Deployment of ML model on android app

Solving machine learning problems in a local system is not possible, but making it available to the community is critical; otherwise, the model is only your responsibility. When it is able to serve, you will be able to receive feedback and make necessary modifications. It is quite simple to implement a machine learning model in a Jupyter Notebook. And 90% of the time, a data science practitioner's problem description is presented in the form of a website. We'll learn how to turn any Machine Learning issue statement into an Android app in this article.

The basic approach for deploying machine learning on a website is to build the model in any Python IDE, extract it using a pickle module, and then deploy it in the form of a web app using any web framework like flask or streamlit. The entire implementation, from frontend to backend, is done in Python.

When installing Machine Learning in Android, the above method needs to be tweaked somewhat. We start with a model and pickle it. Because java is popular for developing Android apps and working with Android Studio, we'll use it for our frontend. In the middle, we'll use a Flask API to implement our machine learning model, whose output will be in JSON format (JSON is a universal format that any programming language can understand), and we'll hit the Flask API with our java android app, parse the JSON, and print it in the android frontend.

1. Building a ML model

This is a simple section. We import a dataset, apply a classifier (SVM, Random Forest, KNN etc.) to train the data on, then save the model in pickle format after a little preprocessing.

2. Built Flask API

A user will fill out the form with information, and the form will get a POST request when it is submitted. And when a user submits a post request, Flask API accepts the data and passes it to the machine learning model, which predicts the output class. We'll pass the anticipated class to the Android app as JSON.

3. Test Application using Postman

Postman is an interactive and automatic tool for verifying APIs in your project. It's a Chrome extension that connects to the HTTP API. It operates in the background and allows you to verify that your API meets our requirements. You may contact your API and get the necessary answer by supplying the URL of your running flask API and putting data in the key and value area.

Install Postman and after it has been successfully installed, open it and paste the Flask app's URL into it. Insert the name of the value we're accessing and the value you wish to provide in the key and value area.

4. Create Android App
    a. Install and setup Android Project
    You'll need Android studio to get started with Android development. If you haven't already done so, please download Android Studio. The download is simple; the studio, however, is quite huge, at over 1GB.

    Start the Android studio when it has been installed, and you will be given the option to create a new project. After you click on the new project, you must choose an activity. Select the empty activity and move on to the next step, where you'll need to give your project a name and a storage folder. After updating the project's name, leave the rest of the settings alone and click Finish. Because Android Studio is a little slow, setting up your project will take some time, so please be patient while it sets up all of the project's files.
    b. Create Android UI

    We all know that user interfaces are always built in an XML file. We'll create a comprehensive frontend UI in the XML file titled activity main.

    c. Run your UI using AVD
    AVD stands for Android Virtual Device, which allows you to run your software on Android and have it seem just like your phone. Select pixel-5 from the AVD area in the top-right corner. Select the virtual device via which it will execute by clicking next. You can now download Android-Q and proceed to the next step. You'll get a dialogue asking you to name a device. Click Finish. Finally, when you're ready, press the play button on your device, and it will launch an AVD in which your UI will run. Then, in AVD, run the desired file to observe the changes.
5. Connectivity of API to Android APP

Now you must create a java-based backend for it. The logic we must develop is that you will take the inputs from the Android app, call the API, and then display the answer from the API in the Android app. As a result, there is one issue: the API we created is running locally on your machine, which the Android app is unable to identify. As a result, we'll need to put our API online, and we'll use Heroku to do it.

Connect API to Internet

To use the API, we'll require the Volley library. To install Volley, go to your project's Gradle scripts directory, open the build Gradle file, and add one line of code that will install the required library. When you click sync now in the top right corner, the relevant libraries will be installed in the project directory.

- **Prof. Madhura Ranade**