Confidential Customized for **Lorem Ipsum LLC** Version 1.0

AGE IDENTIFICATION

Lucien Meteumba

I-INTRODUCTION

- There has been a problem where people (mostly young African) cut their age in order to have acceptance into a soccer academy or school.

- Another problem we will try to solve in this project will be to determine the age of old people whose birthdate is not known.

- The goal of this project is to use someone's picture (who wants to get into a soccer academy) to detect his real age.

II-PROBLEM IDENTIFICATION

1 According to wikipedia, Age fraud is age fabrication or the use of false documentation to gain an advantage over opponents.

2 Since in most african countries, records are difficult to verify, African wanting to join a European soccer academic can easily change their age to meet the requirement.

3 With the help of Deep Learning we can now use their photo picture to identify how old they are.That will help Academy Sport to recruit the right people.

II-PROBLEM IDENTIFICATION(Cont’d)

4 Due to the lack of technology and the lack of money in the past (from 1965 downward) in Africa, people were not able to establish a birth certificate for their children.

5 That is why we see most of the old age people in Africa not knowing their age.

6 The model we will be developing will help identify their age.

II-PROBLEM IDENTIFICATION(Cont’d)

7 Our criteria for success will be to build, and evaluate the performance of a model that will be able to predict someone's age based on their image picture.

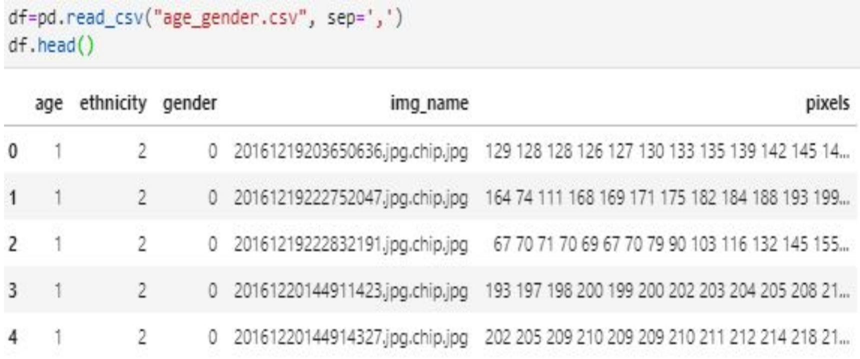
8 We will analyze the data taken from “age_gender.csv” from Kaggle’s Website.

9 The main constraints in this project might be missing values and the fact that our dataset contains less african ethnicity.

10 The intended stakeholders are the european soccer clubs, African society.

III-DATA DESCRIPTION

- The dataset we are using to support this project is “age_gender.csv” from Kaggle’s website.
- This dataset includes a CSV of facial images that are labeled on the basis of age, gender, and ethnicity.The dataset includes 27305 rows and 5 columns.
- ‘age’ is an integer from 0 to 116, indicating the age.
- ‘gender’ is either 0 (male) or 1 (female).
- ‘race’ is an integer from 0 to 4, denoting White, Black, Asian, Indian, and Others (like Hispanic, Latino, Middle Eastern).

