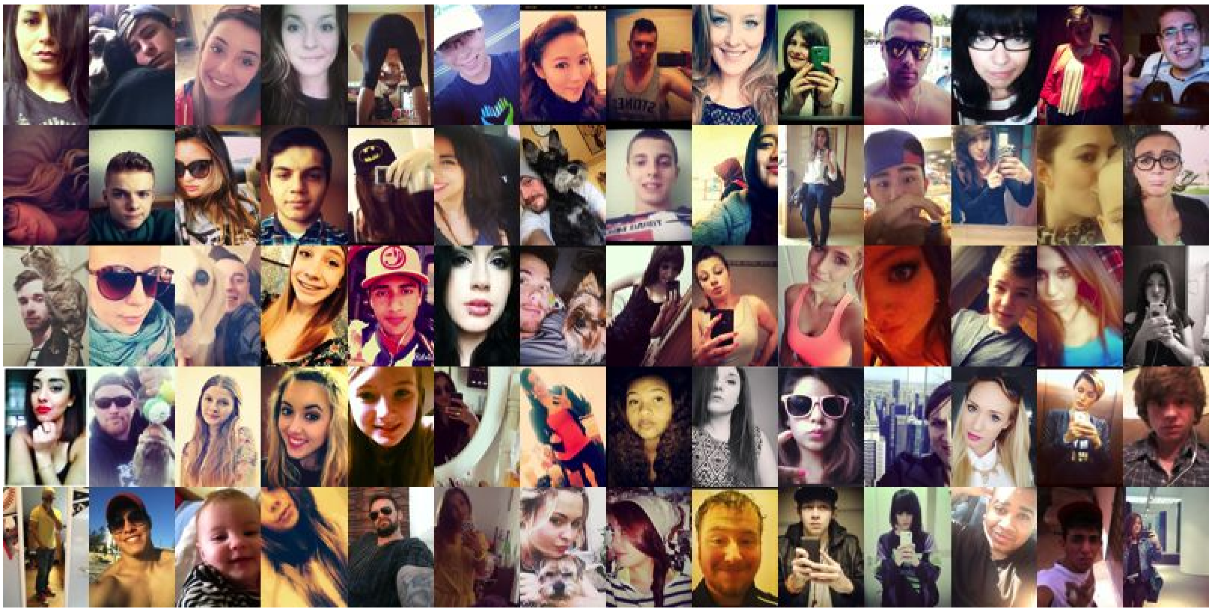
AML Hackathon Report



Team members :

Sanketh Rangreji - 01FB15ECS267

Shahid Ikram - 01FB15ECS274

Nishant Sondhi - 01FB15ECS298

Sumanth S Rao - 01FB15ECS314

TASK 1

TASK 2

Task 1

DATASET :

Selfie dataset contains 46,836 selfie images annotated with 36 different attributes divided into several categories as follows. Gender: is female. Age: baby, child, teenager, youth, middle age, senior. Race: white, black, asian. Face shape: oval, round, heart. Facial gestures: smiling, frowning, mouth open, tongue out, duck face. Hair color: black, blond, brown, red. Hair shape: curly, straight, braid. Accessories: glasses, sunglasses, lipstick, hat, earphone. Misc.: showing cellphone, using mirror, having braces, partial face. Lighting condition: harsh, dim.

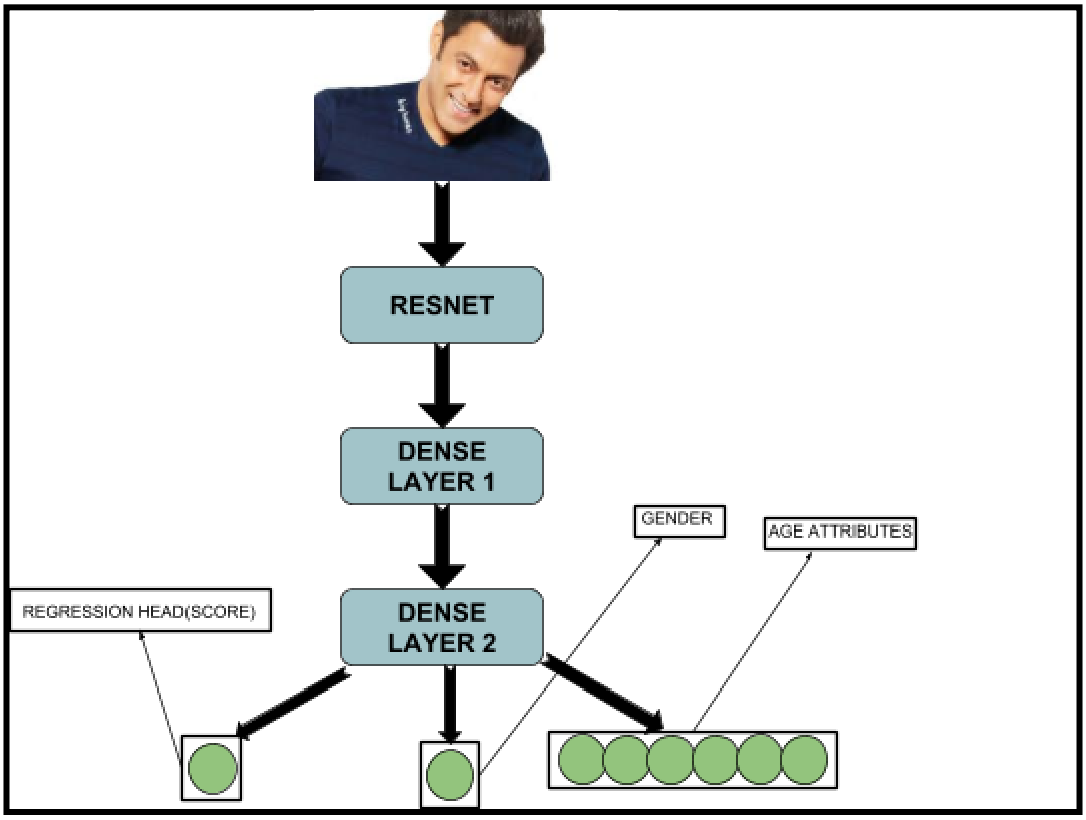
OBJECTIVE :

