

```
In [1]: import pandas as pd
import cv2
import os
import math
import scipy as sp
import PIL
import numpy as np
import tensorflow
```

```
In [5]: from tensorflow.keras import models, layers, Model
from tensorflow.keras.models import Sequential, load_model
from tensorflow.keras.layers import Conv2D, MaxPooling2D
from tensorflow.keras.layers import Dense, Dropout, Flatten, GlobalAveragePooling2D
from tensorflow.keras.layers import Flatten, Dense, Dropout, ZeroPadding2D

from tensorflow.keras.preprocessing.image import ImageDataGenerator
from tensorflow.keras.callbacks import ModelCheckpoint, TensorBoard, ReduceLROnPlateau
from tensorflow.keras import optimizers
from tensorflow.keras.optimizers import SGD
from tensorflow.keras.models import Model
from tensorflow.keras.preprocessing import image
from tensorflow.keras.applications import EfficientNetB4, EfficientNetB6, ResNet50V2
#from keras_tuner.tuners import RandomSearch

import scikitplot as skplt
from sklearn.metrics import roc_auc_score
import matplotlib.pyplot as plt
```

```
In [6]: import logging
import os
import warnings

import matplotlib.pyplot as plt
import matplotlib.style as style
import numpy as np
import pandas as pd
import seaborn as sns
import tensorflow as tf
from sklearn import preprocessing
from sklearn.model_selection import train_test_split
```

```
In [7]: os.getcwd()
```

```
Out[7]: '/content'
```

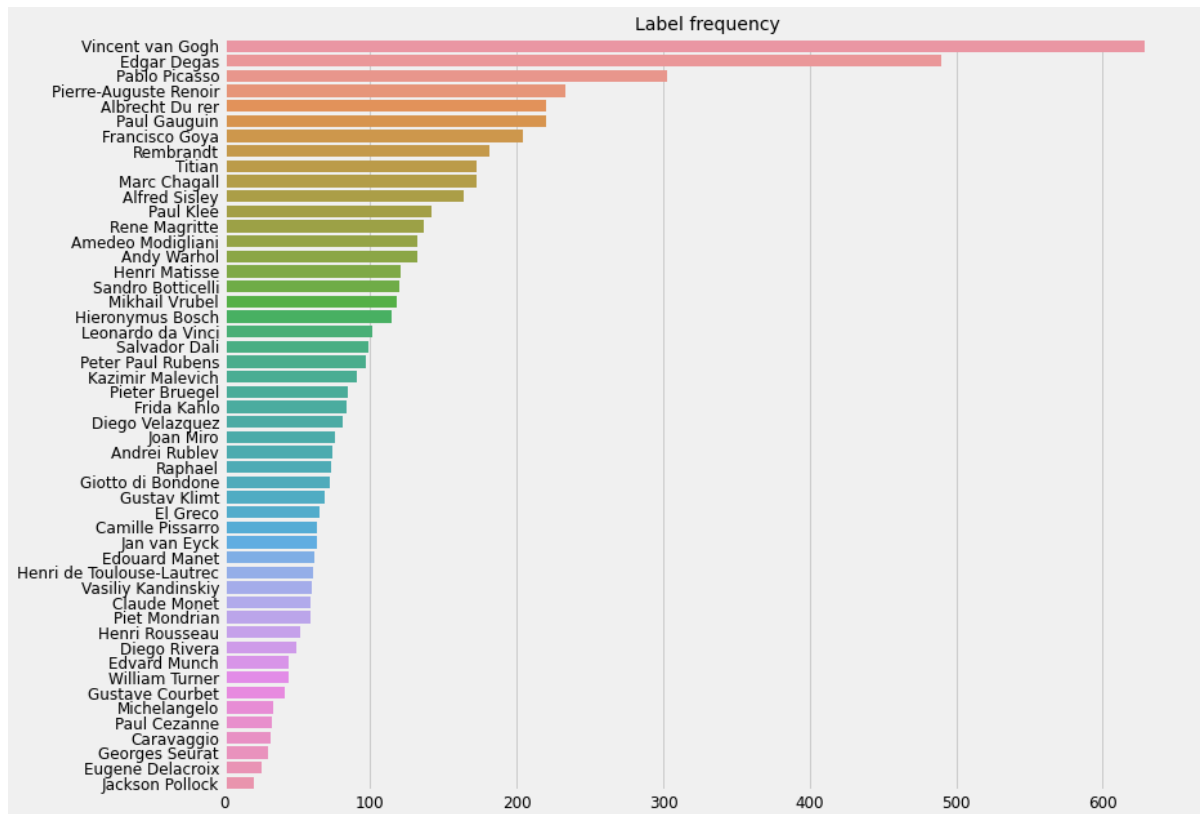
```
In [36]: df = pd.read_csv("/content/drive/MyDrive/Colab Notebooks/art_project/train.csv")
df.head()
```

```
Out[36]:
```