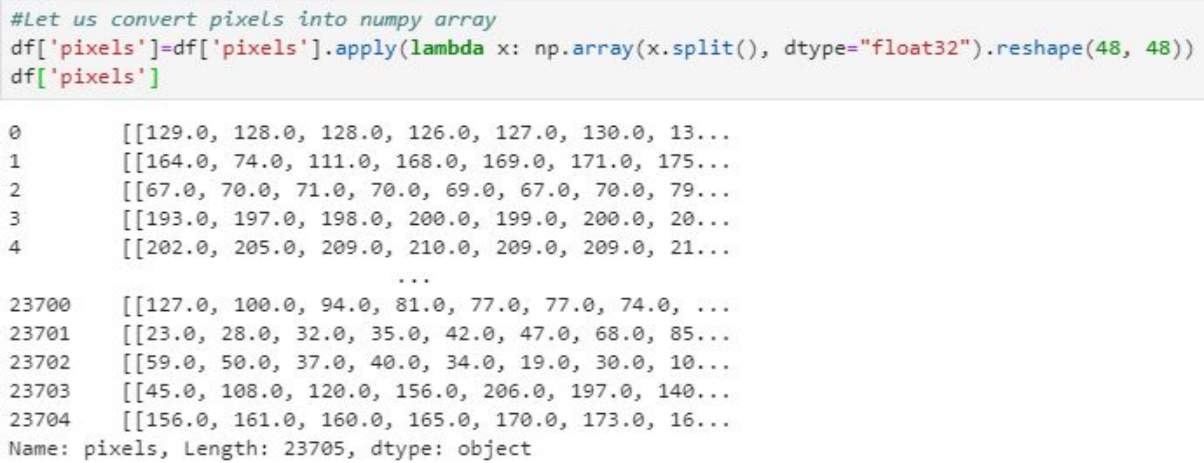


```
df=pd.read_csv("age_gender.csv", sep=',')
df.head()
```

	age	ethnicity	gender	img_name	pixels
0	1	2	0	20161219203650636.jpg.chip.jpg	129 128 128 126 127 130 133 135 139 142 145 14...
1	1	2	0	20161219222752047.jpg.chip.jpg	164 74 111 168 169 171 175 182 184 188 193 199...
2	1	2	0	20161219222832191.jpg.chip.jpg	67 70 71 70 69 67 70 79 90 103 116 132 145 155...
3	1	2	0	20161220144911423.jpg.chip.jpg	193 197 198 200 199 200 202 203 204 205 208 21...
4	1	2	0	20161220144914327.jpg.chip.jpg	202 205 209 210 209 209 210 211 212 214 218 21...

IV-EXPLORATORY DATA ANALYSIS

- We started by reshaping our pixels columns.

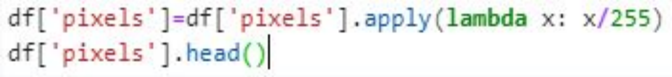


```
#Let us convert pixels into numpy array
df['pixels']=df['pixels'].apply(lambda x: np.array(x.split(), dtype="float32").reshape(48, 48))
df['pixels']
```

```
0        [[129.0, 128.0, 128.0, 126.0, 127.0, 130.0, 13...
1        [[164.0, 74.0, 111.0, 168.0, 169.0, 171.0, 175...
2        [[67.0, 70.0, 71.0, 70.0, 69.0, 67.0, 70.0, 79...
3        [[193.0, 197.0, 198.0, 200.0, 199.0, 200.0, 20...
4        [[202.0, 205.0, 209.0, 210.0, 209.0, 209.0, 21...
                               ...
23700    [[127.0, 100.0, 94.0, 81.0, 77.0, 77.0, 74.0, ...
23701    [[23.0, 28.0, 32.0, 35.0, 42.0, 47.0, 68.0, 85...
23702    [[59.0, 50.0, 37.0, 40.0, 34.0, 19.0, 30.0, 10...
23703    [[45.0, 108.0, 120.0, 156.0, 206.0, 197.0, 140...
23704    [[156.0, 161.0, 160.0, 165.0, 170.0, 173.0, 16...
Name: pixels, Length: 23705, dtype: object
```

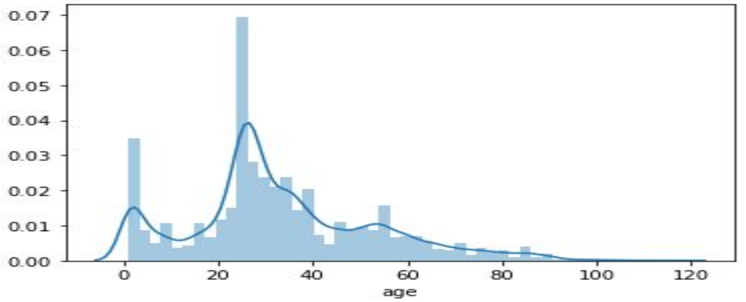
IV-EXPLORATORY DATA ANALYSIS(Cont’d)

- Then we normalized our pixels data.



