Tell us what your idea is.

Describe in 250 words what the feature or service will do and how you'll use Machine Learning to push the bar:

Jenny is a Launcher that at the center of the home screen shows the apps the user needs exactly at that moment, making it faster and more convenient to find and launch apps.

The Launcher uses device sensors and data (WiFi/Bluetooth connectivity, GPS, time of day, day of the week, headset connectivity, battery/charging state and notifications) to determine in which context specific apps is launched.

Each time an app is launched the app name and a data/sensor snapshot is stored. The On-Device Machine Learning (TensorFlow Lite) finds patterns by comparing historical snapshots with app launches. Over time it's able to predict the apps the user wants to launch from real time snapshots. The result is placed right at the center of the home screen.

Because On-Device ML is used it will also work when the device is offline. Another benefit is that the user doesn't have to remember specific app names (f.x. What's the name of that meditation app). The app is just conveniently brought into focus when the user needs it.

Anonymized data is uploaded (optional) to Firebase Cloud Firestore and BigQuery. In the cloud the ML model can be further trained and improved models can be deployed to TensorFlow Lite on the phone.

Big focus on privacy with following options:

- *Opt-out of data upload to the cloud*
- *Exclude specific apps from being logged*
- *Granular grant or deny data/sensor types being used in ML. The algorithm is dynamic enough to work with the data available*

Possible enhancements:

- *Make it even more convenient to launch apps: Use notification bubbles (introduced in Android 10) or notifications*
- *Integrate Slices (Interactive content from apps)*
- *Show system and app actions: F.x. Change volume, Call a contact, Start navigation to Work*
- *Classify/Tag apps: Group apps and improve search: F.x. search on Music will return "Spotify"*

Tell us how you plan on bringing it to life.

Describe where your project is, how you could use Google's help in the endeavor, and how you plan on using On-Device ML technology to bring the concept to life. The best submissions have a great idea combined with a concrete path of where you plan on going, which should include:

- *(1) any potential sample code you've already written,*
  - *Work in progress: Adaptation of AOSP Launcher3 to work as a standalone Play Store app*
    https://github.com/cmunter/Jenny/tree/master/app/src/main/java/com/midsto/app/jenny
  - *Work in progress: Sensor collector sample code*
    https://github.com/cmunter/Jenny/tree/master/app/src/main/java/com/midsto/app/samplecode
- *(2) a list of the ways you could use Google's help,*
  - *Architecture of the Machine Learning model*
  - *Since user's trust and their willingness to contribute data is essential for the project to succeed, anonymizing user data is important. Here I could use Google's guidance to find the most optimal implementation. Current idea is to create an MD5 hash of user sensitive data.*
  - *UX workshop:*
    - *Improve convenience (f.x one-handed use). Discuss possible enhancements*
    - *Make it more delightful (short intro animations)*
    - *Expand center area for more granular results by performing a gesture*
  - *Awareness of the project thereby creating a larger sample size of data*
- *(3) as well as the timeline on how you plan on bringing it to life by May 1, 2020.*
  - *December 2: Adaptation of AOSP Launcher 3 to stand alone Launcher*
  - *December 20: Implement center home screen containing app suggestions*
  - *December 27: Incorporate sensor/data collector*
  - *January 10: Implement App and privacy settings*
  - *January 15: On-Device ML with TensorFlow Lite*
  - *January 30: Send data to the cloud. MD5 hash is used for privacy related data. E. g. WiFi SSID and location*
  - *February 10: Cloud ML analyzing the data uploaded. Allowing improved models to be send to the local ML*
  - *February 30: Welcome/Onboarding wizard*
  - *March 5: Improve UI (Buffer if project is behind schedule)*
  - *March 15: Internal alpha test*
  - *March 20: Bug fixing and improvements from user feedback. Add possible enhancements*
  - *April 2: Internal beta test*
  - *April 7: Bug fixing and improvements from user feedback*

- *May 1: Public Play Store roll out*

Tell us about you.

A great idea is just one part of the equation; we also want to learn a bit more about you. Share with us some of your other projects so we can get an idea of how we can assist you with your project.

I have worked professionally with Android since 2012 and before that it was my primary hobby. I've used many different Firebase services (Firebase Authentication, Firebase Realtime Database, Cloud Firestore, Crashlytics, Storage, Functions, App Engine) and Google cloud platform products (IoT core, Pub/Sub, BigQuery) in several projects that are in production.

My experience with Android is both from an app developers perspektive but also from a device manufactures perspektive (f.x.I have handled the MADA contract in several hardware projects).

Mobile phone manufacture (Lumigon)
Last hardware project - Task: Development Manager:

- Lumigon T3
  https://www.youtube.com/watch?v=CsNLmgA9t-0
  https://www.whathifi.com/lumigon/t3/overview

Other hardware projects - Task: Development Manager/Senior Developer:

- Lumigon T2 HD
- Lumigon T2
- Lumigon T1

I've made framework changes and developed system apps in the following Android distributions:
7.1, 7.0, 6.0, 5.1, 4.4, 4.3, 2.3, 2.2
Selected software projects not on the play store (part of Android distribution for the devices):

- Launcher app
- Framework
- Camera app
- System Settings
- Settings Automation app based on device profiles, connectivity and device sensors. Made together with a charging dock
- Security app, hide apps and media content in an encrypted and secure area
- Message app
- Gallery app

- Voice recorder app
- Universal infrared remote app
- Contacts app
- Call screen and other system UI
- Clock widgets
- Watch face on Android Wear
- Location based IoT (MQTT) app on Android Wear

Selected Play Store links:

- Sportler Ski Tracker
- AnyCore (Closed Beta)
- WelcomeBob

Linkedin: https://www.linkedin.com/in/munter/

Next steps.

- Be sure to include this cover letter in your GitHub repository
- Your GitHub repository should be tagged #AndroidDevChallenge
- Don't forget to include other items in your GitHub repository to help us evaluate your submission; you can include prior projects you've worked on, sample code you've already built for this project, or anything else you think could be helpful in evaluating your concept and your ability to build it
- **The final step is to fill out this form to officially submit your proposal.**