	id	img_path	artist
0	0	./train/0000.jpg	Diego Velazquez
1	1	./train/0001.jpg	Vincent van Gogh
2	2	./train/0002.jpg	Claude Monet
3	3	./train/0003.jpg	Edgar Degas
4	4	./train/0004.jpg	Hieronymus Bosch

```
In [37]: # Get label frequencies in descending order
label_freq = df['artist'].apply(lambda s: str(s).split('|')).explode().value_counts()

# Bar plot
style.use("fivethirtyeight")
plt.figure(figsize=(12,10))
sns.barplot(y=label_freq.index.values, x=label_freq, order=label_freq.index)
plt.title("Label frequency", fontsize=14)
plt.xlabel("")
plt.xticks(fontsize=12)
plt.yticks(fontsize=12)
plt.show()
```



```
In [38]: X_train, X_val, y_train, y_val = train_test_split(df, df['artist'].values, test_size=0.2)
print("Number of posters for training: ", len(X_train))
print("Number of posters for validation: ", len(X_val))
```

Number of posters for training: 4728
Number of posters for validation: 1183

```
In [39]: X_train = X_train.sort_values(by=['id'])
X_train['img_path'] = ['/content/drive/MyDrive/Colab Notebooks/art_project/' + path[0] for path in X_train['img_path'].values]
X_train.head()
```

```
Out[39]:
```

	id	img_path	artist
0	0	/content/drive/MyDrive/Colab Notebooks/art_pro...	Diego Velazquez
1	1	/content/drive/MyDrive/Colab Notebooks/art_pro...	Vincent van Gogh
2	2	/content/drive/MyDrive/Colab Notebooks/art_pro...	Claude Monet
4	4	/content/drive/MyDrive/Colab Notebooks/art_pro...	Hieronymus Bosch
5	5	/content/drive/MyDrive/Colab Notebooks/art_pro...	Pierre-Auguste Renoir

```
In [40]: X_val = X_val.sort_values(by=['id'])
X_val['img_path'] = ['/content/drive/MyDrive/Colab Notebooks/art_project/' + path[2:] for path in X_val['img_path'].values]
```

```
X_val.head(-5)
```

Out[40]:

	id	img_path	artist
3	3	/content/drive/MyDrive/Colab Notebooks/art_pro...	Edgar Degas
9	9	/content/drive/MyDrive/Colab Notebooks/art_pro...	Peter Paul Rubens
12	12	/content/drive/MyDrive/Colab Notebooks/art_pro...	Edouard Manet
14	14	/content/drive/MyDrive/Colab Notebooks/art_pro...	Francisco Goya
15	15	/content/drive/MyDrive/Colab Notebooks/art_pro...	Edgar Degas
...
5882	5882	/content/drive/MyDrive/Colab Notebooks/art_pro...	Edgar Degas
5883	5883	/content/drive/MyDrive/Colab Notebooks/art_pro...	Pablo Picasso
5884	5884	/content/drive/MyDrive/Colab Notebooks/art_pro...	Giotto di Bondone
5885	5885	/content/drive/MyDrive/Colab Notebooks/art_pro...	Francisco Goya
5886	5886	/content/drive/MyDrive/Colab Notebooks/art_pro...	Albrecht Du rer

