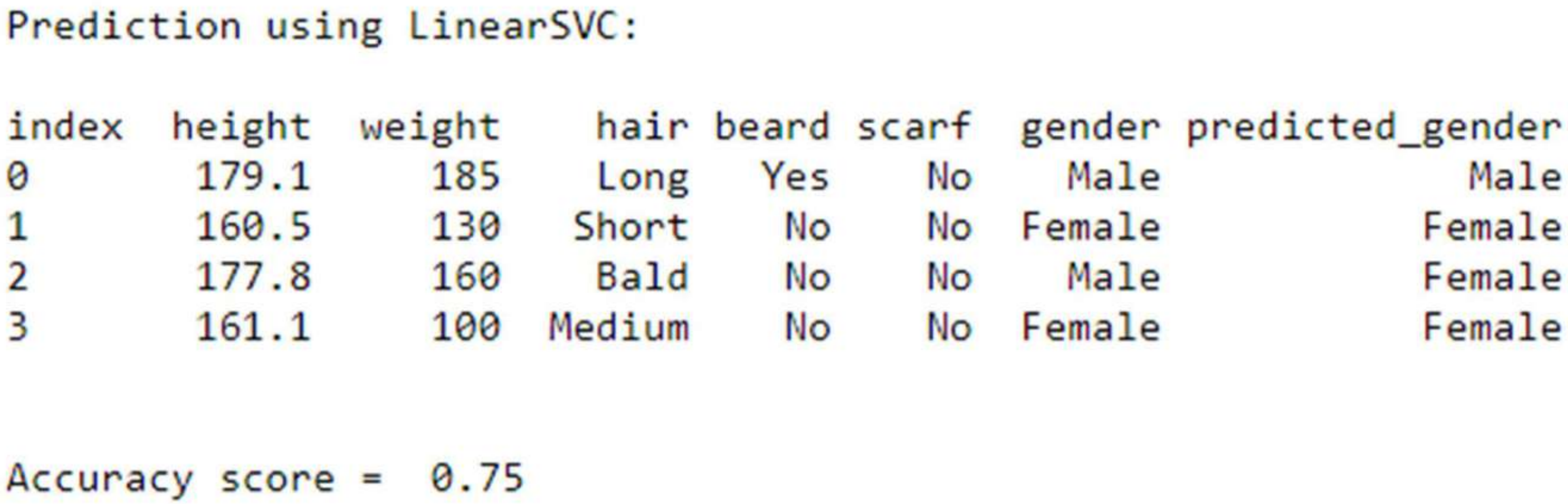
```
Prediction using Logistic Regression:

index  height  weight    hair beard scarf  gender predicted_gender
0       179.1     185    Long   Yes    No    Male             Male
1       160.5     130   Short    No    No  Female           Female
2       177.8     160    Bald    No    No    Male           Female
3       161.1     100  Medium    No    No  Female           Female


Accuracy score =  0.75
```

