Лабораторная работа №3-4, Часть 1: Знакомство с платформой Hugging Face Hub

Цель работы: Освоить базовые принципы работы с платформой Hugging Face Hub - центральным репозиторием моделей, датасетов и приложений машинного обучения. Получить практические навыки поиска, оценки и загрузки моделей и датасетов для задачи текстовой классификации.

Стек технологий:

- **Операционная система:** Ubuntu 24.04 LTS
- **Менеджер пакетов и окружений:** Conda (окружение `mlops-lab`)
- **Библиотеки:** `huggingface_hub`, `datasets`, `transformers`, `pandas`, `numpy`
- **Платформа:** Hugging Face Hub

Теоретическая часть (краткое содержание)

1. Введение в Hugging Face и экосистему Transformers Hugging Face — это компания и сообщество, создавшее самую популярную в мире open-source платформу для машинного обучения. Ключевые продукты:

- **Transformers:** Библиотека, предоставляющая тысячи предобученных моделей для NLP, компьютерного зрения, аудио и других задач.
- **Hugging Face Hub:** Централизованный репозиторий для обмена моделями, датасетами и демо-приложениями (Spaces).
- **Datasets:** Библиотека для простой загрузки и обработки датасетов.

Hugging Face Hub функционирует как "GitHub для ML", где исследователи и инженеры могут:

- **Обнаруживать** предобученные модели и датасеты
- **Совместно работать** над ML-проектами
- **Делиться** своими разработками с сообществом

2. Ключевые концепции платформы

- **Модели (Models):** Предобученные веса архитектур нейронных сетей (BERT, GPT, ResNet и др.) для различных задач.
- **Датасеты (Datasets):** Коллекции данных для обучения и оценки моделей. Могут быть официальными (от создателей) или community-driven.
- **Spaces:** Интерактивные веб-демонстрации моделей с графическим интерфейсом.
- **Tasks:** Стандартизированные типы ML-задач (текстовая классификация, суммирование, перевод и т.д.).

3. Задача текстовой классификации Текстовая классификация — одна из фундаментальных задач NLP, включающая:

- **Классификация тональности** (sentiment analysis)
- **Классификация тем** (topic classification)

- **Определение спама**
- **Категоризация текстов**

Задание на практическую реализацию

Этап 1: Установка необходимых библиотек

1. **Активация окружения и установка пакетов:**

```
conda activate mlops-lab
pip install huggingface_hub datasets transformers pandas numpy
```

Этап 2: Работа с Hugging Face Hub через веб-интерфейс

1. **Знакомство с интерфейсом:**
   - Откройте Hugging Face Hub в браузере.
   - Изучите главную страницу: разделы Models, Datasets, Spaces, Documentation.
2. **Поиск датасета для текстовой классификации:**
   - Перейдите в раздел **Datasets**.
   - В поиске введите "sentiment analysis" или "text classification".
   - Найдите популярные датасеты:
     - **IMDb** - классификация тональности отзывов на фильмы
     - **AG News** - классификация новостей по темам
     - **Emotion** - классификация эмоций в тексте
   - Выберите датасет **emotion** (отметьте количество примеров, лицензию, язык).
3. **Поиск модели для текстовой классификации:**
   - Перейдите в раздел **Models**.
   - В фильтрах выберите:
     - **Task:** Text Classification
     - **Library:** Transformers
     - **Dataset:** emotion (опционально)
   - Изучите доступные модели, обращая внимание на:
     - Количество загрузок
     - Размер модели
     - Язык
     - Метрики качества (если указаны)
   - Выберите модель **bert-base-uncased** или **distilbert-base-uncased** (более легкая версия).