1178 rows × 3 columns

```
In [41]: # batch_size: 한번에 forward & Backword 하는 샘플의 수
batch_size = 32
```

```
# Weight 조절 parameter
LearningRate = 1e-3 # 0.001
Decay = 1e-6

img_width = 224
img_height = 224
```

```
In [42]: # 모델 Return
ResNetModel = tf.keras.applications.ResNet50(include_top=False, weights='imagenet',
x = GlobalAveragePooling2D()(ResNetModel.output)
predictions = Dense(50, activation='softmax')(x)

DeepLearning = Model(inputs=ResNetModel.input, outputs=predictions)

# learning parameter를 더하여 최종 model compile
DeepLearning.compile(optimizer=
    SGD(learning_rate=LearningRate, decay=Decay, momentum=0.9, nesterov=True),
    loss='categorical_crossentropy',
    metrics=['acc']
) # 나이를, MSE
```

```
In [43]: DATAGEN_TRAIN = ImageDataGenerator(
    rescale=1./255,
    rotation_range=20,
    width_shift_range=0.2,
    height_shift_range=0.2,
    shear_range=0.2,
    zoom_range=0.2,
    horizontal_flip=True,
    vertical_flip=True,
    data_format="channels_last",
    validation_split=0.10) # Train / Validation
```

```
# Generator의 instance 생성 (Train)
```

```
TRAIN_GENERATOR = DATAGEN_TRAIN.flow_from_dataframe(  
    dataframe=X_train, x_col='img_path', y_col='class',  
    target_size=(244, 244), class_mode='categorical',  
    batch_size=32, shuffle=True,  
    subset = "training")  
  
VALID_GENERATOR = DATAGEN_TRAIN.flow_from_dataframe(  
    dataframe=X_train, x_col='img_path', y_col='class',  
    target_size=(244, 244), class_mode='categorical',  
    batch_size=32, shuffle=True,  
    subset = "validation")
```

Found 4256 validated image filenames belonging to 50 classes.

Found 472 validated image filenames belonging to 50 classes.

```
In [44]: CP = ModelCheckpoint(filepath='model/' +  
    'ResNet50-Sigmoid-{epoch:03d}-{loss:.4f}-{val_loss:.4f}.hdf5',  
    monitor='val_loss', verbose=1, save_best_only=True, mode='min')  
  
# TB = TensorBoard(log_dir=tensb_directory, write_graph=True, write_images=True, pro  
LR = ReduceLROnPlateau(monitor='val_loss', factor=0.5, patience=5, verbose=1, min_lr=  
CALLBACK = [CP, LR]
```

```
In [45]: DeepLearning.fit(  
    TRAIN_GENERATOR,  
    epochs=20,  
    callbacks=CALLBACK,  
    shuffle=True,  
    validation_data=VALID_GENERATOR)
```

Epoch 1/20
133/133 [=====] - ETA: 0s - loss: 2.9385 - acc: 0.2871
Epoch 1: val_loss improved from inf to 8.04392, saving model to model/ResNet50-Sigmoid-001-2.9385-8.0439.hdf5
133/133 [=====] - 1312s 10s/step - loss: 2.9385 - acc: 0.2871 - val_loss: 8.0439 - val_acc: 0.0487 - lr: 0.0010

Epoch 2/20
133/133 [=====] - ETA: 0s - loss: 1.7880 - acc: 0.5317
Epoch 2: val_loss did not improve from 8.04392
133/133 [=====] - 143s 1s/step - loss: 1.7880 - acc: 0.5317 - val_loss: 13.5986 - val_acc: 0.0127 - lr: 0.0010

Epoch 3/20
133/133 [=====] - ETA: 0s - loss: 1.3197 - acc: 0.6346
Epoch 3: val_loss did not improve from 8.04392
133/133 [=====] - 144s 1s/step - loss: 1.3197 - acc: 0.6346 - val_loss: 12.0645 - val_acc: 0.0233 - lr: 0.0010

Epoch 4/20
133/133 [=====] - ETA: 0s - loss: 1.0344 - acc: 0.7166
Epoch 4: val_loss improved from 8.04392 to 7.74181, saving model to model/ResNet50-Sigmoid-004-1.0344-7.7418.hdf5
133/133 [=====] - 145s 1s/step - loss: 1.0344 - acc: 0.7166 - val_loss: 7.7418 - val_acc: 0.0254 - lr: 0.0010