Step 6: Evaluate ML Algorithms using Test Data



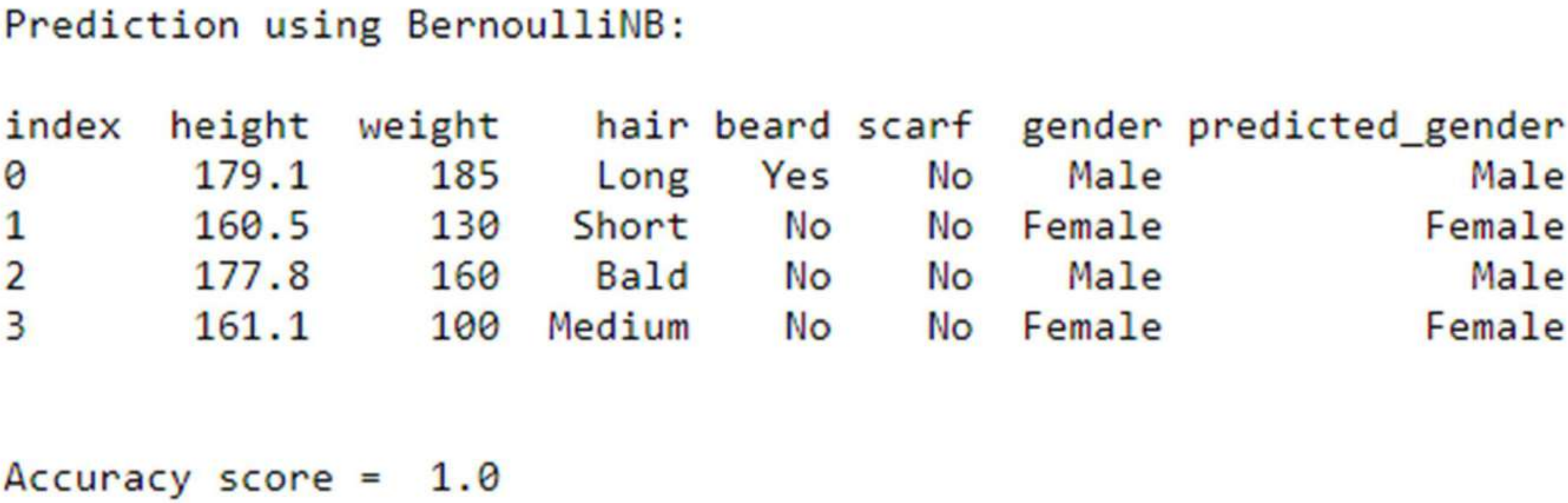
```
Prediction using RandomForestClassifier:

index  height  weight    hair beard scarf  gender predicted_gender
0       179.1     185    Long   Yes    No    Male             Male
1       160.5     130   Short    No    No  Female           Female
2       177.8     160    Bald    No    No    Male             Male
3       161.1     100  Medium    No    No  Female           Female


Accuracy score =  1.0
```

Step 6: Evaluate ML Algorithms using Test Data



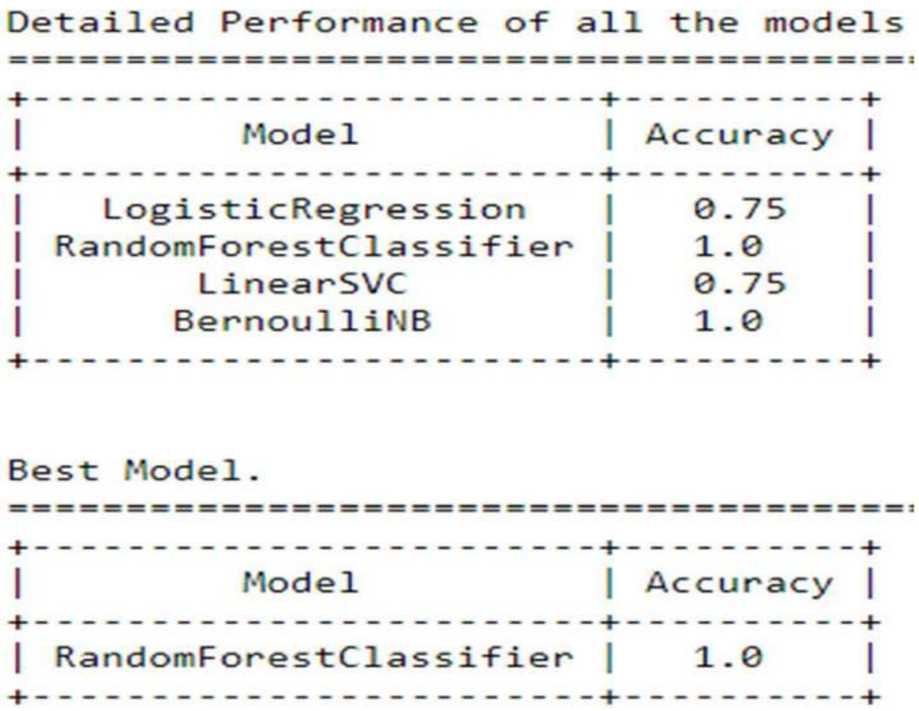
```
Prediction using LinearSVC:

index   height   weight    hair beard scarf  gender predicted_gender
0        179.1      185    Long   Yes    No    Male             Male
1        160.5      130   Short    No    No  Female           Female
2        177.8      160    Bald    No    No    Male           Female
3        161.1      100  Medium    No    No  Female           Female


Accuracy score =  0.75
```

