**StackExchange question quality detection**

Aim: Categorise the StackOverflow questions into various quality classes.

Dataset Creation:

You can download the dataset from [here](https://drive.google.com/drive/folders/15xd3v1mSaeGILRnpUUa2V-r2AbGp26kH?usp=sharing). The folder contains two files, one is the zipped file

containing multiple XML files, another is a readme file describing the details related to each

xml file. You can work on Post.xml only.

You can also use Stack Exchange API to sample the dataset directly from there. You can

store each question’s information as a document in the mongoDB collection, or you can

create a DataFrame to store all the features related to each question.

Create a program to label the data into three categories:

1. Good-Quality questions: Questions for which score is greater than 5 and answer

count is greater than 0 should be labelled as good quality questions.

2. Low-Quality questions: Questions for which the score is between 0 to 5 and having

no answers should be labelled as low-quality questions.

3. Very-low quality questions: Questions which have negative scores

Feature Extraction: Since you are performing the labelling by yourself, any set of features

can be used for the classification scores and answer\_count. Your task is to come up with a

feature set which allows you to get more accuracy.

Parsing the XML file:

There are various ways to parse an XML file. Easiest way is to use [celementtree](https://docs.python.org/3/library/xml.etree.elementtree.html) library of

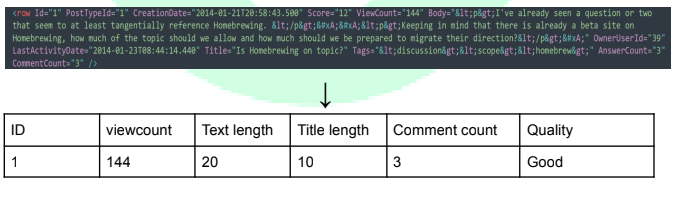
Python to parse the Posts.xml file. Refer to the following [document](https://www.datacamp.com/tutorial/python-xml-elementtree) on how to parse an XML

data using celementtree. We recommend you to play around with it to understand the basic

parsing. At the end, you need to create a feature matrix from Posts.xml. For example, for a

particular question on Posts.XML, you can create following list of features (please see the

below picture)



Exploratory data analysis:

Measure the statistics for the sample dataset created, such as mean votes, mean title

length, mean body length, etc. (Hint: can you use these as features?)

Perform feature engineering on the dataset created by you. Create a correlation matrix (use

the confusion matrix approach) between every parameter.

Preprocessing: Choose the preprocessing steps that boost your prediction.

Prediction:

Your task is to train a classifier which, given a question’s data, categorises the data into one

of these three categories.

Models to choose from: Logistic-regression, Multinomial-Naive Bayes, Random forest.

Highest accuracy will fetch higher marks.

Post-analysis Questions:

1. Clearly mention and explain the preprocessing phase. Why did you choose a

particular pre-processing step?

2. The code should be added to your GitHub repository with a proper readme file.

3. Explain your choice of model and why do you think it performs well?