Epoch 5/20
133/133 [=====] - ETA: 0s - loss: 0.8281 - acc: 0.7761
Epoch 5: val_loss improved from 7.74181 to 6.46535, saving model to model/ResNet50-Sigmoid-005-0.8281-6.4653.hdf5
133/133 [=====] - 143s 1s/step - loss: 0.8281 - acc: 0.7761 - val_loss: 6.4653 - val_acc: 0.0551 - lr: 0.0010

Epoch 6/20
133/133 [=====] - ETA: 0s - loss: 0.6868 - acc: 0.8156
Epoch 6: val_loss improved from 6.46535 to 5.12300, saving model to model/ResNet50-Sigmoid-006-0.6868-5.1230.hdf5
133/133 [=====] - 144s 1s/step - loss: 0.6868 - acc: 0.8156 - val_loss: 5.1230 - val_acc: 0.1292 - lr: 0.0010

Epoch 7/20
133/133 [=====] - ETA: 0s - loss: 0.5709 - acc: 0.8487
Epoch 7: val_loss improved from 5.12300 to 4.29409, saving model to model/ResNet50-Sigmoid-007-0.5709-4.2941.hdf5
133/133 [=====] - 145s 1s/step - loss: 0.5709 - acc: 0.8487 - val_loss: 4.2941 - val_acc: 0.2288 - lr: 0.0010

Epoch 8/20
133/133 [=====] - ETA: 0s - loss: 0.4707 - acc: 0.8790
Epoch 8: val_loss improved from 4.29409 to 2.56186, saving model to model/ResNet50-Sigmoid-008-0.4707-2.5619.hdf5
133/133 [=====] - 144s 1s/step - loss: 0.4707 - acc: 0.8790 - val_loss: 2.5619 - val_acc: 0.4153 - lr: 0.0010

Epoch 9/20
133/133 [=====] - ETA: 0s - loss: 0.3912 - acc: 0.9034
Epoch 9: val_loss improved from 2.56186 to 1.46523, saving model to model/ResNet50-Sigmoid-009-0.3912-1.4652.hdf5
133/133 [=====] - 144s 1s/step - loss: 0.3912 - acc: 0.9034 - val_loss: 1.4652 - val_acc: 0.6017 - lr: 0.0010

Epoch 10/20
133/133 [=====] - ETA: 0s - loss: 0.3301 - acc: 0.9211
Epoch 10: val_loss improved from 1.46523 to 1.20185, saving model to model/ResNet50-Sigmoid-010-0.3301-1.2019.hdf5
133/133 [=====] - 144s 1s/step - loss: 0.3301 - acc: 0.9211 - val_loss: 1.2019 - val_acc: 0.6801 - lr: 0.0010

Epoch 11/20
133/133 [=====] - ETA: 0s - loss: 0.2861 - acc: 0.9300
Epoch 11: val_loss did not improve from 1.20185
133/133 [=====] - 141s 1s/step - loss: 0.2861 - acc: 0.9300 - val_loss: 1.2523 - val_acc: 0.6695 - lr: 0.0010

