

Sentiment Classification for Movie Reviews(EN/JA)

Task Assignment: Sentiment Classification for Movie Reviews

Overview

In this task, you are required to build a classification model that determines whether a movie review expresses a **positive** or **negative** sentiment, using a publicly available dataset.

You will be evaluated on your ability to understand and process the data, build and evaluate appropriate models, and implement the solution as a working service by exposing it via API and containerizing it with Docker.

Dataset

Please use the following Kaggle dataset: [IMDb Dataset of 50K Movie Reviews](#)

This dataset contains approximately 50,000 movie reviews labeled as either "positive" or "negative".

Task Requirements(Mandatory)

1. Data Analysis and Visualization

- Understand the structure and characteristics of the data. Perform visualization as needed.

2. Model Development

- You are free to choose and implement any models, architectures, or frameworks that you consider suitable for this task.

3. Model Evaluation

- Evaluate your models using appropriate metrics.
- If you build multiple models, please compare them and provide analysis based on the evaluation results.

4. Serve the Model as an API

- Provide the trained model as a RESTful API.
- Attach API documentation that clearly describes the endpoints, request format, and response format.

5. Containerization with Docker

- Create a Dockerfile to run the model as a containerized service.
- Include Docker build and run instructions in the `README.md`.

Deliverables

- `.ipynb` or `.py` (Well-structured code with clear comments)
- `README.md` (Explanation of your approach, implementation steps, and how to run the service)
- API documentation
- Dockerfile and Docker run instructions
- Upload all deliverables to a **public GitHub repository**, and share the **repository URL** with us

Bonus Task (Optional)

Design a Mechanism to Detect Model Degradation and Redeploy the Service

- Propose a strategy for detecting performance degradation in production (e.g., data drift, drop in prediction accuracy).
- Additionally, suggest how you would automatically retrain or redeploy the model if such degradation is detected (conceptual level only is fine).

Deadline

Please submit your GitHub repository link **within 7 days** of receiving this assignment.

課題：映画レビューにおける感情分類タスク

概要

この課題では、公開されている映画レビューのデータセットを用いて、レビューがポジティブかネガティブかを判定する分類モデルを構築していただきます。

データの理解と前処理、モデルの選定と構築、評価に加え、構築したモデルを**API**として提供し、**Docker**でコンテナ化することで、実際のサービスとしての実装力を確認します。

使用データ

以下のKaggleデータセットをご利用ください：[IMDb Dataset of 50K Movie Reviews](#)

約5万件の映画レビューに、「ポジティブ」または「ネガティブ」のラベルが付与されています。

課題要件（必須）

1. データの分析・可視化
 - a. データの構造や傾向を把握し、必要に応じて可視化を行ってください。
2. モデル構築
 - a. 本タスクに適切だと考えるモデル、構成、フレームワークを自由に選定し、構築してください（指定はありません）。
3. モデル評価
 - a. 適切な評価指標を用いてモデルを評価してください。
 - b. 複数モデルを構築した場合、評価結果に基づき、モデルの比較・考察を行ってください。
4. モデルを**API**として提供する
 - a. 構築したモデルをRESTful APIとして提供してください。
エンドポイント、リクエスト形式、レスポンス形式を明記した**APIドキュメント**を添付してください。
5. **Docker**によるコンテナ化
 - a. Dockerfileを作成し、モデルをコンテナ上で動作させてください。
 - b. Dockerのビルドおよび実行手順をREADMEに記載してください。

提出物

1. **.ipynb**または**.py**ファイル（コメント付き、整理されたコード）
2. **README.md**（アプローチ、手順、実行方法の説明）

3. **APIドキュメント**

4. **Dockerfileおよび実行手順**

5. 上記一式を**GitHub**の公開リポジトリにアップロードし、**リポジトリURL**を提出してください

ボーナス課題（任意）

モデルの劣化を検知し、再デプロイする仕組みの設計（構想ベースで可）

本番環境でモデルの性能が低下した場合（例：データドリフト、予測精度の低下）に、それをどのように検知するかを設計してください。

また、モデルの再学習や再デプロイを自動で行う仕組みについても、アイデアレベルで構いませんので提案してみてください。

提出期限

課題受領後、**7日以内**にご提出ください。