**\*Please make a copy of this document and include this in your GitHub repository for your submission, using the tag #AndroidDevChallenge\***

**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:*

Learning in developing economies is mostly passive. Teacher walks into the class, introduces a topic no one really cares about and starts talking. Everyone day dreams till the end of the class and almost no one truly learns anything at the end.

The cheapest, easiest and quickest way to convert any classroom from a passive one to an active one is by making the class study some portion of the next topic before coming to class. If they enjoy reading the appletizer, the more active the class will become. They come into class with questions and a fundamental understanding of the concept that is about to be taught. Even the teacher enjoys teaching better.

It is obvious educators do not need extra work creating appletizers from scratch, so they could easily go online, pick any webpage or video they find interesting, convert the page or video into an appletizer and share the “keyword” with their class when next they are in class. Keywords are words chosen by the educators, that are easy or remember. Students use these keywords to access their appletizers (webpage or video the teacher wants them to study before class).

Appletizer uses machine learning to help educators predict how much their class will like an article (webpage) or YouTube video even before they share the article with their class. This helps guide them in selecting only articles or videos with a high probability of being enjoyed by the class as opposed to guessing or not being aware.

You can download .apk file of what I’ve done so far from my gdrive: https://drive.google.com/open?id=1D21xijrvKwua5sOkUVSsyjjQT-vROWXH

**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,*
* *(2) a list of the ways you could use Google’s help,*
* *(3) as well as the timeline on how you plan on bringing it to life by May 1, 2020.*

I am currently working on the full functionality of the application and the models. I should be done by the end of December and start working on the UI/UX. The ML model gets the structure of the article from the URL submitted i.e. the number of sentences in the article, the number of images, captions, presence of a video etc. and uses this as input, and outputs an integer between 1 and 5 to show the educator how likely their class is to enjoy the article.

I have created the model for the webpage and I’m currently working on the model for YouTube videos. It will work with same concept i.e. the data surrounding the video not necessarily the content of the video.

Before launching, I will create model modifications depending on the level of the class (primary, Junior secondary, Tertiary etc.) but for now, I’m using a general model. Although the easiest option would be to use a general model and add to, or subtract from the output of the general model depending on the level of the class then display the final prediction to the educator.

**HELP!!!**

* I need Google’s guidance on a simple, more efficient user experience design for the application.
* I know there are better ways to optimize the models which I may not be aware of right now.
* I will also need advice on how best to monetize the application.

**DONE:**

* App accepts URL and extracts structural information from the link.
* Trained .tflite model to predict how much students are likely to enjoy the webpage or article.
* Feed the extracted data into the model and make a prediction.
* Convert the link into an Appetizer if you wish. An appetizer is a webpage or YouTube video given to students ahead of when it will be taught in class. It may or may not carry a question behind it. Appetiers are share via keywords.
* Students can use keywords to consume appetizer.

**TODO:**

***Dec 2019***

* Create WebView within the application to view URL
* Get URL from Firestore and pass into WebView.
* Create Educator’s history page where they can view details of all their “keywords” and submissions from each student (answers and scores).
* Create ranking mechanism where students can rank webpages and YouTube videos after consumption.
* Add YouTube API and the needed YouTube functionalities.

***Jan 2020***

* Use the ranking data to retrain the model.
* Train YouTube model.
* Handle exceptions.
* Work on UI/UX.
* Clean up codes.

**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 am Daniel from Nigeria and I love building new technologies with implications even though I know they might not become as successful as I wish. Being a tech enthusiast in Nigeria is really difficult but I enjoy the experience because I’m learning every day and I know one day, I’ll have my opportunity.

I have a background as an Android developer and it helped me gain confidence to learn ML which is my new interest.

*Some of my past projects includes:*

**Mocliq.mobi (2009):** The first betting game in Nigeria A 1D 20 player vs game where everyone clicks one button and compete for the 20th click.

The winner wins over the airtime of 15 losers. It grew organically to about 700 players in 3 days. I had to shut it down as my university doesn’t allow the use of mobile phone within the school.

**Ohcrave (2014)**: An event discovery android application where you share upcoming casual plans with your circle while they crave (preorder) for photos you’ll take when you finally carry out your plan. Only people who click the crave icon will receive the photos when they are uploaded.

**Dipitize (2016)**: A 2-player photo fighting game where the photos of each player are displayed side by to the public. The first player to get 10 likes wins the other players airtime (phone credit).

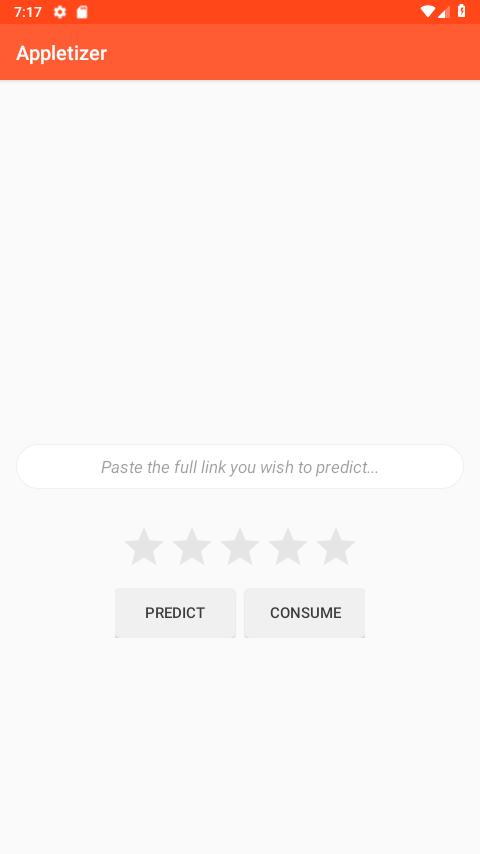
**Cliptize (2019):** Converts web links and YouTube links intoshort, fun, easy-to-pronounce words so teachers can share them verbally in classrooms.

