Day - 1 (Sat, 05-OCT)

04 October 2024 23:32

Github link for codes and session notes :

https://github.com/gridflowai/SL-GenAI-course-4/tree/main

GenAI - pre-reqs measure:-
https://docs.google.com/forms/d/e/1FAIpQLSf1GZNCHaVYGUSmOFyaNDunsN7a4Jy5wz9QvwEpojXxYO9AIA/viewform?usp=sf_link

Comments from participants

Python, Prompt engg basics, AI model , transformer, Exercises on Python, No exercises on AI model. Fine tuning, Text/Image , then may be other DL models. More LAB, more exercises (Biplab)

LangChain was especially interesting and useful. (Jeff)

Best methods / practices, what models are relevant , creating a model and tuning (Chandc)

Industry applications (Finance or medical (text)) - selection of model (Srini)

Pre-requisite was lacking (foundation - AI model) (praveena)

When we will starting coding (chat model) (Bianca)

Gen AI model evaluation Security (Deeshant)

Qs : Logits ? Sigmoid fn or softmax fn

Qs : Classical eval methods (ML/DL)
- Acc
- F1-score
- Pred
- Conf matrix
- Prec, recall
- MAE, MSE, R2…
- Cross validation methods
- Decision boundaries
- Variances in the pred

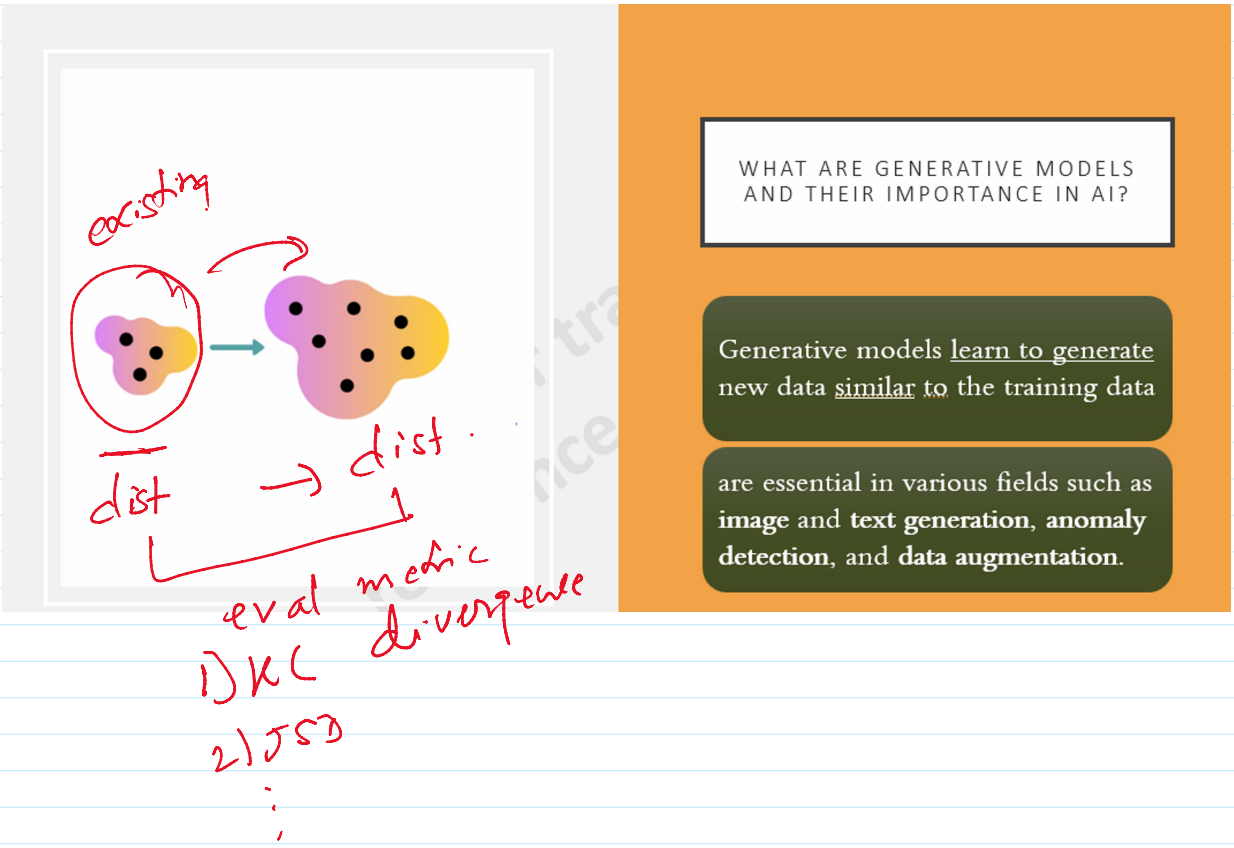
Some inputs on the classical eval methods

