

Липецкий государственный технический университет

Факультет автоматизации и информатики

Кафедра автоматизированных систем управления

ЛАБОРАТОРНАЯ РАБОТА №3

**по дисциплине «Прикладные интеллектуальные системы и экспертные
системы»**

«Классификация текстовых данных»

Студент

Косенков В.Д.

Группы М-ИАП-23

Руководитель

Кургасов В.В.

Доцент

Липецк 2023 г

Цель работы

Получить практические навыки решения задачи классификации текстовых данных в среде Jupiter Notebook. Научиться проводить предварительную обработку текстовых данных, настраивать параметры методов классификации и обучать модели, оценивать точность полученных моделей.

Задание кафедры

Задание:

- 1) Загрузить выборки по варианту из лабораторной работы №2
- 2) Используя GridSearchCV произвести предварительную обработку данных и настройку методов классификации в соответствии с заданием, вывести оптимальные значения параметров и результаты классификации модели (полнота, точность, f1-мера и аккуратности) с данными параметрами. Настройку проводить как на данных со стеммингом, так и на данных, на которых стемминг не применялся.
- 3) По каждому пункту работы занести в отчет программный код и результат вывода.
- 4) Оформить сравнительную таблицу с результатами классификации различными методами с разными настройками. Сделать выводы о наиболее подходящем методе классификации ваших данных с указанием параметров метода и описанием предварительной обработки данных.

Ход работы

Вариант 9

9	RF, LR, KNN
---	-------------

Рисунок 1 - Вариант для выполнения

На рисунке 2 изображен импорт библиотек для загрузки данных.

```
import warnings
from sklearn.datasets import fetch_20newsgroups
warnings.simplefilter(action='ignore', category=FutureWarning)
```

Рисунок 2 - Импорт необходимых библиотек

```
categories = ['comp.windows.x', 'talk.politics.guns', 'talk.politics.misc']
remove = ('headers', 'footers', 'quotes')

twenty_train_full = fetch_20newsgroups(subset='train', categories=categories, shuffle=True, random_state=42, remove=remove)
twenty_test_full = fetch_20newsgroups(subset='test', categories=categories, shuffle=True, random_state=42, remove=remove)
```

Рисунок 3 - Загрузка данных

Код методов анализа на рисунке 4.

✓ RF, KNN, LR

```
▶ from sklearn.tree import DecisionTreeClassifier
from sklearn.neighbors import KNeighborsClassifier
from sklearn.linear_model import LogisticRegression
from sklearn.ensemble import RandomForestClassifier
```

```
[8] stop_words = [None, 'english']
max_features_values = [100, 500, 1000, 5000, 10000]
use_idf = [True, False]
```

```
[9] rf_first = range(1, 5, 1)
    rf_second = range(5, 100, 20)

    rf_tree_max_depth = [*rf_first, *rf_second]
```

```
[10]

knn_parameters = {
    'vect__max_features': max_features_values,
    'vect__stop_words': stop_words,
    'tfidf__use_idf': use_idf,
    'clf__n_neighbors': [3, 5, 7, 9],
    'clf__metric': ['euclidean', 'manhattan'],
}

lr_parameters = {
    'vect__max_features': max_features_values,
    'vect__stop_words': stop_words,
    'tfidf__use_idf': use_idf,
    'clf__solver': ['newton-cg', 'lbfgs', 'sag', 'liblinear'],
    'clf__penalty': ['l2'],
}

rf_parameters = {
    'vect__max_features': max_features_values,
    'vect__stop_words': stop_words,
    'tfidf__use_idf': use_idf,
    'clf__n_estimators': [50, 100, 150],
    'clf__criterion': ['gini', 'entropy'],
    'clf__max_depth': rf_tree_max_depth,
}
```

Рисунок 4 – Параметры для нахождения оптимальных значений классификации

Проведем классификацию методами по варианту и после проведения обучения моделей на обучающем наборе данных рассчитаем характеристики качества классификации по каждому методу. Качество модели дерево решений (RF) для данных без применения стемминга и оптимальные для неё параметры представлены на рисунке 5.



