**به نام خدا**

| **نوع رابطه بین**X**و**y | **مدل پیشنهادی** | **کد نمونه** | **مثال کاربردی** |
| --- | --- | --- | --- |
| **خطی ساده** | Normal Equation (یا LinearRegression) | from sklearn.linear\_model import LinearRegression | پیش‌بینی قیمت خانه براساس متراژ |
| **خطی با داده‌های حجیم/پیچیده** | Gradient Descent (در SGDRegressor) | from sklearn.linear\_model import SGDRegressor | پیش‌بینی فروش با میلیون‌ها داده |
| **غیرخطی با قوانین مشخص** | Decision Tree | from sklearn.tree import DecisionTreeRegressor | تشخیص بیماری براسسن علائم |
| **غیرخطی بسیار پیچیده/تصاویر** | **شبکه عصبی** (Sequential) | from tensorflow.keras import Sequential | تشخیص چهره، پردازش زبان طبیعی |

**توضیحات تکمیلی برای هر مورد**:

۱. **رابطه خطی ساده (**Normal Equation**/**LinearRegression**)**:

* **مثال**: پیش‌بینی قیمت خانه (y) براساس متراژ (X).
* **کد**:

from sklearn.linear\_model import LinearRegression

model = LinearRegression() *# معادل Normal Equation*

model.fit(X, y)

#### ۲. ****رابطه خطی با داده‌های حجیم (****Gradient Descent****)****:

* **مثال**: پیش‌بینی کلیک تبلیغات با ۱۰۰ میلیون رکورد.
* **کد**:

rom sklearn.linear\_model import SGDRegressor

model = SGDRegressor(max\_iter=1000, tol=1e-3) *# نسخه بهینه‌شده Gradient Descent*

model.fit(X, y)

#### ۳. ****رابطه غیرخطی با قوانین قابل تفسیر (****Decision Tree****)****:

* **مثال**: تشخیص وام‌گیرندگان پرریسک براسسن سن و درآمد.
* **کد**:

from sklearn.tree import DecisionTreeClassifier

model = DecisionTreeClassifier(max\_depth=3) *# کنترل پیچیدگی*

model.fit(X, y)

#### ۴. ****رابطه بسیار پیچیده/بدون رابطه واضح (****شبکه عصبی****)****:

* **مثال**: تشخیص خودکار اشیا در تصاویر یا ترجمه ماشینی.
* **کد**:

from tensorflow.keras import Sequential

from tensorflow.keras.layers import Dense

model = Sequential([

Dense(64, activation='relu', input\_shape=(10,)), *# لایه پنهان*

Dense(1, activation='sigmoid') *# لایه خروجی*

])

model.compile(optimizer='adam', loss='binary\_crossentropy')

model.fit(X, y, epochs=10)

### ****نکات کلیدی برای انتخاب نهایی****:

1. **همیشه داده را اول تحلیل کنید**:
   * با نمودار plt.scatter(X, y) ببینید آیا رابطه خطی/غیرخطی است.
2. **مدل‌های ساده را اول امتحان کنید**:
   * همیشه با LinearRegression یا DecisionTree شروع کنید قبل از رفتن به سمت شبکه‌های عصبی.
3. **ارزیابی مدل**:
   * حتماً از معیارهایی مثل accuracy، MSE یا confusion matrix استفاده کنید.