Experiment with multitask learning by training a regression head for predicting the popularity and a classification head for predicting certain attributes of the image.
Also, experiment with transfer learning by using Resnet50 as the backbone.
Experiment with data augmentation to help offset the class imbalance.
Optionally visualize the layers of the network.

ARCHITECTURE:

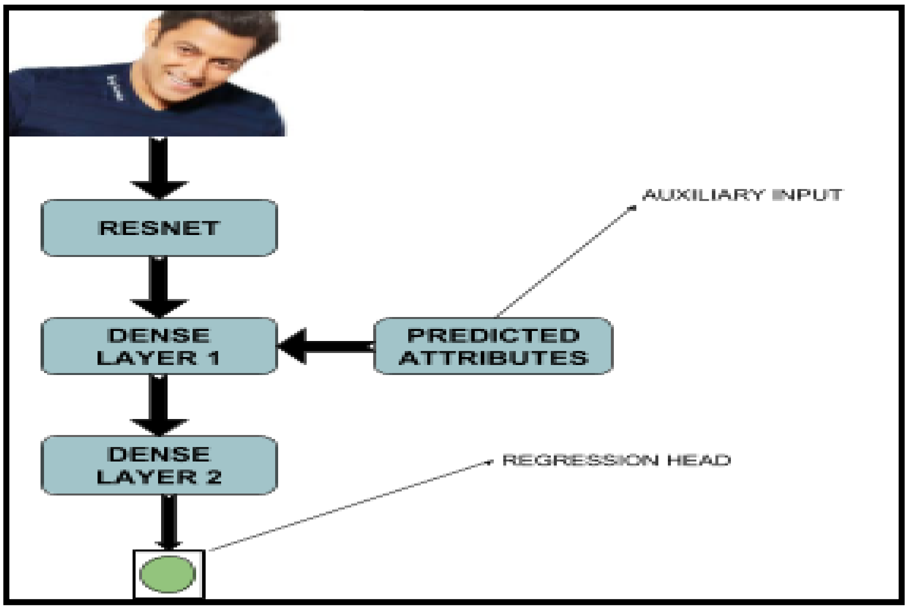
We use the Resnet50 model available as a part of keras as the backbone for our transfer learning model. We replace the fully connected layers with our own custom layers and use the CNN backbone of Resnet50 for feature learning. We incorporate a *fully connected network of two layers each of 1024 hidden units*, followed by one or more output Dense layers(*we have one-three output layers for each architecture variation)*.

We first experimented with training the regression head and the classification head simultaneously, by treating it as a multi-task learning problem. Here *we used two output layers, one Dense(1) layer with linear activation and mse loss, the other with Dense(7), with sigmoid activation and a binary logistic loss function(similar to binary cross entropy).*



We then trained another model that predicted only popularity, however, we *augmented the input* with the predicted attributes obtained from the previously trained model. The second model was an exact replica of the first(same CNN backbone), but for the last few layers(fully connected portion). Here we *added a 7 vector input layer to the fully connected layer, which was fed with the 7 predicted attributes for each training sample.* This has only one output layer, namely the regression head.

We observed much better results for popularity score, but were unable to ascertain if this was a direct result of using the predicted attributes as features or if it was because the model was now focused only on learning one task, instead of trying to learn more than task simultaneously.



We then experimented with using a *softmax for age(because it was mutually exclusive),* we were able to get better results on predicting the age, even though the accuracy was not as good as when using sigmoid.Here we split the classification head into two, one for classifying gender and one for predicting the age. We retained sigmoid activation for the head classifying gender(was Dense(1)) and used a softmax activation for the classification head for age which was Dense(6).

Checkpoint 1

Observations

- The dataset - The dataset consisted of the selfie images and annotations with respect to certain attributes of the face and popularity of the selfie.

- The attributes labelled with dataset contains 46,836 selfie images annotated with 36 different attributes divided into several categories as follows. Gender: is female. Age: baby, child, teenager, youth, middle age, senior. Race: white, black, Asian. Face shape: oval, round, heart. Facial gestures: smiling, frowning, mouth open, tongue out, duck face. Hair color: black, blond, brown, red. Hair shape: curly, straight, braid. Accessories: glasses, sunglasses, lipstick, hat, earphone. Misc.: showing cellphone, using mirror, having braces, partial face. Lighting condition: harsh, dim.