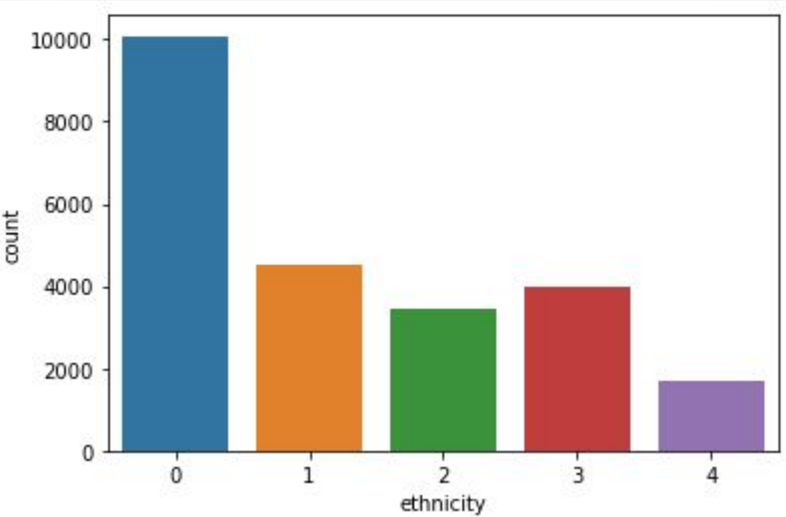
```
df['pixels']=df['pixels'].apply(lambda x: x/255)
df['pixels'].head()
```

- After normalizing the pixels data, we decided to plot the age distribution.



IV-EXPLORATORY DATA ANALYSIS(Cont'd)

- We can see that there are many people in our dataset that are between 0-2 years old and 24-26 years old. We can also observe that there are few people whose age is between 80-120.
- The figure below is the ethnicity distribution.



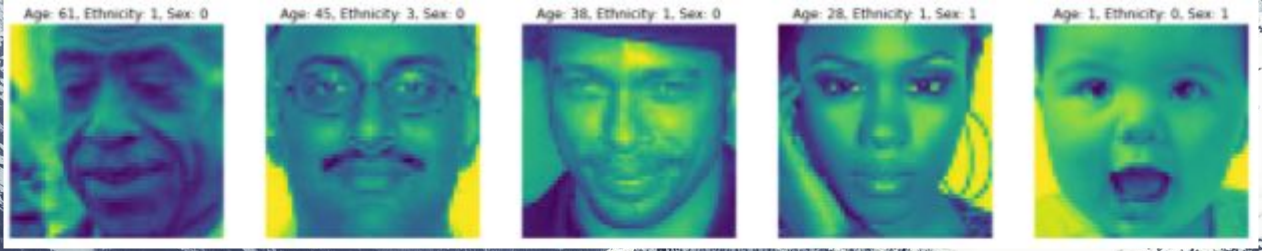
IV-EXPLORATORY DATA ANALYSIS(Cont’d)

- We could see that in this dataset there are more white than any other ethnicity.

- The black ethnicity comes in the second position.

- To be more accurate, 19.09% constitute the percentage of black people in this dataset.

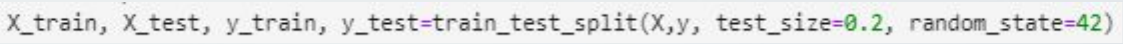
- We decided to visualize some of the images in the dataset. By displaying faces and genders we will be able to identify any anomaly in labeling the data.