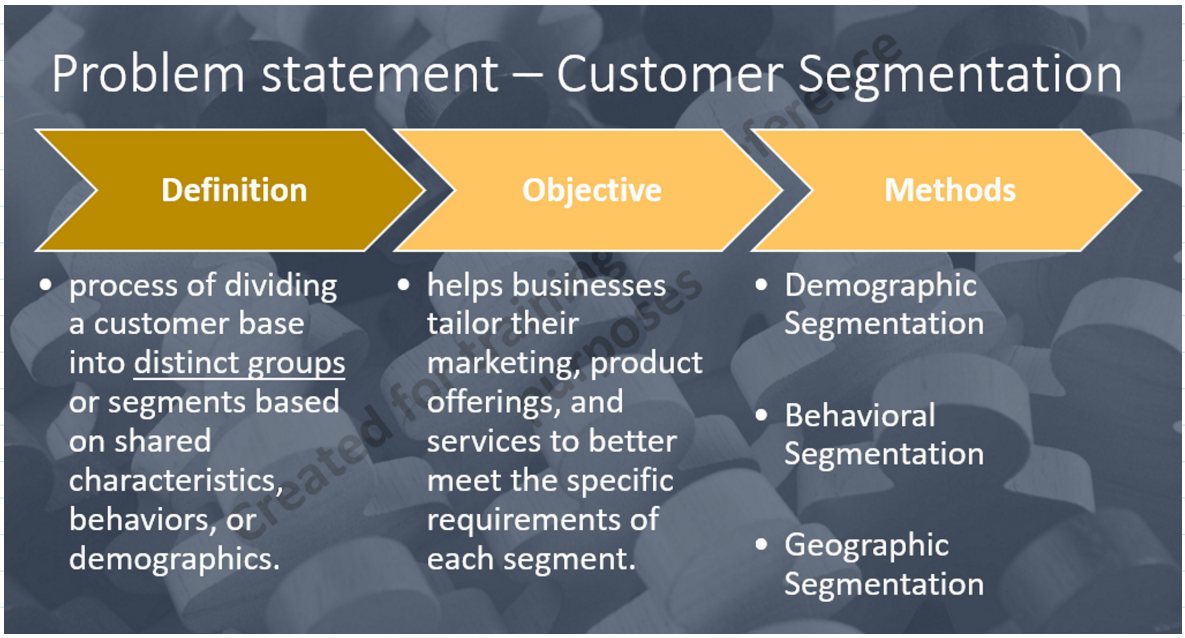
Text/LLM
- Evaluation

Gen model vs Discriminative model

Topics for today

- Overview / quick recap on Gen AI
- connecting to OpenAI & example usage
- understanding basic prompting with OpenAI GPTs (models, endpoints, messages, deciphering response)
- Logits - recap (ML and DL) ** - ML side





Data setup

Introduction:

- four customer classes based on income and spend predictors.

Predictor Columns:

- Column 1: Income
- Column 2: Spend

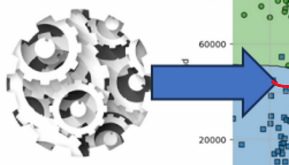
Income and Spend Ranges:

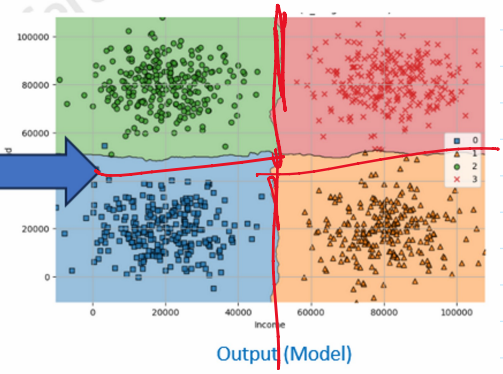
- Income: 1000 to 100000
- Spend: 1000 to 100000

Customer Classes:

- ES (Economically Stressed): Low income, Low spend
- SC (Savings Customer): High income, Low spend
- RC (Risk Customer): Low income, High spend
- PC (Prudent Customer): High income, High spend

Machine learning – Input and output





Input Data

Income	Spend	Customer Type

Learning from Data

Map {Income, Spend} to {Customer Type}

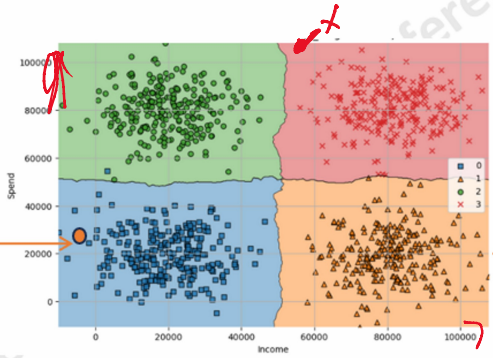
Output (Model)

Classified data using class boundaries

How the decision boundaries help?

Predict customer type of a New customer

Income	Spend
32805.917934	9034.764100



Output (Model)

Classified data using class boundaries

Prediction (customer type)

Class	target
ES	0

Predicted Class: [0]

Prediction Probabilities: [[0.6, 0.4, 0., 0.]]

i.e.
New customer is closer/ more similar to customers of ES type than other types

Actual labels vs predicted labels

Test Data :
100 samples

	Income	Spend	Class	target
251	65284.796931	16704.322641	SC	1
536	7619.595991	85279.986831	RC	2
284	90165.273152	24076.408727	SC	1
111	8401.540579	28816.735024	ES	0
700	25389.476934	82954.129496	RC	2
502	14324.214015	85275.182847	RC	2

Actual label

Compare

Number of predictions = 100

Class	target
ES	0
ES	0
ES	0
SC	1
RC	2
PC	3

Predicted label

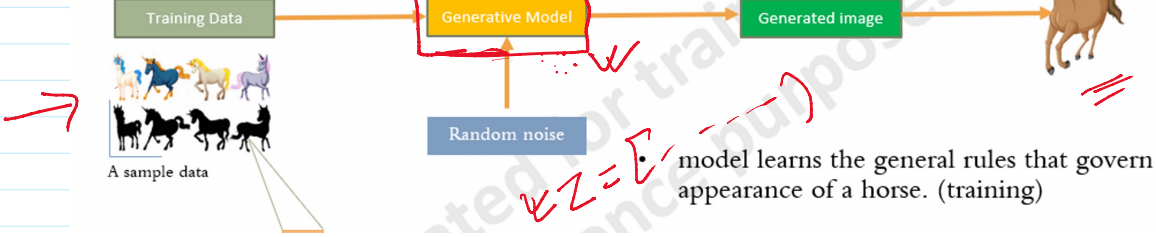
$$\text{Accuracy} = \frac{\text{Number of Correct Predictions}}{\text{Total Number of Predictions}}$$

$$\text{Accuracy} = \frac{90}{100} = 0.9$$

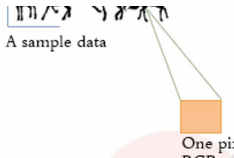
So, the accuracy of the model in this scenario is 0.9 or 90%.

GENERATIVE - EXAMPLE (IMAGES)