- Nature of dataset

  The dataset is highly skewed with respect distribution across classes, the data is skewed towards negative samples in many classes.

  There is a scarcity of samples in some classes like baby, senior etc.,

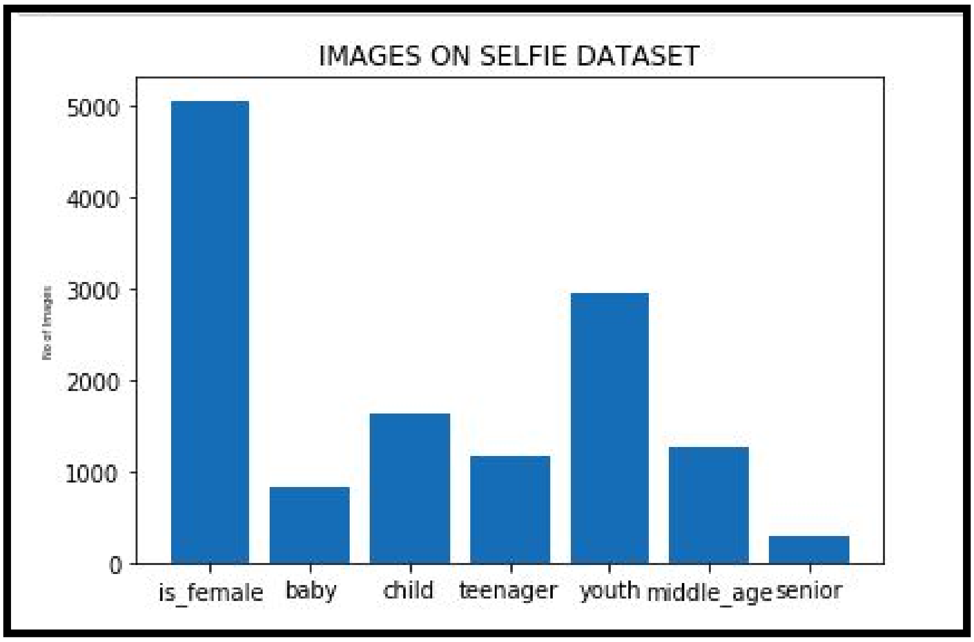
  The data has to be augmented and sampled accordingly correcting the skew.

- Quality of the dataset

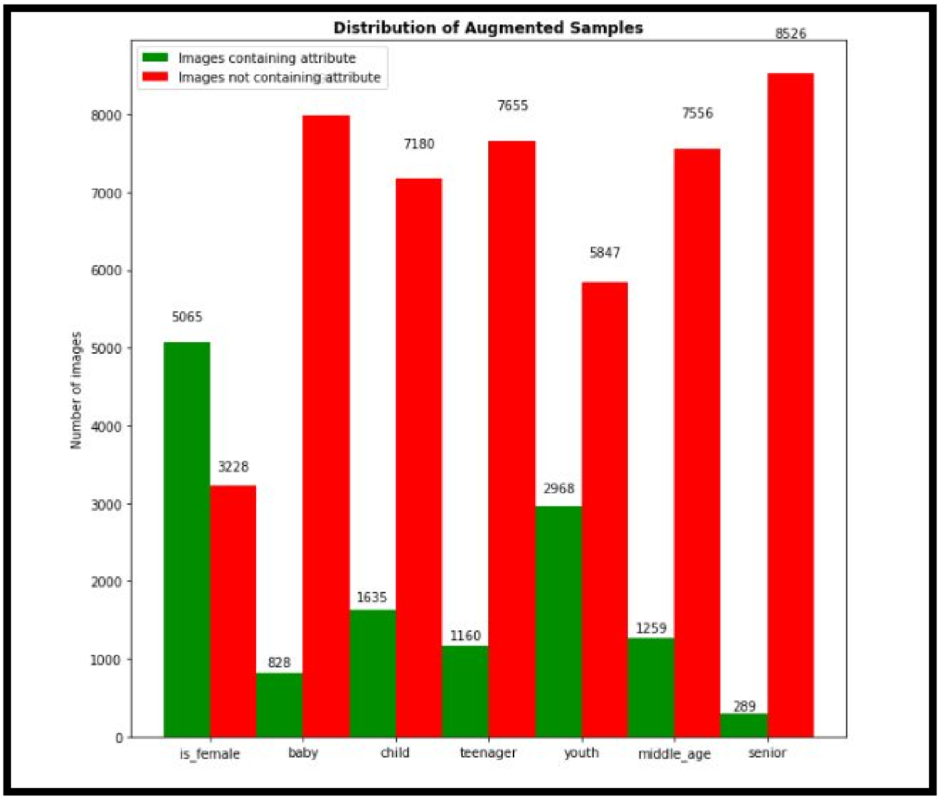
  The dataset quality seems to be very good for the problem statement provided.