Step 6: Evaluate ML Algorithms using Test Data



```
Prediction using BernoulliNB:

index  height  weight    hair beard scarf  gender predicted_gender
0       179.1     185    Long   Yes    No    Male             Male
1       160.5     130   Short    No    No  Female           Female
2       177.8     160    Bald    No    No    Male             Male
3       161.1     100  Medium    No    No  Female           Female


Accuracy score =  1.0
```

Step 7: Selection of Best Model

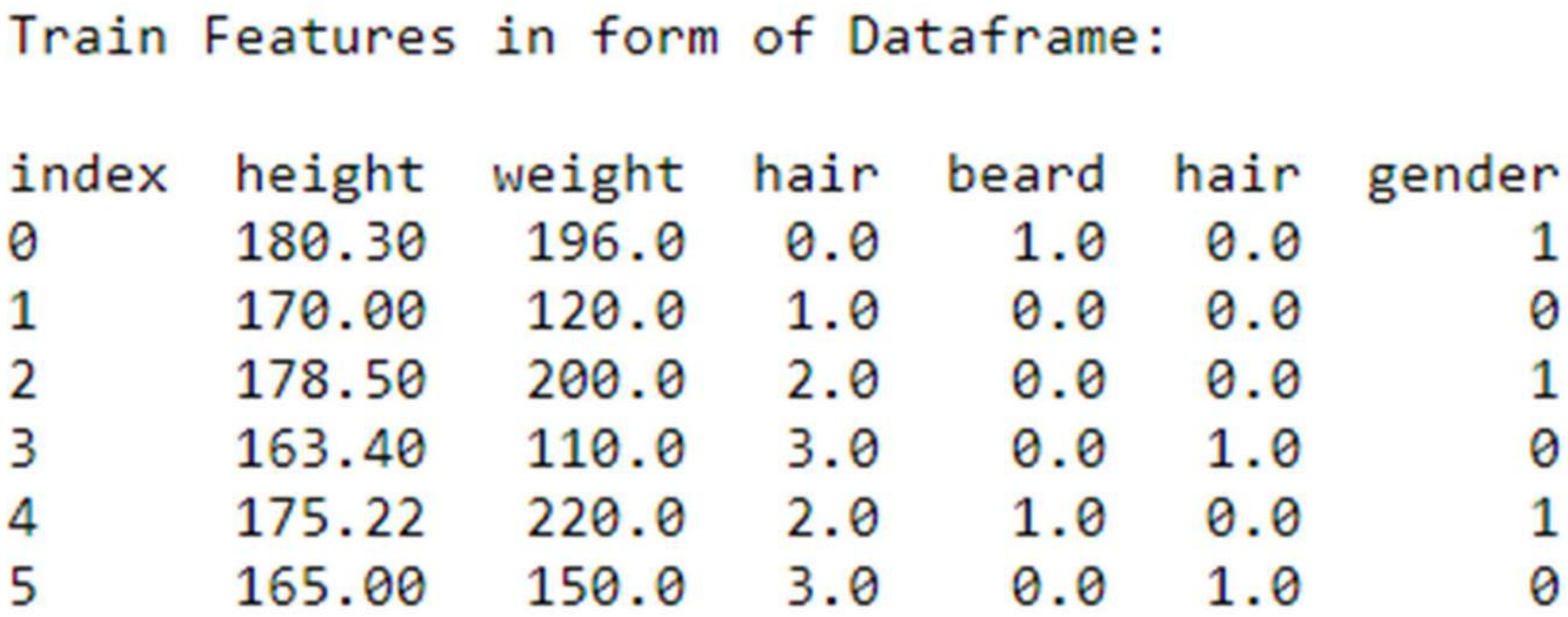


```
Detailed Performance of all the models
======================================
+------------------------+----------+
|         Model          | Accuracy |
+------------------------+----------+
|   LogisticRegression   |   0.75   |
| RandomForestClassifier |   1.0    |
|       LinearSVC        |   0.75   |
|      BernoulliNB       |   1.0    |
+------------------------+----------+


Best Model.
======================================
+------------------------+----------+
|         Model          | Accuracy |
+------------------------+----------+
| RandomForestClassifier |   1.0    |
+------------------------+----------+
```

