Applied MSc in Data Analytics

Applied MSc in Data Science & Artificial Intelligence

Applied MSc in Data Engineering & Artificial Intelligence

Course: Python Machine Learning Labs

Project: Develop an end-to-end Machine Learning Pipeline

Instructor: Assan Sanogo

Project Overview:

The aim of the course project is to ensure students are comfortable enough developing an end-to-end pipeline to answer a given problem or use case.
The project is a group project. Since this is a mixed course (DA/DS/DE), students are encouraged to form mixed groups to benefit from the competences of their teammates when working on different components of the pipeline. An ideal group is a group of 3 members: 1 Data Analyst, 1 Data Scientist and 1 Data Engineer. However, since this is not anideal world, and group formation relies heavily on student distribution, groups of 2-4 are allowed (group diversity is still **highly encouraged**).

- Project: English quality prediction

This project is based on a dataset of 7000+ essays graded by English specialists. This data problem is close to a real-world situation as it requires to be cleaned, an EDA must be thoroughly done so that the team can engineer relevant features.

This project is a NLP problem that will be the foundation of an English program used by the company *Easy Sailing Language Training*. Their ambition is to have a reliable tool to assess new students' ability to write in English according to the IELTS grading system. In turn it would help prospective students in knowing how much time they need to invest to get to the next level.

You are encouraged to experiment with extra libraries like Spacy, NLTK, scikit-learn, HuggingFace…

Project Summary:

Develop an end-to-end pipeline that process an essay and outputs a grade describing the level of English proficiency.

Disclaimer:

If after analyzing the dataset struggling with the dataset, you might want to reframe the problem as a classification problem (poor, average, great).

Project Objectives:

Using the provided dataset, you are asked to train a model that predicts an essay's rating. The project can be submitted as a Jupyter Notebook (at least) and should include exploratory analysis of the data, feature engineering and selection, model training and evaluation and finally, deployment.

You may use additional resources as you see fit (provided you can justify how they can serve your solution). You can even consult similar solutions from the Internet. **However, this comes with a big responsibility: any submission that is over-plagiarised or does not reflect personal work will not be accepted**.

Project Resources:

All resources necessary to the project are organized in the .zip folder

- Training set
- Valid set
- Methodology for dataset creation
- Dataset description

Dataset: https://www.transferxl.com/download/08vSFcz3B7fXPr

Links:

- https://paperswithcode.com/
- https://www.analyticsvidhya.com/blog/2021/04/a-guide-to-feature-engineering-in-nlp/
- https://spacy.io/usage/spacy-101

Tips:

This project is "open" and you are encouraged to use your creativity. In particular, the dataset differs with traditional datasets, because you have to handcraft all the features your model will use. This leads to ask yourself, how are you going to clean the dataset, what makes a text complex? what makes a text well written? How do we measure clarity?
What tool/library enables me to compute these measures?

Below are some features you could consider to measure complexity of a text

Many measures exist to compute text complexity based on **words** and **word structure:**

- Flesch reading ease
- Gunning Fog
- Automated Readibility index (ARI)
- Smog Index
- Flesch-Kincaid
- Coleman-Liau
- Dale-Chall Readability

Complexity can also be measured **lexically**:

- Words sophistication thank to a corpus (AWL)
- Words frequency (tf-idf)
- Lexical diversity
- Lexical variation features (ratio of words tagged as adjectives, nouns or verbs) over total number of words.

Complexity can also be measured via the **syntactic structure** of the sentences:

- Roots of Sentence tree
- Length of Sentence tree
- Average number of Connections of Sentence tree at the root level
- Length of clauses

The **Quality** of a text should also include:

- misspelling score
- slur usage
- overusage of punctuation

There are apps online that you can easily “hack” via Selenium to handcraft features:

https://app.readable.com/text/?demo
https://hub.cathoven.com/?scene=analyser&core=cefr (be careful - limited usage)
http://www.roadtogrammar.com/textanalysis/

Project Evaluation:

Both projects will be evaluated using the following rubric. It contains the required items for a complete submission. The grading system is over 5 and the final grade will be transformed to a grade over 100.

- Data analysis (data processing, data cleaning, exploratory analysis, plots of relevant attributes) and feature selection (feature engineering, feature pruning, choice justification) **[1 point]**
- Model training (motivation for selected model, comparison of different models) and evaluation (evaluation metric, results interpretation) **[1 point]**
- Project report (short report explaining the approach and results) **[1 point]**
- Project reproducibility (requirements file with necessary packages, README file for running the project) **[1 point]**
- Project hosting and deployment (Github, Docker, AWS, Heroku or any other method) **[1 point]**