AIML Online

Frequently Asked Questions in Problem Statement

Course: Supervised Learning

PART - A [30 Marks]

** Direct or Self-explanatory questions are not covered in this FAQ.*

1. Data Understanding:

1 C. Compare Column names of all the 3 DataFrames and clearly write observations. [1 Mark]

→ *Compare the column names of all the three dataframes. As we are going to merge datasets by rows, checking the column names, order and type is mandatory. Use a simple compare operator to check whether all 3 dataframes have the same column names. And write your observations from the result.*

1 D. Print DataTypes of all the 3 DataFrames. [1 Mark]

→ *Print the datatypes of all the 3 dataframes and write your observations.*

1 E. Observe and share variation in 'Class' feature of all the 3 DaraFrames. [1 Mark]

→ *Check the 'Class' variable's distribution and categories.*

2. Data Preparation and Exploration:

2 A. Unify all the variations in 'Class' feature for all the 3 DataFrames. [1 Marks]

→ *Unify the variations reported in the previous step 1.E.*

Example - If the 'Class' variable of 'normal' dataframe has 'Normal', 'normal' or 'Nrml' replace them with 'normal'. Similarly, check and unify the 'class' for type_s and type_h dataframes.

2 B. Combine all the 3 DataFrames to form a single DataFrame [1 Marks]

→ *Combine the 3 datasets into 1. Look at the checkpoint that the final dataframe should have 310 rows and 7 columns.*

3. Data Analysis:

3 C. Visualize a pairplot with 3 classes distinguished by colors and share insights. [2 Marks]

→ *Create a pairplot for the given variables and the color of the data points in the pairplot should be distinguished by 'Class' categories.*

4. Model Building:

4 D. Print all the possible performance metrics for both train and test data. [2 Marks]

→ *Print the performance metric of classification models that include accuracy, precision, recall, F1 score etc.*

5. Performance Improvement:

5 A. Experiment with various parameters to improve performance of the base model. [2 Marks]

→ *So far you would have run the default model, now you can tune the model by changing the parameters in KNeighborsClassifier() or svm function. Firstly, self-explore what are the parameters available in the models and check how you can fine-tune it by changing the options. You have to just research a bit and do it. (Detailed parameter tuning will be covered in feature engineering course)*

Reference link for Hyperparameter tuning for a KNN problem - https://medium.datadriveninvestor.com/k-nearest-neighbors-in-python-hyperparameters-tuning-716734bc557f

You can explore and tune the hyperparameters for other models too.

PART - B [30 Marks]

1. Data Understanding and Preparation:

1 D. Change Datatype of below features to 'Object' [1 Marks]

'CreditCard', 'InternetBanking', 'FixedDepositAccount', 'Security', 'Level', 'HiddenScore'.

[Reason behind performing this operation: - Values in these features are binary i.e. 1/0. But DataType is 'int'/'float' which is not expected.]

→ *The variables are of object type with Binary or multi-class outputs like 0,1 or 1,2,3 etc. Hence, convert them to 'Object' type*

2. Data Exploration and Analysis:

2 A. Visualize distribution of Target variable ‘LoanOnCard’ and clearly share insights. [2 Marks]

→ *Plot a suitable plot to display distribution of Target variable.*

2 C. Check for unexpected values in each categorical variable and impute them with the best suitable value. [2 Marks]

→ *Unexpected values mean if all values in a feature are 0/1 then ‘?’, ‘a’, 1.5 are unexpected values which needs treatment*

3. Data Preparation and model building:

3 D. Print evaluation metrics for the model and clearly share insights. [1 Marks]

→ *Print the performance metric of classification models that include accuracy, precision, recall, F1 score etc.*

3 E. Balance the data using the right balancing technique. [2 Marks]

→ *Target balancing can be done by upsampling the minority class or downsampling the majority class or by using SMOTE as per target distribution. You can research a bit and do this task.*

4. Performance Improvement:

4 A. Train a base model each for SVM, KNN. [4 Marks]

→*You have to build a base model without tuning any parameters.*

4 B. Tune parameters for each of the models wherever required and finalize a model. [3 Marks]

(Optional: Experiment with various Hyperparameters - Research required)

→ *Tune the parameters as performed in Part A, Question 5 A.*

You can tune the model by changing the parameters in KNeighborsClassifier() or svm function. Firstly, self-explore what are the parameters available in the models and check how you can fine-tune it by changing the options. You have to just research a bit and do it. (Detailed parameter tuning will be covered in feature engineering course)

Reference link for Hyperparameter tuning for a KNN problem - https://medium.datadriveninvestor.com/k-nearest-neighbors-in-python-hyperparameters-tuning-716734bc557f

You can explore and tune the hyperparameters for other models too.

4 C. Print evaluation metrics for final model. [1 Marks]

→ *Print the performance metric of the final model that includes accuracy, precision, recall, F1 score etc.*

4 D. Share improvement achieved from base model to final model. [2 Marks]

→ *Show the performance improvement of that model (comparing its base model & final model performance report)***.**