Senior Data Scientist Challenge

PART 1

Music Classification Mini-Project:
A client has requested us to build a composer classifier based on live captured audio that can be set to stream chunks of data to our model at 15, 30, 60-second intervals, as desired. For this project, the client has collected and annotated several hundred audio files and have saved them as simple midi files. The provided PS1 Folder contains the known midi files pertaining to four (4) composers: Bach, Beethoven, Schubert, and Brahms.

Goal:
The goal of this project is to develop a classifier/pipeline that is able to determine which midi files in the provided PS2 folder are not written by the four (4) composers above (it is a small number).

Details:
In the interest of time, 30-seconds is the desired default setting. Analysis of how data quantity effects performance is not expected, but will highlight important client requirements.

The midi files contain additional information that may not be available at inference time; ensure that your algorithm can support this.

The name of the file before the first underscore is the composition name.

The attached dataset is provided in midi format and taken from Musicnet. (https://arxiv.org/pdf/1611.09827.pdf).

Optimal model performance is not expected, and neither is the use of the entire dataset and information. Classical ML approach(es) are recommended as the main form of analysis as setting up a thorough pipeline will be evaluated by the client more favorably than a poorly implemented state-of-the-art model that is not well-validated or documented. A deep learning approach may be attempted based on results to showcase additional capabilities.

The client exercise is geared towards a truncated, real-world scenario, demonstrating: (1) Efficacy: How fast you can build solutions and the quantity of analysis shown; (2) Quality: How well you showcase your understanding of the problem, data, and code; (3) Robustness: The completeness of your analysis flow and the robustness of the validation process.

In particular, the client will consider overall problem understanding and set up of the analysis; the completeness of your EDA and model building/tuning/validation; result interpretation and analysis approach and outcomes; code quality and completeness; and final conclusions and recommendations based on the brief sprint.

Using Open-source Libraries/Packages:

Any open-source code/libraries and publicly available information is encouraged providing you cite the material.

Return & Due Date:
Please email back the final notebook within 72 hours of receiving this exercise. Please return a brief abstract of the solution and findings. **For ease of review, please save your analysis in both an HTML file as well as ipynb.**

“

PART 2

Pandas Basics:
The dataset for this part of the interview can be found at Link, You will see two .csv files which will form the basis of this part.

Please use a Jupyter notebook and perform the following manipulations in order.

1. Concat:
    - Vertically concatenate the two csvs into a new dataframe **df_full**
    - For missing values in the test set use a random selection of values in the same column from the train set.
2. Filter:
    - On **df_full** select only the weekdays, where the relative humidity is >50 and absolute humidity <1.4 denote this **df_filtered**.
3. Apply:
    - On **df_filtered** create a column called “apply_step3” with values sensor 2 * sensor 3 if sensor 4 < median value of sensor 4, else sensor 2 * sensor 4 - mean average of the calendar month value of (sensor 4)
4. Groupby:
    - On **df_full** table find the min/max/mean/count temp over each day as a table denote as **df_group**
5. Create a column in **df_full**:
    - Create a function to read in a dictionary with 5 unequal splits that span: [0-0.2), [0.2-0.5), [0.5-0.8), [0.8-0.85), [0.85-1] and make the function generic to arbitrary splits.
    - Apply this function to the max normed sensor 1 column, denote the resulting column *Truth*.
    - Apply it to a max normed sensor 2 column and denote the resulting column *Predicted*
6. Merge: Starting from **df_full** let’s look at the differences to the same day last year for available days
    - How many days overlap in the two latest years?
    - What is the average of the daily difference of abs humidity?
    - What is the difference in average of sensor 1 data between the two years?
7. Plots on **df_full**:
    - histogram of absolute humidity
    - sensor 1 - 5 histograms overlaid on same figure
    - scatter plot of sensor 1 vs sensor 2 coloured blue if weekday, red otherwise
    - correlation matrix of all sensors against each other with histograms of the sensor values along diagonal
    - confusion matrix of predicted vs truth

8. Create data split: Create a column “train_val_set” which contains a random 70% denoted “train”/30% “val” split of the data from the **train.csv** ensure that each day is not split between sets.
9. Save all resulting tables listed in **bold** as CSVs.

Return & Due Date:
Please email back your submission within 72 hours of receiving this exercise.

REVIEW

Assessment response will be provided with 24-48 hours depending on timing and current activity.

Since the role as Senior Data Scientist requires the combined skills of technical execution, critical thinking, and clear communication, the following will be considered:

- Technical knowledge (show relevant and deep knowledge on the topic through problem solving, EDA, modeling, analysis volume, code quality)
- Analytical Skills (showcases logic and analysis robustness)
- Communication (shows clarity in approach, formatting, communication, response to conclusions or recommendations)