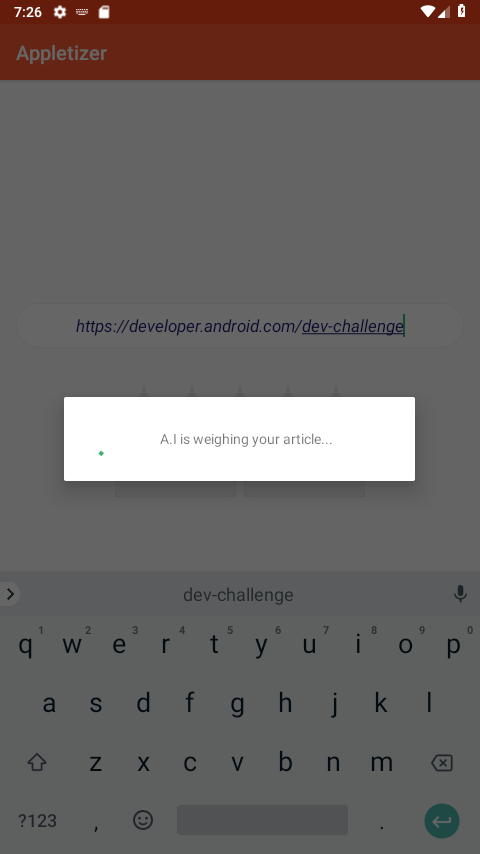
**Ivie Rental AI counter (2019)**: I also built an android application for mom that helps her count the number of plastic chairs in a stack. She does an event business where customers rent 1 full stack (12 chairs) for $0.8 for a day. They usually return the chairs without stacking them properly, so I trained a model that recognizes the number of chairs in a stack without needing to count it and outputs the number quickly. It helps reduce the stress of counting or the risk of guessing.

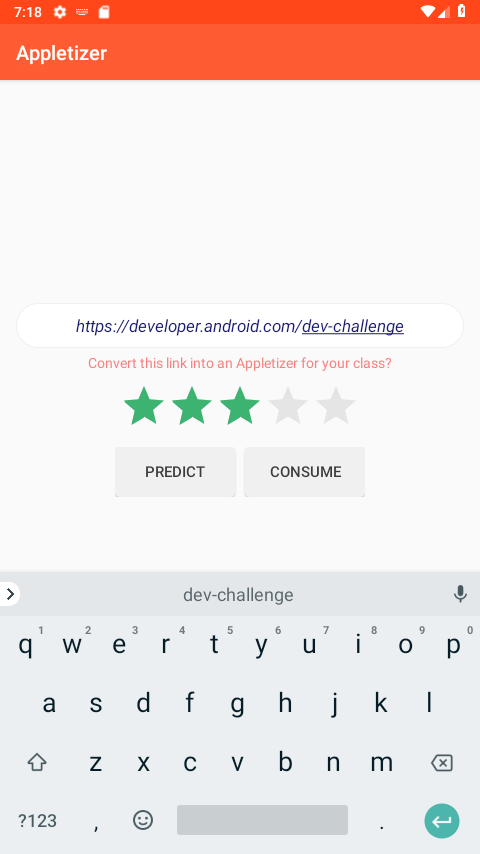
**Dating Scam Predictor (2019)**: Because of the high rate of online romantic scams from Nigeria, I created an android application that takes in answers from a worried user and predicts if the user is currently in the middle of an online relationship scam.

**Next steps.**

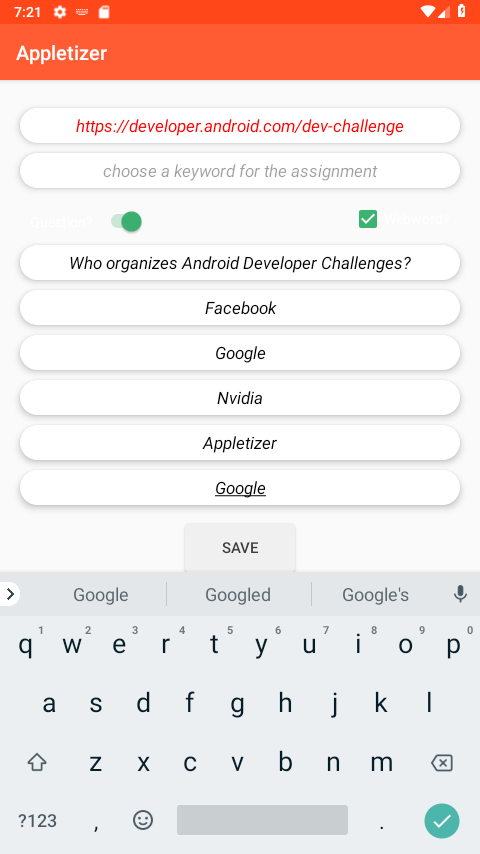
* Be sure to include this cover letter in your GitHub repository
* Your GitHub repository should be tagged #AndroidDevChallenge
* Don’t forget to include other items in your GitHub repository to help us evaluate your submission; you can include prior projects you've worked on, sample code you've already 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.**](https://docs.google.com/forms/d/e/1FAIpQLSe43koQL33IzgxXQl29Ex3AhFuqd4hQzxLiXREqwRkDGtx1vA/viewform?usp=sf_link)







Prediction for developer.android.com/dev-challenge article



Converting a link into an appletizer