Visualisations

The initial distribution of positive data samples across samples

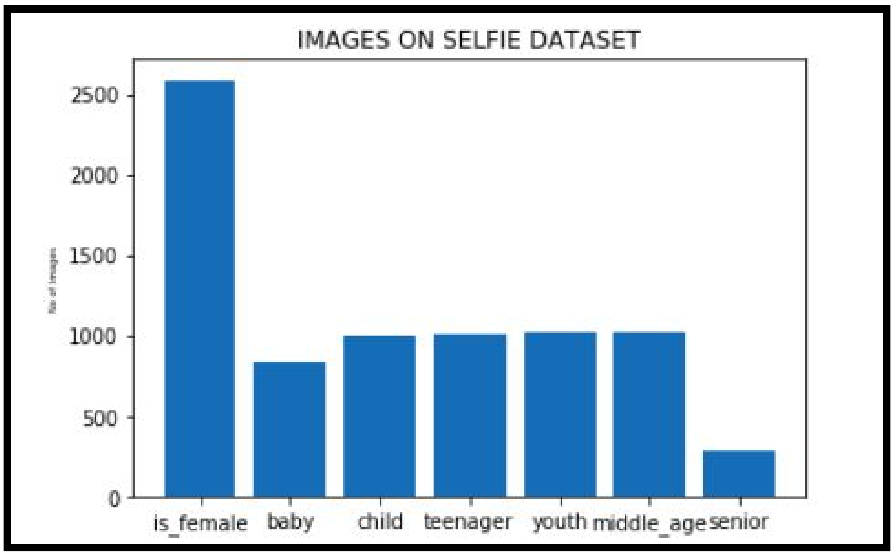


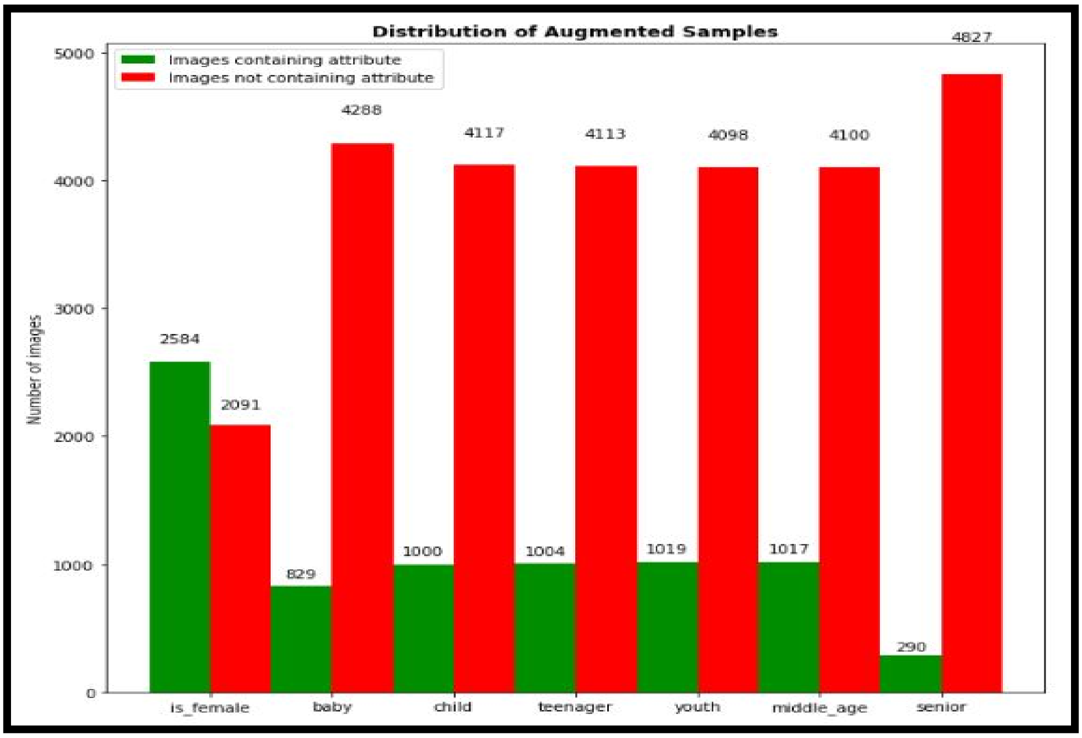
The data distribution across samples for all categories



