horizontal line

AML Hackathon Report

Team #13 - EPOCALYPSE

Friday, 21.12.2018

**─**

# Team members

Parashara R 01FB15ECS**202**

Priyank Bhandia 01FB15ECS**216**

Rakesh K 01FB15ECS**230**

# Problem#1 (Selfie dataset analysis)

## Problem Statement/Objectives

We were given a selfie dataset with 46500 images along with 36 attributes including various factors like the age, gender, race, hair color etc and we were tasked with the following:

1. Visualizing the input dataset given and correcting the skew in the dataset if any.
2. Building an architecture to predict the attributes for age(baby, child, teenager, youth,middle\_age, senior) and gender(is\_female) along with the popularity index for a given image using two different approaches
3. Testing the models against images available in the social media sites like Instagram.

## Environment Used

We used **Google Colab** which was useful for sharing code online. In that, we used the Nvidia K-80 GPU for training the dataset.

## 

## Workflow/Steps

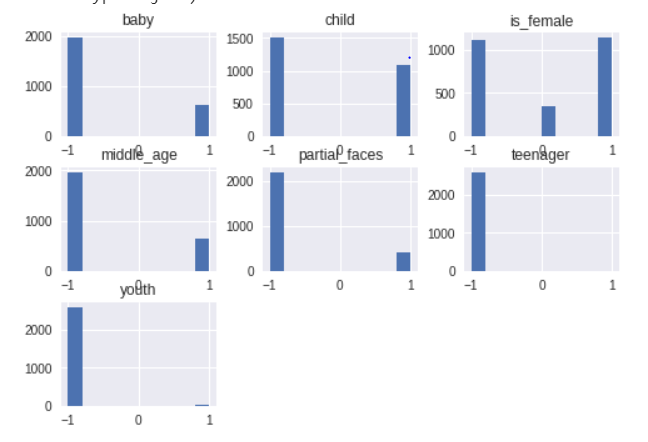
# Checkpoint # 1

### Data Visualisation

* We used visualized the dataset using Pandas, and we plotted histograms for all the attributes but noticed an imbalance in many of the attributes.



* We used keras’s “image-data-generator” data augmentation module for correcting the skew in many of the attributes.After which the data distribution looked like the following.(Note: the graph below is only a sample of 2k images from the total sample of augmented images)



As is clear from the above plots, the dataset is heavily skewed for most of the attributes.

The data imbalance is seen in:

1. Gender: more females than males
2. Partial Faces: there are more complete faces than there are partial faces

The Data augmentation techniques used were:

We used keras’s “image-data-generator” data augmentation module for correcting the skew in many of the attributes. Thus we were able to skew and rotate images.