Age: 61, Ethnicity: 1, Sex: 0
Age: 45, Ethnicity: 3, Sex: 0
Age: 38, Ethnicity: 1, Sex: 0
Age: 28, Ethnicity: 1, Sex: 1
Age: 1, Ethnicity: 0, Sex: 1

V- PRE_PROCESSING AND TRAINING DATA DEVELOPMENT

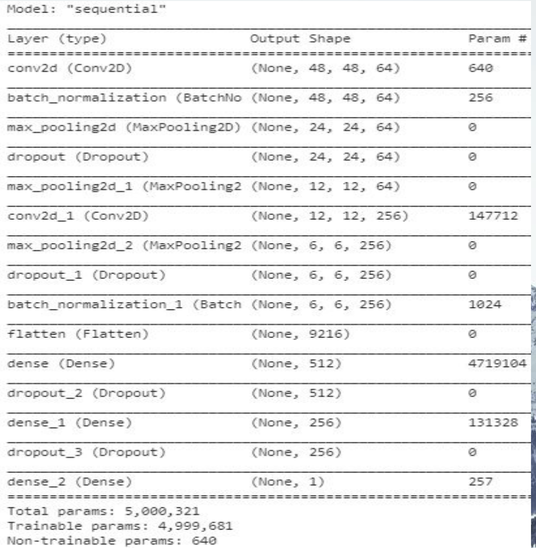
- We converted our image from 1D to 3D.

- We split the data into training and testing.



```
X_train, X_test, y_train, y_test=train_test_split(X,y, test_size=0.2, random_state=42)
```

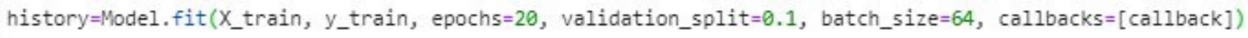
- Then we build the model as we can see the summary below



```
Model: "sequential"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
conv2d (Conv2D)              (None, 48, 48, 64)        640
_________________________________________________________________
batch_normalization (BatchNo (None, 48, 48, 64)        256
_________________________________________________________________
max_pooling2d (MaxPooling2D) (None, 24, 24, 64)        0
_________________________________________________________________
dropout (Dropout)            (None, 24, 24, 64)        0
_________________________________________________________________
max_pooling2d_1 (MaxPooling2 (None, 12, 12, 64)        0
_________________________________________________________________
conv2d_1 (Conv2D)            (None, 12, 12, 256)       147712
_________________________________________________________________
max_pooling2d_2 (MaxPooling2 (None, 6, 6, 256)         0
_________________________________________________________________
dropout_1 (Dropout)          (None, 6, 6, 256)         0
_________________________________________________________________
batch_normalization_1 (Batch (None, 6, 6, 256)         1024
_________________________________________________________________
flatten (Flatten)            (None, 9216)              0
_________________________________________________________________
dense (Dense)                (None, 512)               4719104
_________________________________________________________________
dropout_2 (Dropout)          (None, 512)               0
_________________________________________________________________
dense_1 (Dense)              (None, 256)               131328
_________________________________________________________________
dropout_3 (Dropout)          (None, 256)               0
_________________________________________________________________
dense_2 (Dense)              (None, 1)                 257
=================================================================
Total params: 5,000,321
Trainable params: 4,999,681
Non-trainable params: 640
```

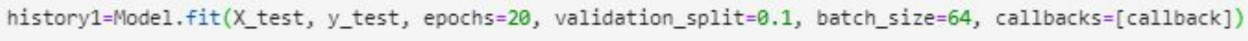
V- PRE_PROCESSING AND TRAINING DATA DEVELOPMENT(Cont’d)

- We fitted our model by passing the input features X_train and y_train. We also passed the validation set equal to 10% of the training set and 20 epochs.



