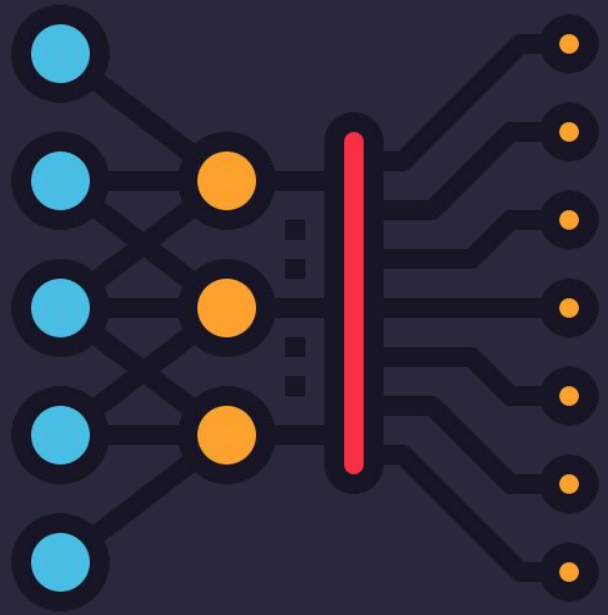
ML NLP

ШИФТ 2023



Задача: классификация текста

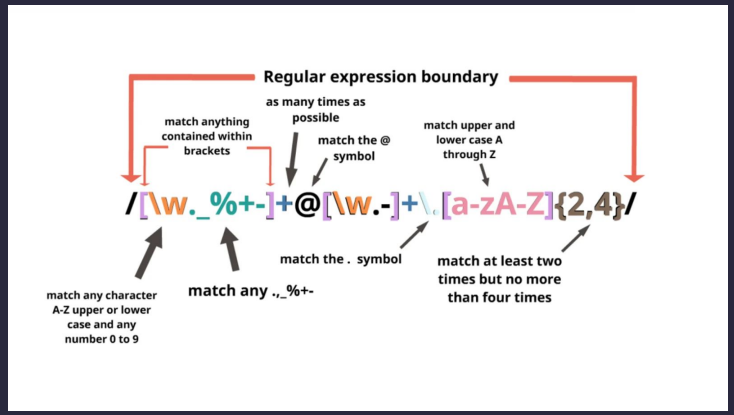
```
text = text.sample(frac=1)
text.head(5)
```

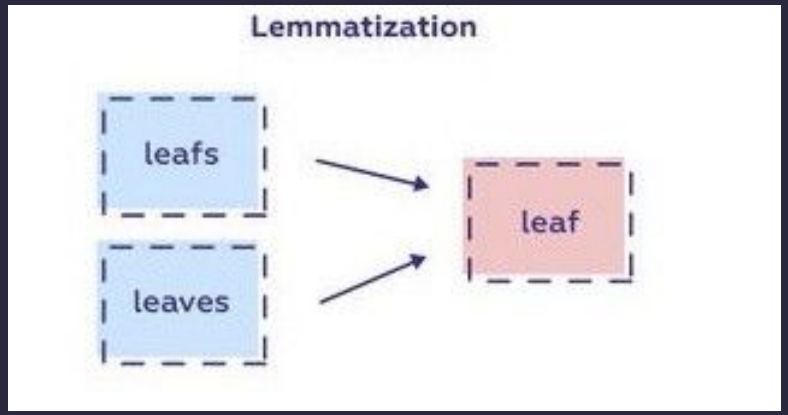
	Unnamed: 0	text	label
6404	6404	Peres demands early Israeli elections Israel #...	0
92798	92798	Phillies Keep Lidle Around for Two More Years ...	1
110782	110782	All This, and They Take Pictures, Too What wil...	3
46702	46702	Toyota: Resistance Is Futile Here #39;s a them...	2
48106	48106	Michigan's Henne heats up BLOOMINGTON, Ind. --...	1

```
text['label'].value_counts()
```