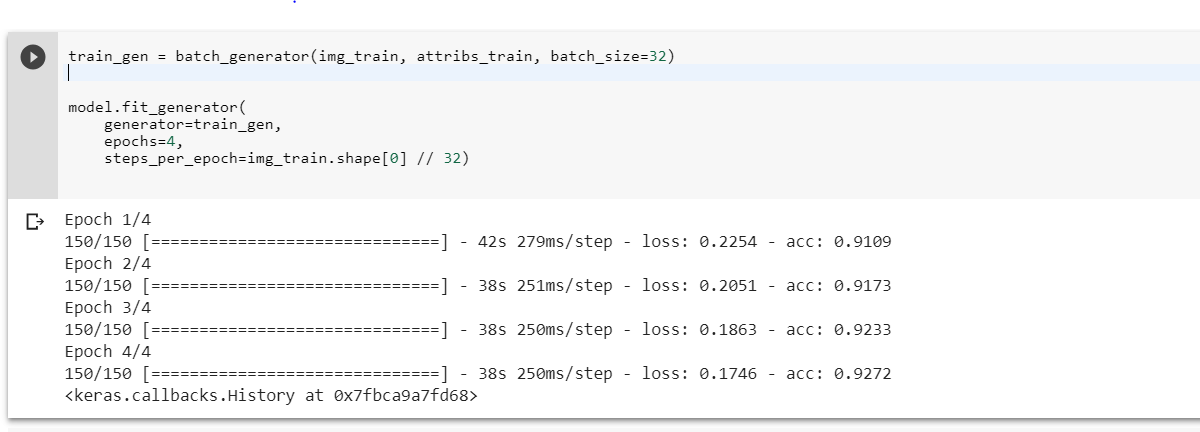
# Checkpoint #2

### Architecture

The General approach

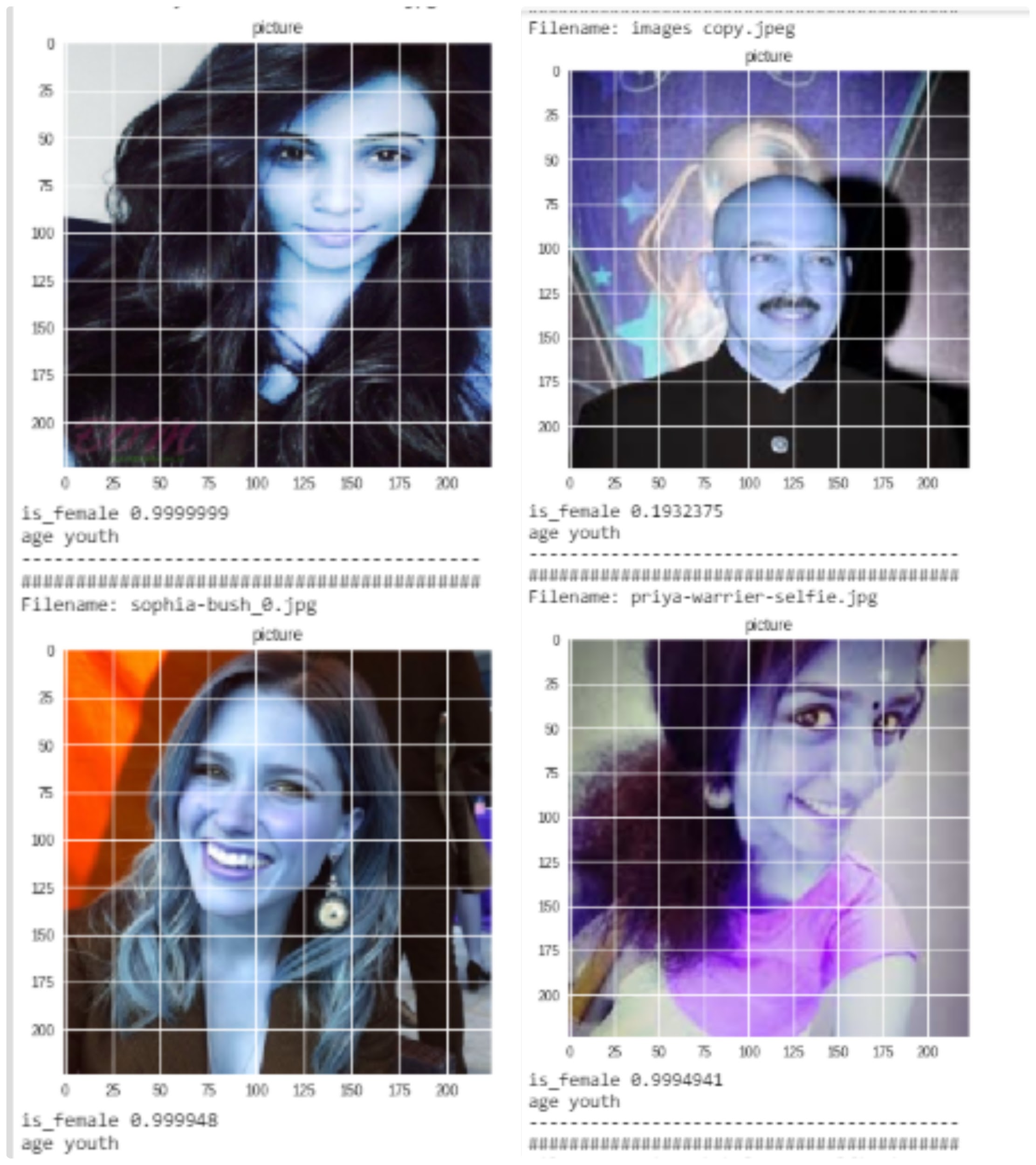
1. Predicting only the attributes of age and gender

