trustwise

Individual Assessment

Technical Take-Home Assignment

Purpose

This document outlines a development task that has been assigned to ascertain the range of skills present in technical applicants. The applicant will need to deliver the source code and associated artifacts, and be prepared to deliver a short presentation regarding the design and implementation of the assessment. The quality of the deliverable and presentation will be the primary way to evaluate the applicant's skills and suitability for the role.

Problem

The goal is to design and deploy a REST API that receives a string of text (which could be an output from an LLM), scores it using machine learning models, and returns the results. The received text and returned results should be logged to a database and displayed in a web UI. Design and document an appropriate interface for the REST API.

Commonly used models are:

1. "Vectara" https://huggingface.co/vectara/hallucination_evaluation_model?text=A+man+walks+into+a+bar+and+buys+a+drink+%5BSEP%5D+A+bloke+swigs+alcohol+at+a+pub
2. "Toxicity" https://huggingface.co/s-nlp/roberta_toxicity_classifier?text=I+like+you.+I+love+you
3. "Emotion" https://huggingface.co/SamLowe/roberta-base-go_emotions
4. "Gibberish" https://huggingface.co/wajidlinux99/gibberish-text-detector?text=I+like+you.+I+love+you
5. "Education" https://huggingface.co/HuggingFaceFW/fineweb-edu-classifier

Please work only on the TWO models specified in the email sent to you, which is the scope for this assignment.

All models are open-source, and can be accessed from HuggingFace, which is the leading open source platform for language models. This task doesn't require prior experience using HuggingFace, since calculating scores is detailed extensively in their documentation.

The web UI should contain details such as a graph showing the scores logged and a table showing the history of received text and results. You are free to use any frontend framework you are comfortable with.

Finally, the REST API and web UI should be containerized in docker, so that they can be deployed to any cloud provider. **You don't have to deploy the containers!** Providing suitable build instructions including Dockerfiles is sufficient.

Example usage of REST API:

Received text: “I like you. I love you”
Response:
Vectara: 0.816
Toxicity: 0.000
Emotion: Love: 0.955
Gibberish: Clean: 0.873
Education: 0.437

Note: We prioritize originality and quality of code over quantity. If we detect that ideas or code have been copied from another source or other students, your submission will be rejected.

Delivery Requirements

- The source code is to be written in the applicant's choice of programming language.
- Source code is to be made available for assessment.
- The project should supply build instructions.
- The project should deliver an appropriate level of architecture, technical documentation to reflect the nature, scope of the project at a level of detail that the team feels appropriate.
    - Any suitable methodology and standards can be adopted.
- The applicant should look to deliver a suite of quality metrics that illustrate the functional / non functional aspect of the project.
- At the end of the assignment, the individual will be asked to present their project. A 30 minute time slot will be allocated followed by a 15 minute Q&A.
- The style, structure and contents of the presentation are left to the discretion of the applicant
- Optionally, record a short video showing the functionality that you developed, voice over in the video is optional.
- In case of any questions or clarifications, please contact abhishek@trustwise.ai or manisha@trustwise.ai
- Upload your code on GitHub and share links with abhishek@trustwise.ai, manisha@trustwise.ai and shiv@trustwise.ai

Delivery Schedule

- The applicant will be given up to 1 week to deliver the project, but you are encouraged to complete it faster, to expedite the hiring process.
- Post completion of the code and associated artifacts, the deliverable will be assessed over 1 week.

- If the applicant's delivery passes the assessment they will be asked to present the solution to the Trustwise team.
- During the delivery period, the applicant can engage with the Trustwise team to help answer any questions and guide the implementation.

Success Criteria

- The delivery meets the scope of the functional and non-functional requirements
- The quality of the deliverable is deemed fit for purpose and reflective of the skills expected for a technical team working at the level indicated during their application.
- The Trustwise assessor will make the final subjective decision on the quality of the delivery and determine if the engagement has been a success.