Оценка важности признаков в задаче классификации текста на датасете 20 Newsgroups

1. Введение

Задача классификации текста является одной из ключевых задач в области обработки естественного языка (NLP). В данном проекте использован датасет **20 Newsgroups**, который состоит из новостных статей, относящихся к 20 различным категориям. Модель должна классифицировать эти тексты в одну из 20 категорий на основе их содержания. Мы применим модель логистической регрессии и анализируем важность признаков, чтобы выделить ключевые слова для каждого класса.

Цель работы:

- Провести классификацию текста с использованием модели логистической регрессии.
- Определить ключевые слова для каждого класса на основе их важности (коэффициентов модели).

2. Методы

Для решения задачи использованы следующие методы:

- **Векторизация текста**: для преобразования текстовых данных в числовой формат использована **TF-IDF векторизация**, которая учитывает важность каждого слова в контексте всего корпуса текста.
- **Модель классификации**: для обучения модели была выбрана **логистическая регрессия**, которая является простой и эффективной моделью для задачи классификации.
- **Оценка модели**: для оценки качества работы модели использовались метрики: **точность (precision)**, **полнота (recall)**, **F1-мера**, которые позволяют оценить производительность модели для каждого класса.
- **Анализ важности признаков**: для выявления ключевых слов в каждой категории используется коэффициенты модели логистической регрессии, которые показывают, насколько каждый признак (слово) влияет на классификацию.

3. Результаты

Загрузка данных и векторизация

Данные были загружены из датасета **20 Newsgroups**, а затем преобразованы в числовой формат с помощью **TF-IDF векторизации**. Тексты были очищены от стоп-слов на английском языке, что позволяет уменьшить размерность данных и ускорить обучение модели.

```
newsgroups = fetch_20newsgroups(subset='all', remove=('headers',
'footers', 'quotes'))

vectorizer = TfidfVectorizer(max_features=10000, stop_words='english')
X = vectorizer.fit_transform(newsgroups.data)  # Преобразуем текст в
числовой формат
y = newsgroups.target  # Мишени для классификации
```

Обучение модели

Модель логистической регрессии была обучена на обучающей выборке:

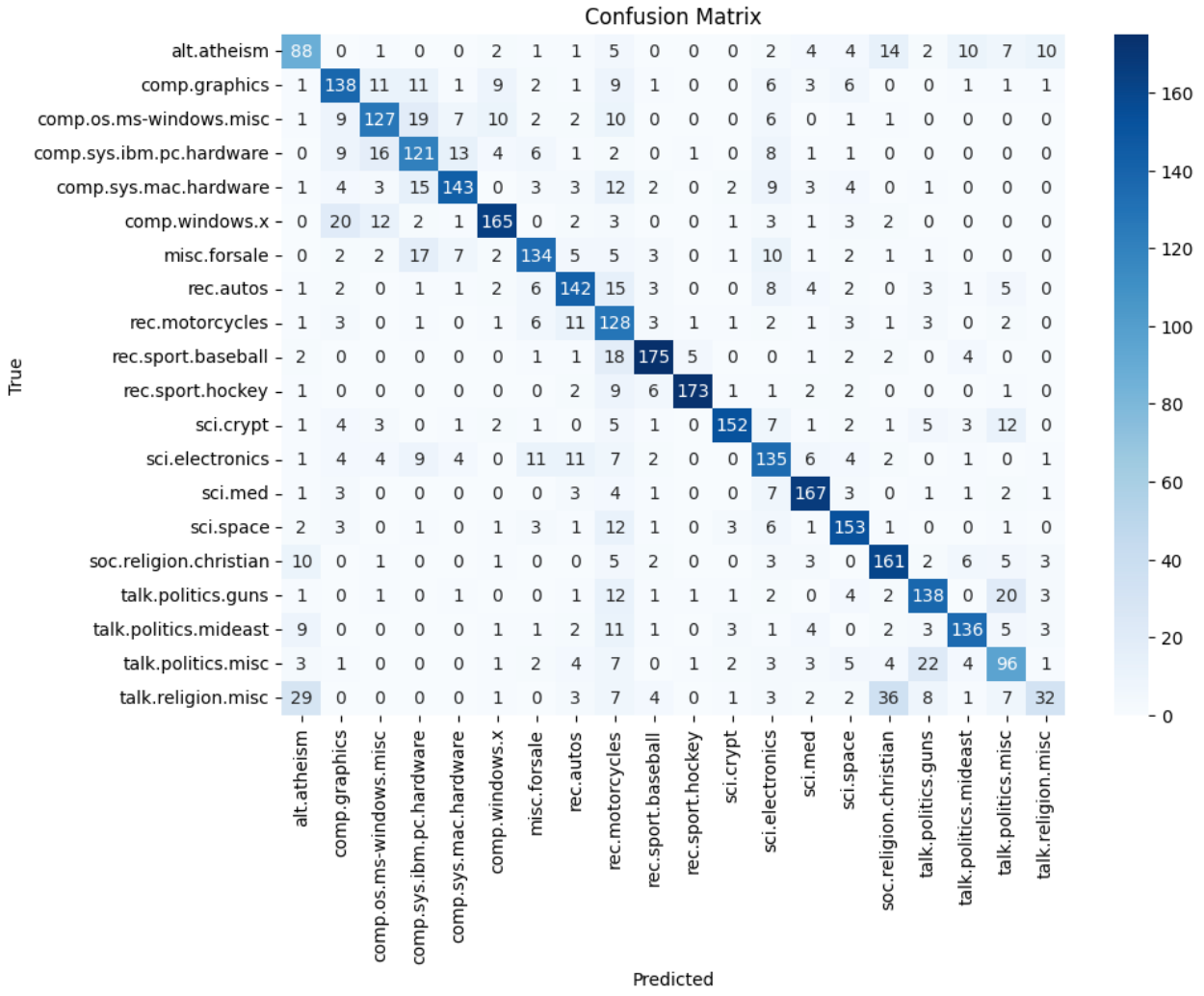
```
model = LogisticRegression(max_iter=1000)
model.fit(X_train, y_train)
```