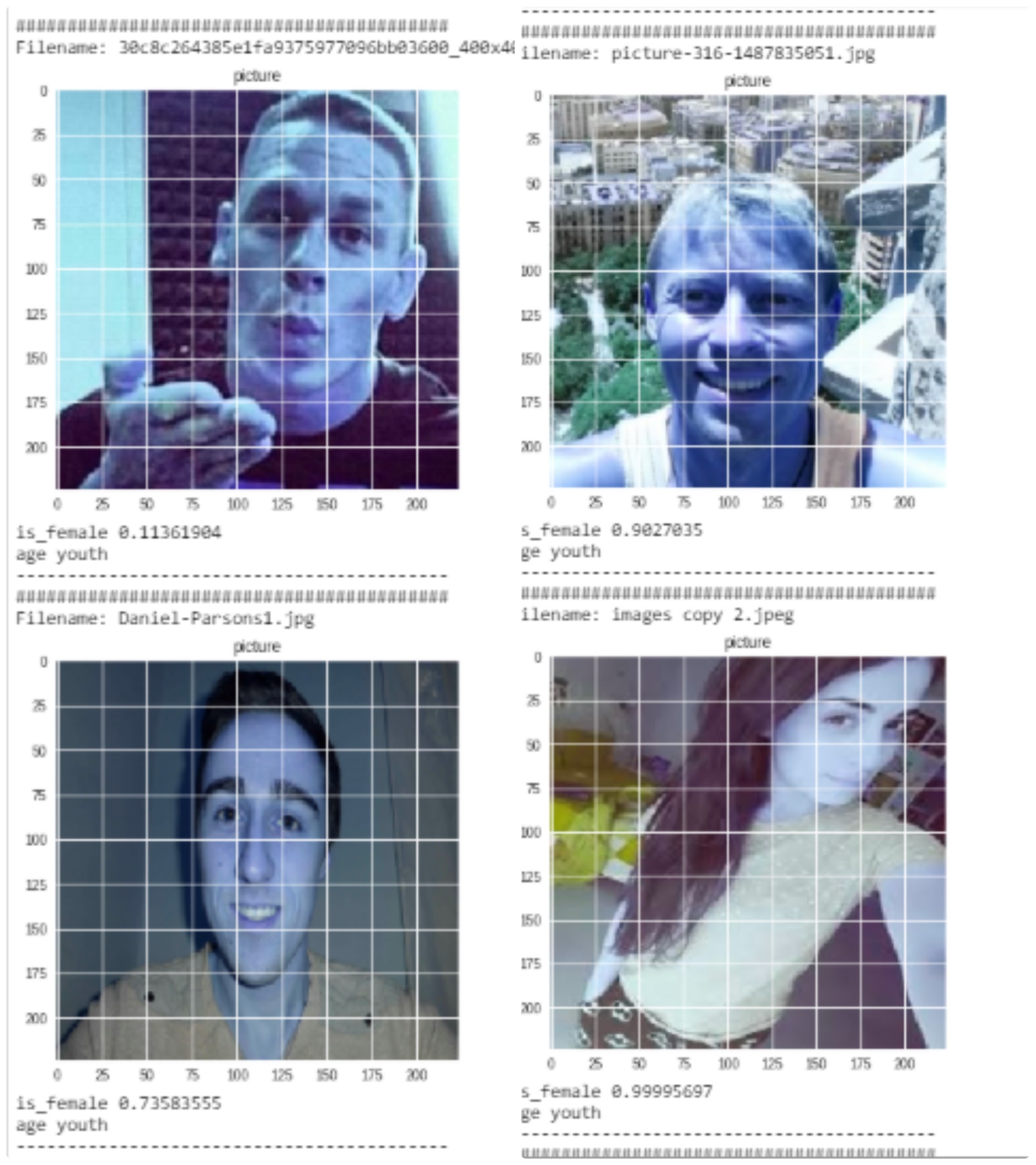
* Our first approach was to just predict the 7 attributes in a single output head using a categorical cross entropy loss function with a sigmoidal output.
  + Although the reported loss and accuracy were good it was not performing to the standards required on the external dataset.
  + We changed the loss function from categorical to binary cross-entropy as we realized this model is multilabel classification problem and not that of multiclass classification. The accuracy after performing this step still only came to about 30 %.
  + So we decided to make the age attribute, one hot vector of size 6 instead of using the way the dataset had represented it by replacing the -1s with 0s and the accuracy suddenly shot up to 70% on the dataset.
* The second approach involved using 2 output heads for the age and gender attribute.
  + Initially, we struggled with giving both the output heads different loss functions but once we realized how to we used the “categorical cross entropy” loss function for both the output heads.
  + We also changed the age from a Dense(1) to Dense(2) as the activation we were using was a sigmoid.
  + This approach increased the training accuracy to well over 88% and shot the loss down for each attribute to below 0.1.



