Work Report: Creating a New Project for Developing an API to Utilize Trained Models.

Prepared by [Nahid Shahab] on [2025-02-24]

Here is my report link on GitHub: GitHub Link

CryptoPredictAPI Project Report

1. Introduction

The **CryptoPredictAPI** project is a Flask-based API that fetches real-time cryptocurrency data from the **Binance** exchange, computes technical indicators, and predicts future prices using an **LSTM deep learning model**. This API can be used in automated trading systems and market analysis.

---

2. Key Features

- **Fetch live data** from Binance (OHLCV: Open, High, Low, Close, Volume)
- **Compute technical indicators** such as **Simple Moving Average (SMA), Relative Strength Index (RSI),** and **Bollinger Bands**
- **Predict prices using an LSTM model** trained on historical data
- **Provide a RESTful API** for seamless integration with other systems
- **Customize the machine learning model** for better predictions

---

3. Installation & Setup

3.1. Prerequisites

Ensure the following dependencies are installed on your system:

- **Python 3.8+**
- **pip** (Python package manager)
- **Git** (for cloning the repository)

3.2. Installation Steps

1. Clone the Repository

```
git clone https://github.com/NShahab/CryptoPredictAPI.git
cd CryptoPredictAPI
```

2. Create a Virtual Environment (Optional but Recommended)

```
python -m venv venv
source venv/bin/activate   # On Windows: venv\Scripts\activate
```

3. Install Dependencies

```
pip install -r requirements.txt
```

---

4. Usage

4.1. Running the API

To start the API, run the following command:

```
python app.py
```

The API will be available at **http://127.0.0.1:5000/**.

4.2. API Documentation

4.2.1. Price Prediction Endpoint (`GET /predict_price`)

This endpoint fetches real-time data, computes technical indicators, and predicts the future price using the **LSTM** model.

Parameters:

- `symbol` (optional): The trading pair (default: `BTCUSDT`)
- `interval` (optional): Time interval (default: `4h`)

Example Request:

```
GET /predict_price?symbol=BTCUSDT&interval=4h
```

Example Response:

```
{
  "symbol": "BTCUSDT",
  "interval": "4h",
  "predicted_price": 45000.25
}
```

---

5. LSTM Model Training

If you want to train your own **LSTM** model:

1. Collect historical cryptocurrency data.
2. Train the model using **TensorFlow/Keras**.
3. Replace the existing model file `models/model_LSTM_4h.keras` with your trained model.

---

6. Deployment

6.1. Deploying on GitHub Codespaces

1. **Open the repository in GitHub Codespaces.**
2. **Install dependencies:**
3. `pip install -r requirements.txt`
4. **Run the API:**
5. `python app.py`
6. **Access the API via the public URL** provided by GitHub Codespaces.

---

7. Contributing to the Project

- **Improvements & Suggestions:** Your pull requests are welcome to enhance the project.
- **Reporting Issues:** If you encounter any problems, please report them in **GitHub Issues**.

---

8. License

This project is licensed under the **MIT License**.

---

9. Resources & Related Links

- GitHub Repository
- Binance API Documentation
- LSTM Training with TensorFlow

Prepared by: The **CryptoPredictAPI** Development Team