Epoch 12/20

```

133/133 [=====] - ETA: 0s - loss: 0.2444 - acc: 0.9389
Epoch 12: val_loss did not improve from 1.20185
133/133 [=====] - 142s 1s/step - loss: 0.2444 - acc: 0.9389
- val_loss: 1.2149 - val_acc: 0.7055 - lr: 0.0010
Epoch 13/20
133/133 [=====] - ETA: 0s - loss: 0.2089 - acc: 0.9509
Epoch 13: val_loss did not improve from 1.20185
133/133 [=====] - 141s 1s/step - loss: 0.2089 - acc: 0.9509
- val_loss: 1.2368 - val_acc: 0.6864 - lr: 0.0010
Epoch 14/20
133/133 [=====] - ETA: 0s - loss: 0.1746 - acc: 0.9624
Epoch 14: val_loss did not improve from 1.20185
133/133 [=====] - 141s 1s/step - loss: 0.1746 - acc: 0.9624
- val_loss: 1.3023 - val_acc: 0.6737 - lr: 0.0010
Epoch 15/20
133/133 [=====] - ETA: 0s - loss: 0.1571 - acc: 0.9638
Epoch 15: val_loss improved from 1.20185 to 1.17322, saving model to model/ResNet50-
Sigmoid-015-0.1571-1.1732.hdf5
133/133 [=====] - 142s 1s/step - loss: 0.1571 - acc: 0.9638
- val_loss: 1.1732 - val_acc: 0.6970 - lr: 0.0010
Epoch 16/20
133/133 [=====] - ETA: 0s - loss: 0.1405 - acc: 0.9706
Epoch 16: val_loss did not improve from 1.17322
133/133 [=====] - 141s 1s/step - loss: 0.1405 - acc: 0.9706
- val_loss: 1.3708 - val_acc: 0.6547 - lr: 0.0010
Epoch 17/20
133/133 [=====] - ETA: 0s - loss: 0.1301 - acc: 0.9744
Epoch 17: val_loss did not improve from 1.17322
133/133 [=====] - 139s 1s/step - loss: 0.1301 - acc: 0.9744
- val_loss: 1.1956 - val_acc: 0.6907 - lr: 0.0010
Epoch 18/20
133/133 [=====] - ETA: 0s - loss: 0.1075 - acc: 0.9796
Epoch 18: val_loss did not improve from 1.17322
133/133 [=====] - 141s 1s/step - loss: 0.1075 - acc: 0.9796
- val_loss: 1.1890 - val_acc: 0.7246 - lr: 0.0010
Epoch 19/20
133/133 [=====] - ETA: 0s - loss: 0.0958 - acc: 0.9819
Epoch 19: val_loss did not improve from 1.17322
133/133 [=====] - 141s 1s/step - loss: 0.0958 - acc: 0.9819
- val_loss: 1.2673 - val_acc: 0.6992 - lr: 0.0010
Epoch 20/20
133/133 [=====] - ETA: 0s - loss: 0.1016 - acc: 0.9800
Epoch 20: val_loss did not improve from 1.17322

Epoch 20: ReduceLROnPlateau reducing learning rate to 0.0005000000237487257.
133/133 [=====] - 141s 1s/step - loss: 0.1016 - acc: 0.9800
- val_loss: 1.1782 - val_acc: 0.7246 - lr: 0.0010
<keras.callbacks.History at 0x7f459c6dbc90>

```

Out[45]:

In [48]: `os.listdir()`

Out[48]: `['.config', 'drive', 'model', 'sample_data']`

In [47]: `X_train.head(1)`

Out[47]:

	id	img_path	artist
--	----	----------	--------

0	0	/content/drive/MyDrive/Colab Notebooks/art_pro...	Diego Velazquez
---	---	---	-----------------

/content/drive/MyDrive/Colab Notebooks/art_project/test

In [4]:

```
In [49]: DeepLearning.load_weights('model/ResNet50-Sigmoid-015-0.1571-1.1732.hdf5')
```

```
In [50]: X_test = pd.read_csv('/content/drive/MyDrive/Colab Notebooks/art_project/test.csv')
```

```
In [51]: X_test['img_path'] = ['/content/drive/MyDrive/Colab Notebooks/art_project/' + path[2
```

```
In [52]: X_test.head()
```

Out[52]:

	id	img_path
--	----	----------

0	TEST_00000	/content/drive/MyDrive/Colab Notebooks/art_pro...
---	------------	---

1	TEST_00001	/content/drive/MyDrive/Colab Notebooks/art_pro...
---	------------	---

2	TEST_00002	/content/drive/MyDrive/Colab Notebooks/art_pro...
---	------------	---

3	TEST_00003	/content/drive/MyDrive/Colab Notebooks/art_pro...
---	------------	---

4	TEST_00004	/content/drive/MyDrive/Colab Notebooks/art_pro...
---	------------	---

```
In [60]: DATAGEN_TEST = ImageDataGenerator(  
    rescale=1./255,  
    data_format="channels_last")  
  
TEST_GENERATOR = DATAGEN_TEST.flow_from_dataframe(  
    dataframe=X_test, x_col='img_path', y_col='i  
    target_size=(244, 244), class_mode='raw',  
    batch_size=32, shuffle=False)
```

Found 12670 validated image filenames.

```
In [ ]: DeepLearning.evaluate(TEST_GENERATOR)
```

```
In [61]: # Prediction  
TEST_Prediction = DeepLearning.predict_generator(TEST_GENERATOR, verbose=1)
```

/usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:2: UserWarning: `Model.predict_generator` is deprecated and will be removed in a future version. Please use `Model.predict`, which supports generators.