Data distribution after Data augmentation

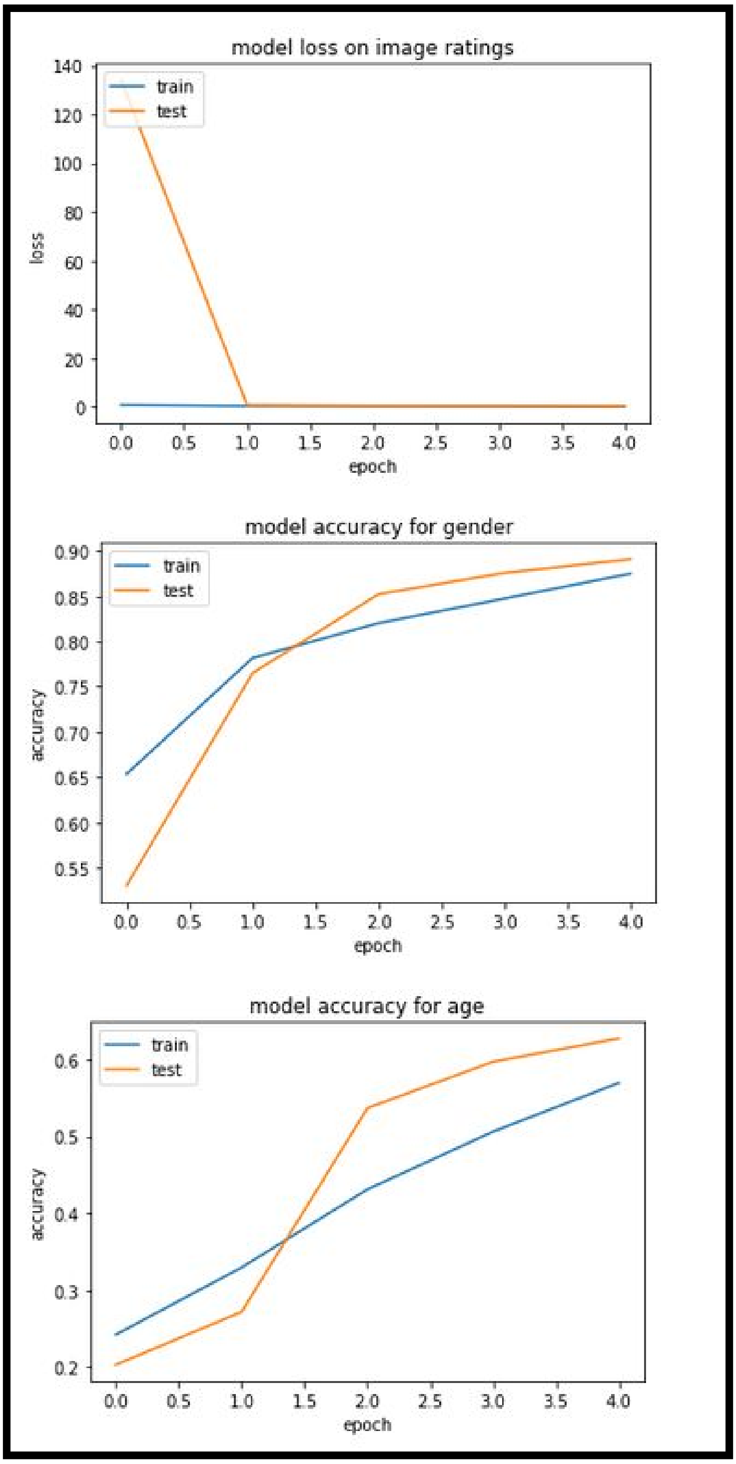
- We saw that the distribution across classes for each class is equalised by adding augmented data.
- We made sure every class has its fair share of representation.
- We augmented the samples for each class to be at least 1000 and made the distribution uniform for all the classes.
- For both the tasks and all subtasks we used a training sample for 6000 images, this was due to computational limitations as data sizes greater than 7k was generating a resource exhaustion error, this was because all training was done on a laptop with GPU Nvidia 950M 2GB with compute capability of 5.2, with 8 GB RAM. We were however were able to successfully leverage transfer learning to achieve satisfactory results.





Checkpoint 2:

Validation vs training loss(Architecture 1 - multi-task):



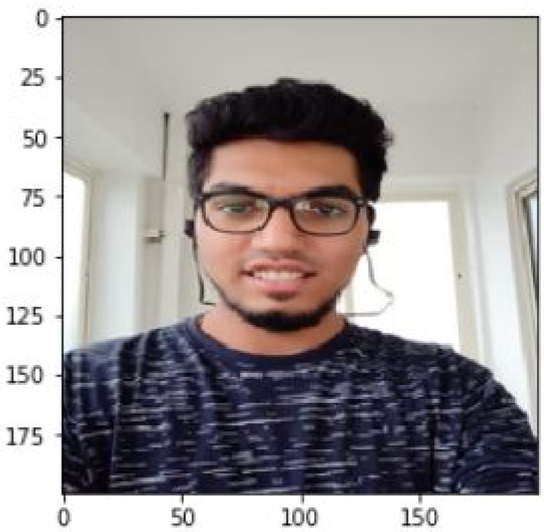
Accuracy and Loss:

```
********************************************************
loss for rating                     0.4464027030855163
accuracy for age gender             91.73789175487312
accuracy for age classification     69.23076923076923
********************************************************
```

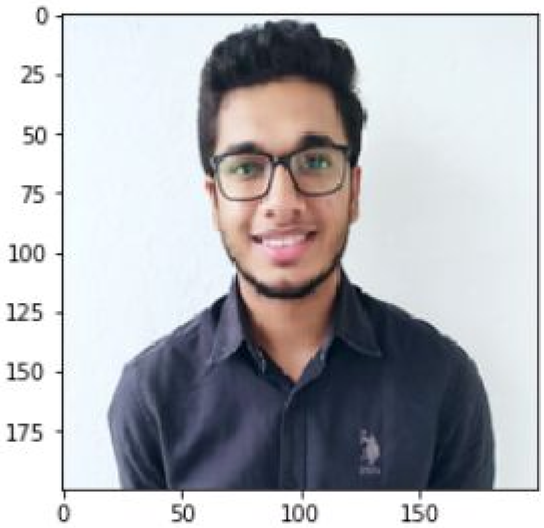
Checkpoint 3:

Custom test cases(on our own team member selfies):

```
In [31]: predict(model, "shahid1")

         Image Rating      4.6829886
         female:           no


         youth
```

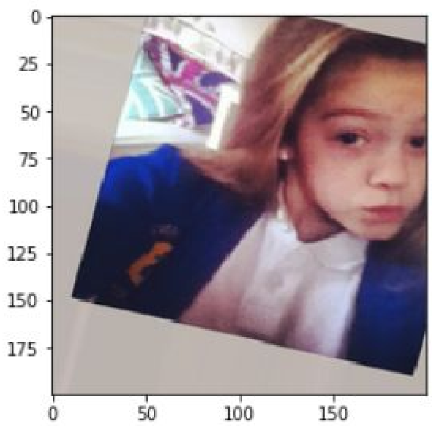


```
In [32]: predict(model, "sumanth1")

Image Rating     4.374714
female:          no


youth
```

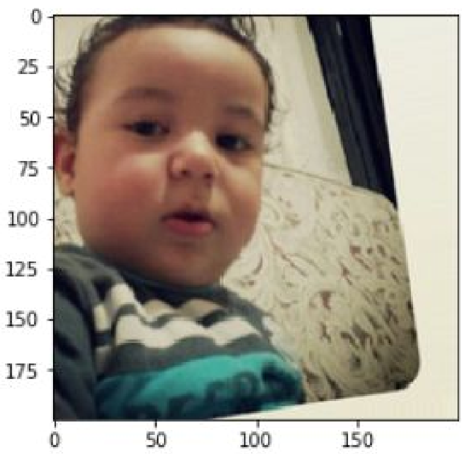


Predictions for gender and age on out of sample images:

```
In [26]: predict(model, "10175241_1408781076058160_1032737994_a0")

Image Rating     4.4858375
female:          yes


child
```



```
In [27]: predict(model, "10004346_477351965700359_810393995_a0")

Image Rating     2.8381429
female:          no


baby
```



Checkpoint 4:

Regression loss for multi-task model:

loss 0.4464027030855163

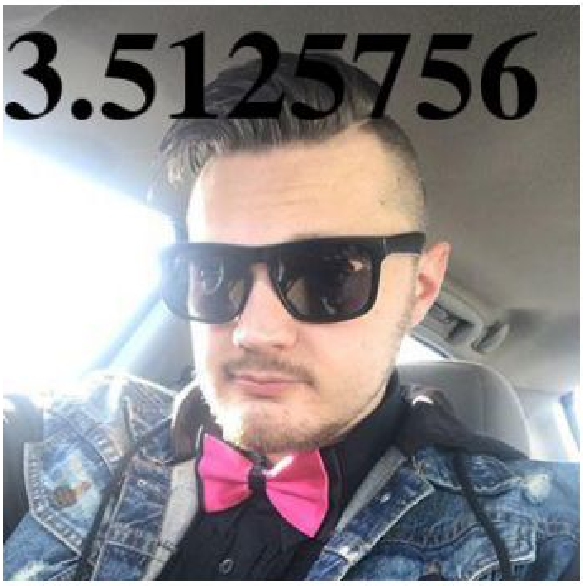
Regression loss for model with auxiliary input as predicted attributes:

loss 0.5246079049558721

Predicted popularity scores for public images:

```
In [50]: Image.open("insta_result/ (1).jpg")
Out[50]:
```


```
In [52]: Image.open("insta_result/ (5).jpg")
Out[52]:
```



Task 2:

To classify the selfie image as belonging to one of the following classes: “Poor”, “Average”, “Great” and also to identify the key attributes contributing to a good selfie.

Approach:

The first step is to convert the continuous-valued regression problem into a classification one by bucketing a range of values into a bin/class. We do this using pandas’ pd.qcut function, which provides us with three equally divided intervals based on quantiles. We then augment the dataset with this label. We build on the same classifier we used, by using the same Resnet50 backbone with a classification head using softmax as activation.

The latter part of the Task 2 pertaining to selecting the most important feature,we use the Chi Squared Test and Logistic Regression for the same.

The chi-square test is a statistical test of independence to determine the dependency of two variables.Since we are using categorical data,hence we have used chi-square test for the same.

In statistics, the logistic model is a widely used statistical model that, in its basic form, uses a logistic function to model a binary dependent variable; many more complex extensions exist.

Both of these approaches were used to extract the most important features.

