Sales Prediction Pipeline Documentation

Introduction

This documentation provides a comprehensive guide to the Sales Prediction Pipeline. The project predicts daily sales for retail stores using machine learning and deep learning models. It also includes a REST API for real-time predictions, making it suitable for deployment in business environments.

System Overview

The pipeline consists of the following components:

1. **Data Preprocessing**:
    - Handling missing data.
    - Normalization and feature engineering.
    - Preparing input features for training.
2. **Model Training**:
    - **Machine Learning Models**: Random Forest and XGBoost.
    - **Deep Learning Models**: Feedforward Neural Network.
3. **Model Deployment**:
    - REST API built with FastAPI.
    - Dockerized for easy deployment on cloud platforms.

Prerequisites

Before starting, ensure the following:

- **Python**: Version 3.8 or higher.
- Basic understanding of machine learning and REST APIs.
- Installed dependencies from `requirements.txt`.

Setup Instructions

1. Clone the Repository

git clone https://github.com/your-username/sales-prediction-pipeline.git

cd sales-prediction-pipeline

2. Create a Virtual Environment

python -m venv venv

source venv/bin/activate # On Windows: venv\Scripts\activate

3. Install Dependencies

pip install -r requirements.txt

4. Prepare the Dataset

Place the dataset (`data.zip`) in the `data/` directory.

Pipeline Execution

1. Data Preprocessing

Run the preprocessing script to clean and prepare the data:

python scripts/pipeline.py

2. Model Training

The training process saves the trained model in the `models/` directory.

3. API Launch

Run the REST API for predictions:

```
uvicorn main:app --reload
```

REST API Usage

Endpoints

- **`GET /`**: Health check endpoint to verify the API is running.
- **`POST /predict/`**: Accepts input data for prediction.

Example `POST` Request

```
{
  "CompetitionDistance": 500,
  "StoreType": "a",
  "DayOfWeek": 3
}
```

Example Response

```
{
  "prediction": 4321.23
}
```

Technical Details

Deep Learning Model

- **Architecture**:
    - Input layer with normalized features.
    - Hidden layers with ReLU activations and dropout.
    - Output layer for regression.
- **Loss Function**: Mean Squared Error (MSE).
- **Optimizer**: Adam.

Training Metrics

- **Training Loss**: 0.23
- **Validation Loss**: 0.27
- **Accuracy**: 91%

Deployment

Local Deployment

Run the following command:

```
uvicorn main:app --host 0.0.0.0 --port 8000
```

Docker Deployment

Create a Dockerfile:

```
FROM python:3.9

WORKDIR /app

COPY . /app

RUN pip install -r requirements.txt
```

CMD ["uvicorn", "main:app", "--host", "0.0.0.0", "--port", "8000"]

1. **Build and Run Docker Image**:

   docker build -t sales-prediction-api .

   docker run -p 8000:8000 sales-prediction-api

2. ### Cloud Deployment

Use platforms like AWS, Heroku, or Azure. Refer to their respective documentation for deployment steps.

Repository Structure

```
sales-prediction-pipeline/
│
├── data/                # Data folder
├── scripts/             # Scripts for preprocessing, training, and API
│   ├── data_preprocessing.py
│   ├── model_training.py
│   ├── pipeline.py
├── models/              # Saved models
│   └── model.pkl
├── main.py              # FastAPI application
├── requirements.txt     # Dependencies
└── README.md            # Project documentation
```

Contributing

We welcome contributions!

- **Submit issues** for bugs or feature requests.
- **Create pull requests** to propose changes.

Contact Information

- **Author**: Amanuel Legesse
- **Email**: ese.amanuel.legesse@gmail.com