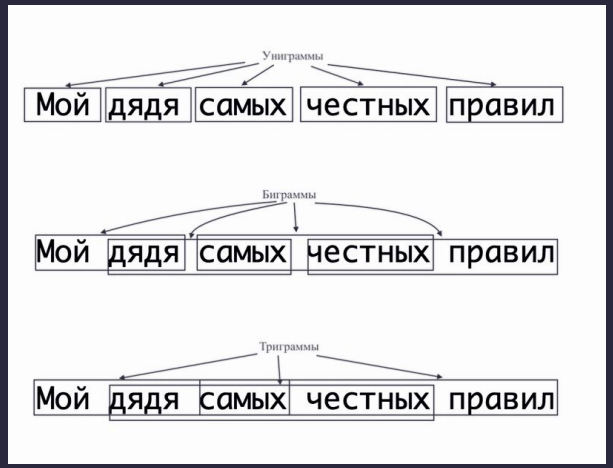
```
2    30000
3    30000
1    30000
0    30000
Name: label, dtype: int64
```

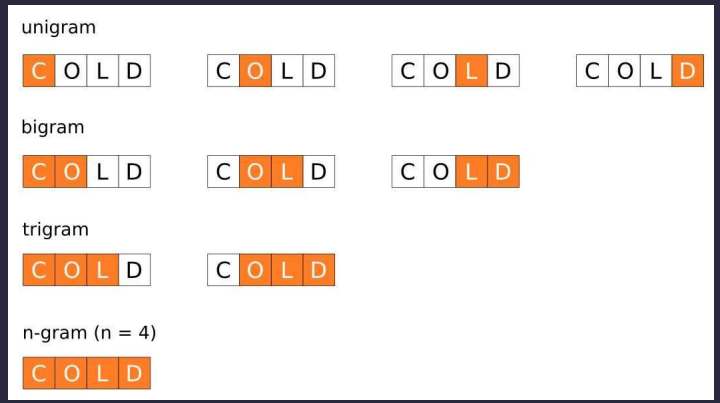
Обработка данных





N-граммы


Униграммы
Мой дядя самых честных правил
Биграммы
Мой дядя самых честных правил
Триграммы
Мой дядя самых честных правил


unigram
C O L D
C O L D
C O L D
C O L D
bigram
C O L D
C O L D
C O L D
trigram
C O L D
C O L D
n-gram (n = 4)
C O L D

Векторизация

$$w_{x,y} = tf_{x,y} \times \log\left(\frac{N}{df_x}\right)$$

TF-IDF

Term x within document y

$tf_{x,y}$ = frequency of x in y

df_x = number of documents containing x

N = total number of documents

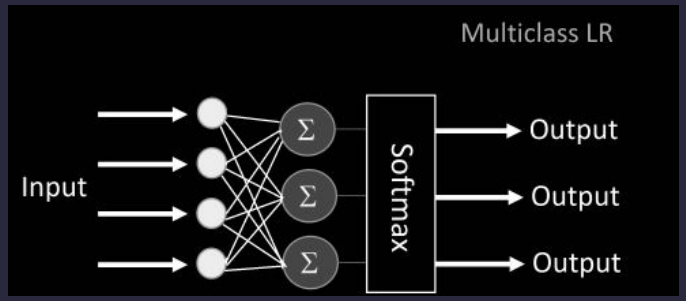
Count Vectorizer

	blue	bright	sky	sun
Doc1	1	0	1	0
Doc2	0	1	0	1

TD-IDF Vectorizer

	blue	bright	sky	sun
Doc1	0.707107	0.000000	0.707107	0.000000
Doc2	0.000000	0.707107	0.000000	0.707107

