UNIL | Université de Lausanne

Project - Big-Scale Analytics

Professor: Michalis Vlachos

Deliverables due: Consult syllabus in Moodle. **Number of group participants:** 3

Foreword

We will reimplement the project we did last semester, LingoRank, but we will use cloud-based services. The goal is to predict the difficulty of some text in French. You can find the training and test data **here: https://github.com/michalis0/BigScaleAnalytics/tree/master/project/data.**

Learning Goals

Our goal is to learn how to use cloud-based services:

- Train predictive models on the cloud.
- Deploy a docker application on the cloud
- Use storage and functions on the cloud.

Milestone 1 - Creating/Evaluating the model – 8 May

Create a predictive model that can be used to predict how easy or difficult a French text is. You may model the problem as a classification problem (e.g. mapping a level from A1 to C2) or as a regression problem, eg A1 = 1, A2 = 2, B1 = 3 and so on). For this, we recommend using the various cloud services from Microsoft or Google, such as Google AutoML, that can be used for:

- **Text classification and sentiment analysis:** https://cloud.google.com/natural-language/automl/docs/features
- **Regression:** https://cloud.google.com/automl-tables/docs/beginners-guide

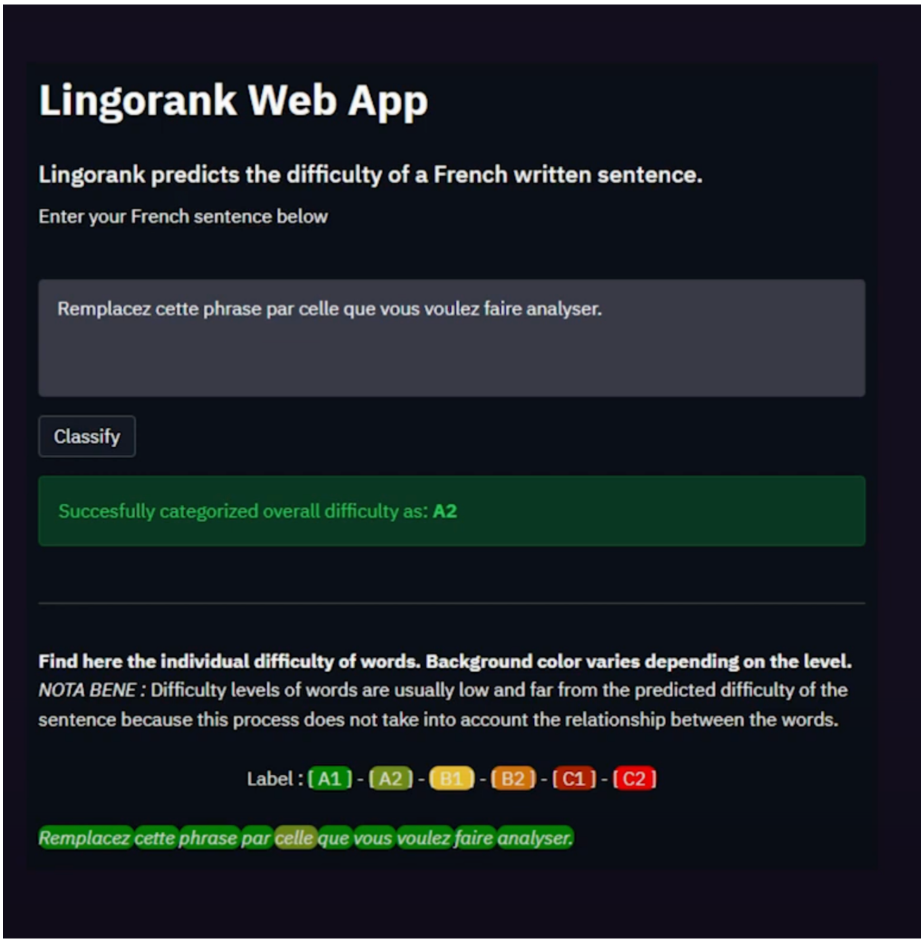
Watch the video presentation of the previous class that describe those services:
Text classification with Google AutoML : https://www.youtube.com/watch?v=Pc1YP-qYm_I
A. Sentiment analysis with Google AutoML : https://www.youtube.com/watch?v=ZHC-kOKHU9E

Deliverable: A callable API point which, given a text in French, will return its difficulty. Wrap this as a callable function. We also expect an update in the README of your GitHub. Show the confusion matrix, if you model as a classification problem. What is the accuracy and various metrics reported? Do you understand what they mean? Think deeper and try to solve any problems that exist. What is the best accuracy that you get? Depending on how you model the problem, pick the right metric and annotated data to evaluate it. For example, if you model the problem as regression, you may use an RMSE metric; if you model as classification, use F1-score.

Milestone 2 - Docker and Deployment – 30 May

Build a simple UI where a user can input some text, click a button and gets back the result on the difficulty. Create this UI with Flask and a UI tool (eg https://streamlit.io). Packetize it with docker. Deploy it on the google cloud. Provide the URL of this cloud address. You can see an example interface below.