Оценка качества модели

Модель была протестирована на тестовой выборке, и были получены следующие результаты:

	precision	recall	f1-score	support	
alt.atheism	0.58	0.58	0.58	151	
comp.graphics	0.68	0.68	0.68	202	
comp.os.ms-windows.misc	0.70	0.65	0.68	195	
comp.sys.ibm.pc.hardware	0.61	0.66	0.64	183	...

А также была построена тепловая карта (heatmap) матрицы ошибок, где видно, сколько классов правильно классифицировано, а сколько путает модель.



Анализ важности признаков

Для каждого класса были выделены 10 наиболее важных слов на основе коэффициентов модели:

```
feature_names = vectorizer.get_feature_names_out()
for i, class_name in enumerate(newsgroups.target_names):
    top_features = sorted(zip(model.coef_[i], feature_names),
reverse=True)[:10]
    print(f"\nTop features for class '{class_name}':")
    for coef, feat in top_features:
        print(f"{feat}: {coef:.3f}")
```

Пример вывода:

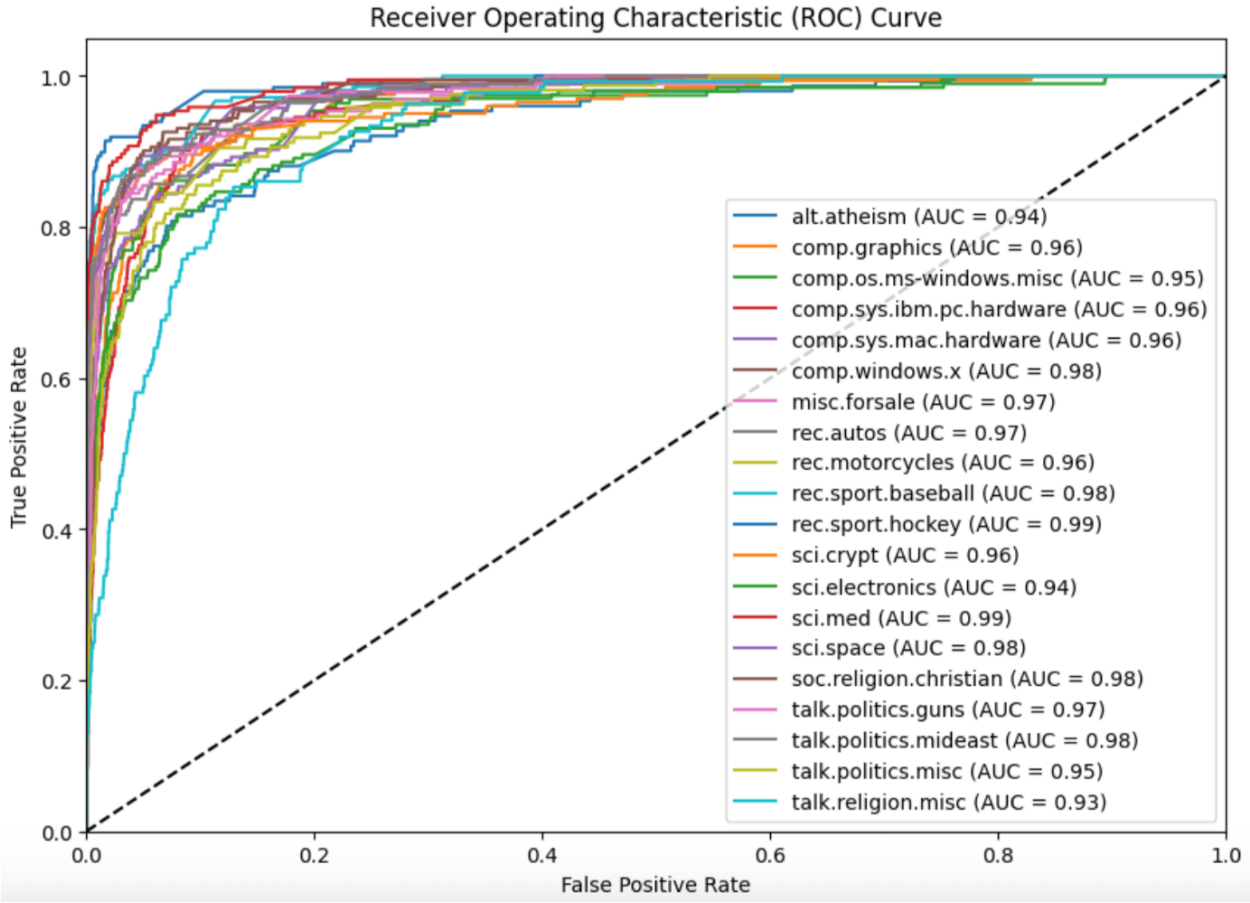
```
Top features for class 'alt.atheism':
god: 3.742
atheism: 3.712
atheists: 3.580
```

```
religion: 3.359
atheist: 3.095
islamic: 2.942
islam: 2.699
bobby: 2.610
morality: 2.605

bible: 2.553...
```