Step 8: Application Phase

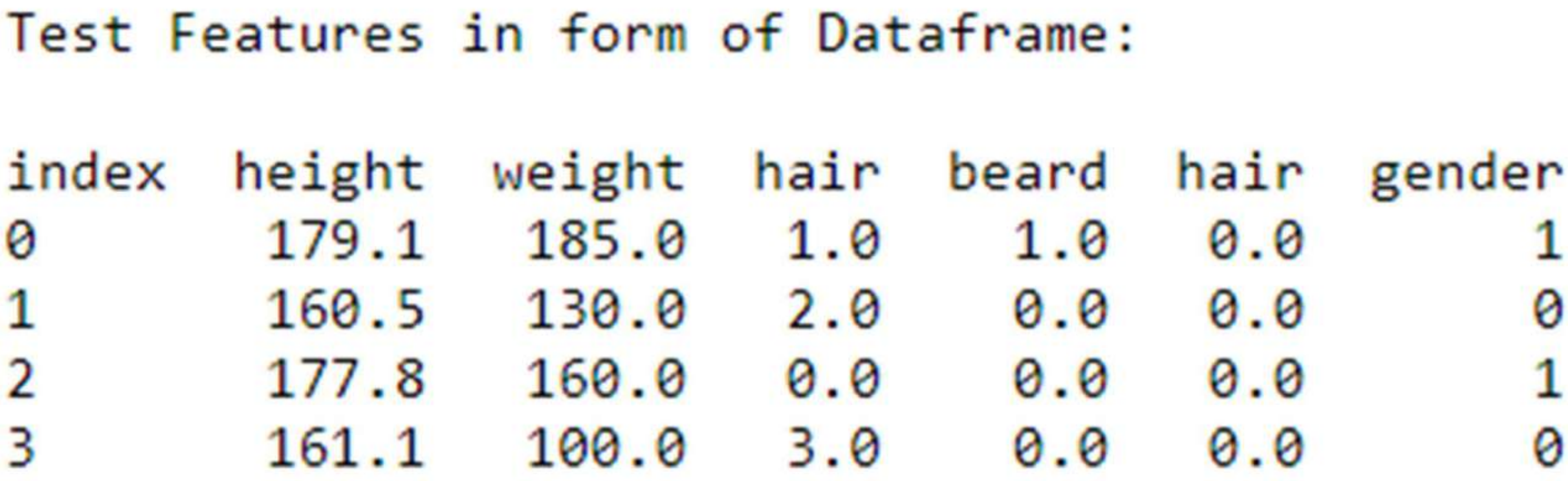
PHASE 3: APPLICATION PHASE

Step 8.1: Combine Data (Train+Test)



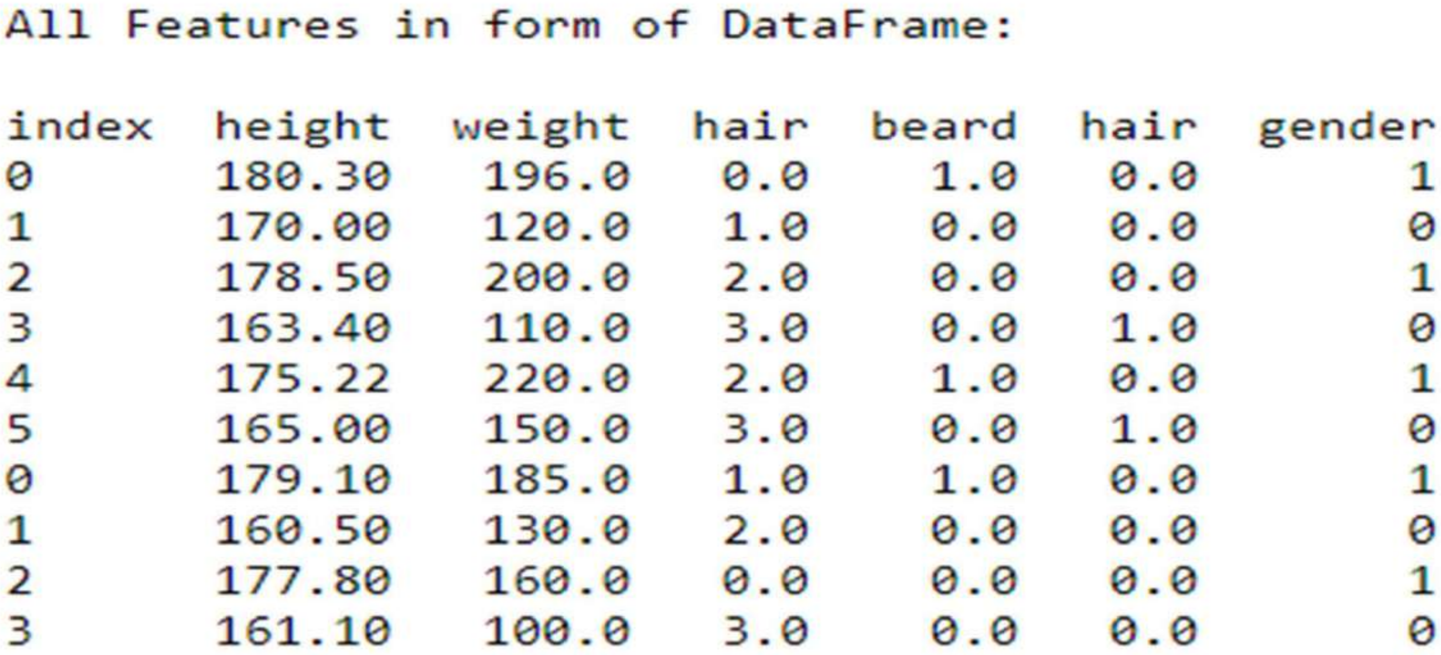
```
Train Features in form of Dataframe:

index  height  weight  hair  beard  hair  gender
0      180.30   196.0   0.0    1.0   0.0       1
1      170.00   120.0   1.0    0.0   0.0       0
2      178.50   200.0   2.0    0.0   0.0       1
3      163.40   110.0   3.0    0.0   1.0       0
4      175.22   220.0   2.0    1.0   0.0       1
5      165.00   150.0   3.0    0.0   1.0       0
```

