Московский Государственный Технический Университет имени Н.Э. Баумана

Факультет Информатика и системы управления

Кафедра ИУ-5 «Системы обработки информации и управления»

Отчёт по рубежному контролю № 2

По дисциплине

«Методы Машинного Обучения»

Выполнил студент Ли Хао
Группа ИУ5И-23М

Москва 2024г

Необходимо решить задачу классификации текстов на основе любого выбранного Вами датасета (кроме примера, который рассматривался в лекции). Классификация может быть бинарной или многоклассовой. Целевой признак из выбранного Вами датасета может иметь любой физический смысл, примером является задача анализа тональности текста.

Необходимо сформировать два варианта векторизации признаков - на основе CountVectorizer и на основе TfidfVectorizer.

В качестве классификаторов необходимо использовать два классификатора по варианту для группы:

LinearSVC и LogisticRegression

```python
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.feature_extraction.text import CountVectorizer, TfidfVectorizer
from sklearn.svm import LinearSVC
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import classification_report

# 人工数据
data = {
    'text': [
        'This is a great movie', 'Terrible film, would not recommend',
        'Loved the cinematography', 'The plot was very boring',
        'Amazing acting by the lead', 'Not worth watching',
        'Fantastic storyline and performances', 'The movie was too slow',
        'Brilliant direction', 'Waste of time'
    ],
    'label': [1, 0, 1, 0, 1, 0, 1, 0, 1, 0]
}

df = pd.DataFrame(data)

# 将数据分为训练集和测试集
X_train, X_test, y_train, y_test = train_test_split(df['text'], df['label'], test_size=0.3, random_state=42)

# 使用 CountVectorizer 进行特征向量化
count_vectorizer = CountVectorizer()
X_train_counts = count_vectorizer.fit_transform(X_train)
X_test_counts = count_vectorizer.transform(X_test)

# 使用 TfidfVectorizer 进行特征向量化
tfidf_vectorizer = TfidfVectorizer()
X_train_tfidf = tfidf_vectorizer.fit_transform(X_train)
X_test_tfidf = tfidf_vectorizer.transform(X_test)
```

```
# 使用 CountVectorizer 进行模型训练和评估

# LinearSVC
svc_count = LinearSVC()
svc_count.fit(X_train_counts, y_train)
y_pred_svc_count = svc_count.predict(X_test_counts)
print("LinearSVC 使用 CountVectorizer:")
print(classification_report(y_test, y_pred_svc_count))

# LogisticRegression
lr_count = LogisticRegression(max_iter=1000)
lr_count.fit(X_train_counts, y_train)
y_pred_lr_count = lr_count.predict(X_test_counts)
print("LogisticRegression 使用 CountVectorizer:")
print(classification_report(y_test, y_pred_lr_count))

# 使用 TfidfVectorizer 进行模型训练和评估

# LinearSVC
svc_tfidf = LinearSVC()
svc_tfidf.fit(X_train_tfidf, y_train)
y_pred_svc_tfidf = svc_tfidf.predict(X_test_tfidf)
print("LinearSVC 使用 TfidfVectorizer:")
print(classification_report(y_test, y_pred_svc_tfidf))

# LogisticRegression
lr_tfidf = LogisticRegression(max_iter=1000)
lr_tfidf.fit(X_train_tfidf, y_train)
y_pred_lr_tfidf = lr_tfidf.predict(X_test_tfidf)
print("LogisticRegression 使用 TfidfVectorizer:")
print(classification_report(y_test, y_pred_lr_tfidf))
```

```
LinearSVC 使用 CountVectorizer:
              precision    recall  f1-score   support

           0       0.00      0.00      0.00         2
           1       0.33      1.00      0.50         1

    accuracy                           0.33         3
   macro avg       0.17      0.50      0.25         3
weighted avg       0.11      0.33      0.17         3

LogisticRegression 使用 CountVectorizer:
              precision    recall  f1-score   support

           0       0.00      0.00      0.00         2
           1       0.33      1.00      0.50         1

    accuracy                           0.33         3
   macro avg       0.17      0.50      0.25         3
weighted avg       0.11      0.33      0.17         3

LinearSVC 使用 TfidfVectorizer:
              precision    recall  f1-score   support

           0       0.00      0.00      0.00         2
           1       0.33      1.00      0.50         1

    accuracy                           0.33         3
   macro avg       0.17      0.50      0.25         3
weighted avg       0.11      0.33      0.17         3

LogisticRegression 使用 TfidfVectorizer:
              precision    recall  f1-score   support

           0       0.00      0.00      0.00         2
           1       0.33      1.00      0.50         1

    accuracy                           0.33         3
   macro avg       0.17      0.50      0.25         3
weighted avg       0.11      0.33      0.17         3
```