Обучение модели



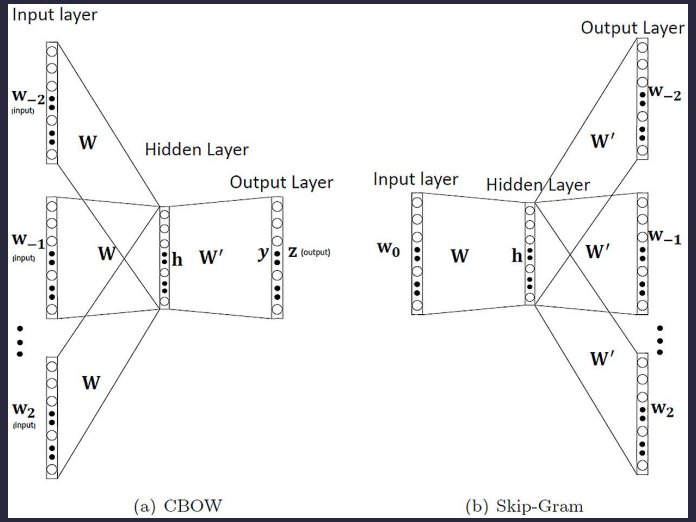

```
accuracy_score(y_test, y_pred)
```

```
0.7004444444444444
```

```
roc_auc_score(y_test, y_pred_proba, multi_class='ovr')
```

```
0.8911401736212246
```

Как улучшить? Word2Vec!



```
new_vec = wordvecs[index_dict['ipo']] + wordvecs[index_dict['usa']]
```

```
[303]: get_k_similar_words('king', dmat)

[303]: [(23811, 'ascended', 0.6263747409057054),
        (10970, 'monarch', 0.6274749884660479),
        (47798, 'gustaf', 0.6457156889185767),
        (6137, 'dancer', 0.6732183974841426),
        (621, 'throne', 0.7134248435099957),
        (45027, 'enzon', 0.7174322782321756),
        (8037, 'basheer', 0.7308761029677215),
        (9918, 'amman', 0.7324373211698257),
        (56579, 'badawi', 0.7412418577421584),
        (39065, 'norodom', 0.7475253900591954)]
```

```
for t in top_k_ind:
    print(id2tok[t.item()])
```

```
ipo
usa
flotation
adwords
conoco
initial
dreamworks
stock
skg
debut
value
incredibles
pixar
openworld
gbrowser
leaseholder
orbitz
playboy
unicom
nastech
```

Фиксируем улучшение

```
accuracy_score(y_test, y_pred)
```

```
0.7004444444444444
```

```
roc_auc_score(y_test, y_pred_proba, multi_class='ovr')
```

```
0.8911401736212246
```

→

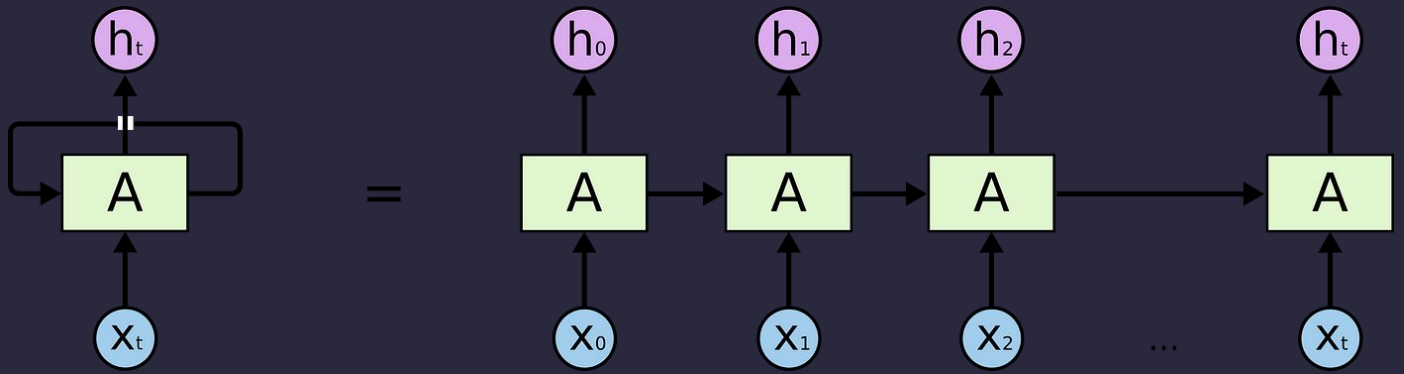
```
accuracy_score(y_test, y_pred)
```

```
0.8874
```

