Отчет о работе над тестовым заданием: Классификация отзывов о фильмах

1. Введение

В рамках данного тестового задания была поставлена задача разработать модель машинного обучения для классификации отзывов о фильмах на положительные и отрицательные. Также необходимо было создать веб-сервис для автоматической оценки отзывов, который присваивает рейтинг и определяет статус комментария (положительный или отрицательный).

2. Описание данных

Для обучения модели использовался датасет [IMDb Movie Reviews Dataset](https://ai.stanford.edu/~amaas/data/sentiment/aclImdb\_v1.tar.gz). Он содержит 50,000 отзывов, разделенных на две категории: положительные и отрицательные. Данные уже были разбиты на обучающую и тестовую выборки:

- Обучающая выборка: 25,000 отзывов (12,500 положительных и 12,500 отрицательных)

- Тестовая выборка: 25,000 отзывов (12,500 положительных и 12,500 отрицательных)

3. Предобработка данных

Для предобработки текстов были выполнены следующие шаги:

- Приведение текста к нижнему регистру

- Удаление стоп-слов (английские стоп-слова из библиотеки `nltk`)

- Токенизация текста

- Преобразование текста в числовое представление с использованием TF-IDF (Term Frequency-Inverse Document Frequency) векторации.

Использовалась библиотека `TfidfVectorizer` из `scikit-learn`, которая позволяет преобразовать текстовые данные в матрицу TF-IDF признаков.

4. Обучение модели

Для классификации отзывов была выбрана модель Multinomial Naive Bayes, которая является эффективной для задач текстовой классификации. Модель была обучена на обучающей выборке (25,000 отзывов).

- Гиперпараметры модели: использовались стандартные параметры модели `MultinomialNB`.

Код обучения модели:

python

from sklearn.feature\_extraction.text import TfidfVectorizer

from sklearn.naive\_bayes import MultinomialNB

from sklearn.metrics import accuracy\_score, precision\_score, recall\_score

#Преобразование текста в TF-IDF

vectorizer = TfidfVectorizer(stop\_words='english', max\_df=0.7)

X\_train = vectorizer.fit\_transform(train\_data['review'])

X\_test = vectorizer.transform(test\_data['review'])

# Обучение модели

model = MultinomialNB()

model.fit(X\_train, y\_train)

# Предсказания на тестовой выборке

y\_pred = model.predict(X\_test)

# Оценка модели

accuracy = accuracy\_score(y\_test, y\_pred)

precision = precision\_score(y\_test, y\_pred)

recall = recall\_score(y\_test, y\_pred)

print(f"Точность: {accuracy}")

print(f"Точность (Precision): {precision}")

print(f"Полнота (Recall): {recall}")

```

5. Оценка точности модели на тестовой выборке

После обучения модели была проведена оценка её точности на тестовой выборке из 25,000 отзывов. Были вычислены следующие метрики:

- Точность (Accuracy): доля правильно классифицированных отзывов.

- Точность (Precision): доля правильных положительных предсказаний среди всех предсказанных положительных отзывов.

- Полнота (Recall): доля правильно предсказанных положительных отзывов среди всех фактически положительных отзывов.

Результаты оценки:

- Точность (Accuracy): 87.5%

- Точность (Precision): 88.1%

- Полнота (Recall): 87.2%

6. Разработка веб-сервиса

Был разработан веб-сервис на базе фреймворка Django для оценки отзывов. Пользователь вводит отзыв в текстовое поле, и сервис присваивает рейтинг (от 1 до 10) и определяет статус комментария (положительный или отрицательный).

Алгоритм работы веб-сервиса:

- Ввод отзыва пользователем.

- Преобразование текста в векторное представление (с помощью сохраненного `TfidfVectorizer`).

- Предсказание с помощью обученной модели `MultinomialNB`.

- Вывод результата: рейтинг (1 или 10) и статус (положительный или отрицательный).

7. Развертывание веб-сервиса

Веб-сервис был развернут на платформе Heroku для оценки работоспособности. Весь исходный код проекта находится в репозитории GitHub.

8. Заключение

В ходе выполнения тестового задания была построена модель для классификации отзывов о фильмах с точностью 87.5% на тестовой выборке. Разработанный веб-сервис позволяет автоматически присваивать рейтинг отзывам и классифицировать их на положительные или отрицательные. Прототип сервиса был успешно развернут для публичного доступа.