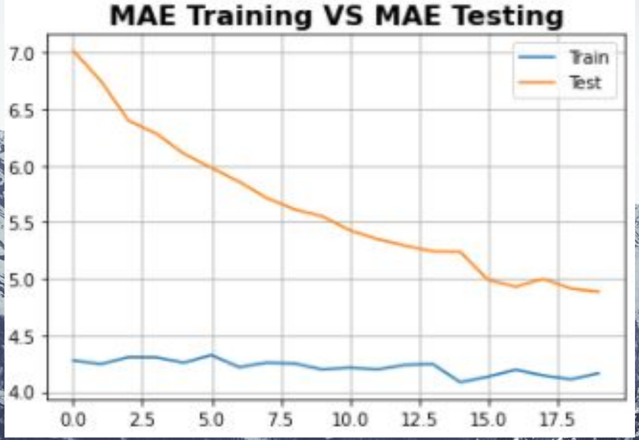
```
history=Model.fit(X_train, y_train, epochs=20, validation_split=0.1, batch_size=64, callbacks=[callback])
```

- We decided to do the same operation with the testing set ie X_test and y_test.



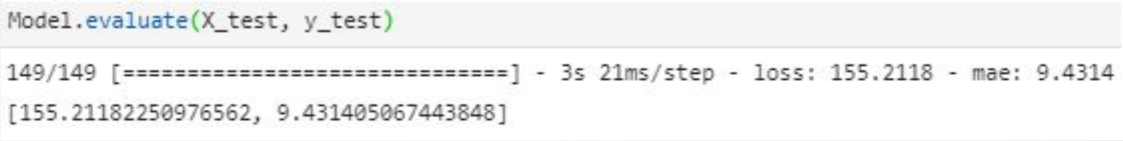
```
history1=Model.fit(X_test, y_test, epochs=20, validation_split=0.1, batch_size=64, callbacks=[callback])
```

- We plotted the MAE of the training data VS MAE of the testing data.



V- PRE_PROCESSING AND TRAINING DATA DEVELOPMENT(Cont’d)

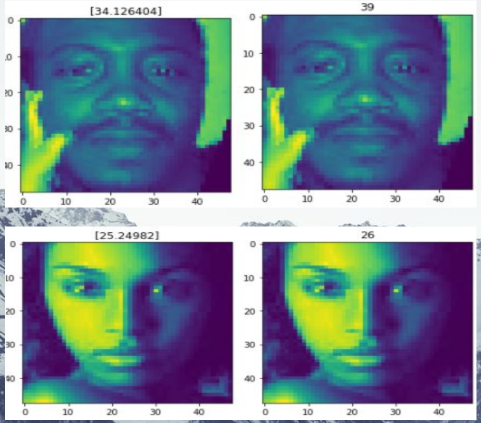
- We evaluated our model’s validation “mae” on the test data to estimate the generalization error before deploying the model to production.

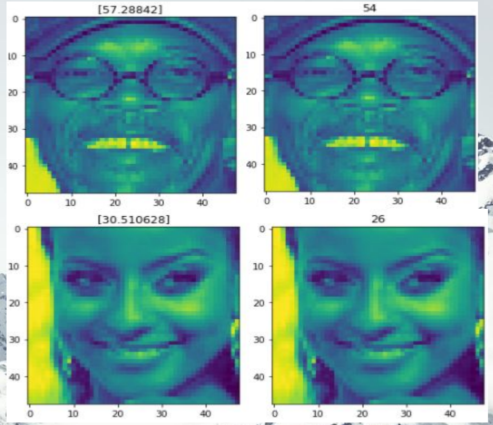


```
Model.evaluate(X_test, y_test)
```

```
149/149 [==============================] - 3s 21ms/step - loss: 155.2118 - mae: 9.4314
[155.21182250976562, 9.431405067443848]
```

- We finally predicted the age of few people in our dataset using our model and compared it with their real age.





VII- NEXT STEPS

- To have a better performance metrics we can use the Transfer learning technique.
- Transfer learning is when we build a new model on top of a pre-trained model.
- Another challenge of mine is to convert the picture we will take into the same nature as the one that was given to us in the dataset.
- We will be able to put our analysis into a software product where we will be able to take people's pictures and the software will output their age.

VII- NEXT STEPS(Cont’d)

- We can use the internet to create a database where we will collect only black people's photos and corresponding age since our project is targeted towards africans.

- That will help us add more data beneath our model in order to increase the prediction and update our model.

THANK YOU