**This image shows decreasing loss and increasing accuracy with successive epochs.**

# Checkpoint # 3





**This image shows the reported values for our external validation dataset.**

# 

# Checkpoint # 4

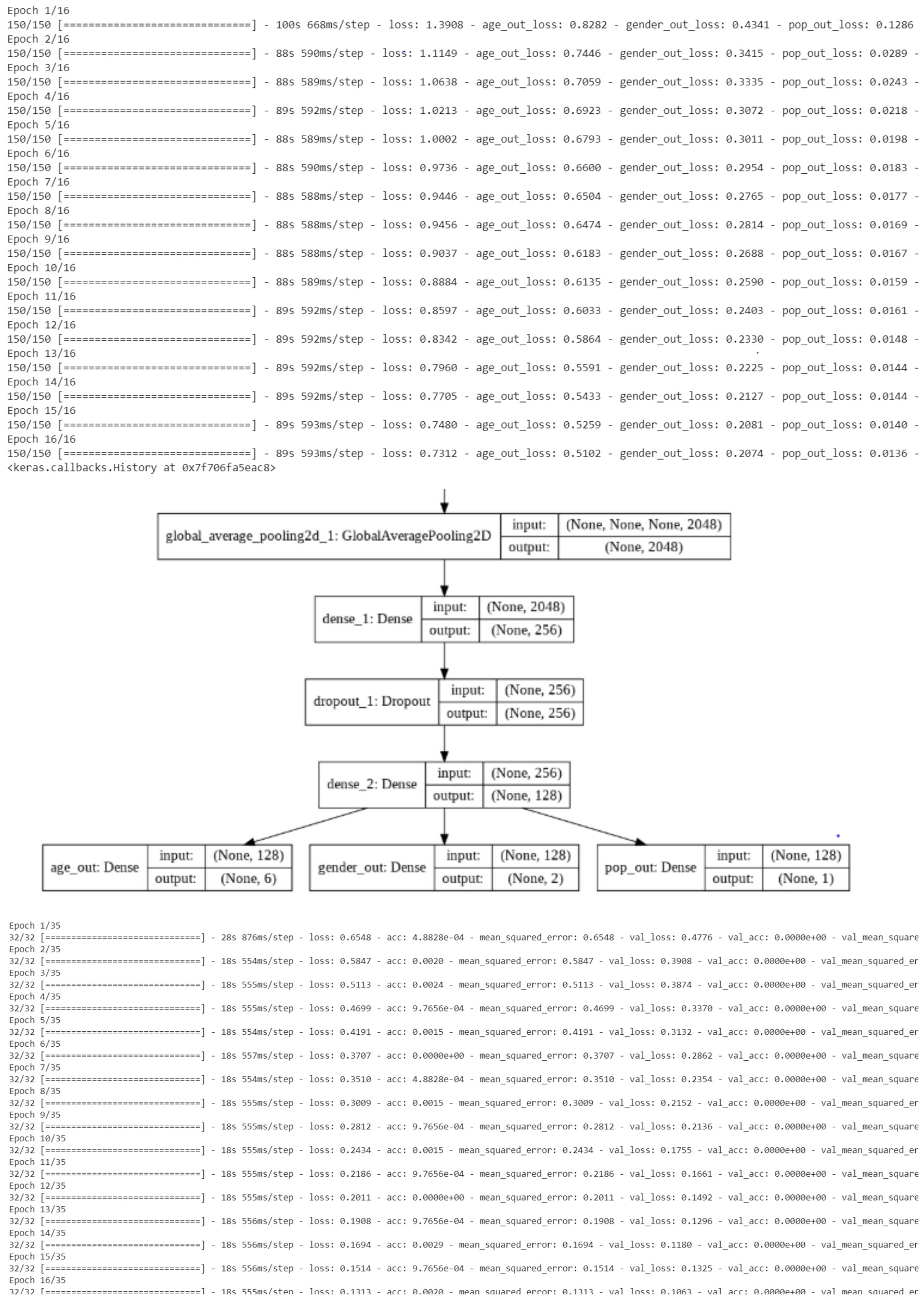
1. Predicting the popularity index using multihead approach

For predicting the popularity index we added another output head to the model described in the above step.Which had a linear activation function and is a layer of Dense(1).