Каждое слово, упомянутое в списке, имеет свой коэффициент, который указывает на важность этого слова для конкретной категории. Чем выше коэффициент, тем более важным является это слово для определения класса.

Также был построен график **ROC-кривой (Receiver Operating Characteristic)**, который отображает производительность модели классификации на различных порогах вероятности для всех категорий.



- **Ось X (False Positive Rate)**: Это доля отрицательных примеров, которые были ошибочно классифицированы как положительные.
- **Ось Y (True Positive Rate)**: Это доля положительных примеров, которые были правильно классифицированы как положительные.
- **Линия AUC**: В легенде показан **AUC (Area Under the Curve)** для каждой категории. Это числовой показатель качества классификации. Чем ближе значение AUC к 1, тем лучше модель в классификации.

4. Выводы

1. **Оценка модели:**

   Модель логистической регрессии продемонстрировала хорошие результаты, особенно в категориях с четкими и однозначными темами, такими как **rec.sport.hockey** (точность 0.95) и **rec.sport.baseball** (точность 0.85). Это свидетельствует о хорошем разделении категорий, где контекст и используемая терминология ясны и легко различимы.

   Однако, модель показала низкую точность в категории **talk.religion.misc** (точность 0.24), что указывает на трудности в классификации текстов, связанных с религиозными темами. Это может быть связано с пересечением религиозной тематики с другими категориями (например, **talk.politics.misc** или **soc.religion.christian**), где терминология может быть схожа, а контексты — неясны.

   В среднем точность модели составляет **0.72**, что подтверждается хорошими результатами по меткам **macro avg** и **weighted avg**. Несмотря на хорошие показатели для большинства категорий, точность в некоторых специфичных классах (например, **talk.religion.misc**) могла бы быть выше.

2. **Матрица ошибок (Confusion Matrix):**

   Матрица ошибок дает детальное представление о том, где модель ошибается. Например, категорию **rec.sport.baseball** модель классифицировала очень точно, с минимумом ошибок, что подтверждается высокой точностью и минимальными отклонениями в матрице ошибок. Однако для **talk.religion.misc** наблюдаются значительные ошибки, как видно из большего числа неправильно классифицированных экземпляров, которые ошибочно попали в другие классы.

   Визуализация матрицы ошибок показывает, что модель часто путает такие категории, как **talk.politics.mideast** и **talk.politics.guns**, что объясняется схожими терминами и темами. Это подтверждает необходимость дальнейшей работы над улучшением классификации в таких категориях, возможно, через дополнительную предобработку или применение других моделей.

3. **Анализ важности признаков:**

   Анализ ключевых слов для каждой категории подтверждает, что модель правильно выделяет важнейшие термины, которые помогают в классификации. Для категории **alt.atheism** выделены такие слова, как **god**, **atheism**, **bible**, что логично для данной темы. В категории **comp.graphics** наибольшее значение имеют слова, такие как **graphics**, **3d**, **image**, что соответствует тематике компьютерной графики.

   Для более сложных категорий, таких как **talk.religion.misc**, выделенные признаки, такие как **god**, **atheism**, **christian** и другие религиозные термины, показывают, что

модель работает с этими темами, но в силу схожести текстов в различных категориях точность могла бы быть выше.

4. **Перспективы и рекомендации:**

   Для улучшения классификации и повышения точности модели, особенно в трудных категориях, таких как **talk.religion.misc**, можно рассмотреть следующие шаги:

   1) Использование **n-gram** векторизации для учета последовательностей слов, что может улучшить различие схожих категорий.
   2) Применение более сложных классификаторов, таких как **случайный лес** (Random Forest) или **машины опорных векторов** (SVM), которые могут лучше справляться с пересекающимися категориями.
   3) Дополнительная предобработка текстов, например, **стемминг** или **лемматизация**, для уменьшения влияния сходных слов и улучшения разделимости категорий.

5. **Заключение:**

   Модель логистической регрессии показала хорошую производительность, достигнув хороших результатов в четко определенных темах. Однако точность в некоторых сложных категориях могла бы быть выше. Применение более мощных алгоритмов и методов улучшения предобработки данных может помочь повысить точность и общее качество классификации.