**Инструкция по развертыванию и использованию торгового бота FaunaBOT**

**1. Обзор системы**

FaunaBOT представляет собой программное обеспечение для анализа финансовых рынков и прогнозирования движения цен на следующий торговый день. Бот использует исторические данные о ценах, загружаемые через yfinance, для расчета технических индикаторов (скользящие средние, RSI, MACD) и обучения модели машинного обучения (Random Forest или Logistic Regression). На основе обученной модели бот предоставляет прогноз: "Рост" или "Падение" для следующего торгового дня, а также отображает графики цен и индикаторов.

**2. Системные требования**

Для корректной работы FaunaBOT требуется операционная система, поддерживающая Python 3.8 или выше. Рекомендуется использовать Windows, macOS или дистрибутивы Linux.

**3. Установка программного обеспечения**

Для развертывания FaunaBOT необходимо установить следующие программные компоненты:

**3.1. Установка Python**

Если Python не установлен в вашей системе, загрузите и установите его с официального сайта: [https://www.python.org/downloads/](https://www.python.org/downloads/" \t "_blank)

* **Для Windows:** Загрузите исполняемый установщик. Во время установки **обязательно** отметьте опцию "Add Python to PATH" (или "Добавить Python в PATH") для удобства использования из командной строки.
* **Для macOS/Linux:** Используйте пакетный менеджер (например, brew install python3 для macOS, sudo apt-get install python3 для Ubuntu/Debian) или загрузите исходный код/установщик с сайта.

**3.2. Установка необходимых библиотек Python**

Все необходимые библиотеки устанавливаются с помощью пакетного менеджера pip. Откройте командную строку (или PowerShell в Windows, Terminal в macOS/Linux) и выполните следующие команды:

Bash

pip install pandas

pip install yfinance

pip install scikit-learn

pip install matplotlib

pip install numpy

**Краткое описание библиотек:**

* pandas: Для работы с табличными данными (DataFrame) и временными рядами.
* yfinance: Для загрузки исторических финансовых данных с Yahoo Finance.
* scikit-learn: Для реализации алгоритмов машинного обучения (Random Forest, Logistic Regression) и метрик.
* matplotlib: Для построения графиков.
* numpy: Для выполнения численных операций, особенно при ручном расчете индикаторов.

**4. Развертывание бота**

1. **Сохраните код:** Скопируйте полный код бота (который был предоставлен) и сохраните его в файл с именем, например, FaunaBot.py, в любой удобной для вас директории (например, C:\Users\ВашПользователь\Documents\Fauna-Capital-main\).
2. **Перейдите в директорию бота:** Откройте командную строку и перейдите в директорию, где вы сохранили файл FaunaBot.py.
   * **Пример для Windows:**

Bash

cd C:\Users\ВашПользователь\Documents\Fauna-Capital-main\

* + **Пример для macOS/Linux:**

Bash

cd /Users/ВашПользователь/Documents/Fauna-Capital-main/

**5. Запуск бота**

Для запуска бота выполните следующую команду в командной строке, находясь в директории с файлом FaunaBot.py:

Bash

python FaunaBot.py

После запуска появится графический интерфейс пользователя (GUI) бота.

**6. Использование графического интерфейса бота**

Интерфейс бота интуитивно понятен и состоит из следующих элементов:

* **Тикер криптовалюты (например, BTC-USD, AAPL):** Введите символ торгового инструмента, который вы хотите проанализировать. Примеры:
  + BTC-USD для Bitcoin к доллару США.
  + AAPL для акций Apple.
  + GOOGL для акций Google.
  + ^GSPC для индекса S&P 500.
* **Дата начала (YYYY-MM-DD):** Введите дату, с которой вы хотите начать загрузку исторических данных. Формат даты может быть YYYY-MM-DD (например, 2022-01-01) или DD.MM.YYYY (например, 01.01.2022).
  + *Примечание:* Для качественного расчета индикаторов и обучения модели требуется достаточное количество данных (рекомендуется не менее 50-100 дней).
* **Дата окончания (YYYY-MM-DD):** Введите конечную дату для загрузки исторических данных. Формат даты такой же, как и для даты начала.
* **Выбор модели:** Выберите модель машинного обучения для прогнозирования:
  + **Random Forest (Случайный лес):** Более сложная и часто более точная модель, хорошо подходящая для выявления нелинейных зависимостей.
  + **Logistic Regression (Логистическая регрессия):** Более простая и быстрая линейная модель.
* **Кнопка "Запустить анализ и прогноз":** Нажмите эту кнопку, чтобы начать процесс загрузки данных, расчета индикаторов, обучения модели и получения прогноза.

**7. Результаты анализа**

После завершения расчетов появится информационное окно с результатами:

* **Точность модели (X.XX%)**: Показывает, насколько хорошо модель предсказала движение цены на тестовых данных. Значение 50% соответствует случайному угадыванию. Чем выше процент, тем лучше модель.
* **Прогноз на следующий торговый день: Рост 📈 / Падение 📉**: Это основной прогноз бота для следующего торгового дня.

Одновременно с информационным окном будут отображены графики:

* **График 1: Цена закрытия и скользящие средние (MA5, MA10, MA20)**.
* **График 2: Индекс относительной силы (RSI)**.
* **График 3: Схождение/Расхождение скользящих средних (MACD)**.

Эти графики помогут визуально оценить тенденции рынка и поведение индикаторов.

**8. Возможные ошибки и их устранение**

* **"Ошибка загрузки данных: нет данных..."**: Проверьте правильность написания тикера и убедитесь, что выбранный диапазон дат содержит торговые данные. Также убедитесь, что дата начала не находится в будущем и не позже даты окончания.
* **"Неверный формат даты..."**: Введите даты в формате YYYY-MM-DD или DD.MM.YYYY.
* **"Недостаточно данных..."**: Увеличьте диапазон дат для загрузки, чтобы модель могла обучиться на достаточном количестве исторических данных.
* **Проблемы с библиотеками (ModuleNotFoundError)**: Убедитесь, что все библиотеки из пункта 3.2 были успешно установлены. Если ошибка сохраняется, попробуйте переустановить их (pip install --upgrade <имя\_библиотеки>).