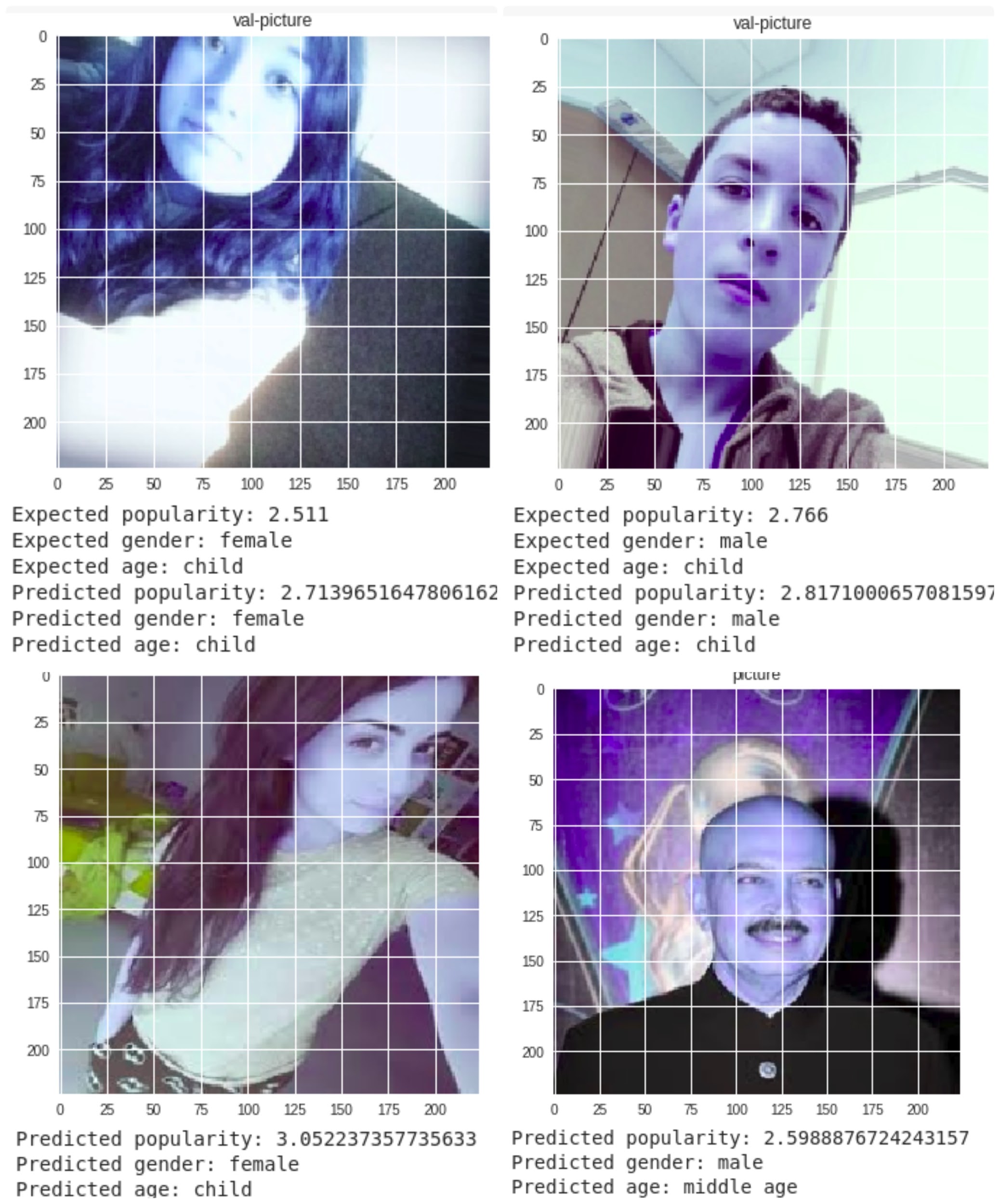
1. Predicting the popularity index using 2 step approach:
   1. The model layers from the attribute predicting model built for Checkpoint #2 are set as non-trainable.
   2. The output heads are then fed to fully connected layers with a regression head to predict the popularity
   3. This is trained on the dataset and results are presented in the following screenshots.