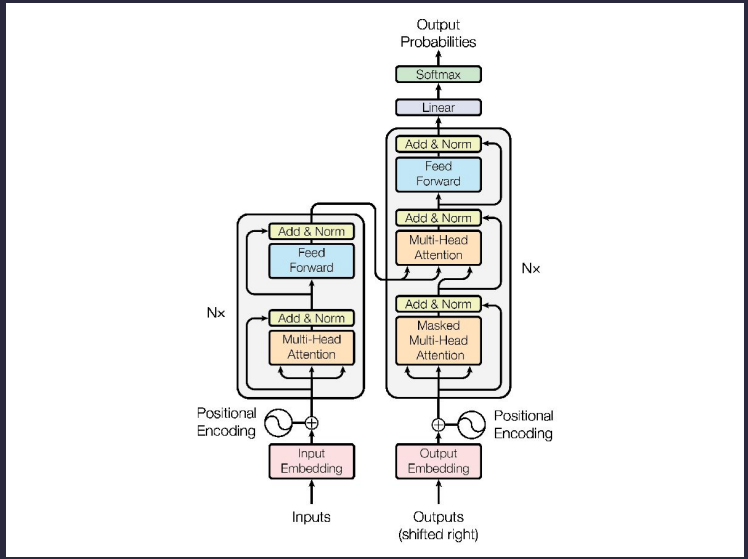
```
roc_auc_score(y_test, y_pred_proba, multi_class='ovr')
```

```
0.9761478672340616
```

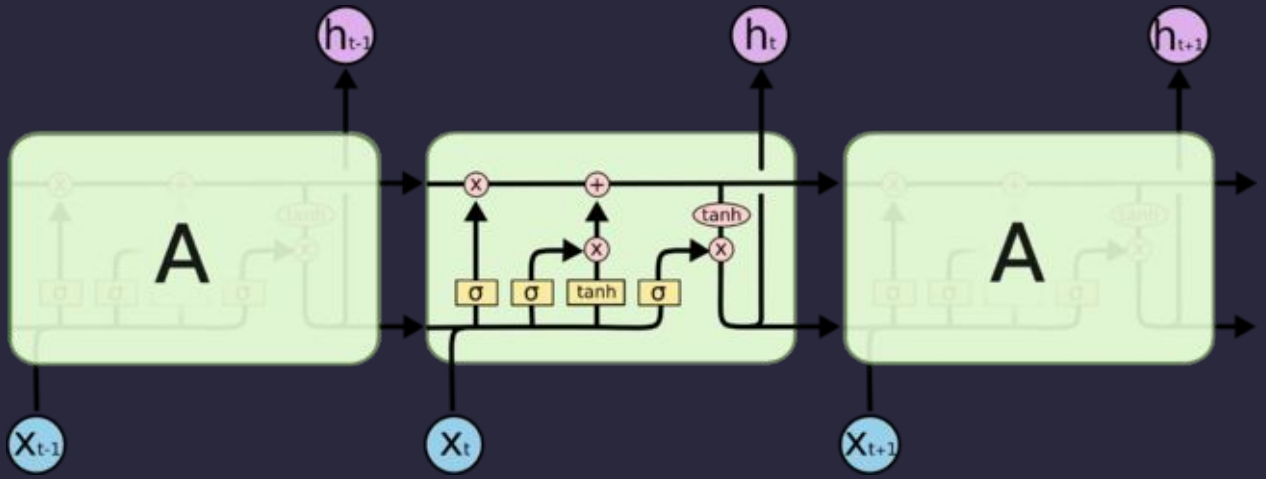
RNN



Трансформеры



Word2Vec + LSTM



```
accuracy_score(y_test, y_pred)
```

```
0.7004444444444444
```

```
accuracy_score(y_test, y_pred)
```

```
0.8874
```

```
max(all_val_acc)
```

```
0.9214583333333334
```