Step 8.1: Combine Data (Train+Test)



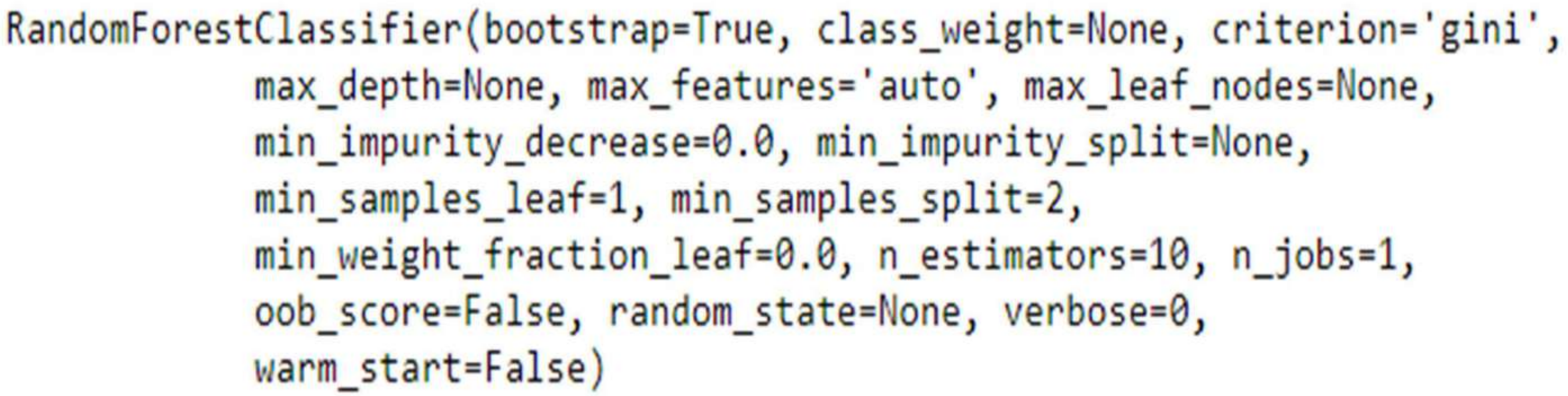
```
Test Features in form of Dataframe:

index   height   weight   hair   beard   hair   gender
0        179.1    185.0    1.0     1.0    0.0        1
1        160.5    130.0    2.0     0.0    0.0        0
2        177.8    160.0    0.0     0.0    0.0        1
3        161.1    100.0    3.0     0.0    0.0        0
```