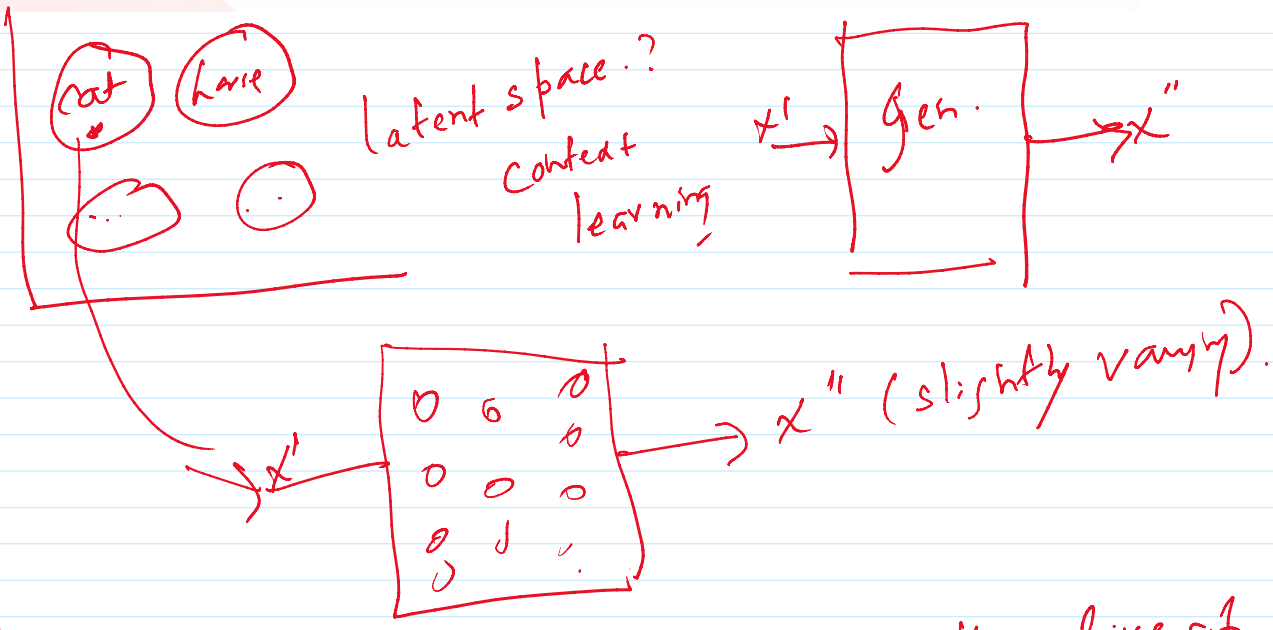
- Generative modeling can be used to generate new images that resemble a specific category, even if they have never existed before.
- For example, in the case of a dataset of horse images



- model learns the general rules that govern the appearance of a horse. (training)



- model learns the general rules that govern the appearance of a horse. (training)
- We pass some random data to the model (to predict)
- model can then generate new images of horses that look realistic based on the learned rules.



eval

x | x' good ?

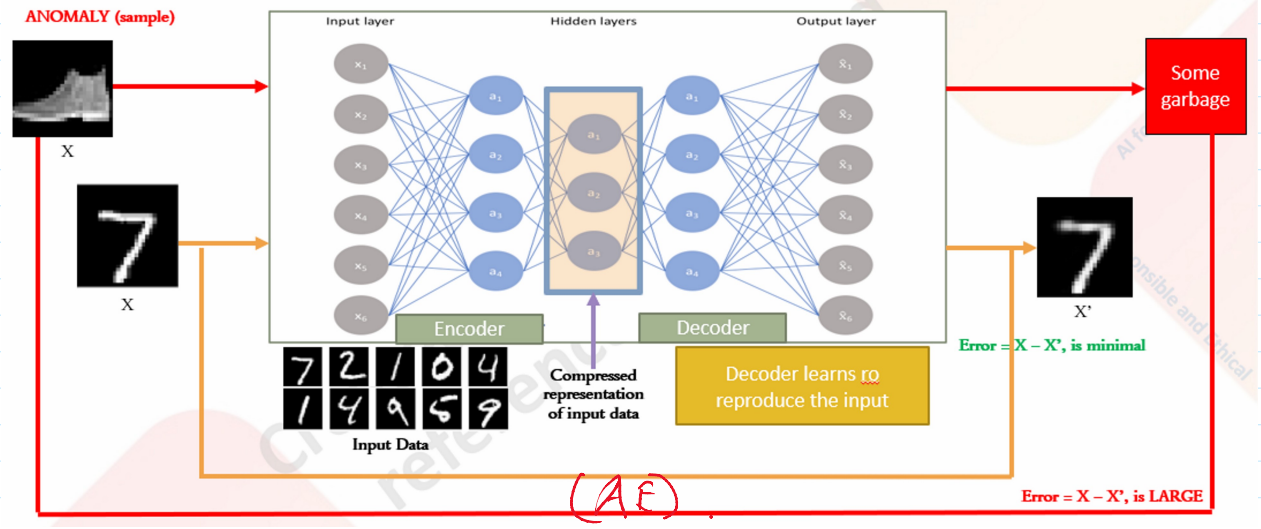
Precision Vs. diversity

FSD.

inception score.

x (text) | x' (reference)

GENERATIVE - EXAMPLE (ANOMALY DETECTION)



(AE).

cats
x

(delta)
acceptable.
(human).

cats
x'

mse
=

0.735

x'

BM
RBM.

Accuracy

x''

.935