Architecture

Input:

A vector representing the image scaled down to 200*200*3

Output:

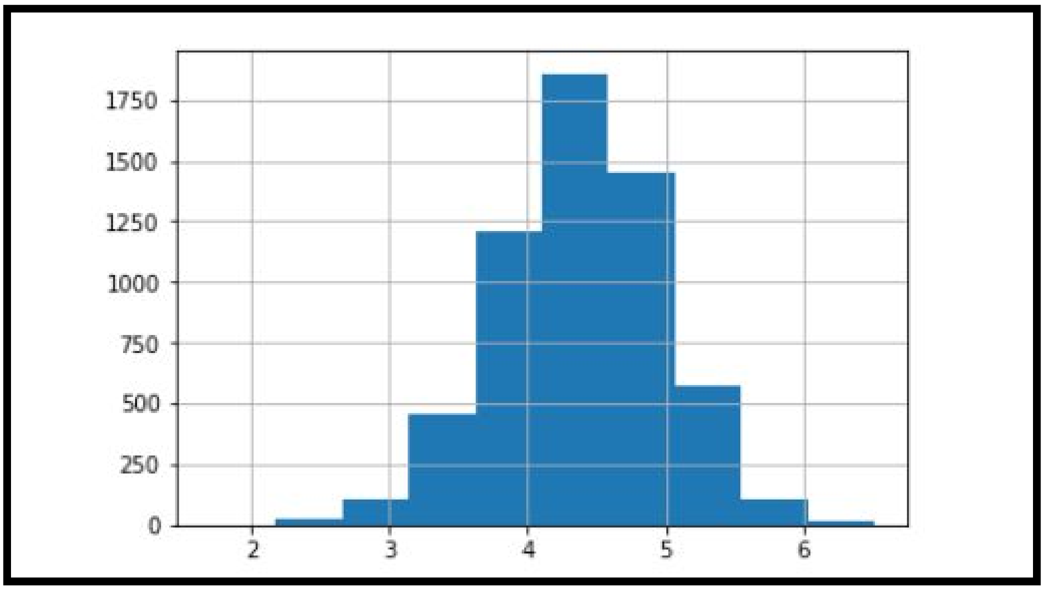
A class label indicating if the image quality is Good, Bad or Average

Training:

The input image is passed to the resNet architecture with the first 30 layers freeze and the model is added with the custom dense layers. The output layer is a softmax layer with three labels for the three quality classes of the image.

Results and Observations:

Histogram distribution of popularity scores in training samples

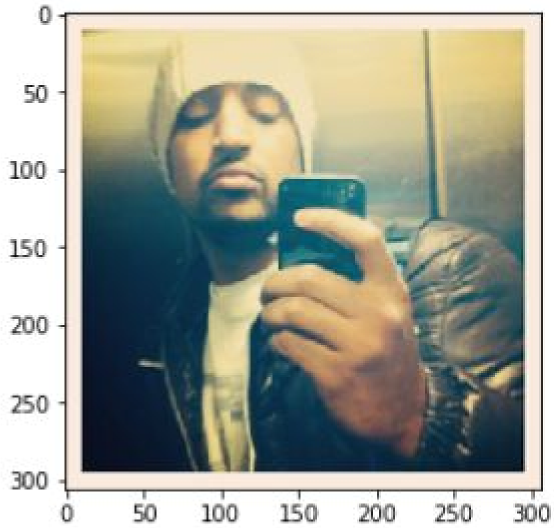


Loss(first value) and accuracy(second value) for the classification

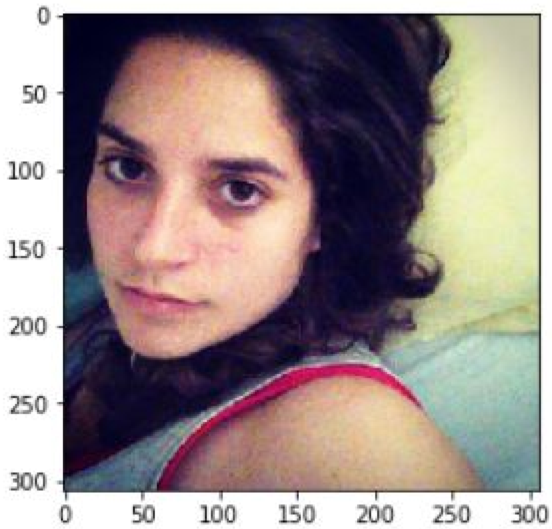
```
print(scores)
[0.9263633772524821, 0.5403225806451613]
```

Predictions for out of sample images

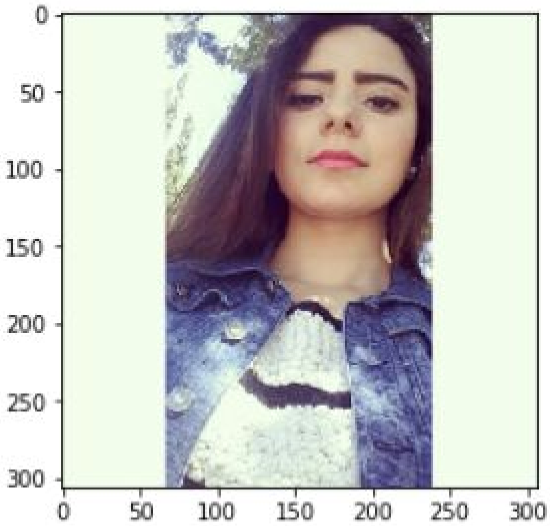
```
In [72]: predict("fed5374e66d211e3ad1e12ed570e9379_6")
Out[72]: 'poor'
```




```
In [81]: predict("fe40cfca3a6611e39fda22000a1f8ae8_6")
Out[81]: 'average'
```

