Capstone Session 12

Deep Learning for Advanced Modeling

Deep Learning for Customer Insights

Aura must offer intuitive analyses that help customers make informed decisions to push relevant ads, services and products based on real-time user sentiments.

ClickO’s customers

Banking Organization

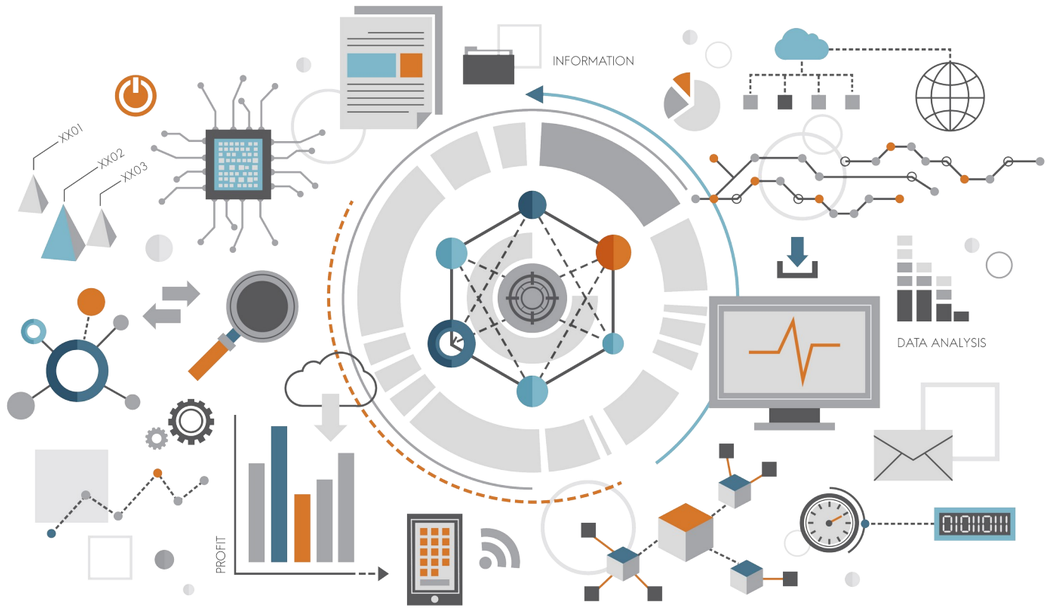
Movie Theater Owner

E-Commerce Company

Dental Clinic

Project Statement

Build necessary data aggregation, wrangling, and visualization modules for Aura using the Healthcare dataset.



- Identify customers who churn the bank
- Detect humans wearing face masks
- Classify customer product reviews
- Denoise dirty documents

Session 9 12: Dataset Description

Variable	Description
Image File Name	Name of the image file [name is a number from 1-116]

Session 12: Enhancing Dental X-rays with Autoencoders

Task: Build an autoencoder model to improve the clarity of dental X-rays (denoise the X-rays) using the dental panoramic dataset.

Dataset: Dental-Panaromic-Autoencoder.npz

This dataset consists of anonymized and de-identified panoramic dental X-rays of 116 patients, taken at Noor Medical Imaging Center, Qom, Iran. The subjects cover a wide range of dental conditions from healthy, to partial, and complete edentulous cases. The mandibles of all cases are manually segmented by two dentists. The dataset is saved in compressed NumPy format which when loaded using numpy.load will give you x_train, y_train, x_test and y_test NumPy arrays respectively.

Session 12: Enhancing Dental X-rays with Autoencoders

Task A:

- Load the Dental-Panaromic-Autoencoder.npz dataset using NumPy.load.
- Note: NPZ is a file format by NumPy that provides storage of array data using gzip compression.
- Extract x_train, y_train, x_test and y_test numpy arrays from the dataset
- A noisy version of the dataset will be created by applying random noise to each image.
  - With a noise factor of 0.2, add noise to the signal by multiplying the noise factor and random values from a normal distribution.
  - Hint:
  - noise_factor = 0.2
  - x_train_noisy = x_train + noise_factor * tf.random.normal(shape=x_train.shape)
  - x_test_noisy = x_test + noise_factor * tf.random.normal(shape=x_test.shape)
  - Clip the signal values between 0 and 1

Session 12: Enhancing Dental X-rays with Autoencoders

Task A:

- Plot the first 5 Xray images from original images (x_train)
- Plot the first 5 Xray images from noise images (x_train_noisy)
- Train an autoencoder using the noisy image as the input and the original image as the destination. The images are shaped 256x256 in RGB scale
  - Create a Denoise class inherited from Keras Model

  The class has an encoder member variable of the following layers:
  - Input Layer of shape 256*256*3
  - Conv2D with 64 filters, kernel size 3,3, activation relu and same padding and strides 2
  - Conv2D with 32 filters, kernel size 3,3, activation relu and same padding and strides 2
  - The class has a decoder member variable of below layers
  - Conv2DTranspose with 32 filters, kernel size 3,3, activation relu and same padding and stride 2

Session 12: Enhancing Dental X-rays with Autoencoders

Task A:

- Train an autoencoder using the noisy image as the input and the original image as the destination. The images are shaped 256x256 in RGB scale *(cont.)*
  - Conv2DTranspose with 64 filters, kernel size 3,3, activation relu and same padding and stride 2
  - Conv2D with 1 filter, kernel size 3,3, activation sigmoid and same padding
  - Create a call member function to the class which will pass the input to the encoder member function and the output of the encoder to the decoder member function
  - Initialize the autoencoder object of the class Denoise
  - Compile the autoencoder with Adam optimizer and MeanSquaredError as loss
  - Train the autoencoder with X as x_train_noisy and Y as x_train for 50 epochs and validation data as x_test_noisy and x_test
- Plot training and validation MAE and loss against epochs

Session 12: Enhancing Dental X-rays with Autoencoders

Task B:

- Evaluate the autoencoder model on x_test
    - Pass the x_test into the encoder
    - Pass the encoded images into the decoder. This step gives the decoded images that are reconstructed from the original.
- Plot both the first 10 noisy images (x_test_noisy) and images without noise produced by the autoencoder. Check how well the AutoEncoder model has done the job with denoising.

Thank You