Step 8.1: Combine Data (Train+Test)



```
All Features in form of DataFrame:

index    height   weight   hair   beard   hair   gender
0        180.30    196.0    0.0     1.0    0.0        1
1        170.00    120.0    1.0     0.0    0.0        0
2        178.50    200.0    2.0     0.0    0.0        1
3        163.40    110.0    3.0     0.0    1.0        0
4        175.22    220.0    2.0     1.0    0.0        1
5        165.00    150.0    3.0     0.0    1.0        0
0        179.10    185.0    1.0     1.0    0.0        1
1        160.50    130.0    2.0     0.0    0.0        0
2        177.80    160.0    0.0     0.0    0.0        1
3        161.10    100.0    3.0     0.0    0.0        0
```

Step 8.2: Train Best Model on All Data



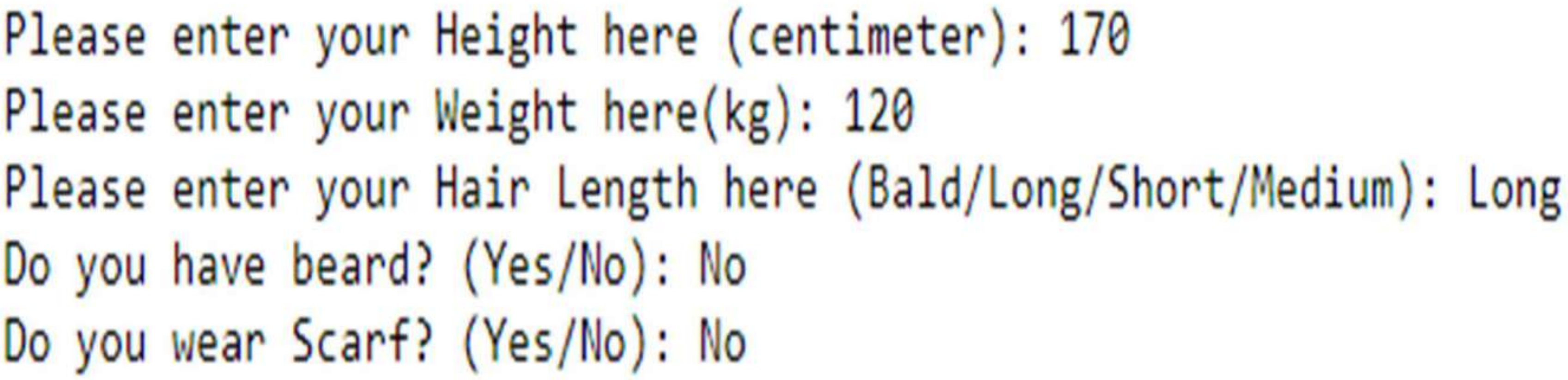
```
RandomForestClassifier(bootstrap=True, class_weight=None, criterion='gini',
            max_depth=None, max_features='auto', max_leaf_nodes=None,
            min_impurity_decrease=0.0, min_impurity_split=None,
            min_samples_leaf=1, min_samples_split=2,
            min_weight_fraction_leaf=0.0, n_estimators=10, n_jobs=1,
            oob_score=False, random_state=None, verbose=0,
            warm_start=False)
```

Step 9: Make Predictions on Unseen/New Data

Making Predictions on Unseen/New Data

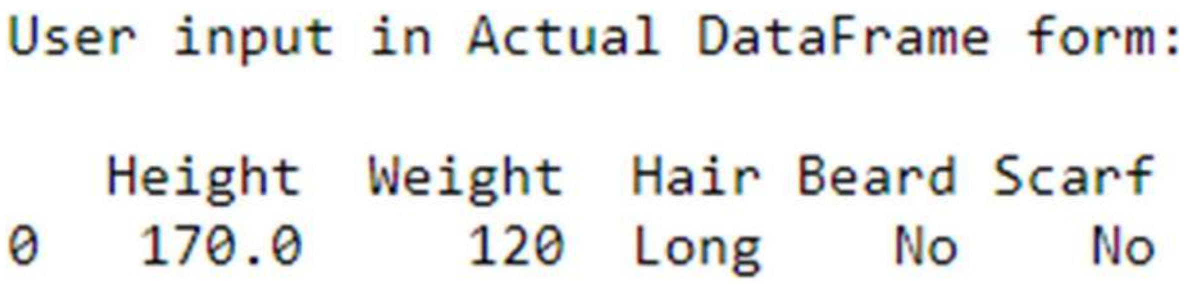
Step 9.1: Load the Trained Model (saved in Step 8.3)