Дерево решений (RF)

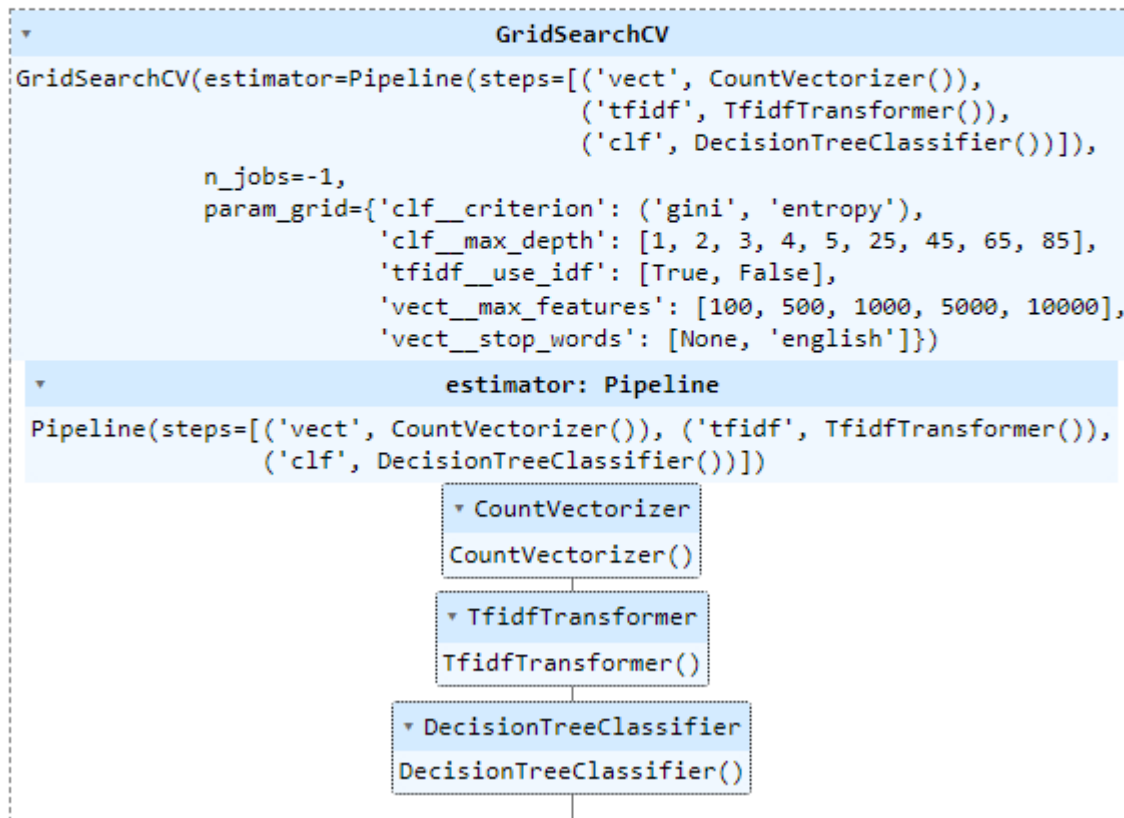
	precision	recall	f1-score	support
comp.windows.x	0.72	0.53	0.61	395
talk.politics.guns	0.43	0.69	0.53	364
talk.politics.misc	0.42	0.26	0.32	310
accuracy			0.51	1069
macro avg	0.52	0.49	0.49	1069
weighted avg	0.53	0.51	0.50	1069

```
{'clf__criterion': 'entropy', 'clf__max_depth': 25, 'tfidf__use_idf': True, 'vect__max_features': 500, 'vect__stop_words': 'english'}
```

