

Python Full Stack

21 Jan

`python -m venv LPU` → To create a virtual environment

`.\LPU\bin\activate` → To activate the environment

`deactivate` → to deactivate the environment

`rm -r LPU` -> to remove directory recursively, all the files & sub folders

`mv oldname newname` → To rename the Folder

'm' → stands for message

'venv' → virtual environment

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 tries to find the pattern.

ReEnforcement Learning → Which tries to learn from the experience. Reward based learning.



Overfitting → Where the model gives a lot 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 → $\frac{TP+TN}{TP+TN+FP+FN}$

Precision → $\frac{TP}{TP+FP}$ //Predicted mai kitna sahi predict kia.

Recall → $\frac{TP}{TP+FN}$ //Actual mai kitna sahi predict kia.

F1 Score → $2 * (\text{Precision} * \text{Recall}) / (\text{Precision} + \text{Recall})$

`from sklearn.metrics import classification_matrix` → Classification Matrix

23 Jan

```
from sklearn.metrics import confusion_matrix
av = ['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))
"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

