Липецкий государственный технический университет

Факультет автоматизации и информатики

Кафедра автоматизированных систем управления

ЛАБОРАТОРНАЯ РАБОТА №4

по дисциплине «Прикладные интеллектуальные системы и экспертные системы»

Классификация текстовых данных

Студент Мамедов Р. В.

Группа М-ИАП-23-1

Руководитель Кургасов В.В.

доцент, канд. пед. наук

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

1) Загрузить выборки по варианту из лабораторной работы №12.

2) Используя GridSearchCV произвести предварительную обработку данных и настройку методов классификации в соответствие 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.

Вариант №12

Классы RF, SVM, LR

Случайный лес (RF):

• количество деревьев решений,

• критерий (параметр criterion: ‘gini’, ‘entropy’),

• глубина дерева (параметр max_depth от 1 до 5 с шагом 1, далее до 100 с шагом 20).

Логистическая регрессия (LR):

• метод нахождения экстремума (параметр solver: ‘newton-cg’, ‘lbfgs’, ‘sag’, ‘liblinear’),

• регуляризация (параметр penalty: ‘L1’, ‘L2’)

Обратить внимание, что разные виды регуляризации работают с разными методами нахождения экстремума.

Метод опорных векторов (SVM):

• функция потерь (параметр loss: ‘hinge’, ‘squared_hinge’),

• регуляризация (параметр penalty: ‘L1’, ‘L2’)

Обратить внимание, что разные виды регуляризации работают с разными функциями потерь.

Ход работы

Загрузим обучающую и тестовую выборку в соответствии с вариантом.

```
categories = ['comp.graphics', 'sci.crypt', 'sci.electronics']
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)
```

Зададим параметры, которые будем варьировать, чтобы найти наиболее оптимальные.

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

rf_first = range(1, 5, 1)
rf_second = range(5, 100, 20)

rf_tree_max_depth = [*rf_first, *rf_second]

parameters_rf = {
    'vect__max_features': max_features_values,
    'vect__stop_words': stop_words,
    'tfidf__use_idf': use_idf,
    'clf__n_estimators': range(1, 10, 1),
    'clf__criterion': ('gini', 'entropy'),
    'clf__max_depth': rf_tree_max_depth,
}

parameters_lr = {
    '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']
}

parameters_lr_l1 = {
    'vect__max_features': max_features_values,
```

```
    'vect__stop_words': stop_words,
    'tfidf__use_idf': use_idf,
    'clf__solver': ['liblinear'],  # Используем только 'liblinear' для l1
    'clf__penalty': ['l1'],
}

parameters_svm = {
    'vect__max_features': max_features_values,
    'vect__stop_words': stop_words,
    'tfidf__use_idf': use_idf,
}
```

Проведем классификацию методами RF, LR и SVM. После проведения обучения моделей на обучающем наборе данныхрассчитаем характеристики качества классификации по каждому методу.

Качество модели случайного леса для данныхm без применения стемминга и оптимальные для неё параметры представлены на рисунке 1.

Случайный лес (RF) без стемминга

	precision	recall	f1-score	support
comp.graphics	0.76	0.79	0.77	389
sci.crypt	0.86	0.70	0.77	396
sci.electronics	0.64	0.74	0.69	393
accuracy			0.74	1178
macro avg	0.75	0.74	0.74	1178
weighted avg	0.75	0.74	0.74	1178

Рисунок 1 – Результаты работы классификации методом случайного леса без применения стемминга

Случайный лес (RF) со стеммингом

	precision	recall	f1-score	support
comp.graphics	0.67	0.65	0.66	389
sci.crypt	0.87	0.58	0.69	396
sci.electronics	0.53	0.73	0.62	393
accuracy			0.65	1178
macro avg	0.69	0.65	0.66	1178
weighted avg	0.69	0.65	0.66	1178

Рисунок 2 – Результаты работы классификации методом случайного леса с применением стемминга

Качество модели логистической регрессии для данных без применения стемминга и оптимальные для неё параметры представлены на рисунке 3.

Логистическая регрессия (LR) без стемминга

	precision	recall	f1-score	support
comp.graphics	0.80	0.79	0.80	389
sci.crypt	0.89	0.69	0.78	396
sci.electronics	0.67	0.82	0.74	393
accuracy			0.77	1178
macro avg	0.79	0.77	0.77	1178
weighted avg	0.79	0.77	0.77	1178

Рисунок 3 - Качество модели линейной регрессии для данных без применения стемминга и оптимальные для неё параметры

Логистическая регрессия (LR) без стемминга

	precision	recall	f1-score	support
comp.graphics	0.82	0.87	0.85	389
sci.crypt	0.90	0.76	0.83	396
sci.electronics	0.75	0.83	0.79	393
accuracy			0.82	1178
macro avg	0.83	0.82	0.82	1178
weighted avg	0.83	0.82	0.82	1178

Рисунок 4 - Качество модели линейной регрессии для данных с применением стемминга и оптимальные для неё параметры

Качество модели метода опорных векторов L1 для данных без применениястемминга и оптимальные для неё параметры представлены на рисунке 5.

```
                precision    recall  f1-score   support

  comp.graphics       0.83      0.86      0.85       389
      sci.crypt       0.93      0.74      0.83       396
sci.electronics       0.74      0.86      0.79       393

       accuracy                           0.82      1178
      macro avg       0.83      0.82      0.82      1178
   weighted avg       0.83      0.82      0.82      1178

{'tfidf__use_idf': True, 'vect__max_features': 5000, 'vect__stop_words': 'english'}
```

Рисунок 5 – Качество модели метода опорных векторов для данных без применения стемминга и оптимальные для неё параметры

```
Метод опорных векторов (SVM) со стеммингом

                precision    recall  f1-score   support

  comp.graphics       0.79      0.81      0.80       389
      sci.crypt       0.87      0.67      0.76       396
sci.electronics       0.67      0.81      0.73       393

       accuracy                           0.76      1178
      macro avg       0.78      0.76      0.76      1178
   weighted avg       0.78      0.76      0.76      1178
```

Рисунок 6 – Качество модели метода опорных векторов для данных с применением стемминга и оптимальные для неё параметры

Вывод

В результате выполнения данной лабораторной работы я получил практические навыки решения задачи классификации текстовых данных в среде Jupiter Notebook.

Также научился проводить предварительную обработку текстовых данных, настраивать параметры методов классификации и обучать модели, оценивать точность полученных моделей.

Мною были применены следующие методы: случайного леса (RF), Логистической регрессии (LR) и метод опорных векторов (SVM).

Наилучшей точностью классификации для данного набора данных обладают модели с методом опорных векторов без и с применением стемминга. Их точность составляет 82%. Параметры для данных моделей представлены соответственно на рисунке 5.