Рисунок 5 - Дерево решений (DT) без стемминга

Дерево решений (DT) со стеммингом представлено на рисунке 6.

```
# Дерево решений (RF) со стеммингом
text_clf_dt_stem = Pipeline([('vect', CountVectorizer()),
                             ('tfidf', TfidfTransformer()),
                             ('clf', DecisionTreeClassifier())])
gscv_dt_stem = GridSearchCV(text_clf_dt_stem, param_grid=dt_parameters, n_jobs=-1)
gscv_dt_stem.fit(stem_train[:subset_size], subset_target)
```



Дерево решений (RF) со стеммингом

	precision	recall	f1-score	support
comp.windows.x	0.78	0.43	0.56	395
talk.politics.guns	0.40	0.73	0.52	364
talk.politics.misc	0.39	0.24	0.30	310
accuracy			0.48	1069
macro avg	0.52	0.47	0.46	1069
weighted avg	0.54	0.48	0.47	1069

```
{'clf__criterion': 'entropy', 'clf__max_depth': 85, 'tfidf__use_idf': False, 'vect__max_features': 5000, 'vect__stop_words': 'english'}
```

Рисунок 6 - Дерево решений (RF) со стеммингом представлено на рисунке 6.

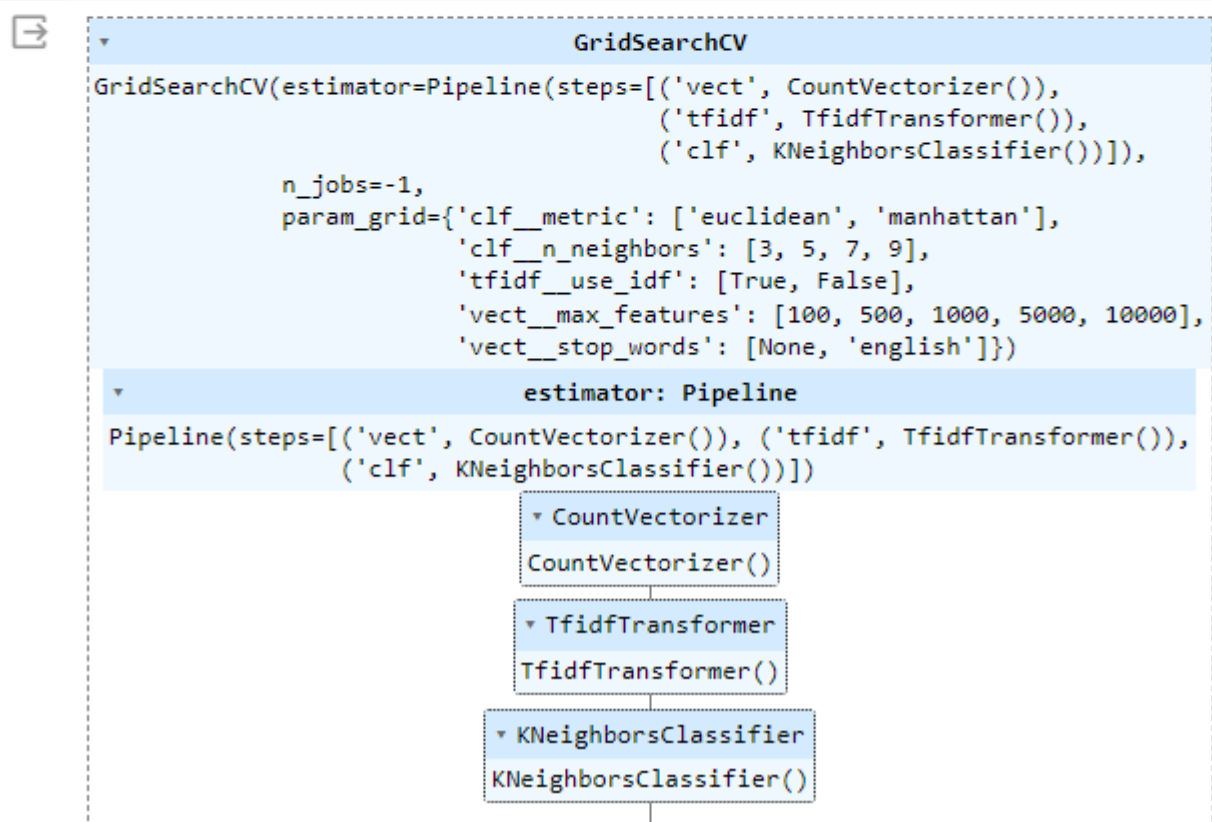
Качество модели К-ближайших соседей (KNN) для данных без применения стемминга и оптимальные для неё параметры представлены на рисунке 7.