Этап 3: Программная работа с Hugging Face Hub

1. **Создание Python-скрипта для исследования:**

```
touch hf_hub_exploration.py
```

2. **Написание кода для загрузки датасета:**

```
from datasets import load_dataset
from huggingface_hub import list_models, list_datasets
import pandas as pd

# Исследование доступных датасетов
print("Доступные датасеты для текстовой классификации:")
datasets = list_datasets(filter="task_categories:text-classification")
for dataset in datasets:
    print(f"- {dataset.id}")

# Загрузка датасета emotion
print("\nЗагрузка датасета emotion...")
dataset = load_dataset("emotion")

# Исследование структуры датасета
print(f"\nСтруктура датасета: {dataset}")
print(f"\nПримеры из train split:")
train_df = pd.DataFrame(dataset['train'][:5])
print(train_df)

# Анализ распределения классов
print("\nРаспределение классов в тренировочных данных:")
label_counts = pd.DataFrame(dataset['train']['label']).value_counts()
print(label_counts)
```

3. **Написание кода для исследования моделей:**

```
# Исследование доступных моделей
print("\n\nДоступные модели для текстовой классификации:")
models = list_models(
    filter="task:text-classification",
    sort="downloads",
    direction=-1,
    limit=5
)

for model in models:
    print(f"\nМодель: {model.id}")
    print(f"Загрузок: {model.downloads}")
    print(f"Тэги: {model.tags}")
    if model.pipeline_tag:
        print(f"Тип задачи: {model.pipeline_tag}")
```

4. **Загрузка выбранной модели:**

```
from transformers import AutoTokenizer, AutoModelForSequenceClassification

# Загрузка токенизатора и модели
model_name = "distilbert-base-uncased"
print(f"\nЗагрузка модели {model_name}...")

tokenizer = AutoTokenizer.from_pretrained(model_name)
model = AutoModelForSequenceClassification.from_pretrained(
    model_name,
    num_labels=6  # Количество классов в датасете emotion
)

print("Модель и токенизатор успешно загружены!")
print(f"Размер словаря: {tokenizer.vocab_size}")
print(f"Архитектура модели: {model.__class__.__name__}")
```

5. **Тестирование работы токенизатора:**

```
# Тестирование токенизатора
test_text = "I am feeling very happy today!"
print(f"\nТекст для теста: {test_text}")

tokens = tokenizer(test_text, return_tensors="pt")
print(f"Токены: {tokens}")
print(f"Декодированные токены: {tokenizer.decode(tokens['input_ids'][0])}")
```

6. **Запуск скрипта:**

```
python hf_hub_exploration.py
```

Этап 4: Сохранение локальных копий

1. **Создание директории для проекта:**

```
mkdir text-classification-project
cd text-classification-project
```

2. **Сохранение информации о выбранных ресурсах:**

```
echo "Датасет: emotion" > resources.txt
echo "Модель: distilbert-base-uncased" >> resources.txt
```

```
echo "Количество классов: 6" >> resources.txt
```

Требования к оформлению и отчету

Критерии оценки для Части 1:

- **Удовлетворительно:** Успешно выполнены Этапы 1-2 (установка библиотек, исследование Hub через веб-интерфейс, выбор датасета и модели). Создан файл `resources.txt` с информацией о выбранных ресурсах.
- **Хорошо:** Дополнительно успешно выполнен Этап 3 (написан и запущен скрипт `hf_hub_exploration.py`, который загружает датасет и информацию о моделях). Продемонстрировано понимание структуры датасета.
- **Отлично:** Все задания выполнены в полном объеме. Скрипт дополнен функционалом анализа датасета (статистика по длине текстов, визуализация распределения классов) и тестирования работы модели на примерах из датасета.

Рекомендуемая литература

1. **Официальная документация Hugging Face:** https://huggingface.co/docs
2. **Hugging Face Transformers Documentation:** https://huggingface.co/docs/transformers
3. **Hugging Face Datasets Documentation:** https://huggingface.co/docs/datasets
4. **Статья "Getting Started with Hugging Face":** https://towardsdatascience.com/getting-started-with-hugging-face-transformers-
5. **Книга "Natural Language Processing with Transformers":** Lewis et al. (Главы 1-2)