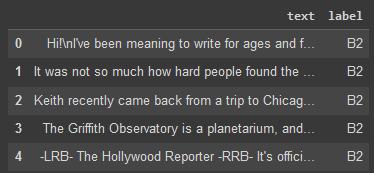
Импорт библиотек и загрузка набора данных

from sklearn.feature\_extraction.text import CountVectorizer, TfidfVectorizer  
from sklearn.model\_selection import train\_test\_split  
from sklearn.neighbors import KNeighborsRegressor, KNeighborsClassifier  
from sklearn.linear\_model import LogisticRegression  
from sklearn.metrics import accuracy\_score  
import pandas as pd  
import time

data = pd.read\_csv('cefr\_leveled\_texts.csv')

data.head()



mask = data.isna()  
props = mask.sum()  
props

text 0  
label 0  
dtype: int64

X, Y = data['text'], data['label']  
X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, Y, test\_size = 0.2, random\_state = 42)  
time\_arr = []  
  
count\_vect = CountVectorizer()  
X\_train\_counts = count\_vect.fit\_transform(X\_train)  
X\_test\_counts = count\_vect.transform(X\_test)  
  
tfidf\_vect = TfidfVectorizer()  
X\_train\_tfidf = tfidf\_vect.fit\_transform(X\_train)  
X\_test\_tfidf = tfidf\_vect.transform(X\_test)

gbc = KNeighborsClassifier()  
start\_time = time.time()  
gbc.fit(X\_train\_counts, y\_train)  
train\_time = time.time() - start\_time  
time\_arr.append(train\_time)  
pred\_gbc\_counts = gbc.predict(X\_test\_counts)  
print("Toyxocte CountVectorizer mu KNeighborsClassifier: ", accuracy\_score(y\_test, pred\_gbc\_counts))

Toyxocte CountVectorizer mu KNeighborsClassifier: 0.47491638795986624

lr = LogisticRegression(max\_iter = 1900)  
start\_time = time.time()  
lr.fit(X\_train\_counts, y\_train)  
train\_time = time.time() - start\_time  
time\_arr.append(train\_time)  
pred\_lr\_counts = lr.predict(X\_test\_counts)  
print("ToyHocte CountVectorizer m Logistic Regression:", accuracy\_score(y\_test, pred\_lr\_counts))

ToyHocte CountVectorizer m Logistic Regression: 0.6053511705685619

gbc = KNeighborsClassifier()  
start\_time = time.time()  
gbc.fit(X\_train\_tfidf, y\_train)  
train\_time = time.time() - start\_time  
time\_arr.append(train\_time)  
pred\_gbc\_tfidf = gbc.predict(X\_test\_tfidf)  
print("Touwnocts TfidfVvectorizer m KNeighborsClassifier: ", accuracy\_score(y\_test, pred\_gbc\_tfidf))

Touwnocts TfidfVvectorizer m KNeighborsClassifier: 0.4882943143812709

lr = LogisticRegression(max\_iter = 1980)  
start\_time = time.time()  
lr.fit(X\_train\_tfidf, y\_train)  
train\_time = time.time() - start\_time  
time\_arr.append(train\_time)  
pred\_lr\_tfidf = lr.predict(X\_test\_tfidf)  
print("Toyxocte TfidfVvectorizer u Logistic Regression: ", accuracy\_score(y\_test, pred\_lr\_tfidf))

Toyxocte TfidfVvectorizer u Logistic Regression: 0.5986622073578596

from tabulate import tabulate  
  
data = [  
["(CountVectorizer и LogisticRegression)", accuracy\_score(y\_test, pred\_lr\_counts), time\_arr[0]],  
["(Countvectorizer и KNeighborsClassifier)", accuracy\_score(y\_test, pred\_gbc\_counts), time\_arr[1]],  
["(1fidfvectorizer и LogisticRegression)", accuracy\_score(y\_test, pred\_lr\_tfidf), time\_arr[2]],  
["(1fidfVectorizer и KNeighborsClassifier)", accuracy\_score(y\_test, pred\_gbc\_tfidf), time\_arr[3]]  
]  
  
sorted\_data = sorted(data, key=lambda x: x[1], reverse=True)  
  
print(tabulate(sorted\_data,['KКомбинация', 'Точность', 'Время обмена'], tablefmt="grid"))

+------------------------------------------+------------+----------------+  
| KКомбинация | Точность | Время обмена |  
+==========================================+============+================+  
| (CountVectorizer и LogisticRegression) | 0.605351 | 0.00476503 |  
+------------------------------------------+------------+----------------+  
| (1fidfvectorizer и LogisticRegression) | 0.598662 | 0.00658631 |  
+------------------------------------------+------------+----------------+  
| (1fidfVectorizer и KNeighborsClassifier) | 0.488294 | 5.35414 |  
+------------------------------------------+------------+----------------+  
| (Countvectorizer и KNeighborsClassifier) | 0.474916 | 57.1838 |  
+------------------------------------------+------------+----------------+