✓ К-ближайших соседей (KNN):

- количество ближайших соседей, • метрика (евклидова, городских кварталов)

✓ Без использования стэмминга

```
# К-ближайших соседей (KNN) без стемминга
text_clf_knn = Pipeline([('vect', CountVectorizer()),
                        ('tfidf', TfidfTransformer()),
                        ('clf', KNeighborsClassifier())])
gscv_knn = GridSearchCV(text_clf_knn, param_grid=knn_parameters, n_jobs=-1)
gscv_knn.fit(subset_data, subset_target)
```



К-ближайших соседей (KNN)

	precision	recall	f1-score	support
comp.windows.x	0.91	0.89	0.90	395
talk.politics.guns	0.54	0.93	0.68	364
talk.politics.misc	0.81	0.14	0.24	310
accuracy			0.69	1069
macro avg	0.76	0.66	0.61	1069
weighted avg	0.76	0.69	0.64	1069

{'clf__metric': 'euclidean', 'clf__n_neighbors': 9, 'tfidf__use_idf': True, 'vect__max_features': 10000, 'vect__stop_words': 'english'}

Рисунок 7 - К-ближайших соседей (KNN) без стэмминга

Теперь К-ближайших соседей (KNN) со стэммингом, это показано на рисунке 8.



Рисунок 8 - К-ближайших соседей (KNN) со стэммингом

Качество модели Логистическая регрессия (LR) для данных без применения стемминга и оптимальные для неё параметры представлены на рисунке 9.


```
# Логистическая регрессия (LR) без стемминга
text_clf_lr = Pipeline([('vect', CountVectorizer()),
                        ('tfidf', TfidfTransformer()),
                        ('clf', LogisticRegression())])
gscv_lr = GridSearchCV(text_clf_lr, param_grid=lr_parameters, n_jobs=-1)
gscv_lr.fit(subset_data, subset_target)
```



Логистическая регрессия (LR)