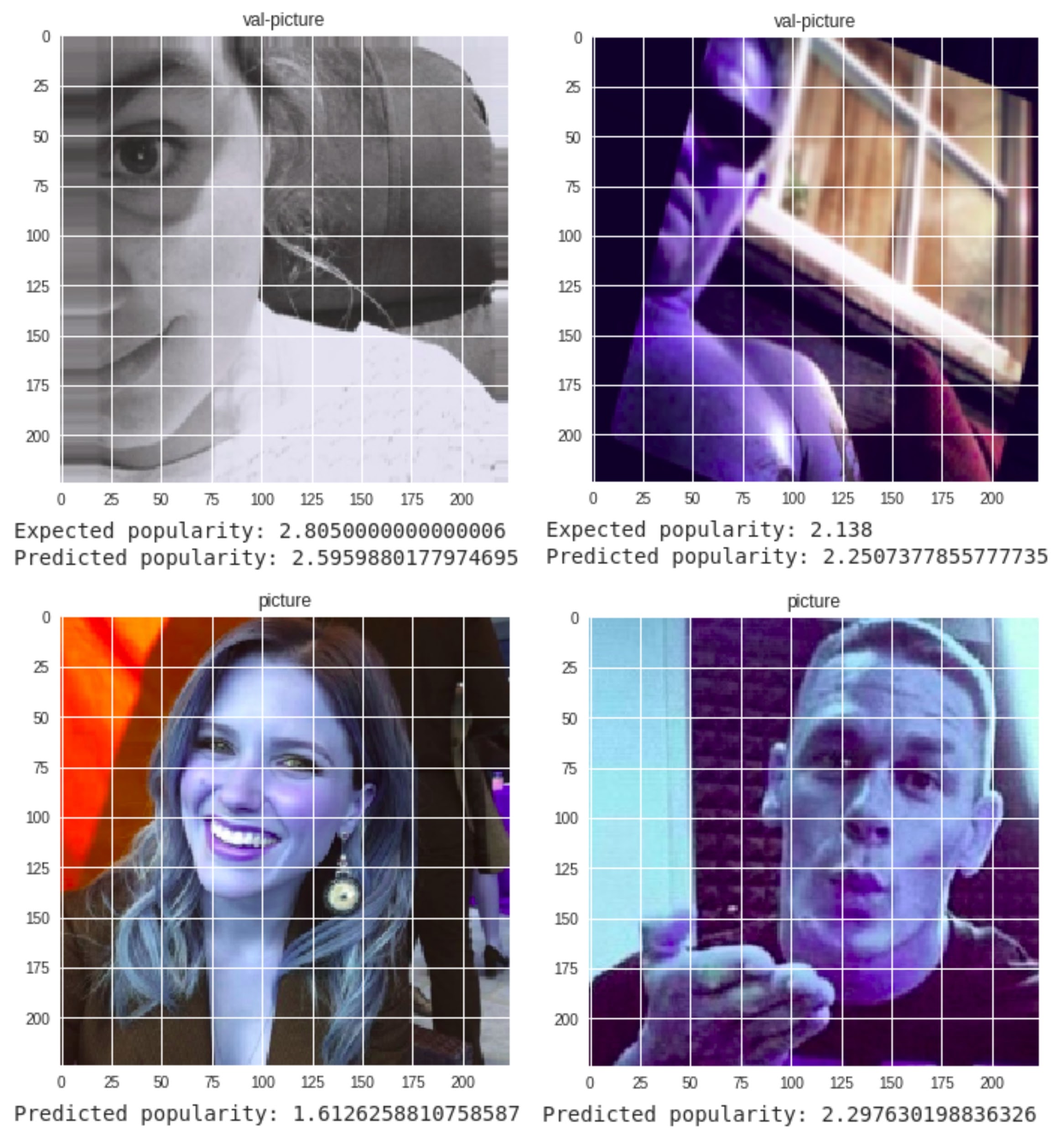
### 

### 



### Results





# Problem#2 (Classifying the selfie as good or bad)

The approach taken in solving this problem is:

1. We used pandas to find the buckets of popularity. We found that the cutoff values were:

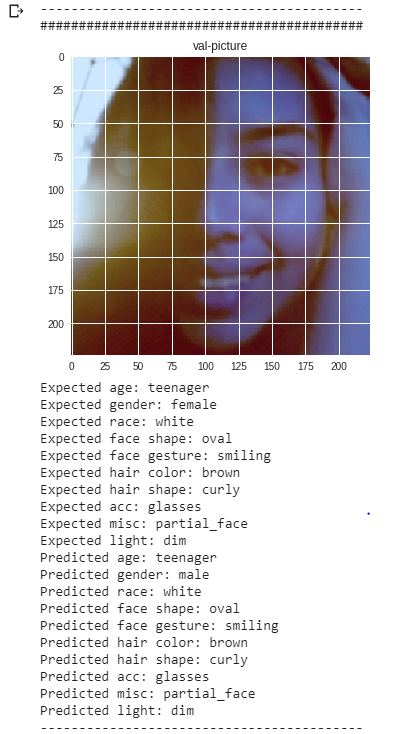
**For the normalised popularity scores,**

**Bad: 0.00 - 0.487**

**Medium- 0.487- 0.595**

**Good- 0.595-1.00**

1. We rebuilt the classifier from the first problem to now have 10 output heads for all the 36 attributes
2. Then we added more dense layers on top of this model to predict the popularity classes.
3. The results are reported below:



1. The results are shown below:

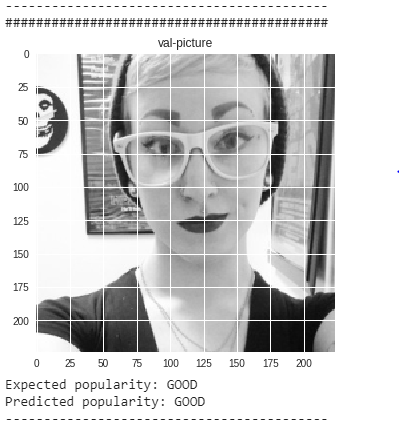
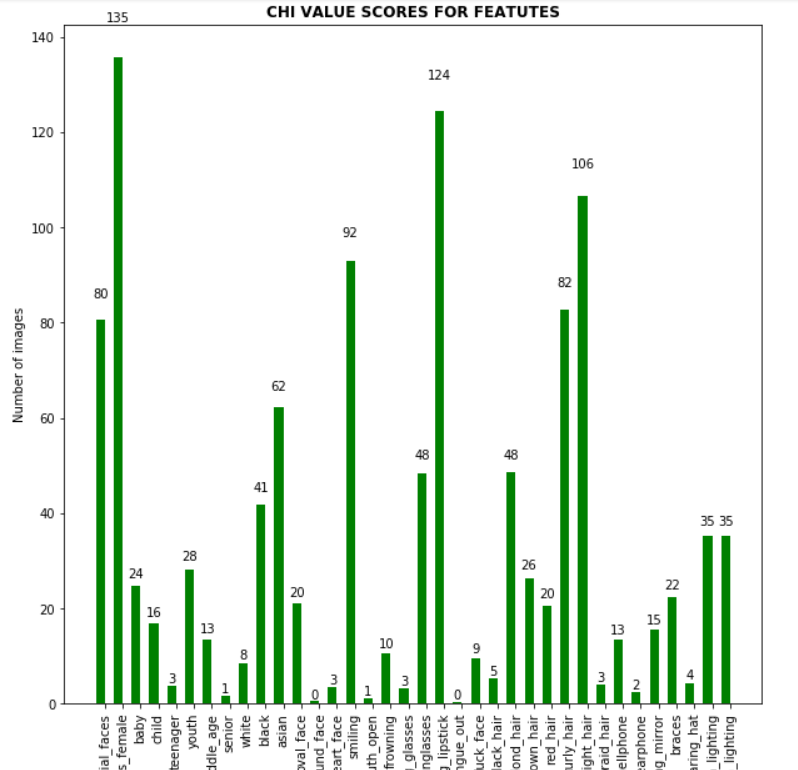


Fig: Prediction from the dataset



Fig: the top 4 attributes for predicting the popularity index

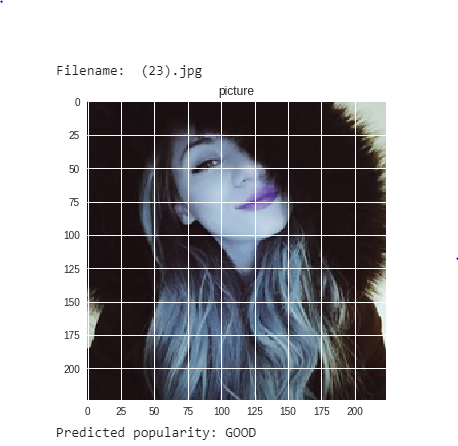


Fig: Prediction from an external image