Step 9.2: Take Input from User



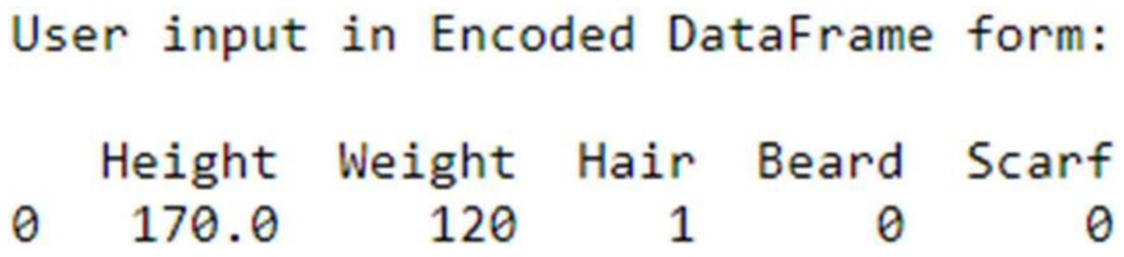
```
Please enter your Height here (centimeter): 170
Please enter your Weight here(kg): 120
Please enter your Hair Length here (Bald/Long/Short/Medium): Long
Do you have beard? (Yes/No): No
Do you wear Scarf? (Yes/No): No
```

Step 9.3: Convert User Input into Feature Vector (Same ss Feature Vector of Trained Model)



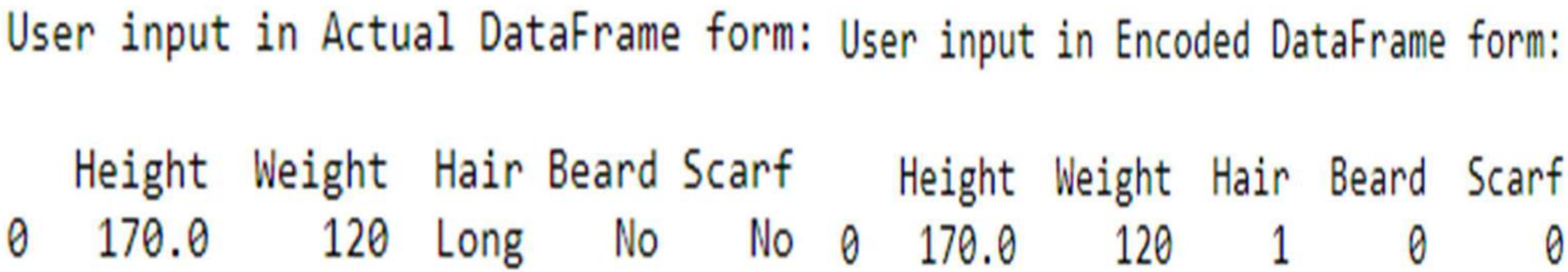
```
User input in Actual DataFrame form:

   Height  Weight  Hair Beard Scarf
0   170.0     120  Long    No    No
```

Step 9.3: Convert User Input into Feature Vector (Same ss Feature Vector of Trained Model)



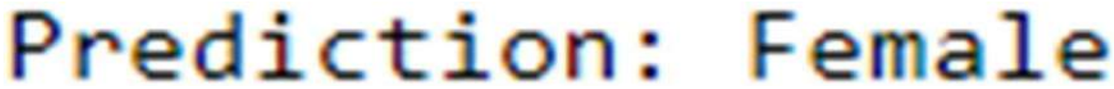
```
User input in Encoded DataFrame form:

   Height  Weight  Hair  Beard  Scarf
0   170.0     120     1      0      0
```

Step 9.3: Convert User Input into Feature Vector (Same ss Feature Vector of Trained Model)



```
User input in Actual DataFrame form: User input in Encoded DataFrame form:

   Height  Weight  Hair Beard Scarf      Height  Weight  Hair  Beard  Scarf
0   170.0     120  Long    No    No   0   170.0     120     1      0      0
```

Step 9.4: Apply Trained Model on Feature Vector of Unseen Data and Output Prediction to User



```
Prediction: Female
```