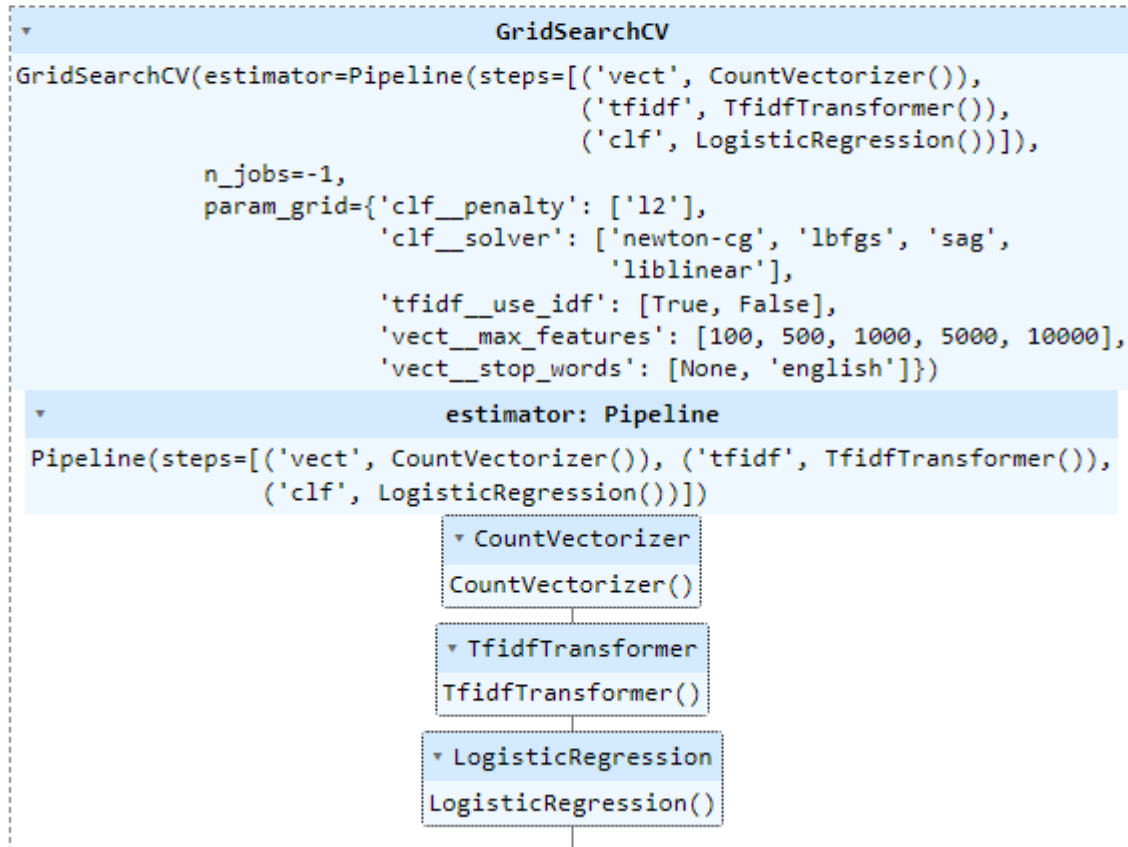
	precision	recall	f1-score	support
comp.windows.x	0.78	0.97	0.86	395
talk.politics.guns	0.61	0.77	0.68	364
talk.politics.misc	0.73	0.27	0.40	310
accuracy			0.70	1069
macro avg	0.71	0.67	0.65	1069
weighted avg	0.71	0.70	0.67	1069

```
{'clf__penalty': 'l2', 'clf__solver': 'newton-cg', 'tfidf__use_idf': False, 'vect__max_features': 5000, 'vect__stop_words': 'english'}
```

Рисунок 9 - Логистическая регрессия (LR) без стэмминга

Логистическая регрессия (LR) без стэмминга показана на рисунке 10.

```
# Логистическая регрессия (LR) со стеммингом
text_clf_lr_stem = Pipeline([('vect', CountVectorizer()),
                             ('tfidf', TfidfTransformer()),
                             ('clf', LogisticRegression())])
gscv_lr_stem = GridSearchCV(text_clf_lr_stem, param_grid=lr_parameters, n_jobs=-1)
gscv_lr_stem.fit(stem_train[:subset_size], subset_target)
```



Логистическая регрессия (LR) со стеммингом

	precision	recall	f1-score	support
comp.windows.x	0.68	0.98	0.80	395
talk.politics.guns	0.60	0.71	0.65	364
talk.politics.misc	0.76	0.17	0.28	310
accuracy			0.65	1069
macro avg	0.68	0.62	0.58	1069
weighted avg	0.68	0.65	0.60	1069

```
{'clf__penalty': 'l2', 'clf__solver': 'newton-cg', 'tfidf__use_idf': True, 'vect__max_features': 5000, 'vect__stop_words': 'english'}
```

Рисунок 10 - Логистическая регрессия (LR) со стэммингом

Вывод

В результате выполнения работы получены практические навыки обработки текстовых данных в среде Jupiter Notebook. Проведена предварительная обработка текстовых данных и выявлены параметры обработки, позволяющие добиться наилучшей точности классификации.