Example of Streamlit user interface for your application



Lingorank Web App

Lingorank predicts the difficulty of a French written sentence.

Enter your French sentence below

Remplacez cette phrase par celle que vous voulez faire analyser.

Classify

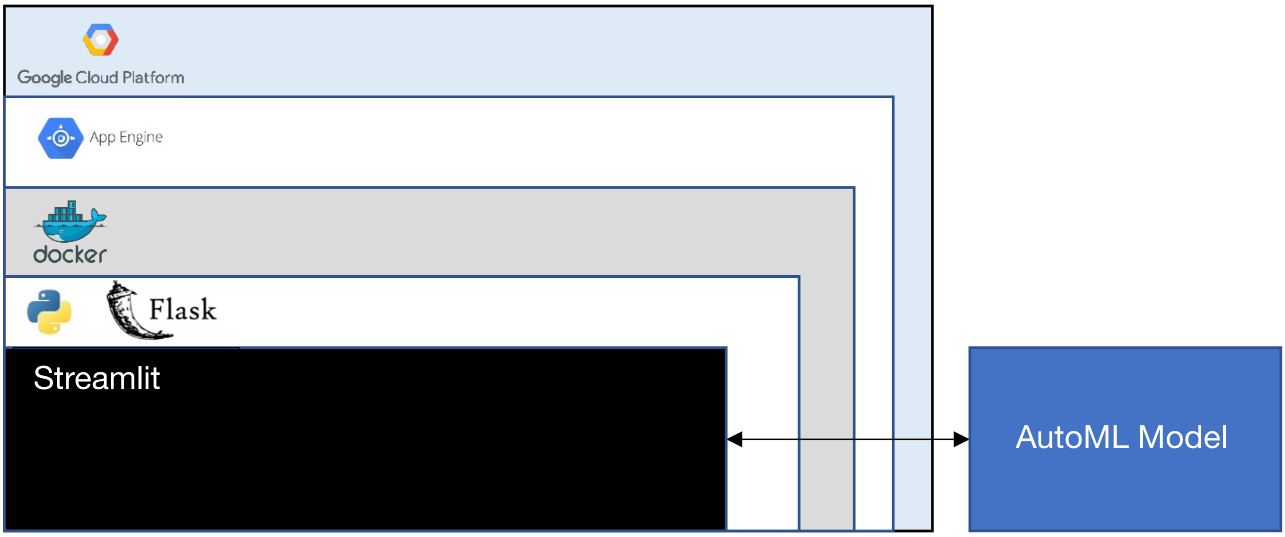
Succesfully categorized overall difficulty as: **A2**

Find here the individual difficulty of words. Background color varies depending on the level.
NOTA BENE : Difficulty levels of words are usually low and far from the predicted difficulty of the sentence because this process does not take into account the relationship between the words.

Label : [A1] - [A2] - [B1] - [B2] - [C1] - [C2]

Remplacez cette phrase par celle que vous voulez faire analyser.

Architecture of your machine learning model running on Google App Engine



I. **Github:** A project **GitHub page** which will be updated accordingly during the different milestones. The main README and the notebook should reflect your current progress (and the relevant code). Note that, as we progress through the semester, you may need to revisit the previous parts. This is ok. You should evaluate your final result. Some part of your grade will also be influenced by the accuracy of your solution on the test set **evaluation.** In your README, mention your score for the metric and test data prepared by your TAs.

II. **API Services:** Provide the **endpoints of all the services** that you use and give examples how to call them. Clearly state what is the URL of your main service. Your main service should accept a sentence in French and return back its level. You may also have created other services (this is common when you design using a micro-services architecture). Clearly state all the web-service APIs that you use.

III. **Tools:** Mention which technical tools you used: Flask, docker, cloud service, etc.

IV. **UI:** A usable UI in the cloud where we can try your service. We expect to see (at least) a simple webpage with a UI where you enter a text and it returns back its easiness/difficulty level. Feel free to get creative here!

V. **Video:** Create a YouTube video of your solution and embed it in your notebook. Imagine you are giving a presentation or a tutorial. The video should explain:

  V.A. The general architecture (cloud provider, technology, APIs, etc.)

  V.B. How you modelled the problem.

  V.C. Show the UI that you have created

  V.D. An evaluation of your solution (accuracy, precision, recall, F1-score, etc.)

  V.E. Post the video link to #project channel in Slack. All projects will also be **presented live** by the group during the last class.

Grading Scheme

I. **75%** of grade: GitHub quality, code quality, solution (UI+services) and evaluation

I. **25%** of grade from video + live presentation

Logistics and Deadlines

A. Register your group on Moodle (three people)

B. Deadlines for the different milestones are found on Moodle.

Useful Resources

https://www.youtube.com/watch?v=RT7SBYA5jTk