Python Full Stack

21 Jan

`python -m venv LPU` → To create a virtual enviroment
`.\LPU\bin\activate` → To acivate the enviroment

`deactivate` → to deactivate the enviroment
`rm -r LPU` - > to remove directory recursively, all the file & sub folders
`mv oldname newname` → To rename the Folder

'm' → stands for message
'venv' → virtual enviroment

22 Jan

Wider Range → Class
Derived → Objects

Animals → Class
Lion, Tiger → Objects

'.' → Location, LEKE AAO

`python` `filename.py` → To run the file

Supervised Learning → Where we have both input and output. The eq is y = mx + c.
UnSupervised Learning → Where we don't have the output. Not label data. The model try to find the pattern.
ReEnforcement Learning → Which tries to learn from the experience. Reward based learning.

Overfitting → Where the model gives alot of accuracy on the training data but fails in testing data.
Underfitting → Which fails at the training data as well as testing data.
TradeOff → Which performs well on training data as well as testing data.

sklearn → warehouse of ML.

```
                PV
              Yes  No
            _________
                    | TP  | FN |
                Yes  |_____|____|
          AV         | FP  | TN |
                No   |_____|____|
```

```
                  PV
                No  Yes
              _________
            |  TN | FP |
      No    |_____|____|
   AV       |  FN | TP |
      Yes   |_____|____|

                  AV
                Yes  No
              _________
            |  TP | FP |
      Yes   |_____|____|
   PV       |  FN | TN |
      No    |_____|____|

                  AV
                No  Yes
              _________
            |  TN | FN |
      No    |_____|____|
   PV       |  FP | TP |
      Yes   |_____|____|
```

💡 Accuracy Score → TP+TN / TP+TN+FP+FN
Precicison → TP / TP+FP //Predicted mai kitna sahi predict kia.
Recall → TP / TP+FN //Actual mai kitna sahi predict kia.
F1 Score → 2 * (Precision * Recall / Precision + Recall)

`from sklearn.metrics import classification_marix` → Classification Matrix

23 Jan

```
from sklearn.metrics import confusion_matrixav = ['dog','dog','dog','dog','not_dog','not_dog','not_dog','not_dog']
pv = ['dog','dog','dog','dog','not_dog','not_dog','not_dog','not_dog']
print(confusion_matrix(pv,av))
```

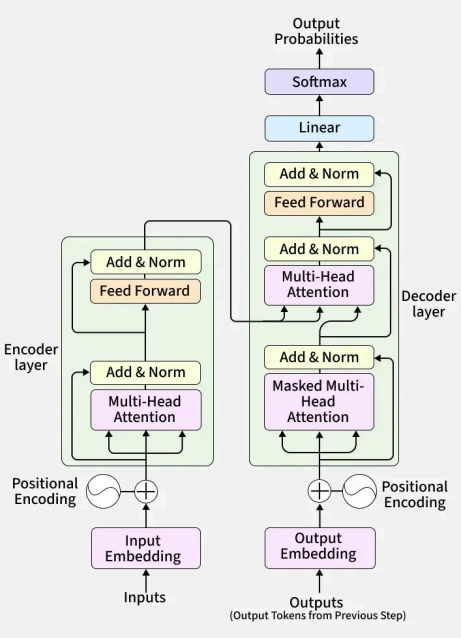
4	0
0	4

```
av = ['dog','dog','dog','dog','dog','not_dog','dog','not_dog','not_dog','not_dog']
pv = ['dog','dog','dog','dog','dog','dog','not_dog','not_dog','not_dog','not_dog']
print(confusion_matrix(pv,av))
```

5	1
1	3

"Transformers in Python" **primarily refers to the use of the powerful Hugging Face `transformers` library, which provides access to thousands of state-of-the-art pre-trained AI models for various tasks**

💡 **"Transformers in Python"** primarily refers to the use of the powerful Hugging Face `transformers` library, which provides access to thousands of state-of-the-art pre-trained AI models for various tasks



ANN	Tabular Data
CNN	Image Data
RNN	Sequential Data
ISTM	No Memory