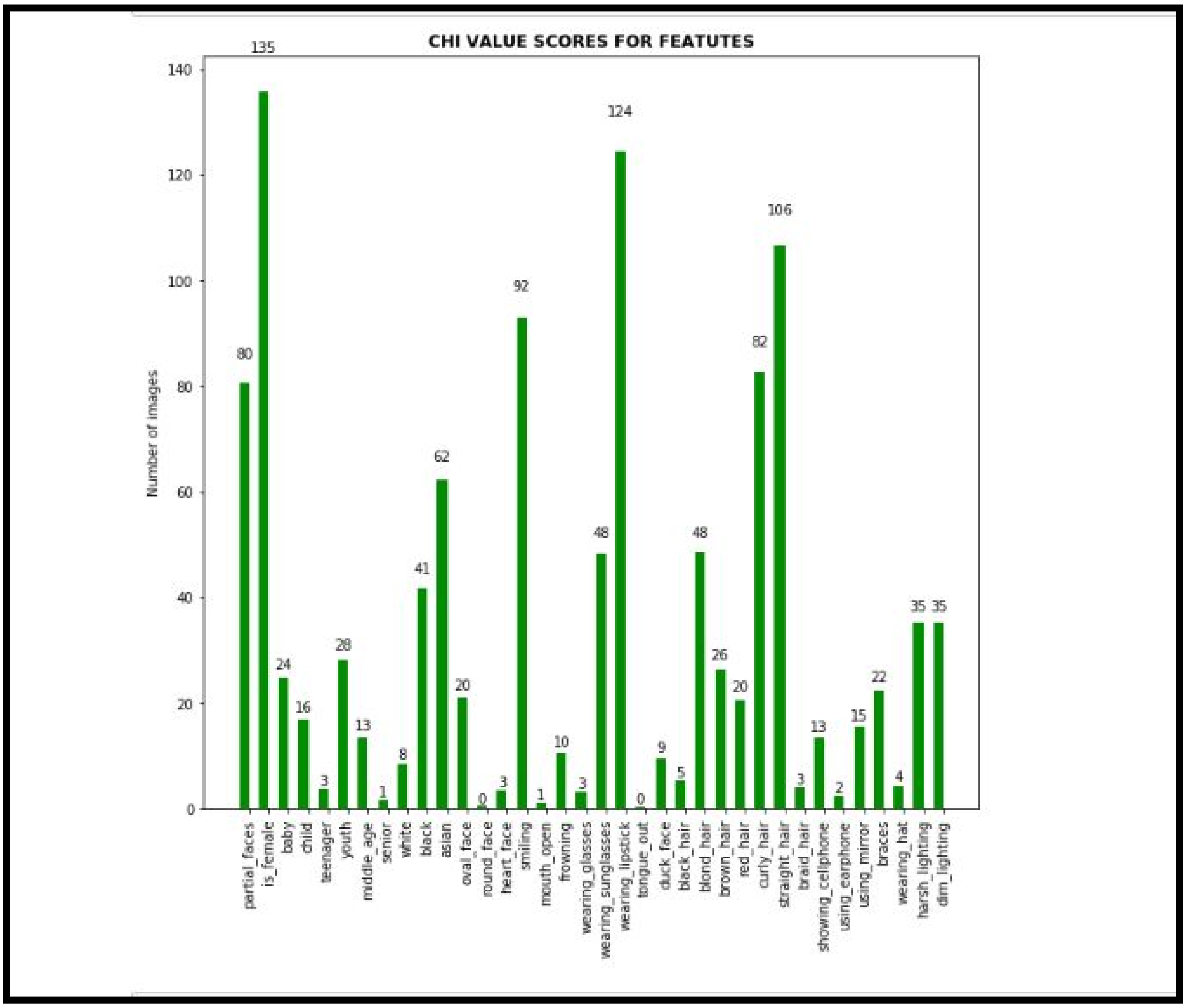


```
In [78]: predict("929053_1374671502815858_704281817_a")
Out[78]: 'good'
```



DATA ANALYSIS AND INFERENCE

Results



THE RANK ORDERING OF THE ATTRIBUTES

```
The top attributes are :
RANK            Attribute                       SCORE
1                is_female                       135.78389756318245
2                wearing_lipstick                124.35392335572422
3                straight_hair                   106.60675590709525
4                smiling                         92.92656260454172
5                curly_hair                      82.83312971899248
6                partial_faces                   80.75266458419698
7                asian                           62.18823357117863
```

THE RANK OUTPUT FOR ATTRIBUTES

```
1
[ 80.753 135.784  24.783  16.824   3.667  28.082  13.593   1.704   8.412
  41.937  62.188  20.97    0.525   3.533  92.927   1.166  10.576   3.185
  48.494 124.354   0.387   9.581   5.25   48.512  26.337  20.686  82.833
 106.607   3.919  13.415   2.393  15.51   22.427   4.333  35.139  35.139]
```