

## TF- IDF Vectorisation Technique (Tf(t')X IDF(t'))

```
[1]: from sklearn.feature_extraction.text import TfidfVectorizer

[2]: documents = [
    'i love machine learning and nlp',
    'machine learning is always fun',
    'i love coding as well',
    'i can code my own machine learning model'
];

[3]: print('documents:',documents)

documents: ['i love machine learning and nlp', 'machine learning is always fun', 'i love coding as well', 'i can code my own machine learning mode
l']

[4]: vectorizer = TfidfVectorizer()

[5]: transformed_matrix = vectorizer.fit_transform(documents)

[6]: print('Transformed Matrix :',transformed_matrix)

Transformed Matrix : <Compressed Sparse Row sparse matrix of dtype 'float64'
with 21 stored elements and shape (4, 16)>
  Coords      Values
(0, 9)        0.4253047588435995
(0, 10)       0.34432085671547236
(0, 8)        0.34432085671547236
(0, 1)        0.5394451581794177
(0, 13)       0.5394451581794177
(1, 10)       0.3267976803045626
(1, 8)        0.3267976803045626
(1, 7)        0.5119917161748856
(1, 0)        0.5119917161748856
(1, 6)        0.5119917161748856
(2, 9)        0.41428875116588965
(2, 5)        0.5254727492640658
(2, 2)        0.5254727492640658
(2, 15)       0.5254727492640658
(3, 10)       0.2646962975587471
(3, 8)        0.2646962975587471
(3, 3)        0.41469789970950716
(3, 4)        0.41469789970950716
(3, 12)       0.41469789970950716
(3, 14)       0.41469789970950716
(3, 11)       0.41469789970950716

[8]: print('Extracting Features :',vectorizer.get_feature_names_out()) # Feature Extraction

Extracting Features : ['always' 'and' 'as' 'can' 'code' 'coding' 'fun' 'is' 'learning' 'love'
'machine' 'model' 'my' 'nlp' 'own' 'well']

[ ]: print('Vector Maxtrix Display : ',transformed_matrix.t)
```