Learning Roadmap for AI Health Risk Predictor App

To build the AI Health Risk Predictor app, you'll need to enhance your knowledge and skills in Android development and machine learning. Here's a learning roadmap tailored to your project:

1. Android Development (Java)

Focus on improving your core Android development skills since this will form the foundation of your app.

Key Areas to Learn:

- Form Design and User Input:
  - Learn to create user-friendly forms using EditText, RadioButton, Spinner, and CheckBox.
  - Validate user inputs in real-time (e.g., checking numeric ranges).

- Navigation and UI Design:
  - Understand RecyclerView, ViewModel, and LiveData for better UI design.
  - Explore material design principles for modern and engaging interfaces.

- Database Integration:
  - Learn how to use SQLite or Room for storing data like user inputs and recommendations.

- Charts and Graphs:
  - Use libraries like MPAndroidChart to visualize prediction results.

- Background Tasks:
  - Learn about WorkManager or AlarmManager for notifications and periodic tasks.

- Permissions and APIs:
  - Handle runtime permissions (e.g., camera access for scanning documents).
  - Integrate APIs if external data is needed (e.g., medical advice).

2. Machine Learning Basics

You need basic ML knowledge to train, evaluate, and integrate models into Android.

Key Areas to Learn:

- ML Concepts:
  - Study basic classification algorithms like Logistic Regression, Decision Trees, or Neural Networks.
  - Understand data preprocessing (e.g., cleaning, scaling).

- Training the Model:
  - Use Python and libraries like TensorFlow or scikit-learn to train models.
  - Work with health datasets (e.g., PIMA Indian Diabetes dataset).

- Exporting the Model:
  - Learn how to convert models into TensorFlow Lite format for Android integration.

- TensorFlow Lite in Android:
  - Explore how to load and run a TFLite model in Java using TensorFlow Lite APIs.
  - Familiarize yourself with ML Kit as an alternative for lightweight tasks.

3. Backend Integration

You might need a backend for advanced features like storing user data or updating health tips.

Key Areas to Learn:

- Firebase:
  - Use Firebase Realtime Database or Cloud Firestore for cloud data storage.
  - Utilize Firebase Authentication for user sign-ins.

- APIs:
  - Learn to work with RESTful APIs using Retrofit or Volley for fetching real-time data (e.g., health tips).

4. Advanced Android Topics

These will help you polish the app and add advanced features.

- Material Design:
  - Use libraries and components like Navigation Drawer, BottomNavigationView, and Snackbar.

- Push Notifications:
  - Implement notifications with Firebase Cloud Messaging (FCM).

- Sensors:
  - Explore sensors like pedometers if you plan to track activity.

5. Tools and Libraries

Here are some tools and libraries you should explore:

- TensorFlow Lite: For ML model integration.
- MPAndroidChart: For graphing and charting.
- Retrofit: For working with APIs.

- Room: For local database management.

- Material Design Components: For a modern UI.

Recommended Learning Resources:

- Android Development:

  - Android Developers Official Documentation (https://developer.android.com/docs)

  - Courses on platforms like Udemy or Coursera for Android (Java).

- Machine Learning:

  - TensorFlow Lite Tutorials (https://www.tensorflow.org/lite)

  - Coursera's "Machine Learning" by Andrew Ng for fundamentals.

- Hands-On Practice:

  - Build smaller apps focusing on individual features (e.g., using forms, loading TFLite models).

Once you've mastered these areas, you'll have the skills to build your app effectively.