396/396 [=====] - 3120s 8s/step

```
In [62]: TEST_Prediction
```

```
Out[62]: array([[4.3478703e-05, 4.3665199e-04, 1.3298820e-03, ..., 1.4711377e-04,  
    3.8490002e-03, 1.7981105e-04],  
    [4.5779203e-08, 4.2928708e-08, 9.8870510e-01, ..., 1.6035719e-08,  
    9.6711638e-06, 4.3315995e-07],  
    [2.1293246e-04, 4.8408488e-06, 2.3275219e-05, ..., 3.8956859e-06,  
    3.8254244e-04, 2.1760596e-05],  
    ...,  
    [4.3233907e-05, 1.3361792e-05, 1.2111354e-01, ..., 3.5921991e-04,  
    4.0082091e-06, 3.5090445e-05],  
    [1.5671427e-05, 1.5789061e-06, 3.0304373e-06, ..., 1.2904723e-06,  
    1.9172086e-05, 2.8163382e-05],  
    [3.3157721e-10, 5.2229426e-10, 1.4852722e-07, ..., 1.4515211e-10,  
    9.9981254e-01, 9.8249305e-11]], dtype=float32)
```

```
In [66]: artist_num = np.argmax(TEST_Prediction, axis = 1)
```

```
In [67]: artist_num
```

```
Out[67]: array([33,  2, 29, ..., 33, 46, 48])
```

```
In [68]: num_list = pd.DataFrame(artist_num, columns = ['artist'])  
num_list
```

```
Out[68]:
```

	artist
0	33
1	2
2	29
3	0
4	48
...	...
12665	10
12666	16
12667	33
12668	46
12669	48

12670 rows × 1 columns

```
In [70]: num_list.to_csv("ansT.csv", index = False)
```

여기부터 sample_submission에 맞춰 csv로 저장하는 코드

```
In [2]: import numpy as np  
import pandas as pd
```

```
In [15]: df = pd.read_csv("C:/Users/Yedam/Downloads/ansT.csv")  
df.head(3)
```

```
Out[15]:
```

	artist
0	33
1	2
2	29

```
In [5]: pic = pd.read_csv("C:/Users/Yedam/Desktop/art_project/sample_submission.csv")  
pic.head(3)
```


Out[5]:

	id	artist
0	TEST_00000	Edgar Degas
1	TEST_00001	Edgar Degas
2	TEST_00002	Edgar Degas

```
In [6]: artist_info = pd.read_csv("C:/Users/Yedam/Desktop/art_project/artists_info.csv")
artist_info.head(3)
```

Out[6]:

	name	years	genre	nationality
0	Amedeo Modigliani	1884 - 1920	Expressionism	Italian
1	Vasiliy Kandinskiy	1866 - 1944	Expressionism,Abstractionism	Russian
2	Diego Rivera	1886 - 1957	Social Realism,Muralism	Mexican

```
In [7]: con = pd.concat([pic, df], axis=1)
con.head()
```

Out[7]:

	id	artist	num
0	TEST_00000	Edgar Degas	33
1	TEST_00001	Edgar Degas	2
2	TEST_00002	Edgar Degas	29
3	TEST_00003	Edgar Degas	0
4	TEST_00004	Edgar Degas	48

```
In [8]: pre_ans = con.drop('artist', axis=1)
pre_ans.head()
```

Out[8]:

	id	num
0	TEST_00000	33
1	TEST_00001	2
2	TEST_00002	29
3	TEST_00003	0
4	TEST_00004	48

```
In [9]: num_50 = np.arange(50)
num_50
```

