برای ایجاد یک برنامه هوش مصنوعی با هدف ارائه پیشنهاد سرمایه‌گذاری روی ارزهای دیجیتال، شما باید یک رویکرد گام‌به‌گام و جامع را دنبال کنید. در زیر مراحل کلی این فرآیند توضیح داده شده است:

---

### 1. \*\*تعریف هدف و جمع‌آوری نیازمندی‌ها\*\*

- مشخص کنید برنامه شما قرار است چه کارهایی انجام دهد (مثلاً پیش‌بینی قیمت، شناسایی بهترین زمان خرید یا فروش، تحلیل ریسک، و ارائه دلایل تحلیل).

- ویژگی‌های جدیدی که می‌خواهید اضافه کنید را تعریف کنید، مثل یادگیری از داده‌ها، مدیریت سرمایه، یا تحلیل دلایل رشد یا افت ارزها.

---

### 2. \*\*جمع‌آوری داده‌ها\*\*

- \*\*منابع داده:\*\*

از منابع معتبر برای داده‌های قیمت، حجم معاملات، و سایر ویژگی‌ها استفاده کنید. APIهایی مانند Binance، CoinMarketCap، و CoinGecko مناسب هستند.

- \*\*داده‌های تکمیلی:\*\*

اخبار، رسانه‌های اجتماعی (مثل توییتر و ردیت)، و تحلیل‌های تکنیکال.

- \*\*ایجاد پایگاه داده:\*\*

داده‌های جمع‌آوری‌شده را ذخیره و به‌صورت ساختاریافته سازماندهی کنید (مثل استفاده از SQLite یا Pandas برای ذخیره موقت).

---

### 3. \*\*پیش‌پردازش داده‌ها\*\*

- \*\*تمیزسازی داده‌ها:\*\*

حذف نویز، داده‌های تکراری یا مقادیر گمشده.

- \*\*استخراج ویژگی‌ها:\*\*

ویژگی‌های مرتبط مانند میانگین متحرک، شاخص قدرت نسبی (RSI)، و تحلیل حجمی را اضافه کنید.

- \*\*عادی‌سازی:\*\*

داده‌ها را برای مدل یادگیری ماشینی آماده کنید (مانند مقیاس‌بندی بین ۰ و ۱).

---

### 4. \*\*طراحی مدل هوش مصنوعی\*\*

- \*\*انتخاب الگوریتم:\*\*

- مدل‌های پیش‌بینی زمانی: LSTM یا GRU برای داده‌های سری زمانی.

- مدل‌های تصمیم‌گیری: Random Forest یا Gradient Boosting.

- \*\*یادگیری عمیق:\*\*

اگر قصد دارید از الگوریتم‌های پیچیده‌تر استفاده کنید، شبکه‌های عصبی عمیق (Deep Learning) می‌توانند مفید باشند.

- \*\*تست و آموزش مدل:\*\*

داده‌ها را به مجموعه‌های آموزش و آزمایش تقسیم کنید و مدل را آموزش دهید.

---

### 5. \*\*ارائه تحلیل و پیشنهادات\*\*

- \*\*نمایش اطلاعات:\*\*

یک داشبورد تعاملی برای نمایش نمودارها و تحلیل‌ها طراحی کنید.

- \*\*توصیه‌های سرمایه‌گذاری:\*\*

مدل را طوری تنظیم کنید که بهترین ارزها و زمان‌های سرمایه‌گذاری را همراه با توضیحات ارائه دهد.

- \*\*مدیریت ریسک:\*\*

برنامه باید به کاربران در مدیریت سرمایه کمک کند (مثلاً پیشنهاد تخصیص سرمایه به ارزهای مختلف).

---

### 6. \*\*بهبود مستمر\*\*

- \*\*اضافه کردن یادگیری مداوم:\*\*

مدل شما باید داده‌های جدید را یاد بگیرد و خود را به‌روزرسانی کند.

- \*\*حذف داده‌های قدیمی:\*\*

برای بهینه نگه‌داشتن سیستم، داده‌های غیرضروری را به‌صورت دوره‌ای پاک کنید.

- \*\*ایده‌های جدید:\*\*

- تحلیل روانشناسی بازار.

- بررسی عوامل اجتماعی و اقتصادی.

- پیش‌بینی تغییرات بزرگ بازار (Bull یا Bear Market).

---

### 7. \*\*پیاده‌سازی رابط کاربری (UI)\*\*

- از فریم‌ورک‌های سبک مانند \*\*Tkinter\*\* یا ابزارهای پیشرفته‌تر مثل \*\*Dash\*\* برای طراحی واسط کاربری استفاده کنید.

- دکمه‌های جداگانه برای تحلیل و جستجو طراحی کنید تا کاربران به‌راحتی به داده‌ها دسترسی داشته باشند.

---

### 8. \*\*تست و ارزیابی\*\*

- برنامه را با داده‌های واقعی و کاربران آزمایشی تست کنید.

- عملکرد مدل را ارزیابی کنید و نقاط ضعف آن را شناسایی کنید.

---

### 9. \*\*انتشار و پشتیبانی\*\*

- برنامه را روی یک پلتفرم (مانند وب یا موبایل) منتشر کنید.

- به‌روزرسانی‌های منظم و رفع اشکالات را در نظر بگیرید.

---

### پیشنهاد ابزارها و تکنولوژی‌ها:

- \*\*Python Libraries:\*\*

- درخواست داده: `requests`، `ccxt` (برای Binance).

- پردازش داده: `pandas`، `numpy`.

- یادگیری ماشین: `tensorflow`، `keras` یا `scikit-learn`.

- رسم نمودار: `matplotlib`، `plotly`.

- \*\*بانک داده:\*\* SQLite یا MySQL.

- \*\*APIهای مورد نیاز:\*\* Binance، CoinGecko، و CoinMarketCap.

اگر نیاز به کدنویسی یا جزئیات بیشتری دارید، خوشحال می‌شوم کمک کنم. 😊