Out[9]:

```
array([ 0,  1,  2,  3,  4,  5,  6,  7,  8,  9, 10, 11, 12, 13, 14, 15, 16,
        17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33,
        34, 35, 36, 37, 38, 39, 40, 41, 42, 43, 44, 45, 46, 47, 48, 49])
```

```
In [10]: new_df = pd.DataFrame(artist_info['name'])
new_df['num'] = num_50
new_df.head(3)
```

Out[10]:

	name	num
0	Amedeo Modigliani	0
1	Vasiliy Kandinskiy	1
2	Diego Rivera	2

```
In [37]: list1 = df.to_dict()
list2 = list1['artist'].values()
ans_num = list(list2)
ans_num[:5]
```

Out[37]: [33, 2, 29, 0, 48]

```
In [38]: dict1 = new_df['name'].to_dict()
dict1
```

```
Out[38]: {0: 'Amedeo Modigliani',
 1: 'Vasiliy Kandinskiy',
 2: 'Diego Rivera',
 3: 'Claude Monet',
 4: 'Rene Magritte',
 5: 'Salvador Dali',
 6: 'Edouard Manet',
 7: 'Andrei Rublev',
 8: 'Vincent van Gogh',
 9: 'Gustav Klimt',
10: 'Hieronymus Bosch',
11: 'Kazimir Malevich',
12: 'Mikhail Vrubel',
13: 'Pablo Picasso',
14: 'Peter Paul Rubens',
15: 'Pierre-Auguste Renoir',
16: 'Francisco Goya',
17: 'Frida Kahlo',
18: 'El Greco',
19: 'Albrecht Dürer',
20: 'Alfred Sisley',
21: 'Pieter Bruegel',
22: 'Marc Chagall',
23: 'Giotto di Bondone',
24: 'Sandro Botticelli',
25: 'Caravaggio',
26: 'Leonardo da Vinci',
27: 'Diego Velazquez',
28: 'Henri Matisse',
29: 'Jan van Eyck',
30: 'Edgar Degas',
31: 'Rembrandt',
32: 'Titian',
33: 'Henri de Toulouse-Lautrec',
34: 'Gustave Courbet',
35: 'Camille Pissarro',
36: 'William Turner',
37: 'Edvard Munch',
38: 'Paul Cezanne',
39: 'Eugene Delacroix',
40: 'Henri Rousseau',
41: 'Georges Seurat',
42: 'Paul Klee',
43: 'Piet Mondrian',
44: 'Joan Miro',
45: 'Andy Warhol',
46: 'Paul Gauguin',
47: 'Raphael',
48: 'Michelangelo',
49: 'Jackson Pollock'}
```

```
In [ ]: artist_name = []
for i in ans_num:
    artist_name.append(dict1[i])
artist_name
```

```
In [41]: last_ans = pd.read_csv("C:/Users/Yedam/Desktop/art_project/sample_submission.csv")

last_ans.rename(columns = {'artist': 'artist_name'}, inplace=True)
last_ans.head(3)
```

Out[41]:

	id	artist_name
0	TEST_00000	Edgar Degas
1	TEST_00001	Edgar Degas
2	TEST_00002	Edgar Degas

```
In [42]: last_ans['artist'] = artist_name
last_ans.head()
```

Out[42]:

	id	artist_name	artist
0	TEST_00000	Edgar Degas	Henri de Toulouse-Lautrec
1	TEST_00001	Edgar Degas	Diego Rivera
2	TEST_00002	Edgar Degas	Jan van Eyck
3	TEST_00003	Edgar Degas	Amedeo Modigliani
4	TEST_00004	Edgar Degas	Michelangelo

```
In [43]: last_ans.drop('artist_name', axis=1, inplace=True)
last_ans.head()
```

Out[43]:

	id	artist
0	TEST_00000	Henri de Toulouse-Lautrec
1	TEST_00001	Diego Rivera
2	TEST_00002	Jan van Eyck
3	TEST_00003	Amedeo Modigliani
4	TEST_00004	Michelangelo

```
In [44]: last_ans.to_csv("answer1.csv", index=False)
```